

ORACLE® HYPERION FINANCIAL MANAGEMENT,
FUSION EDITION

RELEASE 11.1.1.3

OBJECT GUIDE

ORACLE®
ENTERPRISE PERFORMANCE
MANAGEMENT SYSTEM

Financial Management Object Guide, 11.1.1.3

Copyright © 2000, 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	39
Prerequisites for Developers and Users	39
COM Data Type Mapping	40
Features Exposed by the Objects	40
Application Server Tier Type Libraries	41
Type Libraries for Loading and Extracting	42
Control for Selecting Dimension Members	42
Additional Architectural Considerations	43
About Subcubes	43
Examples of Subcubes	44
Data Across Multiple Application Servers: Subcube Caching	44
Values Returned in Arguments	45
Error Handling	45
Chapter 2. About the Type Libraries and Objects	47
Type Library References	47
Objects, Properties, and Methods	49
HsxClient Type Library Overview	49
HsxClientUI Type Library Overview	52
HsxServer Type Library Overview	53
HsvSession Type Library Overview	54
HsvMetadata Type Library Overview	55
HsvData Type Library Overview	69
HsvCalculate Type Library Overview	74
HsvJournals Type Library Overview	75
HsvSecurityAccess Type Library Overview	79
HsvSystemInfo Type Library Overview	84
HsvProcessFlow Type Library Overview	88
HsvReports Type Library Overview	91
HsvMDArrays Type Library Overview	93
HsvDataCubes Type Library Overview	102
HsvStarSchemaACM Type Library Overview	104

HsvICM Type Library	105
Type Libraries for Loading and Extracting Information	108
HsvPOVSelection Type Library	115
HsvResourceManager Type Library Overview	117
Chapter 3. Getting Started	119
Logging a User on to an Application	119
Working with Metadata	120
Getting and Setting Cell Data	121
Getting and Setting User Preferences for Number Formatting	121
Error Handling	122
Loading and Extracting Application Information	122
Chapter 4. HsxClient Type Library	123
HsxClient Object Methods	123
AuthenticateCSSToken	124
AuthenticateSecurityAgentCredentials	124
AuthenticateUserCredentials	124
AuthenticateUserOnCluster	124
AuthenticateUserOnClusterSSO	124
AuthenticateUserOnServer	124
CreateApplication	124
CreateApplicationCAS	124
CreateApplicationCASWithAccessCode	126
CreateObjectOnCluster	126
CreateObjectOnServer	126
DeleteApplication	126
DeleteApplicationWithAccessCode	127
DeleteSystemErrors	127
DetermineWindowsLoggedInUser	127
DisableNewConnections	128
DoesUserHaveCreateApplicationRights	128
DoesUserHaveSystemAdminRights	129
EnableNewConnections	129
EnumProhibitConnections	130
EnumProvisioningProjects	131
EnumRegisteredClusterNames	132
EnumRegisteredServerNames	132
EnumUserAppPreferences	132
EnumUsersOnSystem	132

EnumUsersOnSystemEx	134
EnumUsersOnSystemEx2	136
GetApplicationFolder	137
GetClusterInfo	138
GetHFMErrLogRecordSet	138
GetLogonInfo	138
GetLogonInfoSSO	138
GetServer	139
GetServerOnCluster	139
GetSSOTokenUsingWebSecurityAgentCredentials	139
GetWebSecurityAgentSettings	139
IsValidApplication	139
KillUsers	140
OpenApplication	140
RegisterApplicationCAS	141
RegisterApplicationCASWithAccessCode	142
RegisterCluster	142
RegisterServer	142
ScriptableEnumRegisteredClusterNames	142
ScriptableEnumRegisteredServerNames	143
ScriptableGetLogonInfoSSO	143
ScriptableOpenApplication	143
SetApplicationFolder	143
SetLogonInfo	143
SetLogonInfoSSO	144
UnregisterAllClusters	144
UnregisterCluster	144
UnregisterServer	145
UpdateUserAppPreferences	145
WarnUsersForShutDown	145

Chapter 5. HsxClientUI Type Library	147
HsxClientUI Object Methods	147
DeleteApplication	148
GetServer	148
Initialize	148
Logon	148
OpenApplication	149
RegisterServer	150

Chapter 6. HsxServer Type Library	151
HsxServer Object Methods	151
DeleteSystemErrors	152
EnumDataSources	152
EnumRegisteredDSNs	152
GetClustersAndServers	153
GetDataSource	154
GetFileTransfer	154
GetHFMErrLogRecordSet	154
GetSystemDataLinkFile	154
GetSystemFolder	154
GetXMLErrorFromDatabase	155
GetXMLErrorsListFromDatabase	155
ScriptableEnumDataSources	156
ScriptableEnumRegisteredDSNs	156
HsxServer Object Properties	156
CSSEnabled	156
Chapter 7. HsvSession Type Library	157
HsvSession Object Properties	157
Calculate	157
Data	157
ICM	158
Journals	158
Metadata	158
ProcessFlow	158
Reports	158
Security	158
SystemInfo	158
HsvSession Object Methods	159
CreateObject	159
GetLicenseExpirationStatus	159
HasSystemChanged	159
HasUserStatusChanged	160
IsBusy	160
IsRunningTasks	160
LockMetadataLoadWithSystemChangeCheck	160
UnlockMetadataLoad	160

Chapter 8. HsvMetadata Type Library	161
About Member IDs	161
HsvMetadata Object Methods	162
EnumConsolidationMethodIDs	162
EnumExtractOptions	163
EnumLanguages	164
EnumLoadOptions	164
Extract	165
GetApplicationAttribute	167
GetApplicationCurrency	168
GetApplicationSettingsTimeStamp	168
GetByIndexValidationAccount	169
GetCellLevelAccountType	170
GetConsolidationMethodDescription	170
GetConsolidationMethodInfo	171
GetConsolidationMethodsTimeStamp	172
GetCurrencyTimeStamp	173
GetCurrencyValueIDForEntityValueCombination	173
GetCurrencyValueIDsForEntityValueCombinations	173
GetDefaultValueOfActiveStatusAccount	174
GetFdmAppName	174
GetFrequencyID	174
GetICPEntitiesAggregationWeight	175
GetSupportSubmissionPhaseForAccountFlag	175
GetSupportSubmissionPhaseForCustom1Flag	175
GetSupportSubmissionPhaseForCustom2Flag	176
GetSupportSubmissionPhaseForCustom3Flag	176
GetSupportSubmissionPhaseForCustom4Flag	176
GetSupportSubmissionPhaseForICPFlag	177
GetUseSubmissionPhaseFlag	177
GetValidationAccount	178
IsCustomMemberValidForAccount	179
IsOrgByPeriodApplication	180
Load	180
LoadWithAccessCode	185
TranslateApplicationAttributeForDisplay	185
HsvMetadata Object Properties	185
Accounts	185
Currencies	186

Custom1	186
Custom2	186
Custom3	186
Custom4	186
Dimension	186
Entities	187
ICPs	187
Periods	187
Scenarios	187
Values	188
Views	188
Years	188
IHsvTreeInfo Interface Methods	188
EnumAllMemberIDs	188
EnumAllMemberLabels	189
EnumAllParentAndChildIDs	189
EnumAllParentAndChildLabels	190
EnumAncestors	191
EnumBaseMemberIDs	192
EnumDefaultAncestors	193
EnumDefaultAncestorsLabels	194
EnumDescendants	194
EnumIDsOfChildren	195
EnumIDsOfChildren	196
EnumMemberLists	197
EnumMembers	197
EnumMembers2	199
EnumMembersWithAttribValue	200
EnumParents	201
EnumSortedIDsOfChildren	201
EnumSortedIDsOfChildren	202
EnumSortedMembers	203
Find	204
FindByDesc	205
FindMatchingMembersFromHierarchy	207
FindMatchingMembersFromHierarchyByDesc	208
FindMatchingMembersFromHierarchyWildcard	209
GetAllPathsToMember	210
GetAttributeValue	211

GetDefaultHierarchyPosition	212
GetDefaultItemIDHierarchy	212
GetDefaultMemberID	214
GetDefaultParent	214
GetDefaultParentLabel	214
GetDescription	215
GetDisplayInfo	216
GetDisplayInfoForSeveralItems	217
GetItemGeneration	218
GetItemID	219
GetItemIDQL	220
GetItemLevel	220
GetLabel	221
GetMemberListID	221
GetMemberListName	222
GetNumBaseMembers	222
GetNumChildren	223
GetNumDescendants	223
GetNumMembers	224
GetNumParents	225
GetQualifiedLabel	225
GetTreeCapabilities	225
GetTreeName	226
GetTreeTimeStamp	226
HasChildren	227
IsMemberABaseOf	228
IsMemberAChildOf	228
IsMemberADescendantOf	229
SortMembersBasedOnList	230
TranslateAttributeValueForDisplay	230
HsvAccounts Object Methods	232
GetAccountType	232
GetCalcAttribute	233
GetICPTopMember	233
GetIsICP	233
GetNumDecimalPlaces	234
GetPlugAccount	235
GetSecurityClassID	236
GetSubmissionGroup	236

GetTopMemberOfValidCustom1Hierarchy	237
GetTopMemberOfValidCustom2Hierarchy	237
GetTopMemberOfValidCustom3Hierarchy	238
GetTopMemberOfValidCustom4Hierarchy	238
GetUserDefined1	239
GetUserDefined2	239
GetUserDefined3	239
GetXBRLTags	240
IsCalculated	240
IsConsolidated	240
IsCustom1AggregationEnabled	241
IsCustom2AggregationEnabled	241
IsCustom3AggregationEnabled	242
IsCustom4AggregationEnabled	242
IsICP	243
IsICPRestricted	243
UsesLineItems	243
HsvCustom Object Methods	244
GetAggregationWeight	244
GetSecurityClassID	244
GetSubmissionGroup	245
GetUserDefined1	245
GetUserDefined2	245
GetUserDefined3	246
IsCalculated	246
IsSwitchSignEnabledForFlow	247
IsSwitchTypeEnabledForFlow	247
HsvEntities Object Methods	247
GetAllowAdjustments	247
GetAllowAdjustmentsFromChildren	248
GetDefaultValueID	248
GetHoldingCompany	249
GetSecurityAsPartnerID	249
GetSecurityClassID	250
GetUserDefined1	251
GetUserDefined2	251
GetUserDefined3	251
IsChild	252
IsDescendant	252

IsICP	253
IsOrgByPeriodFilteringOn	253
IsSecurityAsPartnerEnabled	254
HsvICPs Object Methods	254
GetSecurityClassID	255
GetSubmissionGroup	255
HsvPeriods Object Methods	256
GetBaseFrequency	256
GetFrequency	257
GetNumPeriodsInFrequency	257
GetPeriodFromFrequency	258
GetPeriodsInFrequency	259
HsvScenarios Object Methods	259
GetPhasedSubmissionStartYear	260
EnumPhasedSubmissionStartYears	260
GetDefaultFrequency	260
GetDefaultView	260
GetMaximumReviewLevel	261
GetMissingDataZeroViewForAdjValues	262
GetMissingDataZeroViewForNonAdjValues	262
GetSecurityClassID	263
GetUserDefined1	264
GetUserDefined2	265
GetUserDefined3	265
IsConsolidateYTD	265
IsPhasedSubmissionEnabled	266
SupportsAccountAllocations	266
SupportsEmailAlerting	266
SupportsProcessFlow	267
SupportsTargetSetting	267
SupportsTimePeriodAllocations	267
UsesLineItems	268
HsvValues Object Methods	268
GetCurrencyIDFromValueID	268
GetValueIDFromCurrencyID	269
HsvYears Object Methods	269
GetYearRange	269
HsvCurrencies Object Methods	270
EnumCurrencies	270

EnumCurrencies2	271
GetCurrencyDescription	272
GetCurrencyID	272
GetCurrencyLabel	273
GetCurrencyTranslationOperator	273
GetScale	274
Chapter 9. HsvData Type Library	275
About Cell Statuses	275
Update Mode Constants	276
HsvData Object Methods	276
AddDataToMDDDataBuffer	277
AttachDocumentToCell	278
ClearAllData	279
ClearAllDescriptionsInSubCube	279
ClearDataAuditItems	280
ClearInputData	281
ClearInvalidData	283
CopyInputData	283
CopyInputDataForMultipleEntities	284
DeleteLineItemDetails	285
DetachDocumentFromCell	286
DMELoad	287
DoCellDescriptionsExist	288
DoesCellDescriptionExist	288
DoesDataExist	289
DoesSparseDataExist	290
EnumDataAuditItems	291
EnumDataAuditItems2	293
EnumEntitiesWithDataForScenarioYear	293
EnumExtractOptions	294
EnumLoadOptions	295
Extract	295
ExtractDataAuditItems	299
ExtractDrillableRegions	300
ExtractDrillableRegionsByURLNames	301
FilterMembersThatHaveData	301
FilterMembersThatHaveData2	302
FormatNumberToText	303

FormatNumberToText2	305
FormatStatusToText	307
FormatStoredNumberToText	308
FormatTextToNumber	309
FormatTextToStoredNumber	310
GetAllDescriptionsInSubCube	311
GetAllURLNames	312
GetAttachedDocumentsToCell	312
GetBaseDataForAccount	313
GetCalcStatus	313
GetCalcStatusEx	314
GetCalcStatusStatistics	315
GetCell	316
GetCellDescription	317
GetCellDescriptions	318
GetCellHistory	319
GetCellHistory2	321
GetCellJournalEntries	323
GetCellLineItems	325
GetCells	326
GetCellsWithRowSuppression	327
GetCellsWithRowSuppression2	330
GetCountOfAttachedDocumentsToCell	331
GetCurrencyCube	332
GetDataForAllMetadataCombinations	333
GetLineItems	333
GetMaxCellTextSize	333
GetMembersThatHaveData	333
GetNodeCube	334
GetPhaseSubmissionGridForGivenScenarioPeriod	335
GetStatus	335
GetStatusEx	337
GetStatusUsingPhaseID	338
GetTextCell	339
GetTextCellLineItems	341
GetTextCells	342
GetTextCellsWithRowSuppression	344
GetTextCellsWithRowSuppression2	346
GetTextLineItems	348

GetTransactionData	348
GetUnassignedGroups	349
GetURLByName	349
GetURLsForCell	350
InsertLineItemDetails	352
IsValidCellText	353
Load	353
LoadDrillableRegions	355
SetCalcStatusLocked	357
SetCalcStatusLockedForMultipleProcessUnits	358
SetCalcStatusUnlocked	358
SetCalcStatusUnlockedForMultipleProcessUnits	359
SetCell	359
SetCellDescriptions	361
SetCellLineItems	362
SetCells	363
SetCells2	365
SetCellsLineItems	367
SetFileForLoad	368
SetLineItems	368
SetMinMaxPeriod	368
SetPhaseSubmissionGridForGivenScenarioPeriod	368
SetTextCell	369
SetTextCellLineItems	370
SetTextCells	371
SetTextCellsLineItems	373
SetTextLineItems	374
StartLoad	374
UpdateDataUsingMDDDataBuffer	374

Chapter 10. HsvCalculate Type Library	375
HsvCalculate Object Methods	376
Allocate	376
Allocate2	376
CalcEPU	377
ChartLogic	377
ChartLogic2	378
Consolidate	379
Consolidate2	380

CustomLogic	381
FindOverlappingConsolidation	381
GetCOMDLLRules	382
GetConsolidationProgress	382
GetDefaultExchangeRate	383
GetEPUInfo	384
GetVBScriptRules	385
IsEntityAnEPUOwner	385
LoadCalcManagerRules	386
LoadCalcManagerRules2	386
SetCOMDLLRules	386
SetVBScriptRules	386
SetVBScriptRules2	388
StopConsolidation	390
Translate	390
Translate2	391
Chapter 11. HsvJournals Type Library	393
Journal Column Return Values	393
Template Column Return Values	394
HsvJournals Object Methods	395
ClosePeriod	395
GetItemID	396
GetJournalTemplateItemID	396
GetPeriodStatusList	397
IsPeriodOpen	398
OpenPeriod	398
IHsvJournalsEx Interface Methods	399
AddJournalGroup	399
ApproveJournals	399
DeleteJournals	400
DeleteTemplates	401
EnumJournalGroups	402
EnumJournalGroupsForScenarioYear	402
EnumJournalIDsForExtractFilter	403
GenerateRecurring	404
GetEntityJournals	405
GetJournal	406
GetJournal2	409

GetJournalDisplayData	411
GetJournalLabelsForIDs	412
GetJournalQueryDefinitionIDs	413
GetTemplate	415
GetTemplateDisplayData	417
GetTemplateLabelsForIDs	418
GetTemplateQueryDefinitionIDs	418
GetTextJournal	420
GetTextTemplate	422
GetVariance	425
PostJournals	426
RejectJournals	427
RemoveAllJournalGroups	428
RemoveJournalGroup	428
SaveJournal	428
SaveTemplate	431
SaveTextJournal	434
SaveTextTemplate	437
SubmitJournals	439
UnpostJournals	440
UnsubmitJournals	441
ValidateLineItems	442
ValidateValue	443
IHsvJournalsReport Interface Methods	444
Report Data Array Structures	444
GetReportData	447
GetReportData2	449
Chapter 12. HsvSecurityAccess Type Library	451
Access Rights Constants	451
Role ID Constants	452
Task Constants	452
E-mail Alerting Constants	452
User Groups - User Type Flag Constants	452
Identity Type Constants	453
Search Filters	453
Security Identifiers	454
The Connected User	454
HsvSecurityAccess Object Methods	454

AddApplicationAdministrator	455
AddApplicationAdministrator2	455
AddOrRemoveApplicationAdministrators	455
AddOrRemoveApplicationAdministrators2	455
AddOrRemoveRolesFromUser	456
AddOrRemoveRolesFromUser2	456
AddOrRemoveUsersFromRole	457
AddOrRemoveUsersFromRole2	457
AddSecurityClass	458
AddSecurityClassWithAccessCode	458
AddUser	458
AddUser2	458
AddUserEx	459
AddUserToRole	459
AddUserToRole2	459
AllowRulesLoadForEPMAApp	459
DeleteSecurityClass	459
DeleteSecurityClassWithAccessCode	460
EnumApplicationAdministrators	460
EnumApplicationAdministrators2	460
EnumRoles	460
EnumRolesForPrincipal	461
EnumRolesForUser	461
EnumSecurityClasses	462
EnumSecurityClassRightsForPrincipal	463
EnumUserClassAccess	464
EnumUsers	464
EnumUsers2	464
EnumUsers3	464
EnumUsersInGroup	465
EnumUsersInRole	465
EnumUsersInRole2	465
EnumUsersInRole3	466
EnumUsersInSecurityClass	466
EnumUsersInSecurityClass2	466
EnumUsersInSecurityClass3	467
EnumUsersOrGroups	467
EnumUsersWithFilter	467
EnumUsersWithFilter2	467

GenerateSecurityReportForBiPub	468
GetAllSecurityClassRightsForConnectedUser	468
GetApplicationAdministratorAccessForAllUsers	469
GetApplicationAdministratorAccessForAllUsers2	469
GetConnectedUser	469
GetConnectedUser2	469
GetIdentityTypes	470
GetNumRoles	470
GetOwner	470
GetRoleAccessForAllUsers	470
GetRoleAccessForAllUsers2	471
GetRoleID	471
GetRoleLabel	471
GetRulesMode	472
GetSecurityClassAccessForAllUsers	472
GetSecurityClassAccessForAllUsers2	472
GetSecurityClassID	472
GetSecurityClassLabel	473
GetSecurityClassRightsForConnectedUser	473
GetTaskAccessForConnectedUserFromList	474
GetUserAccessForAllRoles	474
GetUserAccessForAllRoles2	475
GetUserAccessForAllSecurityClasses	475
GetUserAccessForAllSecurityClasses2	475
GetUserID	475
GetUserIDFromSID	476
GetUserInfoFromUniqueID	476
GetUserInfoFromUniqueID2	476
GetUserName	477
GetUserName2	477
GetUserNameFromSID	477
GetUserSID	477
InsertDefaultSecurityClass	478
InsertDefaultSecurityClassWithAccessCode	478
IsApplicationAdministrator	478
IsClassicHFMAApplication	478
IsConnectedUserAllowedToPerformTask	478
IsConnectedUserInRole	479
IsValidWindowsUser	479

RemoveApplicationAdministrator	480
RemoveApplicationAdministrator2	480
RemoveUser	480
RemoveUser2	480
RemoveUserFromRole	481
RemoveUserFromRole2	481
RenameSecurityClass	481
SetManySecurityClassRightsForUser	481
SetManySecurityClassRightsForUser2	482
SetRolesForUser	482
SetRulesMode	483
SetSecurityClassLabel	483
SetSecurityClassLabelWithAccessCode	483
SetSecurityClassRightsForManyUsers	483
SetSecurityClassRightsForManyUsers2	483
SetSecurityClassRightsForUser	484
SetSecurityClassRightsForUser2	484
SetUserClassAccess	484
TakeOwnership	485
IHsvDataSecurity Interface Methods	485
GetCellLevelAccessRights	485
GetProcessUnitAccessRights	486
GetProcessUnitAccessRightsAndState	487
GetProcessUnitAccessRightsAndStateEx	487
GetProcessUnitAccessRightsEx	489
RefreshAccessRightsCache	490
Chapter 13. HsvSystemInfo Type Library	491
HsvSystemInfo Object Methods	491
AddRefToHsxServer	491
AddTaskToAudit	491
AddTaskToRunningTasks	492
AddTaskToRunningTasksAndUpdatePOV	492
CheckAccess	492
ClearAuditTasks	492
ClearAuditTasks2	493
ClearRunningTask	494
DeleteUserParameter	495
DisableNewConnections	495

EnableNewConnections	496
EnumActivityServers	496
EnumActivityUsers	496
EnumAuditTasks	497
EnumAuditTasks2	499
EnumProhibitConnections	501
EnumRunningTasks	502
EnumRunningTasksEx	505
EnumRunningTasksPOV	507
ExtractTaskAudit	509
GetActivityCodeDesc	510
GetActivityUserID	510
GetApplicationDirectory	510
GetApplicationName	511
GetCalcRulesType	512
GetCOMDLLCalcRules	512
GetCurrentActivity	512
GetExtractFileEncoding	512
GetFormattedDateTime	513
GetFormattedResourceString	513
GetKillUsersStatus	513
GetKillUserStatus	513
GetLanguageUserParameters	514
GetLastModifiedDateForArtifact	514
GetModuleName	514
GetNumberFormattingUserParameters	514
GetResourceLanguageUserParameters	515
GetResourceString	515
GetRunningTaskLogFilePathName	515
GetRunningTaskProgress	516
GetRunningTasksCount	516
GetRunningTaskStatus	516
GetServerName	517
GetUserName	517
GetUserParameter	518
GetVBScriptCalcRules	518
GetVBScriptMemberListRules	519
GetWorkingDirectory	520
IsScheduledTaskReadyToRun	520

KeepRunningTaskStillAlive	520
KillUsers	520
OutputSystemInfo	521
ReleaseHsxServer	521
SetApplicationDirectory	521
SetCOMDLLCalcRules	521
SetCurrentActivity	521
SetCurrentModule	521
SetCurrentModuleEx	522
SetExtractFileEncoding	522
SetLanguageUserParameters	522
SetNumberFormattingUserParameters	523
SetResourceLanguageForCurrentSession	523
SetResourceLanguageUserParameters	524
SetUserParameter	524
SetVBScriptCalcRules	525
SetVBScriptMemberListRules	527
StopRunningTask	529
UpdateRunningTaskLogFilePathName	529
UpdateRunningTaskPOV	529
UpdateRunningTaskProgress	529
UpdateRunningTaskProgressDetails	529
UpdateRunningTaskStatus	529
WarnUsersForShutDown	529

Chapter 14. HsvProcessFlow Type Library	531
Process Units versus Submission Phases	531
Approve	532
Approve2	533
ApproveEx	534
GetGroupPhaseFromCell	536
GetHistory	537
GetHistory2	538
GetPhasedSubmissionHistory	540
GetPhasedSubmissionState	541
GetPhasedSubmissionStateUsingPhaseID	542
GetState	543
PhasedSubmissionApprove	544
PhasedSubmissionApprove2	545

PhasedSubmissionApproveEx	547
PhasedSubmissionGetHistory2	548
PhasedSubmissionGetHistory2UsingPhaseID	549
PhasedSubmissionProcessManagementChangeStateForMultipleEntities2	551
PhasedSubmissionPromote	552
PhasedSubmissionPromote2	554
PhasedSubmissionPublish	555
PhasedSubmissionPublish2	556
PhasedSubmissionPublishEx	558
PhasedSubmissionReject	559
PhasedSubmissionReject2	560
PhasedSubmissionSignOff	561
PhasedSubmissionSignOff2	562
PhasedSubmissionStart	564
PhasedSubmissionStart2	565
PhasedSubmissionStartEx	567
PhasedSubmissionSubmit	569
PhasedSubmissionSubmit2	570
ProcessManagementChangeStateForMultipleEntities	571
ProcessManagementChangeStateForMultipleEntities2	572
ProcessManagementChangeStateForMultipleEntitiesEx	574
Promote	575
Promote2	576
Publish	578
Publish2	579
PublishEx	581
Reject	582
Reject2	583
SignOff	584
SignOff2	585
Start	586
Start2	588
StartEx	589
Submit	590
Submit2	591
Chapter 15. HsvReports Type Library	595
CheckReportSecurityClass	595
CheckReportSecurityClass2	596

CheckSecurityRole	596
DeleteDocuments	597
DeleteReport	597
DeleteReports	598
EnumDocuments	598
EnumDocumentsEx	600
EnumReports	602
GetDocument	603
GetDocumentEx	603
GetReport	604
SaveDocument	605
SaveDocumentEx	606
SetReport	607
Chapter 16. HsvMDArrays Type Library	609
Types of Indexes	610
HsvMDDataBuffer Object Methods	610
BeginEnumeration	611
CreateDataIndexList	611
CreateDataIndexListEx	613
EndEnumeration	613
EraseRecordFromPMBuffer	613
GetCheckLineItemDetailsForCaseInsensitiveDuplicates	613
GetCubeIndexFromPOV	614
GetCubePOVFromIndex	614
GetData	615
GetDataAtIndex	616
GetDescription	618
GetDescriptionAtIndex	619
GetLineItems	620
GetLineItemsAtIndex	621
GetNumCellsForData	622
GetNumCellsForDescriptions	623
GetNumCellsForLineItems	623
GetNumPeriodsInCubeForData	624
GetNumPeriodsInCubeForDescriptions	624
GetNumPeriodsInCubeForLineItems	625
GetPeriodIndexFromPOVForData	625
GetPeriodIndexFromPOVForDescriptions	626

GetPeriodIndexFromPOVForLineItems	627
GetPeriodPOVFromIndexForData	628
GetPeriodPOVFromIndexForDescriptions	629
GetPeriodPOVFromIndexForLineItems	629
GetPMErrorRecordCount	630
GetPMRecordCount	630
GetRecordFromPMBuffer	630
GetRecordFromPMErrorBuffer	630
GetSortedNature	630
InsertDataAtBeginning	630
InsertDataAtEnd	631
InsertDescriptionAtBeginning	632
InsertDescriptionAtEnd	633
InsertLineItemsAtBeginning	633
InsertLineItemsAtEnd	634
InsertRecordIntoPMBuffer	635
InsertRecordIntoPMErrorBuffer	635
RemoveAll	635
SetCheckLineItemDetailsForCaseInsensitiveDuplicates	635
SetData	636
SetDescription	637
SetGrowByAmount	638
SetLineItems	638
SetSortedNature	639
Sort	639
HsvMDDataBufferLite Object Methods	639
BeginEnumeration	640
CreateDataIndexList	641
EraseRecordFromPMBuffer	643
EndEnumeration	643
GetCheckLineItemDetailsForCaseInsensitiveDuplicates	643
GetCubeIndexFromPOV	643
GetCubePOVFromIndex	644
GetData	645
GetDataAtIndex	646
GetDescription	647
GetDescriptionAtIndex	648
GetLineItems	649
GetLineItemsAtIndex	650

GetNumCellsForData	652
GetNumCellsForDescriptions	652
GetNumCellsForLineItems	653
GetNumPeriodsInCubeForData	653
GetNumPeriodsInCubeForDescriptions	654
GetNumPeriodsInCubeForLineItems	654
GetPeriodIndexFromPOVForData	654
GetPeriodIndexFromPOVForDescriptions	656
GetPeriodIndexFromPOVForLineItems	657
GetPeriodPOVFromIndexForData	658
GetPeriodPOVFromIndexForDescriptions	658
GetPeriodPOVFromIndexForLineItems	659
GetPMErrorRecordCount	659
GetPMRecordCount	659
GetRecordFromPMBuffer	659
GetRecordFromPMErrorBuffer	659
GetSortedNature	659
InsertDataAtEnd	660
InsertDescriptionAtEnd	661
InsertLineItemsAtEnd	661
InsertRecordIntoPMBuffer	662
InsertRecordIntoPMErrorBuffer	662
RemoveAll	662
SetCheckLineItemDetailsForCaseInsensitiveDuplicates	663
SetData	663
SetDescription	664
SetGrowByAmount	665
SetLineItems	665
SetSortedNature	666
Sort	666
IHsvMDDataBufferLite Interface	667
SetMinCubesInCache	667
HsvTransactionData Object Methods	667
BeginDataEnum	668
BeginQueryEnum	669
EndDataEnum	669
EndQueryEnum	669
GetFixedDimensionMembers	669
GetQueryItem	669

GetTransactionData	669
Initialize	672
SetAccessRight	672
SetQueryItem	672
SetTransactionData	673
HsvMDIndexList Object Methods	673
GetItem	673
GetNumItems	674
HsvICTransactionsData Object Methods	674
AddAccountCustomCombination	675
AddICTransactionData	675
AddQueryDimensionMemberID	675
AddQueryField	675
AddQueryFieldItem	675
AddQueryOrderField	675
BeginDataEnum	676
BeginQueryFieldEnum	676
BeginQueryOrderEnum	676
EndDataEnum	676
EndQueryFieldEnum	677
EndQueryOrderEnum	677
GetAccessRights	677
GetAccountCustomCombination	677
GetAccountCustomCombinationTotal	677
GetEntityPartnerOption	677
GetErrorStatus	678
GetFilterOptions	678
GetFixedDimensionMembers	678
GetICTransactionCell	679
GetICTransactionData	679
GetMatchCode	681
GetNumTransactionsCached	681
GetPagingOption	682
GetPartnerAsEntityList	682
GetPartnerQueryDimensionMemberIDs	682
GetQueryDimensionMemberIDs	682
GetQueryFieldInformation	682
GetQueryFieldItem	682
GetQueryOrderField	682

GetTotalTransactions	682
GetTransGroupType	682
Initialize	682
InitializeSequenceIdMap	683
IsEntityInPartnerAsEntityList	684
IsICTransactionValid	684
RemoveQueryOrder	684
SetAccessRights	684
SetEntityPartnerOption	684
SetErrorStatus	685
SetFilterOptions	685
SetICTransactionData	685
SetPagingOption	685
SetPartnerAsEntityList	685
SetPartnerQueryDimensionMemberIDs	685
SetQueryDimensionMemberIDs	685
SetTotalTransactions	685
SortByCell	685
UnInitialize	685
Chapter 17. HsvDataCubes Type Library	687
About Subcube Items	687
HsvCurrencyCube Object Methods	688
BeginEnumerationOfStoredData	688
EndEnumerationOfStoredData	688
GetCell	689
GetFixedDimensionMembers	690
GetOneCellFromStoredItem	690
GetPOVFromStoredItem	692
HsvNodeCube Object Methods	693
BeginEnumerationOfStoredData	693
EndEnumerationOfStoredData	694
GetCell	694
GetFixedDimensionMembers	695
GetOneCellFromStoredItem	695
GetPOVFromStoredItem	697
Chapter 18. HsvStarSchemaACM Type Library	699
HsvStarSchemaACM Object Methods	699
CreateStarSchema	700

DeleteStarSchema	702
EnumApplicationStarSchemas	703
EnumRegisteredDSNs	703
GetAsynchronousTaskStatus	703
GetExtractLogData	704
GetPersistedSettings	705
QuitAsynchronousTask	705
SetDefaultTablePrefix	705
SetPersistedSettings	705
SetSession	705
TestSQLConnection	706
IHsvStarSchemaTemplates Interface	706
DeleteTemplate	706
EnumTemplates	706
GetTemplate	707
SetTemplate	707
Chapter 19. HsvICM Type Library	711
HsvICM Object Methods	711
AllMVTransPosted	711
CheckReportSecurity	712
CreateICTransaction	712
DeleteICReasonCode	714
DoesCellSupportICTransactionDetail	714
GetColumnFilter	715
GetCurrencyInfo	715
GetEntitiesContacts	716
GetEntityCurrencyID	716
GetICReasonCodeID	717
GetICReasonCodeLabel	717
GetICReasonCodes	717
GetICTransactions	718
GetMonitorICDetails	718
GetMonitorICSummary	719
GetMonitorICTransactions	720
GetRowFilter	721
GetTransCurrencyID	721
GetUnMatchTransactions	721
IsOneSideOfTransactionGroupWriteable	722

MatchAutoAccounts	723
MatchAutoIDs	724
NoTransMatchedOrPosted	725
ProcessAllICTrans	725
ProcessICTransactions	726
SaveColumnFilter	728
SaveICReasonCode	728
SaveICTransaction	728
SaveRowFilter	730
SetReasonCodeToICTransactions	730
UnMatchICTransactions	731
IHsvAdminICM Interface Methods	732
CheckSecurityForICExtract	732
CloseICPeriod	732
GetICEntitiesLockStatus	733
GetLockStatusICEntities	733
GetLockStatusICEntity	733
GetSettingsICPeriod	734
GetSettingsICPeriods	735
LoadICTransactions	737
LockICEntity	739
OpenICPeriod	739
OpenICPeriod2	740
SavePeriodsSettings	740
UnlockICEntity	740
UpdatePeriodSettings	741
UpdatePeriodSettings2	741
Chapter 20. Type Libraries for Loading and Extracting Information	743
Load and Extract Option Interfaces	743
IHsvLoadExtractOptions Interface Properties	744
IHsvLoadExtractOption Interface Properties	745
Common Properties and Methods	747
Common Load and Extract Properties	748
Common Load and Extract Methods	749
HsvSecurityLoadACV Type Library	752
IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvSecurityLoadACV	753
Loading Security Information	753
Extracting Security Information	754

Security Load Options	755
Security Extract Options	756
HsvSecurityLoadACV Object Properties	757
HsvSecurityLoadACV Object Methods	757
HsvMetadataLoadACV Type Library	758
IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvMetadataLoadACV	759
Loading Metadata	759
Extracting Metadata	760
Metadata Load Options	761
Metadata Extract Options	765
HsvMetadataLoadACV Object Properties	768
HsvMetadataLoadACV Object Methods	768
HsvRulesLoadACV Type Library	768
HsvRulesLoadACV Object Methods	769
HsvcDataLoad Type Library	775
IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvcDataLoad	775
Loading Data	776
Extracting Data	777
Data Load Options	778
Data Extract Options	781
HsvcDataLoad Object Properties	783
HsvcDataLoad Object Methods	783
HsvJournalLoadACV Type Library	784
IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvJournalLoadACV	785
Loading Journals	785
Extracting Journals	786
Journal Load Options	789
Journal Extract Options (Unfiltered)	789
Journal Extract Options (Filtered)	791
HsvJournalLoadACV Object Properties	793
HsvJournalLoadACV Object Methods	793
Chapter 21. HsvPOVSelection Type Library	795
HsvPOVSelection Object Methods	796
CheckAllItems	796
CheckAllItemsUsingSelectedList	796
EnableDimension	797

GetCheckedItems	798
GetListInfo	798
GetMember	799
GetNumCheckedItems	800
GetOBPInfo	801
Initialize	801
InitializeEx	802
InitializeHTTP	802
InitializeWithConnection	802
SelectDimension	802
SetCheckedItems	802
SetDimUnk	804
SetListInfo	804
SetMember	805
SetOBPInfo	805
SetResourceAndLanguage	806
ShowCheckBoxes	806
HsvPOVSelection Object Events	807
CheckBoxChanged	807
DimensionWasSelected	808
MultiCheckBoxChanged	809
SelectionChanged	809
Chapter 22. Error Handling and the HsvResourceManager Type Library	811
Error Handling with Financial Management	811
HsvResourceManager Object Methods	812
Handling Errors with the HsvResourceManager Object	812
System Message Detail Strings	813
GetAvailableLanguages	813
GetCurrentHSSRegistrationVersion	814
GetCurrentVersion	814
GetCurrentVersionInUserDisplayFormat	814
GetFormattedDateTime	815
GetFormattedDateTimeForLanguage	815
GetFormattedError	816
GetFormattedErrorWithLineFeed	817
GetFormattedResourceString	818
GetHelpDirectoryForLanguageID	818
GetHFMLanguageIdFromUserLanguages	818

GetLanguageCountryCodeFromLanguageId	819
GetLocaleIdFromLanguageId	819
GetResourceString	820
GetResourceStringFromHR	820
GetUserDisplayDataTimeFormats	820
GetUserLanguageFromHFMLanguageId	820
GetWindowsDateFormat	821
GetWindowsDateFormatForLocale	821
Initialize	821
Error Message Lookup Utility	822
Appendix A. Constants: The HFMConstants Type Library	823
Metadata-Related Constants	823
Account Dimension Constants	824
Custom Dimension Constants	827
Consolidation Method Constants	829
Currency Attribute Constants	831
Entity Dimension Constants	831
View Dimension Constants	832
Supported Language Constants	833
Dimension-Related Constants	833
Intercompany Partner Constants	835
Period Dimension Constants	836
Scenario Attribute Constants	837
Member List Constants	838
Value Dimension Constants	838
IHsvTreeInfo Interface Constants	840
Application Setting Attribute ID Constants	840
Cell Status Constants	841
Cell Calculation Status Constants	842
Cell Status Constants	842
Cell Metadata Status Constants	843
Subcube Period Calculation Status Constants	843
Transaction Dimension Constants	845
Cell Transaction Type Constants	846
Calculation Status Types	846
Additional Status Information Constant	847
Journal-Related Constants	847
Journal Action Constants	848

Balance Type Constants	848
Journal Column Display Constants	848
Debit/Credit Constants	849
Period Status Constants	849
Journal Report Display Option Constants	850
Journal Report Sort Option Constants	850
Journal Report Total Flag Constants	850
Journal Status Constants	851
Journal Type Constants	851
Journal Web Session Parameter Constants	851
Template Column Display Constants	851
Template Type Constants	852
Journal and Template Type Constants	852
Process Management Constants	853
Process Management Action Constants	853
Process Management Review Level Constants	853
Report Column Constants	854
Consolidation Type Constants	855
Tier Constants	855
Security Constants	856
Access Rights Constants	856
Role Constants	856
Task Constants	858
Web Constants	863
Data Explorer Task Constants	863
Data Explorer Process Management Constants	864
Data Grid Definition Constants	865
Data Grid Member Expansion Mode Constants	865
Data Grid Dimension Expansion Mode Constants	866
Data Grid Transaction Information Constants	866
Data Information Display Constants	866
Data Display Page Constant	867
Member Display Constants	867
Metadata Information Constants	867
Document Type Constants	868
Document File Type Constants	869
Extracted File Encoding Constants	869
User Activity Constants	870
Task Status Constants	872

Log Severity Constants	873
Intercompany Transaction Constants	873
Lock Status Constants	874
Period Status Constants	874
Match/Validate Before Post Constants	874
Match Option Constants	874
Posting Status Constants	875
Matching Status Constants	875
Processing Action Constants	875
Transaction Load Mode Constants	876
Filtering and Sorting Options	876
Process Status Constants	877
Event Constants	877
Miscellaneous Constants	877
Showing Public and Private Documents	878
Date and Time Format Constants	878
Invalid ID Constant	879
Maximum String Length Constants	879
Aggregation Type Constants	880
Member ID Range	881
Number Defaults Constants	881
Share Calculation Ownership Constants	881
Share Calculation Types Constants	881
Validation Type Constants	882
Default Security Class Constant	883
Module ID Constants	883
Appendix B. Change History	887
Changes for Release 11.1.3	887
HsvData Type Library	888
HsvJournals Type Library	888
HsvMetadata Type Library	888
HsvResourceManager Type Library	888
HsvSecurityAccess Type Library	888
HsvSystemInfo Type Library	889
HsxClient Type Library	889
HsxServer Type Library	889
Changes for Release 11.1.1	889
HsvCalculate Type Library	889

HsvDataCubes Type Library	890
HsvICM Type Library	890
HsvMDArrays Type Library	890
HsvMetadata Type Library	890
HsvResourceManager Type Library	890
HsvRulesLoadACV Type Library	890
HsvSecurityAccess Type Library	890
HsvSession Type Library	891
HsvSystemInfo Type Library	891
Changes for Release 9.3.1	891
HsvSession Type Library	891
HsvData Type Library	891
HsvSecurityAccess Type Library	891
HsvcDataLoad Type Library	892
Changes for Release 9.3.0.1	892
Classic Applications	892
HsxClient Type Library	893
HsxClientUI Type Library	893
HsvMetadata Type Library	893
HsvData Type Library	894
HsvSecurityAccess Type Library	894
HsvProcessFlow Type Library	895
HsvSystemInfo Type Library	895
HsvMDArrays Type Library	896
HsvICM Type Library	896
HsvMetadataLoadACV Type Library	896
HsvcDataLoad Type Library	897
HsvSecurityLoadACV Type Library	897
Changes for Release 4.1	897
HsxClient Type Library	897
HsxServer Type Library	898
HsvSession Type Library	898
HsvMetadata Type Library	898
HsvData Type Library	898
HsvCalculate Type Library	898
HsvSecurityAccess Type Library	899
HsvSystemInfo Type Library	900
HsvICM Type Library	900
HsvMDArrays Type Library	901

HsvResourceManager Type Library	902
Changes for Release 4.0	902
HsvSession Type Library	902
HsvMetadata Type Library	903
HsvData Type Library	903
HsvCalculate Type Library	904
HsvJournals Type Library	904
HsvSystemInfo Type Library	904
HsvProcessFlow Type Library	905
HsvReports Type Library	905
HsvStarSchemaACM Type Library	905
HsvICM Type Library	906
HsvMDArrays Type Library	906
HsvRulesLoadACV Type Library	906
HsvcDataLoad Type Library	907
HsvMetadataLoadACV Type Library	907
HsvResourceManager Type Library	907
HsxClient Type Library	907
Changes for Release 3.5.1	907
HsvMetadata Type Library	907
HsvData Type Library	908
HsvPOVSelection Type Library	908
HsvResourceManager Type Library	908
Documentation Errors	908
Changes for Release 3.5	908
HsxClient Type Library	909
HsvMetadata Type Library	909
HsvData Type Library	909
HsvSystemInfo Type Library	910
HsvStarSchemaACM Type Library	910
HsvReports Type Library	910
HsvMDArrays Type Library	910
HsvMetadataLoadACV Type Library	910
Changes for Release 3.4	910
HsxClient Type Library	911
HsxServer Type Library	911
HsvSession Type Library	911
HsvMetadata Type Library	911
HsvData Type Library	912

HsvJournals Type Library	912
HsvSystemInfo Type Library	913
HsvProcessFlow Type Library	913
HsvReports Type Library	913
HsvJournalLoadACV Type Library	913
HsvMDArrays Type Library	914
Changes for Release 3.0.4	914
HsxClient Type Library	914
HsxServer Type Library	914
HsvMetadata Type Library	915
HsvData Type Library	915
HsvSystemInfo Type Library	915
HsvReports Type Library	915
HsvRulesLoadACV Type Library	916
Changes for Release 3.0	916
HsxClient Type Library	916
HsxServer Type Library	917
HsvMetadata Type Library	917
HsvData Type Library	918
HsvSecurityAccess Type Library	918
HsvSystemInfo Type Library	918
HsvMDArrays Type Library	919
HsvMetadataLoadACV Type Library	919
HsvRulesLoadACV Type Library	919
Error Numbers	919
Changes for Release 2.2	919
HsxClient, HsxServer, and HsxClientUI References	920
HsvData Type Library	920
HsvCalculate Type Library	920
HsvMDArrays Type Library	921
Changes for Release 2.1	921
HsvData Type Library	921
HsvJournals Type Library	921
HsvReports Type Library	922
HsvMDArrays Type Library	922
HsvDataCubes Type Library	922
HsvRulesLoadACV Type Library	923
HsvSecurityAccess Type Library	923
HsxClient Type Library	923

Changes for Release 2.0	923
Security: HsxSecurity and HsvSecurityAccess Type Libraries	924
HsxClient Type Library	924
HsxClientUI Type Library	925
HsxServer Type Library	925
HsvSession Type Library	925
HFMConstants Type Library	925
HsvMetadata Type Library	925
HsvData Type Library	926
HsvCalculate Type Library	926
HsvJournals Type Library	927
HsvProcessFlow Type Library	927
HsvReports Type Library	927
HsvSystemInfo Type Library	928
HsvMDArrays Type Library	928
HsvRulesLoadACV Type Library	928
HsvcDataLoad Type Library	929
HsvJournalLoadACV Type Library	929
Changes for Release 1.2	929
HsvProcessFlow Type Library	929
HsvData Type Library	929
HsvMetadata Type Library	930
HsvReports Type Library	931
HsvMetadataLoadACV Type Library	931
Changes for Release 1.1	931
HsvMetadata Type Library	931
HsxSecurity Type Library	932
HsvReports Type Library	932
HsvMDArrays Type Library	932
HsvDataCubes Type Library	932
HsvMetadataLoadACV Type Library	933
HsvSecurityLoadACV Type Library	933
Index	935

1

Overview

In This Chapter

Prerequisites for Developers and Users.....	39
COM Data Type Mapping.....	40
Features Exposed by the Objects.....	40
Application Server Tier Type Libraries	41
Type Libraries for Loading and Extracting.....	42
Control for Selecting Dimension Members	42
Additional Architectural Considerations.....	43
About Subcubes	43
Data Across Multiple Application Servers: Subcube Caching.....	44
Values Returned in Arguments.....	45
Error Handling	45

You can use the objects described in this book to programmatically extend Oracle Hyperion Financial Management, Fusion Edition. This chapter describes various concepts that apply to the object model.

Prerequisites for Developers and Users

The following prerequisites apply to developers who want to use the object model:

- You must install the Financial Management Windows client on your computer.
- You must either install the Financial Management application server files on your computer or you must have read access to a file system on which the application server files are installed. Access to the application server files is required so that your projects can reference the object model's type libraries.

Users who will work with programs developed against the object model must install the Financial Management Windows client on their computers.

Note: The code examples are in Visual Basic 6 format. Using the example code in Visual Basic 2005 may yield errors.

COM Data Type Mapping

Method and property descriptions in this guide describe types with Visual Basic 6 data types. For users of languages other than Visual Basic 6, the following table maps the Visual Basic 6 types used in this guide to COM types:

Visual Basic 6 Types	COM Types
Boolean	VARIANT_BOOL
Byte	Byte
String	BSTR
Double	Double
Single	Float
Integer	short
Long	long
Object	IDispatch
Object	IUnknown
Variant (array)	Variant *
Variant	Variant

Features Exposed by the Objects

The objects covered in this book expose many Financial Management features, including the following features:

- Logging in
- Opening Financial Management applications
- Registering and unregistering servers
- Getting attributes of dimension members
- Getting and setting data for a cell
- Getting and setting data for arrays of cells
- Executing consolidations, translations, and calculations
- Opening and closing periods
- Processing journals
- Defining application security
- Process Management
- Loading and extracting member lists, rules, data, and journals

- Extracting metadata
- Extended Analytics
- A control that provides a user interface for selecting dimension members

Application Server Tier Type Libraries

After an HsvSession object is instantiated, the client can initiate tasks such as consolidations, data entry, and so on. When a task runs, subordinate objects of the HsvSession object are used on the application server tier.

The following table lists the type libraries that contain these subordinate objects and the features to which the libraries correspond.

Table 1 Features Exposed by HsvSession’s Subordinate Objects

Type Library	Corresponding Features
HsvMetadata	Metadata for an application; used to get dimension member properties.
HsvData	Data in an application; used to get and set data.
HsvCalculate	Calculations, consolidations, and translations.
HsvJournals	Journal-related features of Financial Management.
HsvSecurityAccess	Security definition features of Financial Management.
HsvSystemInfo	Various features; used to get and set application names, get the application server name, and get and set other system-related information.
HsvProcessFlow	Process Management-related features of Financial Management.
HsvReports	Reporting-related features of Financial Management.
HsvICM	Intercompany transaction features of Financial Management.
HsvMDArrays	Provides helper methods for working with data, transaction data generated by statutory consolidations, and intercompany transactions.
HsvDataCubes	Provides access to data at the subcube level. This library’s objects are subordinate to the HsvData object.

Note: The objects in these type libraries are introduced in [Chapter 2, “About the Type Libraries and Objects.”](#) The chapter includes a brief description of each object’s properties and methods.

The following figure illustrates the relationship between the application server type libraries.

Type Library Relationships—Application Server Libraries

Note: The HsvMDArrays type library is not directly related to these objects. The type library provides various helper methods, but technically is not a branch of the hierarchy pictured above. Also, the HsvStarSchemaACM library is omitted from the diagram because the HsvStarSchemaACM object is not a child of the HsvSession object; instead, HsvStarSchemaACM instances are associated with HsvSession instances using HsvStarSchemaACM.[SetSession](#).

Type Libraries for Loading and Extracting

Financial Management exposes objects for loading and extracting security information, member lists, rules, data, and journals, and for extracting metadata. These libraries expose client tier Application Components (ACVs) that transfer information between tiers by talking to application server tier Application Components (ACMs). Each ACV contains a `SetSession` method; `SetSession` returns an HsvSession object reference that points to the application for which information is to be loaded or extracted.

The following table lists the type libraries for loading and extracting information.

Table 2 Type Libraries for Loading and Extracting

Type Library	Corresponding Features
HsvSecurityLoadACV	Loading and extracting security information.
HsvMetadataLoadACV	Extracting metadata.
HsvRulesLoadACV	Loading and extracting member lists and rules.
HsvcDataLoad	Loading and extracting data.
HsvJournalLoadACV	Loading and extracting journals and journal templates.

Note: The objects in these type libraries are introduced in [Chapter 2, “About the Type Libraries and Objects.”](#) The chapter includes a brief description of each object’s properties and methods.

Control for Selecting Dimension Members

The HsvPOVSelection type library exposes a control that provides a graphical user interface for selecting members. This is a tabbed control that includes one tab per dimension. You use the [Initialize](#) method to associate the control with the HsvSession object reference that represents the application for which dimension members will be displayed.

Note: For more information, see [Chapter 21, “HsvPOVSelection Type Library.”](#)

Additional Architectural Considerations

Here are additional architectural issues to keep in mind when programming with the Financial Management objects:

- Financial Management is a session-based system, not a transaction-based system. A client remains logged onto an application server, enabling users to perform multiple transactions before logging off.
- Security and authentication is automatically applied by Financial Management. The objects enable you to programmatically *define* application security; however, as a programmer you do not need to do anything to *enforce* the security that has been defined for an application.
- Since an application can have multiple application servers, Financial Management automatically checks the data in the database before it updates a cell's data. This “synchronization” of data across application servers is automatically applied by Financial Management; you do not need to write any code to do this. For more information, see [“Data Across Multiple Application Servers: Subcube Caching” on page 44.](#)

About Subcubes

Several Financial Management 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 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 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 would store 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 would store the cells for the year 2002, Actual scenario, Connecticut child entity, UnitedStates parent entity, and [Contribution] Value dimension member.

Data Across Multiple Application Servers: Subcube Caching

Since an application can have multiple application servers, Financial Management keeps an application's data synchronized across servers by caching subcubes of data into RAM.

Whenever a cell is viewed or set by a client, the data for the cell's subcube is placed in the data cache, which is located in the application server's HsvDataSource process. When a user updates a cell's data, Financial Management performs the following process to make sure that data is kept in sync:

1. The subcube is updated and locked on the application server to which the client is connected.
2. The application server checks the time stamps of the subcube cells on the data server to see if the subcube has been updated by *another* application server since the current application server last cached the subcube.
3. If the subcube has been updated by another application server, the data cache is updated with the other application server's data.
4. The data cache is updated with the client's cell.
5. The database is updated.

Values Returned in Arguments

Most of the methods in this document return values in arguments passed by reference instead of in traditional return values on the left-hand side. This convention provides flexibility, allowing methods to return multiple values. To indicate how the Financial Management methods return values, the method descriptions follow these notational conventions:

- If a method is a function, its arguments are enclosed within parentheses; otherwise the method is a subroutine. Note that many Financial Management functions *also* return values in `ByRef` arguments.
- Visual Basic defaults to arguments passed by reference. If an argument overrides this default and is passed by value, the argument description will indicate this with the `ByVal` keyword; otherwise the argument is passed by reference.

Error Handling

The methods in this document return HRESULT error numbers that can be trapped in error handling. In Visual Basic, these error numbers are returned in the `Err` object's `Number` property.

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.

Note: When a method successfully executes, 0 is returned by `Err.Number`.

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 22, “Error Handling and the HsvResourceManager Type Library.”](#)

2

About the Type Libraries and Objects

In This Chapter

Type Library References	47
Objects, Properties, and Methods	49

This chapter explains the Financial Management type library references you are required to make in projects, and summarizes the objects, methods, and properties that these type libraries contain.

Type Library References

Financial Management's objects are contained by several type libraries. When creating a project, you must use Visual Basic's References dialog box to reference *each* type library that applies to the project. You do not have to reference all of the type libraries, only those that apply to the project.

You *must* reference the following type libraries in most projects. These type libraries contain objects for connecting to the client and application server tiers, and for opening applications:

- HsxClient
- HsxServer
- HsvSession

For the other type libraries, you must reference those libraries that contain methods or properties that you will use in a project. For example, if a project requires access to data and metadata, you must also reference the HsvMetadata and HsvData type libraries.

Note: There are certain HsxClient methods that do not require references to the HsxServer library.

The following table lists the type libraries described in this guide.

Table 3 Type Libraries

Type Library	Description
HsxClient	Provides client tier functionality such as logging on and opening applications.

Type Library	Description
HsxClientUI	Provides dialog boxes for some of the client tier functionality.
HsxServer	Gets application server tier information such as the server's Financial Management applications and data sources.
HsvSession	Provides access to Financial Management applications. Use HsvSession's properties to access subordinate objects such as HsvData and HsvMetadata.
HsvMetadata	Gets information about an application's metadata. HsvMetadata contains many objects, enabling you to get information about dimension members.
HsvData	Gets and sets data for an application.
HsvCalculate	Executes consolidations, calculations, and translations.
HsvJournals	Processes journals, opens and closes periods, and generates reports.
HsvSecurityAccess	Defines an application's security.
HsvSystemInfo	Gets and sets various types of system information such as application directories and server names.
HsvProcessFlow	Exposes the Process Management features, enabling you to take actions for and get histories of process units.
HsvReports	Gets report information, enumerates an application's reports, and saves and deletes reports.
HsvICM	Exposes the intercompany transaction features.
HsvMDArrays	Supplements the HsvData object by enabling you to manage data in arrays of cells, and also enables you to obtain the transaction data generated by statutory consolidations.
HsvDataCubes	Provides access to cell data and information at the subcube level.
HsvStarSchemaACM	Exposes the Extended Analytics features.
HsvSecurityLoadACV	Loads and extracts security settings, using load and extract files on client computers.
HsvMetadataLoadACV	Extracts metadata using extract files on client computers.
HsvRulesLoadACV	Loads and extracts rules and member lists, using load and extract files on client computers.
HsvcDataLoad	Loads and extracts data, using load and extract files on client computers.
HsvJournalLoadACV	Loads and extracts journals and journal templates, using load and extract files on client computers.
HsvPOVSelection	Exposes an insertable control that provides a user interface for selecting dimension members.
HsvResourceManager	Provides an interface to the Resource Manager. Note: 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.

Objects, Properties, and Methods

The following topics introduce the objects, properties, and methods for the type libraries provided by Financial Management. The following type libraries are described:

- [“HsxClient Type Library Overview” on page 49](#)
- [“HsxClientUI Type Library Overview” on page 52](#)
- [“HsxServer Type Library Overview” on page 53](#)
- [“HsvSession Type Library Overview” on page 54](#)
- [“HsvMetadata Type Library Overview” on page 55](#)
- [“HsvData Type Library Overview” on page 69](#)
- [“HsvCalculate Type Library Overview” on page 74](#)
- [“HsvJournals Type Library Overview” on page 75](#)
- [“HsvSecurityAccess Type Library Overview” on page 79](#)
- [“HsvSystemInfo Type Library Overview” on page 84](#)
- [“HsvProcessFlow Type Library Overview” on page 88](#)
- [“HsvReports Type Library Overview” on page 91](#)
- [“HsvMDArrays Type Library Overview” on page 93](#)
- [“HsvDataCubes Type Library Overview” on page 102](#)
- [“HsvStarSchemaACM Type Library Overview” on page 104](#)
- [“HsvICM Type Library” on page 105](#)
- [“Type Libraries for Loading and Extracting Information” on page 108](#)
- [“HsvPOVSelection Type Library” on page 115](#)
- [“HsvResourceManager Type Library Overview” on page 117](#)

HsxClient Type Library Overview

The HsxClient type library contains one object, the HsxClient object. The HsxClient object connects to applications and provides other client tier functionality.

Assign HsxClient object references with the `Set` keyword as shown in the following example:

```
Dim cHsxClient As HsxClient
Set cHsxClient = New HsxClient
```

Use the `SetLogonInfoSSO` and `OpenApplication` methods to log on and to connect to applications. Note that `OpenApplication` returns object references to the HsxServer and HsvSession objects.

Tip: The HsxClientUI type library’s methods provide dialogs that offer similar functionality to some of the HsxClient methods. The HsxClientUI methods are introduced in [“HsxClientUI Type Library Overview” on page 52](#).

The following table lists the HsxClient object's methods.

Table 4 HsxClient Object Methods

Method	Description
AuthenticateCSSToken	<i>For internal use.</i>
AuthenticateSecurityAgentCredentials	<i>For internal use.</i>
AuthenticateUserCredentials	<i>For internal use.</i>
AuthenticateUserOnCluster	<i>For internal use.</i>
AuthenticateUserOnClusterSSO	<i>For internal use.</i>
AuthenticateUserOnServer	<i>For internal use.</i>
CreateApplication	<i>Deprecated.</i>
CreateApplicationCAS	Creates a Classic application.
CreateApplicationCASWithAccessCode	<i>For internal use.</i>
CreateObjectOnCluster	Instantiates an object on a cluster or application server.
CreateObjectOnServer	<i>Deprecated - use CreateObjectOnCluster.</i>
DeleteApplication	Deletes a Classic application.
DeleteApplicationWithAccessCode	<i>For internal use.</i>
DeleteSystemErrors	Deletes records of system errors for a cluster.
DetermineWindowsLoggedOnUser	Returns the domain name and the username of the user who is logged onto Windows.
DisableNewConnections	Disables new Financial Management connections to a cluster for the specified application and application server criteria.
DoesUserHaveCreateApplicationRights	Indicates whether the connected user is a member of the Creator group for a given application server cluster.
DoesUserHaveSystemAdminRights	Indicates whether the connected user is a member of the Administrator group for a given application server cluster.
EnableNewConnections	Enables new Financial Management connections to a cluster for the specified application and application server criteria.
EnumProhibitConnections	Returns information on the applications, application servers, and users for which connections have been disabled on a given cluster.
EnumProvisioningProjects	Returns the names of the provisioning projects for Oracle's Hyperion® Shared Services associated with a given cluster.
EnumRegisteredClusterNames	Returns an array containing the names of the clusters or application servers that have been registered for the client.

Method	Description
EnumRegisteredServerNames	<i>Deprecated</i> - use EnumRegisteredClusterNames.
EnumUserAppPreferences	<i>For internal use.</i>
EnumUsersOnSystem	Returns the usernames of and other information applicable to the users logged on to a given cluster.
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.
EnumUsersOnSystemEx2	Returns the usernames and other information applicable to all users logged on to / logged out from a given cluster.
GetApplicationFolder	Returns the path of an application's local storage folder.
GetClusterInfo	Returns the name of the cluster to which the specified application server is assigned.
GetHFMErrLogRecordSet	<i>For internal use.</i>
GetLogonInfo	<i>Deprecated</i> - use GetLogonInfoSSO.
GetLogonInfoSSO	Gets the domain, username, and external authentication token for the connected user.
GetServer	<i>Deprecated</i> - use GetServerOnCluster.
GetServerOnCluster	Returns an object reference to the HsxServer object that represents the specified cluster or server.
GetSSOTokenUsingWebSecurityAgentCredentials	<i>For internal use.</i>
GetWebSecurityAgentSettings	<i>For internal use.</i>
IsValidApplication	Indicates whether the specified application exists on the specified cluster.
KillUsers	Logs off users from a cluster.
OpenApplication	Opens an application. OpenApplication also sets object references to the HsxServer and HsvSession objects.
RegisterApplicationCAS	Registers a Classic application with Shared Services.
RegisterApplicationCASWithAccessCode	<i>For internal use.</i>
RegisterCluster	Registers a cluster or application server for a client.
RegisterServer	<i>Deprecated</i> - use RegisterCluster.
ScriptableEnumRegisteredClusterNames	<i>For internal use.</i>
ScriptableEnumRegisteredServerNames	<i>For internal use.</i>
ScriptableGetLogonInfoSSO	<i>For internal use.</i>

Method	Description
ScriptableOpenApplication	<i>For internal use.</i>
SetApplicationFolder	Sets an application's local storage folder.
SetLogonInfo	<i>Deprecated</i> - use SetLogonInfoSSO.
SetLogonInfoSSO	Sets logon information such as the user's name, password, domain or sets an SSO token.
UnregisterAllClusters	Unregisters all the clusters or application servers that are registered for the client.
UnregisterCluster	Unregisters the specified cluster or application server.
UnregisterServer	<i>Deprecated</i> - use UnregisterCluster.
WarnUsersForShutDown	<i>For internal use.</i>

For detailed descriptions of these methods, see [“HsxClient Object Methods” on page 123](#).

HsxClientUI Type Library Overview

The HsxClientUI type library contains one object, the HsxClientUI object. The HsxClientUI object exposes standard Financial Management dialog boxes for client tier services such as logging on, opening applications, and registering and unregistering application servers.

Assign HsxClientUI object references with the `Set` keyword as shown in the following example:

```
Dim cHsxClientUI As HsxClientUI
Set cHsxClientUI = New HsxClientUI
```

After setting the object reference, call `Initialize` *before* calling any of the HsxClientUI object's other methods.

Tip: The HsxClientUI methods are similar to some of the HsxClient object's methods. The HsxClient methods are listed in [“HsxClient Type Library Overview” on page 49](#).

The following table lists the HsxClientUI object's methods.

Table 5 HsxClientUI Object Methods

Method	Description
DeleteApplication	Displays the Delete Application dialog box.
GetServer	<i>For internal use.</i>
Initialize	Provides the HsxClientUI object with access to the client layer. You must call <code>Initialize</code> before calling any of the other HsxClientUI methods.

Method	Description
Logon	Displays the Logon dialog box.
OpenApplication	Displays the Open Application dialog box.
RegisterServer	Displays the Register Server dialog box.

For detailed descriptions of these methods, see [“HsxClientUI Object Methods” on page 147](#).

HsxServer Type Library Overview

The HsxServer type library contains one object, the HsxServer object. The HsxServer object is used by clients to obtain connections to applications and to return lists of the applications registered on application servers. The HsxServer object also provides access to other application server functionality; for example, the `EnumRegisteredDSNs` method returns the names of the Data Source Names registered on an application server.

To assign HsxServer object references, use the HsxServer object reference returned by the HsxClient methods [OpenApplication](#) or [GetServerOnCluster](#) or the HsxClientUI method [OpenApplication](#).

The following table lists the methods of the HsxServer object, and [“HsxServer Object Property” on page 54](#) describes the HsxServer object’s one property.

Table 6 HsxServer Object Methods

Method	Description
DeleteSystemErrors	<i>For internal use.</i>
EnumDataSources	Returns the names and descriptions of the applications on an application server.
EnumRegisteredDSNs	Returns an array of the registered Data Source Names on the application server.
GetClustersAndServers	Returns the names of the clusters and servers associated with an HsxServer object reference.
GetDataSource	<i>For internal use.</i>
GetFileTransfer	<i>For internal use</i>
GetHFMErrorsLogRecordSet	<i>For internal use.</i>
GetSystemDataLinkFile	Returns the name and path of the data link file for the application server.
GetSystemFolder	Returns the name and path of the system file for the application server.
GetXMLErrorFromDatabase	Returns the message for a system error, given the system error’s reference number.
GetXMLErrorsListFromDatabase	Returns the reference numbers, log types, timestamps, application server names, and application names of system errors. You can filter the errors to be returned by date range, application server name, and application name.

Method	Description
ScriptableEnumDataSources	<i>For internal use.</i>
ScriptableEnumRegisteredDSNs	<i>For internal use.</i>

For detailed descriptions of these methods, see [“HsxServer Object Methods” on page 151](#).

HsxServer Object Property

The HsxServer object contains the `CSSEnabled` property, which indicates whether external authentication is enabled for the cluster.

HsvSession Type Library Overview

The HsvSession type library contains one object, the HsvSession object. The HsvSession object provides properties for accessing subordinate objects, such as the objects used for accessing metadata, data, journals, security, and so on. The HsvSession object also provides a method for instantiating objects such as server-side Application Components (ACMs) on application servers.

HsvSession object references are returned by the `OpenApplication` methods of the HsxClient and HsxClientUI objects. For details and an example, see [“OpenApplication” on page 140](#).

The following table lists the properties of the HsvSession object, and [Table 8](#) lists the object’s methods.

Table 7 HsvSession Object Properties

Property	Description
Calculate	Returns an HsvCalculate object reference.
Data	Returns an HsvData object reference.
ICM	Returns an object reference to the HsvICM object or the IHsvAdminICM interface.
Journals	Returns an HsvJournals, IHsvJournalsEx, or IHsvJournalsReport object reference.
Metadata	Returns an HsvMetadata object reference.
ProcessFlow	Returns an HsvProcessFlow object reference.
Reports	Returns an HsvReports object reference.
Security	Returns an HsvSecurityAccess or IHsvDataSecurity object reference.
SystemInfo	Returns an HsvSystemInfo object reference.

The HsvSession object also contains the following methods.

Table 8 HsvSession Object Methods

Method	Description
CreateObject	Instantiates an object on the application server on which an application has been opened. Use <code>CreateObject</code> to instantiate server-side objects that you develop.
GetLicenseExpirationStatus	<i>Deprecated.</i>
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, <code>HasSystemChanged</code> returns TRUE if a metadata file has been loaded since the last consolidation.
HasUserStatusChanged	Indicates whether the connected user has been logged off by an administrator.
IsBusy	<i>For internal use.</i>
IsRunningTasks	<i>For internal use.</i>
LockMetadataLoadWithSystemChangeCheck	<i>For internal use.</i>
UnlockMetadataLoad	<i>For internal use.</i>

For detailed descriptions of these properties and methods, see [Chapter 7, “HsvSession Type Library.”](#)

HsvMetadata Type Library Overview

The HsvMetadata type library contains several objects. The top-level object in this type library is the HsvMetadata object, which is a child object of the HsvSession object. The HsvMetadata object has several child objects, and contains properties that return object references to these objects. The HsvMetadata object also contains methods for extracting metadata and for returning various types of metadata-related information.

Most of the HsvMetadata object's child objects represent the Financial Management dimensions; these dimension objects are listed in [Table 9](#). The default interface for a dimension object contains the methods, if any, that specifically apply to the dimension. For example, the HsvAccounts object's default interface contains methods that return attributes of Account dimension members.

In addition, each dimension object implements the IHsvTreeInfo interface. The IHsvTreeInfo interface gets IDs and labels of dimension members, as well as parent-child information for Entity dimension members.

The HsvMetadata type library also contains the HsvCurrencies object, which represents an application's currencies.

The following table lists the dimension-related objects in the HsvMetadata type library.

Table 9 HsvMetadata Type Library – Dimension Objects

Object	Description
HsvAccounts	Returns attributes of Account dimension members, and implements the IHsvTreeInfo interface for the Account dimension.
HsvCustom	Returns attributes of the four Custom dimensions, and implements the IHsvTreeInfo interface for the Custom dimensions.
HsvEntities	Returns attributes of Entity dimension members, and implements the IHsvTreeInfo interface for the Entity dimension.
HsvICPs	Implements the IHsvTreeInfo interface for the Intercompany Partner dimension.
HsvPeriods	Returns information about periods and frequencies, and implements the IHsvTreeInfo interface for the Period dimension.
HsvScenarios	Returns attributes of Scenario dimension members, and implements the IHsvTreeInfo interface for the Scenario dimension.
HsvValues	Implements the IHsvTreeInfo interface for the Value dimension.
HsvViews	Implements the IHsvTreeInfo interface for the View dimension.
HsvYears	Provides a method that returns an application's valid range of years, and implements the IHsvTreeInfo interface for the Year dimension.

Note: The default interface for these objects is simply the object name prefixed with I; for example, the default interface for HsvAccounts is the IHsvAccounts interface. This is an irrelevant consideration for Visual Basic, but applies to other languages such as C++.

HsvMetadata Type Library Object Hierarchy

The following diagram illustrates the relationships between the higher-level objects and the HsvMetadata type library's objects.

HsvMetadata Object Hierarchy

HsvMetadata Object Properties and Methods

The HsvMetadata object provides properties that access subordinate objects for the various Financial Management dimensions, and methods that return metadata-related information and that load and extract metadata files on application servers. Assign HsvMetadata object references with the HsvSession object's `Metadata` property; for more information, see [“Metadata” on page 158](#).

The following table lists the HsvMetadata object's properties, and [Table 11](#) lists the object's methods.

Table 10 HsvMetadata Object Properties

Property	Description
Accounts	Sets object references to the HsvAccounts object, and also sets references to the IHsvTreeInfo interface for use with the Account dimension.
Currencies	Sets object references to the HsvCurrencies object.
Custom1	Sets object references to the HsvCustom object and the IHsvTreeInfo interface for use with the Custom 1 dimension.
Custom2	Sets object references to the HsvCustom object and the IHsvTreeInfo interface for use with the Custom 2 dimension.
Custom3	Sets object references to the HsvCustom object and the IHsvTreeInfo interface for use with the Custom 3 dimension.
Custom4	Sets object references to the HsvCustom object and the IHsvTreeInfo interface for use with the Custom 4 dimension.
Dimension	Sets object references to the IHsvTreeInfo interface. This property takes an integer parameter that identifies the dimension to which the object variable applies.
Entities	Sets object references to the HsvEntities object, and also sets references to the IHsvTreeInfo interface for use with the Entity dimension.
ICPs	Sets object references to the HsvICPs object, and also sets references to the IHsvTreeInfo interface for use with the Intercompany Partner dimension.
Periods	Sets object references to the HsvPeriods object, and also sets references to the IHsvTreeInfo interface for use with the Period dimension.
Scenarios	Sets object references to the HsvScenarios object, and also sets references to the IHsvTreeInfo interface for use with the Scenario dimension.
Values	Sets object references to the HsvValues object, and also sets references to the IHsvTreeInfo interface for use with the Value dimension.
Views	Sets object references to the HsvViews object, and also sets references to the IHsvTreeInfo interface for use with the View dimension.
Years	Sets object references to the HsvYears object, and also sets references to the IHsvTreeInfo interface for use with the Year dimension.

For detailed descriptions of these properties, see [“HsvMetadata Object Properties” on page 185](#).

The following table lists the HsvMetadata object’s methods:

Table 11 HsvMetadata Object Methods

Method	Description
ConsolidationMethodIDs	Returns IDs that identify an application’s consolidation methods.

Method	Description
EnumExtractOptions	Returns a two-dimensional array of the metadata extract options that can be passed to <code>Extract</code> . The array includes the options' names and default values.
EnumLanguages	Returns the numeric IDs and labels of an application's languages.
EnumLoadOptions	Returns a two-dimensional array of the metadata load options that can be passed to <code>Load</code> .
Extract	Extracts metadata into a text file. The file will be created on the application server.
GetApplicationAttribute	Returns the raw value of a given application setting attribute.
GetApplicationCurrency	Returns the label of the application's default currency.
GetApplicationSettingsTimeStamp	Returns a timestamp that indicates when the application settings were last updated.
GetByIndexValidationAccount	Returns the member ID of a validation account, given the index of the account.
GetCellLevelAccountType	Returns the account type of a cell, given the member IDs of the cell's Account and Custom dimension members.
GetConsolidationMethodDescription	Returns a consolidation method's description for the specified language.
GetConsolidationMethodInfo	Returns the attributes of a consolidation method, given the method's ID.
GetConsolidationMethodsTimeStamp	Returns a timestamp that indicates when the application's consolidation methods were last updated.
GetCurrencyTimeStamp	Returns a timestamp that indicates when the application's currencies were last updated.
GetCurrencyValueIDForEntityValueCombination	Returns the currency value dimension ID for the specified parent, entity, and value triplet.
GetCurrencyValueIDsForEntityValueCombinations	Returns the currency value dimension IDs for the specified parent, entity, and value triplet.
GetDefaultValueOfActiveStatusAccount	Returns the default value for an application's active status account.
GetFdmAppName	Returns the value of the FDM Application Name setting for the application.
GetFrequencyID	The frequency (YTD, MTD, QTD, and so forth) for which to return an internal numeric ID.
GetICPEntitiesAggregationWeight	<i>For internal use.</i>
GetSupportSubmissionPhaseForAccountFlag	Indicates whether phased submissions are enabled for the Account dimension.

Method	Description
GetSupportSubmissionPhaseForCustom1Flag	Indicates whether phased submissions are enabled for the Custom1 dimension.
GetSupportSubmissionPhaseForCustom2Flag	Indicates whether phased submissions are enabled for the Custom2 dimension.
GetSupportSubmissionPhaseForCustom3Flag	Indicates whether phased submissions are enabled for the Custom3 dimension.
GetSupportSubmissionPhaseForCustom4Flag	Indicates whether phased submissions are enabled for the Custom4 dimension.
GetSupportSubmissionPhaseForCPFlag	Indicates whether phased submissions are enabled for the Intercompany Partner dimension.
GetUseSubmissionPhaseFlag	Indicates whether phased submissions are enabled for the application.
GetValidationAccount	Returns the member ID of an application's Validation Account setting.
IsCustomMemberValidForAccount	Indicates whether a Custom dimension member is valid for an account.
IsOrgByPeriodApplication	Indicates whether the OrgByPeriodApplication application setting is on or off.
Load	Loads metadata into a Classic application, using a load file on the application server.
LoadWithAccessCode	<i>For internal use.</i>
TranslateApplicationAttributeForDisplay	Returns a user-readable string that represents the value of a given application setting attribute.

For detailed descriptions of these methods, see [“HsvMetadata Object Methods” on page 162](#).

HsvAccounts Object Overview

The HsvAccounts object provides access to some of the attributes of Account dimension members. Assign HsvAccounts object references with the `Accounts` property; for more information, see [“Accounts” on page 185](#).

The following table lists the HsvAccounts object's methods.

Table 12 HsvAccounts Object Methods

Method	Description
GetAccountType	Indicates an account's type.
GetCalcAttribute	Returns the value of a given Account dimension member's CalcAttribute attribute.

Method	Description
GetICPTopMember	Returns the member ID of the Intercompany Partner dimension member that has been assigned as an Account dimension member's ICPTopMember attribute.
GetIsICP	Indicates the value assigned to the IsICP attribute of an Account dimension member.
GetNumDecimalPlaces	Returns the maximum number of digits to the right of the decimal point that an account supports.
GetPlugAccount	Returns the member ID of an account's plug account.
GetSecurityClassID	Returns the ID of the security class that has been assigned to an account, given the account's member ID.
GetSubmissionGroup	Returns the value of an account's Submission Group property.
GetTopMemberOfValidCustom1Hierarchy	Returns the member ID of the top member of an account's Custom 1 dimension hierarchy.
GetTopMemberOfValidCustom2Hierarchy	Returns the member ID of the top member of an account's Custom 2 dimension hierarchy.
GetTopMemberOfValidCustom3Hierarchy	Returns the member ID of the top member of an account's Custom 3 dimension hierarchy.
GetTopMemberOfValidCustom4Hierarchy	Returns the member ID of the top member of an account's Custom 4 dimension hierarchy.
GetUserDefined1	Returns an account's UserDefined1 attribute, given the account's member ID.
GetUserDefined2	Returns an account's UserDefined2 attribute, given the account's member ID.
GetUserDefined3	Returns an account's UserDefined3 attribute, given the account's member ID.
GetXBRLTags	Returns an account's XBRLTags attribute, given the account's member ID.
IsCalculated	Indicates whether an account's data is calculated by Financial Management or is manually entered.
IsConsolidated	Returns a Boolean that indicates whether an account's data is consolidated to parent entities.
IsCustom1AggregationEnabled	Returns the value to which an account's EnableCustom1Aggr attribute has been set.
IsCustom2AggregationEnabled	Returns the value to which an account's EnableCustom2Aggr attribute has been set.
IsCustom3AggregationEnabled	Returns the value to which an account's EnableCustom3Aggr attribute has been set.
IsCustom4AggregationEnabled	Returns the value to which an account's EnableCustom4Aggr attribute has been set.

Method	Description
IsICP	Returns a Boolean that indicates whether the account is used in intercompany transactions.
IsICPRestricted	Indicates whether an Account dimension member is restricted from having Intercompany Partner transactions with itself.
UsesLineItems	Indicates whether an account allows line items.

For detailed descriptions of these methods, see [“HsvAccounts Object Methods” on page 232](#).

HsvCustom Object Overview

The HsvCustom object provides access to attributes of the four Custom dimensions. Assign HsvCustom object references with the `Custom1`, `Custom2`, `Custom3`, and `Custom4` properties; for more information, see [“HsvCustom Object Methods” on page 244](#).

The following table lists the methods of the HsvCustom object.

Table 13 HsvCustom Object Methods

Method	Description
GetAggregationWeight	<i>For internal use.</i>
GetSecurityClassID	Returns the ID of the security class assigned to a Custom 1, 2, 3, or 4 dimension member.
GetSubmissionGroup	Returns the value of a Custom dimension member’s Submission Group property.
GetUserDefined1	Returns the value assigned to the UserDefined1 attribute of a Custom 1, 2, 3, or 4 dimension member.
GetUserDefined2	Returns the value assigned to the UserDefined2 attribute of a Custom 1, 2, 3, or 4 dimension member.
GetUserDefined3	Returns the value assigned to the UserDefined3 attribute of a Custom 1, 2, 3, or 4 dimension member.
IsCalculated	Returns a Boolean that indicates whether a Custom dimension member’s data is calculated by Financial Management or is manually entered.
IsSwitchSignEnabledForFlow	Indicates whether a Custom dimension member’s SwitchSignForFlow attribute is enabled.
IsSwitchTypeEnabledForFlow	Indicates whether a Custom dimension member’s SwitchTypeForFlow attribute is enabled.

HsvEntities Object Overview

The HsvEntities object provides access to some of the attributes of Entity dimension members. Assign HsvEntities object references with the `Entities` property; for more information, see [“Entities” on page 187](#).

The following table lists the methods of the HsvEntities object.

Table 14 HsvEntities Object Methods

Method	Description
GetAllowAdjustments	Returns a Boolean that indicates whether an entity allows journal entries.
GetAllowAdjustmentsFromChildren	Returns a Boolean that indicates whether an entity allows journal postings from its children.
GetDefaultValueID	Returns the member ID of the Value dimension member that represents an entity's default currency.
GetHoldingCompany	Returns the member ID of an entity's HoldingCompany attribute.
GetSecurityAsPartnerID	Returns the value assigned to the SecurityAsPartner attribute of an Entity dimension member.
GetSecurityClassID	Returns the ID number of an entity's security class.
GetUserDefined1	Returns the value that has been defined for an entity's UserDefined1 attribute.
GetUserDefined2	Returns the value that has been defined for an entity's UserDefined2 attribute.
GetUserDefined3	Returns the value that has been defined for an entity's UserDefined3 attribute.
IsChild	Indicates whether an entity is a child of another entity.
IsDescendant	Returns a Boolean that indicates whether an entity is a descendant of another entity.
IsICP	Returns a Boolean that indicates whether an entity is an intercompany entity.
IsOrgByPeriodFilteringOn	Returns the current state of organization-by-period filtering.
IsSecurityAsPartnerEnabled	Indicates whether metadata security using entities' SecurityAsPartner attributes is enabled.

For detailed descriptions of these methods, see [“HsvEntities Object Methods” on page 247](#).

HsvICPs Object Overview

The HsvICPs object provides access to some of the attributes of Intercompany Partner dimension members. Assign HsvICPs object references with the ICPs property; for more information, see [“ICPs” on page 187](#).

The following table lists the methods of the HsvICPs object.

Table 15 HsvICPs Object Methods

Method	Description
GetSecurityClassID	Returns the ID of the security class that has been assigned to an Intercompany Partner.

Method	Description
GetSubmissionGroup	Returns the value of an Intercompany Partner dimension member's Submission Group property.

HsvPeriods Object Overview

The HsvPeriods object provides access to some of the frequency-related attributes of Period dimension members. Assign HsvPeriods object references with the `Periods` property; for more information, see [“Periods” on page 187](#).

The following table lists the methods of the HsvPeriods object.

Table 16 HsvPeriods Object Methods

Method	Description
GetBaseFrequency	Returns the internal ID of an application's base frequency.
GetFrequency	Returns the ordinal position of a period within an application's default frequency, and also returns the ID of the default frequency.
GetNumPeriodsInFrequency	Returns the number of periods in a frequency.
GetPeriodFromFrequency	Returns the member ID of the period that is at a given ordinal position within a frequency.
GetPeriodsInFrequency	Returns an array containing the member IDs of a frequency's Period dimension members.

For detailed descriptions of these methods, see [“HsvPeriods Object Methods” on page 256](#).

HsvScenarios Object Overview

The HsvScenarios object provides access to some of the attributes of Scenario dimension members. Assign HsvScenarios object references with the `Scenarios` property; for more information, see [“Scenarios” on page 187](#).

The following table lists the methods of the HsvScenarios object.

Table 17 HsvScenarios Object Methods

Method	Description
GetPhasedSubmissionStartYear	Returns the Phased Submission start year for the scenario.
EnumPhasedSubmissionStartYears	Returns an array of Scenario IDs and their corresponding Phased Submission start years.
GetDefaultFrequency	Returns the number that identifies the default frequency of a Scenario dimension member.
GetDefaultView	Returns the number that identifies the default view of a Scenario dimension member.

Method	Description
GetMaximumReviewLevel	Returns the value assigned to the MaximumReviewLevel attribute of a Scenario dimension member.
GetMissingDataZeroViewForAdjValues	Returns the member ID of the View dimension member that has been assigned as the ZeroViewForAdj attribute of a given Scenario dimension member.
GetMissingDataZeroViewForNonAdjValues	Returns the member ID of the View dimension member that has been assigned as the ZeroViewForNonadj attribute of a given Scenario dimension member.
GetSecurityClassID	Returns the ID of the security class that has been assigned to a scenario, given the scenario's member ID.
GetUserDefined1	Returns the text stored in the UserDefined1 attribute of a Scenario dimension member.
GetUserDefined2	Returns the text stored in the UserDefined2 attribute of a Scenario dimension member.
GetUserDefined3	Returns the text stored in the UserDefined3 attribute of a Scenario dimension member.
IsConsolidateYTD	Indicates whether a given Scenario dimension member supports year-to-date consolidation.
IsPhasedSubmissionEnabled	Indicates whether Phased Submissions is enabled for a given Scenario and Year.
SupportsAccountAllocations	<i>For internal use.</i>
SupportsEmailAlerting	Indicates whether E-mail alerting is enabled for a given Scenario dimension member.
SupportsProcessFlow	Indicates whether Process Management is enabled for a given Scenario dimension member.
SupportsTargetSetting	<i>For internal use.</i>
SupportsTimePeriodAllocations	<i>For internal use.</i>
UsesLineItems	Indicates whether a Scenario dimension member supports line items.

For detailed descriptions of these methods, see [“HsvScenarios Object Methods” on page 259](#).

HsvValues Object Overview

The HsvValues object provides information on Value dimension members. Assign HsvValues object references with the [Values](#) property.

The following table lists the methods of the HsvValues object. For detailed descriptions of these methods, see [“HsvValues Object Methods” on page 268](#).

Table 18 HsvValues Object Methods

Method	Description
GetCurrencyIDFromValueID	Returns a currency ID for a given Value dimension member ID.
GetValueIDFromCurrencyID	Returns the member ID of a Value dimension member that corresponds to a given currency.

HsvViews Object Overview

This object has no member properties or methods; however, like the other dimension objects, it implements the IHsvTreeInfo interface.

Assign HsvViews object references with the `Views` property; for more information, see [“Views” on page 188](#).

HsvYears Object Overview

The HsvYears object provides the `GetYearRange` method, which returns an application’s valid range of years. For details, see [“HsvYears Object Methods” on page 269](#).

Assign HsvYears object references with the `Years` property; for more information, see [“Years” on page 188](#).

IHsvTreeInfo Interface Overview

The IHsvTreeInfo interface is implemented in all of the dimension objects listed in [Table 9 on page 56](#). This interface provides access to internal IDs and labels of dimension members, and also gets information about dimension hierarchies.

To assign IHsvTreeInfo object references, use the HsvMetadata object property for the applicable dimension; these properties are summarized in [Table 10 on page 57](#). For an example of how to set IHsvTreeInfo object references, see [“IHsvTreeInfo Interface Methods” on page 188](#).

The following table summarizes the IHsvTreeInfo interface’s methods.

Table 19 IHsvTreeInfo Interface Methods

Method	Description
EnumAllMemberIDs	Returns numeric IDs that identify all the members of a dimension.
EnumAllMemberLabels	Returns the labels of all the members of a dimension.
EnumAllParentAndChildIDs	Returns arrays of IDs that represent the parent-child relationships of a dimension’s members.
EnumAllParentAndChildLabels	Returns arrays that contain member labels and that represent the parent-child relationships of a dimension’s members.
EnumAncestors	Returns an array containing the member IDs of a given member’s ancestors.

Method	Description
EnumBaseMemberIDs	Returns an array containing the member IDs of a given parent's base-level members. You can also use <code>EnumBaseMemberIDs</code> to get all of a dimension's base-level members.
EnumDefaultAncestors	Returns the member IDs of a given member's default ancestors.
EnumDefaultAncestorsLabels	Returns the labels of a given member's default ancestors.
EnumDescendants	Returns an array containing the member IDs of a given member's descendants.
EnumIDsOfChildren	Returns either the top members of a dimension hierarchy or the child members of a parent member.
EnumIDsOfChildren	Returns children and the parent item of the requested children.
EnumMemberLists	Returns the names of the member lists for a dimension.
EnumMembers	Returns an array containing the member IDs of the dimension members in either the default dimension hierarchy or in a member list.
EnumMembers2	Returns an array containing the member IDs of the dimension members in either the default dimension hierarchy, a static member list, or a dynamic member list for given Scenario, Year, Period, and Entity dimension members.
EnumMembersWithAttribValue	Enumerates the attribute IDs and corresponding labels for a given attribute.
EnumParents	Returns an array containing the member IDs of a given member's parent members.
EnumSortedIDsOfChildren	Returns an array containing the member IDs an Intercompany Partner dimension member's children, with the members sorted according to the specified sorting criteria.
EnumSortedIDsofChildren	Returns Sorted children and the parent item of the specified children.
EnumSortedMembers	Returns an array containing the member IDs of the Intercompany Partner dimension members in a member list, with the members sorted according to the specified sorting criteria.
Find	Returns the member IDs of those members of a member list with labels that match a search string. Each time a member matching the search criteria is found, the member ID of the member and the member ID of its parent are returned.
FindByDesc	Returns the member IDs of those members of a member list with descriptions that match a search string in a given language.
FindMatchingMembersFromHierarchy	Returns the member IDs of the members with labels that match a search string.
FindMatchingMembersFromHierarchyByDesc	Returns the member IDs of the members with descriptions in a given language that match a search string.

Method	Description
FindMatchingMembersFromHierarchyWildcard	Returns the member IDs of the members with labels or descriptions that match a search string; the search string can include wildcard characters.
GetAllPathsToMember	Returns an array of strings that represent the possible paths in a dimension's hierarchy to a given member.
GetAttributeValue	Returns the value of a given member's metadata attribute.
GetDefaultHierarchyPosition	Returns the default position of a member within a dimension's hierarchy.
GetDefaultItemIDHierarchy	Returns an array containing the path of a dimension member and its ancestors. The array contains the member IDs of the members in this path.
GetDefaultMemberID	Returns the member ID of a dimension's default member.
GetDefaultParent	Returns the member ID of a given dimension member's default parent.
GetDefaultParentLabel	Returns the label of a given dimension member's default parent.
GetDescription	Returns the description of a dimension member.
GetDisplayInfo	Returns the label and description of a dimension member.
GetDisplayInfoForSeveralItems	Returns arrays containing the labels and descriptions of dimension members.
GetItemGeneration	Returns the generation of a given member in the dimension hierarchy.
GetItemID	Returns the member ID of a dimension member.
GetItemIDQL	<i>For internal use.</i>
GetItemLevel	Returns the level of a given member in the dimension hierarchy.
GetLabel	Returns the label of a dimension member.
GetMemberListID	Returns the numeric ID of a member list.
GetMemberListName	Returns the name of a member list for which you have the ID number.
GetNumBaseMembers	Returns a count of the base-level members beneath a dimension member.
GetNumChildren	Returns the number of children that are one level beneath a dimension member in a dimension hierarchy.
GetNumDescendants	Returns the number of descendants beneath a member in a dimension's hierarchy, given the member's ID.
GetNumMembers	Returns the number of members in a dimension.

Method	Description
GetNumParents	Returns the number of parents for a dimension member, given the member's ID.
GetQualifiedLabel	<i>For internal use.</i>
GetTreeCapabilities	Returns various properties of a dimension.
GetTreeName	Returns the name of the dimension to which the IHsvTreeInfo interface has been set.
GetTreeTimeStamp	Returns a timestamp that indicates when the dimension was last updated.
HasChildren	Indicates whether a dimension member has child members.
IsMemberABaseOf	Indicates whether a member is in a base-level position beneath another member in a dimension hierarchy.
IsMemberAChildOf	Indicates whether one member is a child of another member.
IsMemberADescendantOf	Indicates whether one member is a descendant beneath another member in a dimension hierarchy.
SortMembersBasedOnList	Filters and sorts member IDs, using the members in a member list as the filtering and sorting criteria. You can also filter and sort against a dimension's default hierarchy instead of a member list.
TranslateAttributeValueForDisplay	Returns a String representation of a dimension member's attribute. However, in cases where the attribute value is itself a dimension member, the member ID of the attribute value is returned instead of the member's label.

For detailed descriptions of these methods, see [“IHsvTreeInfo Interface Methods” on page 188](#).

HsvCurrencies Object Overview

The HsvCurrencies object enables you to obtain information on an application's currencies.

Assign HsvCurrencies object references with the [Currencies](#) property.

The following table lists the methods of the HsvCurrencies object. For detailed descriptions of these methods, see [“HsvCurrencies Object Methods” on page 270](#).

Table 20 HsvCurrencies Object Methods

Method	Description
EnumCurrencies	Returns arrays containing the currency IDs and labels of the application's currencies.
EnumCurrencies2	Returns arrays containing the currency IDs and labels of the application's currencies, with the option to return only those currencies for which the DisplayInCT attribute is enabled.

Method	Description
GetCurrencyDescription	Returns a currency's description in a given language.
GetCurrencyID	Returns the currency ID of a given currency.
GetCurrencyLabel	Returns a currency label, given a currency ID.
GetCurrencyTranslationOperator	Returns the conversion operator for a given currency.
GetScale	Returns the scale of a given currency.

HsvData Type Library Overview

The HsvData type library provides the HsvData object, which gets and sets data in Financial Management applications. The HsvData object is a child of the HsvSession object. Use HsvSession's Data property to set HsvData object references as shown in the following example:

```
Dim cHsvData As HsvData
Set cHsvData = cHsvSession.Data
```

The following table lists the methods of the HsvData object.

Table 21 HsvData Object Methods

Method	Description
AddDataToMDDataBuffer	Adds an application's cell to an HsvMDDataBuffer or HsvMDDataBufferLite object. Any data, description, or line items for the cell will be added.
AttachDocumentToCell	Attaches a previously loaded document to a cell.
ClearAllData	Deletes all data from an application.
ClearAllDescriptionsInSubCube	Removes the cell text from all the cells in a subcube.
ClearDataAuditItems	Deletes the audit history for all data changes that occurred before a given date and time.
ClearInputData	Deletes data from some or all of a subcube's cells.
ClearInvalidData	Scans for or deletes invalid records.
CopyInputData	<i>Deprecated</i> - use CopyInputDataForMultipleEntities.
CopyInputDataForMultipleEntities	Copies data for one or more entities from one set of cells to another set of cells.
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.
DetachDocumentFromCell	Detaches a document from a cell.

Method	Description
DoCellDescriptionsExist	Indicates whether one or more of the cells for a combination of Scenario, Year, and Entity dimension members contain cell text descriptions.
DoesCellDescriptionExist	Indicates whether a cell contains cell text.
DoesDataExist	Indicates whether any data exists in one or more cells.
DoesSparseDataExist	Indicates whether sparse data exists for the specified cell.
EnumDataAuditItems	Returns data audit information from a given range of audit records that meet the specified filtering criteria.
EnumDataAuditItems2	<i>For internal use.</i>
EnumEntitiesWithDataForScenarioYear	Returns a variant array of Entity ID's that have data for a given scenario and year.
EnumExtractOptions	Returns a two-dimensional array of the data extract options that can be passed to <code>HsvData.Extract</code> . The array includes the options' names and default values.
EnumLoadOptions	Returns a two-dimensional array of the data load options that can be passed to <code>HsvData.Load</code> . The array includes the options' names and default values.
Extract	Extracts data into a text file. The file will be created on the application server.
ExtractDataAuditItems	Extracts to a file the data audit information that meets the specified filtering criteria. The filtering criteria include dimension members, date range, application server, and username.
ExtractDrillableRegions	Retrieves all ERPI URL definitions.
ExtractDrillableRegionsByURLNames	Retrieves only the URLs that appear in the input array <i>varabstrURLNames</i> .
FilterMembersThatHaveData	Returns a list of members for the specified dimension for which data exists for the specified POVs.
FilterMembersThatHaveData2	For a specified dimension, returns the filtered of member IDs that have data for any of the specified POVs.
FormatNumberToText	Returns a number as a String. <code>FormatNumberToText</code> scales the number and puts a fixed number of digits to the right of the decimal point.
FormatNumberToText2	Takes a number and returns it as a String, with the option to remove trailing zeroes. <code>FormatNumberToText2</code> scales the number and puts a fixed number of digits to the right of the decimal point.
FormatStatusToText	Returns a string description of the numeric calculation status passed as the argument.

Method	Description
FormatStoredNumberToText	Converts a number passed as a Double to a String, formatting the String with the decimal and thousands delimiters that you specify.
FormatTextToNumber	Returns a number stored in a String as a Double. <i>FormatTextToNumber</i> also scales the returned number.
FormatTextToStoredNumber	Converts a number passed as a String to a Double.
GetAllDescriptionsInSubCube	Returns an array containing the cell text descriptions of one or more cells that have the same Scenario, Year, and Value dimension members.
GetAllURLNames	Retrieves the full list of URL names currently in Financial Management.
GetAttachedDocumentsToCell	Returns the names and paths of the documents attached to a given cell.
GetBaseDataForAccount	<i>For internal use.</i>
GetCalcStatus	Returns the calculation status of a subcube.
GetCalcStatusStatistics	Returns arrays of flags that indicate which calculation statuses apply to the specified entities and periods of a subcube.
GetCell	Returns the data in a cell, as well as the cell's status. The data is returned as a Double.
GetCellDescription	Returns the cell text description of a cell.
GetCellDescriptions	Returns the cell text descriptions of one or more cells.
GetCellHistory	Returns the audit history of a cell's data changes, with cell values returned as Doubles.
GetCellHistory2	Returns the audit history of a cell's data changes, with cell values returned as formatted strings.
GetCellJournalEntries	Returns arrays containing the data in and IDs of the journal entries for a cell.
GetCellLineItems	Returns arrays of the data and descriptions for the specified cell's line items. Data is returned in a Double array.
GetCells	Returns the data and statuses of cells. The cells' data is returned as a Double array.
GetCellsWithRowSuppression	Returns the data and statuses of cells, optionally excluding rows of cells that match specified criteria. You can exclude rows that contain no data, zero, and derived data, as well as rows consisting of invalid intersections.
GetCellsWithRowSuppression2	Returns the data and statuses of cells, optionally excluding rows of cells that match specified criteria.
GetCountOfAttachedDocumentsToCell	Returns a count of the documents attached to a cell.

Method	Description
GetCurrencyCube	Returns an object reference to the HsvCurrencyCube object. The object reference provides access to the subcube identified by the member IDs passed to GetCurrencyCube.
GetDataForAllMetadataCombinations	<i>For internal use.</i>
GetLineItems	<i>Deprecated</i> - superseded by GetCellLineItems.
GetMaxCellTextSize	Returns the maximum number of characters that can be inserted as cell text.
GetMembersThatHaveData	Returns members that have data.
GetNodeCube	Returns an object reference to the HsvNodeCube object. The object reference provides access to the subcube identified by the member IDs passed to GetNodeCube.
GetPhaseSubmissionGridForGivenScenarioPeriod	Returns an array representing the submission groups assigned to the specified combinations of Scenario dimension member, Period dimension members, and submission phases.
GetStatus	Returns the transaction status, metadata status, and calculation status of a cell.
GetStatusEx	Returns the transaction status, metadata status, and calculation status of a cell, as well as additional status information such as whether the cell supports intercompany transactions.
GetStatusUsingPhaseID	Returns the transaction status, metadata status, and calculation status of either a cell or a submission phase.
GetTextCell	Returns the data in a cell, as well as the cell's status. The data is returned as a String.
GetTextCellLineItems	Returns arrays of the data and descriptions for the specified cell's line items. Data is returned in a String array.
GetTextCells	Returns Variant arrays containing the data and statuses of cells. GetTextCells also enables you to specify the scaling and the number of decimals for the returned data.
GetTextCellsWithRowSuppression	Returns the data and statuses of cells, optionally applying scaling and formatting, and excluding rows of cells that match specified criteria. You can exclude rows that contain no data, zero, and derived data, as well as rows consisting of invalid intersections.
GetTextCellsWithRowSuppression2	Returns the data and status for rows that are not suppressed.
GetTextLineItems	<i>Deprecated</i> - superseded by GetTextCellLineItems.
GetTransactionData	Populates an HsvTransactionData object with an array of transaction data.
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.

Method	Description
GetURLByName	Returns an XML block representing the specified URL.
GetURLsForCell	Retrieves all URLs whose regions cover a specified POV.
InsertLineItemDetails	Inserts line item descriptions in the cells that intersect the specified dimension members.
IsValidCellText	Indicates whether a string exceeds the application's maximum cell text size.
IIndexRecordProcessed	<i>For internal use.</i>
Load	Loads data into an application, using a load file on the application server.
LoadDrillableRegions	Loads the definition file for the drillable regions.
SetCalcStatusLocked	Locks the cells for a period in a subcube.
SetCalcStatusLockedForMultipleProcessUnits	Locks the cells for one or more process units.
SetCalcStatusUnlocked	Unlocks the cells for a period in a subcube.
SetCalcStatusUnlockedForMultipleProcessUnits	Unlocks the cells for one or more process units.
SetCell	Sets a cell's data; you can either insert data or set the cell to Null. Cell data is passed as a Double.
SetCellDescriptions	Inserts cell text into one or more cells.
SetCellLineItems	Appends or updates line items for the specified cell. Line item data is passed in a Double array.
SetCells	Sets data for an array of cells. For each cell in the array, you can either insert data or set the cell to Null. The cells' data is passed as a Double array.
SetCells2	Sets data for an array of cells and returns the cells' statuses; if any of the cells are not writable, <code>SetCells2</code> inserts data in the writable cells.
SetCellsLineItems	Appends or updates line items for the specified cells. Line item data is passed in a Double array.
SetFileForLoad	<i>For internal use.</i>
SetLineItems	<i>Deprecated</i> - superseded by <code>SetCellLineItems</code> .
SetMinMaxPeriod	<i>For internal use.</i>
SetPhaseSubmissionGridForGivenScenarioPeriod	Assigns submission groups to the specified combinations of Scenario dimension member, Period dimension members, and submission phases.
SetTextCell	Inserts data into a cell, passing the data as a String. <code>SetTextCell</code> also enables you to scale the data that is passed.

Method	Description
SetTextCellLineItems	Appends or updates line items for the specified cell. Line item data is passed in a String array.
SetTextCells	Inserts data into cells. The cells' data is passed in a String array. <code>SetTextCells</code> also enables you to scale the data that is passed.
SetTextCellsLineItems	Appends or updates line items for the specified cells. Line item data is passed in a String array.
SetTextLineItems	<i>Deprecated</i> – superseded by <code>SetTextCellLineItems</code> .
StartLoad	<i>For internal use.</i>
UpdateDataUsingMDDataBuffer	Inserts an <code>HsvMDDataBuffer</code> or <code>HsvMDDataBufferLite</code> object's cells into the corresponding cells of an application.
varalAggregationMapAddr	<i>For internal use.</i>

For detailed descriptions of these methods, see [“HsvData Object Methods” on page 276](#).

HsvCalculate Type Library Overview

The `HsvCalculate` type library contains one object, the `HsvCalculate` object. Use this object to execute consolidations, translations, and calculations in Financial Management applications.

The `HsvCalculate` object is a child of the `HsvSession` object. Use `HsvSession`'s `Calculate` property to set `HsvCalculate` object references as shown in the following example:

```
Dim cHsvCalculate as HsvCalculate
Set cHsvCalculate = cHsvSession.Calculate
```

The following table lists the methods of the `HsvCalculate` object.

Table 22 HsvCalculate Object Methods

Method	Description
Allocate	Allocates an entity's data for the specified Scenario, Year, Period, and Value dimension members.
Allocate2	Allocates an entity's data across a range of periods for the specified Scenario, Year, and Value dimension members.
CalcEPU	Runs the Equity Pickup for the specified Scenario, Year, and Period.
ChartLogic	Calculates an entity's data for the specified Scenario, Year, Period, and Value dimension members.
ChartLogic2	Calculates an entity's data across a range of periods for the specified Scenario, Year, and Value dimension members.

Method	Description
Consolidate	Consolidates an entity's data for the specified Scenario, Year, and Period dimension members.
Consolidate2	Consolidates an entity's data across a range of periods for the specified Scenario and Year dimension members.
CustomLogic	<i>For internal use.</i>
FindOverlappingConsolidation	Indicates whether any consolidations are currently running or queued for the specified entity, scenario, year, and range of periods. If FindOverlappingConsolidation finds any such consolidations, the consolidations' types and dimension member labels are returned.
GetCOMDLLRules	<i>For internal use.</i>
GetConsolidationProgress	<i>For internal use.</i>
GetDefaultExchangeRate	Returns the exchange rate between two currencies for the specified Point of View.
GetEPUInfo	Retrieves the equity pickup information for the specified Scenario, Year, and Period. The information returns include: owner, owned, percentage ownership, and status.
GetVBScriptRules	Returns the rules that have been loaded into an application. The rules are returned as an array of bytes.
IsEntityAnEPUOwner	Indicates whether the entity is an EPU owner for specified Scenario, Year, and Period.
LoadCalcManagerRules	<i>For internal use.</i>
LoadCalcManagerRules2	<i>For internal use.</i>
SetCOMDLLRules	<i>For internal use.</i>
SetVBScriptRules	Loads or scans a rules file.
SetVBScriptRules2	Loads or scans a rules file, optionally validating whether the rules violate the referential integrity of any intercompany transactions.
StopConsolidation	<i>For internal use.</i>
Translate	Translates an entity's data from one currency to another for the specified Scenario, Year, Period, and Value dimension members.
Translate2	Translates an entity's data from one currency to another across a range of periods for the specified Scenario, Year, and Value dimension members.

For detailed descriptions of these methods, see [“HsvCalculate Object Methods” on page 376](#).

HsvJournals Type Library Overview

The HsvJournals type library provides the HsvJournals object and the IHsvJournalsEx and IHsvJournalsReport interfaces:

- The HsvJournals object is used to open and close periods, to test whether periods are open, and to get internal IDs of journals and templates. The HsvJournals methods are listed in [“HsvJournals Object Overview” on page 76](#).
- The IHsvJournalsEx interface is used to create new journals and templates and to process existing templates. The interface’s methods also get the information contained by existing journals and templates. The IHsvJournalsEx methods are listed in [“IHsvJournalsEx Interface Overview” on page 76](#).

Tip: Many of the IHsvJournalsEx methods take journal or template IDs. Get these IDs with the HsvJournals object’s `GetItemID` and `GetJournalTemplateItemID` methods.
- The IHsvJournalsReport interface gets journal report data, and is described in [“IHsvJournalsReport Interface Overview” on page 79](#).

HsvJournals Object Overview

The HsvJournals object is used to manage periods and to get internal IDs of journals and templates.

The HsvJournals object is a child of the HsvSession object. Use HsvSession’s `Journals` property to assign HsvJournals object references as shown in the following example:

```
Dim cHsvJournals As HsvJournals
Set cHsvJournals = cHsvSession.Journals
```

The following table lists the HsvJournals object’s methods.

Table 23 HsvJournals Object Methods

Method	Description
ClosePeriod	Closes a period, meaning that journals can no longer be posted for the period.
GetItemID	Returns the internal ID of a journal. Use <code>GetItemID</code> with the IHsvJournalsEx interface methods that take journal IDs as arguments.
GetJournalTemplateItemID	Returns the internal ID of a template. Use <code>GetJournalTemplateItemID</code> with the IHsvJournalsEx interface methods that take template IDs as arguments.
GetPeriodStatusList	Returns a two-dimensional array that indicates whether the periods for a scenario and year are opened, unopened, or closed.
IsPeriodOpen	Indicates whether a period is open.
OpenPeriod	Opens a period, enabling users to post journals to the period.

For detailed descriptions of these methods, see [“HsvJournals Object Methods” on page 395](#).

IHsvJournalsEx Interface Overview

The IHsvJournalsEx interface is used to create and process journals and to create templates. The IHsvJournalsEx methods also return the information that journals and templates contain.

The IHsvJournalsEx interface is a child of the HsvSession object. Use HsvSession's Journals property to assign IHsvJournalsEx object references as shown in the following example:

```
Dim cIHsvJournalsEx As IHsvJournalsEx
Set cIHsvJournalsEx = cHsvSession.Journals
```

The following table lists the methods of the IHsvJournalsEx interface.

Table 24 IHsvJournalsEx Interface Methods

Method	Description
AddJournalGroup	Creates a journal group.
ApproveJournals	Approves one or more journals.
DeleteJournals	Deletes one or more journals.
DeleteTemplates	Deletes one or more templates.
EnumJournalGroups	Returns the names and descriptions of an application's journal groups.
EnumJournalGroupsForScenarioYear	Returns the names and descriptions of the journal groups assigned to journals for a given scenario and year.
EnumJournalIDsForExtractFilter	Returns the IDs of journals and templates that match the specified filtering criteria.
GenerateRecurring	Generates a journal from a recurring template.
GetEntityJournals	Returns journal entry information for all journals that match the specified Point of View.
GetJournal	Returns a variety of information for a journal. Line item amounts are returned in a Double array.
GetJournal2	Returns a variety of information for a journal. This is the same as <i>GetJournal</i> except with the addition of the VARIANT_BOOL <i>vbScaleAmounts</i> flag which is used to determine whether the textual representation of the amounts are scaled or not.
GetJournalDisplayData	Returns various types of information for journals. The types of information returned correspond to the display columns in the Columns tab of the Filters And Sorting dialog box in the Process Journals workspace frame.
GetJournalLabelsForIDs	Returns the labels of the journals that contain the specified Scenario and Year dimension members and that correspond to a given set of journal IDs.
GetJournalQueryDefinitionIDs	Returns IDs of one or more journals. You can return IDs of all journals or return only those IDs that meet filtering criteria that you specify.
GetTemplate	Returns a variety of information for a template. Line item amounts are returned in a Double array.
GetTemplateDisplayData	Returns various types of information for templates. The types of information returned correspond to the display columns in the Columns tab of the Filters And Sorting dialog box in the Set Up Journals workspace frame.

Method	Description
GetTemplateLabelsForIDs	Returns the labels of the journal templates for a given set of journal template IDs.
GetTemplateQueryDefinitionIDs	Returns IDs of one or more templates. You can return IDs of all templates or return only those IDs that meet the filtering criteria you specify.
GetTextJournal	Returns a variety of information for a journal. Line item amounts are returned in a String array.
GetTextTemplate	Returns a variety of information for a template. Line item amounts are returned in a String array.
GetVariance	Returns a journal's total debit and credit amounts and the difference between these amounts.
PostJournals	Posts one or more journals.
RejectJournals	Rejects one or more journals.
RemoveAllJournalGroups	Removes all journal groups from an application.
RemoveJournalGroup	Removes the specified journal group.
SaveJournal	Creates a new journal, or saves changes to an existing journal that has a Working or Submitted status. Line item amounts are passed in a Double array. Tip: To create a journal from a template, pass the values returned by <code>GetTemplate</code> to <code>SaveJournal</code> .
SaveTemplate	Creates a new journal template, or updates an existing journal template. Line item amounts are passed in a Double array.
SaveTextJournal	Creates a new journal, or saves changes to an existing journal that has a Working or Submitted status. Line item amounts are passed in a String array. Tip: To create a journal from a template, pass the values returned by <code>GetTextTemplate</code> to <code>SaveTextJournal</code> .
SaveTextTemplate	Creates a new journal template, or updates an existing journal template. Line item amounts are passed in a String array.
SubmitJournals	Submits one or more journals.
UnpostJournals	Unposts one or more journals.
UnsubmitJournals	Unsubmits one or more journals.
ValidateLineItems	Validates whether the values passed would define a valid journal.
ValidateValue	Indicates whether a String resolves to a Double that Financial Management would then allow to be inserted as a journal amount. To evaluate as valid, the String must not represent a negative Double, as Financial Management does not allow the entry of negative amounts in journals.

For detailed descriptions of these methods, see [“IHsvJournalsEx Interface Methods” on page 399](#).

IHsvJournalsReport Interface Overview

The IHsvJournalsReport interface provides methods for returning journal report data. For more information, see [“IHsvJournalsReport Interface Methods” on page 444](#).

The following table lists the methods of the IHsvJournalsReport interface.

Table 25 IHsvJournalsReport Interface Methods

Method	Description
GetReportData	Returns data for journals; only actual journal amounts are returned.
GetReportData2	Returns data for journals; scaled and actual journal amounts are returned.

HsvSecurityAccess Type Library Overview

The HsvSecurityAccess type library enables you to define and to get information about Financial Management security. This type library provides the HsvSecurityAccess object and the IHsvDataSecurity interface:

- The HsvSecurityAccess object is used to get and set application security. The HsvSecurityAccess methods are listed in [“HsvSecurityAccess Object Overview” on page 79](#).
- The IHsvDataSecurity interface returns information about the connected user’s rights to process units and submission phases, and also provides a method that refreshes access rights on application servers. The IHsvDataSecurity methods are listed in [“IHsvDataSecurity Interface Overview” on page 84](#).

HsvSecurityAccess Object Overview

The HsvSecurityAccess object is used to get and set application security. This object provides access to security-related items such as users, roles, and security classes.

The HsvSecurityAccess object is a child of the HsvSession object. Use HsvSession’s Security property to assign HsvSecurityAccess object references as shown in the following example:

```
Dim cHsvSecurityAccess As HsvSecurityAccess
Set cHsvSecurityAccess = cHsvSession.Security
```

The following table lists the methods of the HsvSecurityAccess object.

Table 26 HsvSecurityAccess Object Methods

Method	Description
AddApplicationAdministrator	<i>Deprecated</i> - use AddApplicationAdministrator2.
AddApplicationAdministrator2	Assigns the Application Administrator role to a user.
AddOrRemoveApplicationAdministrators	<i>Deprecated</i> - use AddOrRemoveApplicationAdministrators2.

Method	Description
AddOrRemoveApplicationAdministrators2	Adds or removes one or more users from the Application Administrator role.
AddOrRemoveRolesFromUser	<i>Deprecated</i> - use AddOrRemoveRolesFromUser2.
AddOrRemoveRolesFromUser2	Adds or removes a user from one or more roles.
AddOrRemoveUsersFromRole	<i>Deprecated</i> - use AddOrRemoveUsersFromRole2.
AddOrRemoveUsersFromRole2	Assigns users to or removes them from a given role.
AddSecurityClass	Adds a security class to a Classic application.
AddSecurityClassWithAccessCode	<i>For internal use.</i>
AddUser	<i>Deprecated</i> - use AddUser2.
AddUser2	Adds a user or user group to an application.
AddUserEx	<i>This method is not supported as of Release 4.1.</i>
AddUserToRole	<i>Deprecated</i> - use AddUserToRole2.
AddUserToRole2	Assigns a user to a role.
AllowRulesLoadForEPMAApp	<i>For internal use.</i>
DeleteSecurityClass	Deletes a security class from a Classic application.
DeleteSecurityClassWithAccessCode	<i>For internal use.</i>
EnumApplicationAdministrators	<i>Deprecated</i> - use EnumApplicationAdministrators2.
EnumApplicationAdministrators2	Returns arrays containing the security identifiers and usernames of the users assigned to the Application Administrator role.
EnumRoles	Returns an array containing the names of an application's roles.
EnumRolesForPrincipal	Returns the IDs of a user's or group's roles.
EnumRolesForUser	Returns the localized names of a user's roles.
EnumSecurityClasses	Returns arrays containing an application's security class IDs and names.
EnumSecurityClassRightsForPrincipal	Returns arrays representing a user's access and E-mail alerting rights to security classes.
EnumUserClassAccess	Returns the access and E-mail alerting rights that the specified users have for the specified security classes.
EnumUsers	<i>Deprecated</i> - use EnumUsers2.
EnumUsers2	<i>Deprecated</i> - use EnumUsers3 .
EnumUsers3	Returns arrays containing the security identifiers and usernames of an application's users. If delegated user management is enabled

Method	Description
	in Shared Services, you can filter users who have access rights not granted to the connected user.
EnumUsersInGroup	Returns the security identifiers, usernames, and identity types of the users in a user group.
EnumUsersInRole	<i>Deprecated</i> - use <code>EnumUsersInRole2</code> .
EnumUsersInRole2	Returns the security identifiers and usernames of the users and groups assigned to a given role.
EnumUsersInRole3	Returns the security identifiers and usernames of the users and groups assigned to a given role, and optionally allows you to return this information for users of any groups assigned to the role.
EnumUsersInSecurityClass	<i>For internal use.</i>
EnumUsersInSecurityClass2	<i>For internal use.</i>
EnumUsersInSecurityClass3	<i>For internal use.</i>
EnumUsersOrGroups	Returns the security identifiers, usernames, and identity types of the users in a user group.
EnumUsersWithFilter	<i>Deprecated.</i> Use EnumUsersWithFilter2 .
EnumUsersWithFilter2	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. If delegated user management is enabled in Shared Services, you can filter users who have access rights not granted to the connected user.
GenerateSecurityReportForBiPub	<i>For internal use.</i>
GetAllSecurityClassRightsForConnectedUser	Returns arrays indicating the connected user's rights to all security classes.
GetApplicationAdministratorAccessForAllUsers	<i>Deprecated</i> - use <code>GetApplicationAdministratorAccessForAllUsers2</code> .
GetApplicationAdministratorAccessForAllUsers2	Indicates which of an application's users are assigned to the Application Administrator role.
GetConnectedUser	<i>Deprecated</i> - use <code>GetConnectedUser2</code> .
GetConnectedUser2	Returns the security identifier and username of the connected user.
GetIdentityTypes	Returns the identity types of the specified users.
GetNumRoles	Returns a count of the number of roles in an application.
GetOwner	<i>This method is not supported as of Release 4.1.</i>
GetRoleAccessForAllUsers	<i>Deprecated</i> - use <code>GetRoleAccessForAllUsers2</code> .

Method	Description
GetRoleAccessForAllUsers2	Indicates whether the application's users are assigned to a given role.
GetRoleID	Returns the ID of a role, given the role's name.
GetRoleLabel	Returns the name of a role, given a role ID.
GetRulesMode	<i>For internal use.</i>
GetSecurityClassAccessForAllUsers	<i>Deprecated</i> - use <code>GetSecurityClassAccessForAllUsers2</code> .
GetSecurityClassAccessForAllUsers2	Returns the access rights to a security class for all of an application's users.
GetSecurityClassID	Returns the ID of a security class, given the security class's name.
GetSecurityClassLabel	Returns the name of a security class, given a security class ID.
GetSecurityClassRightsForConnectedUser	Indicates the access rights that the connected user has to a security class.
GetTaskAccessForConnectedUserFromList	Indicates whether the connected user is allowed to perform one or more tasks.
GetUserAccessForAllRoles	<i>Deprecated</i> - use <code>GetUserAccessForAllRoles2</code> .
GetUserAccessForAllRoles2	Indicates whether a user is assigned to the application's roles.
GetUserAccessForAllSecurityClasses	<i>Deprecated</i> - use <code>GetUserAccessForAllSecurityClasses2</code> .
GetUserAccessForAllSecurityClasses2	Returns the level of access rights that a user has for each security class in an application.
GetUserID	<i>Deprecated</i> - use <code>GetUserSID</code> .
GetUserIDFromSID	<i>This method is not supported as of Release 4.1.</i>
GetUserInfoFromUniqueID	<i>Deprecated.</i> 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.
GetUserName	<i>Deprecated</i> - use <code>GetUserName2</code> .
GetUserName2	Returns a username, given a security identifier.
GetUserNameFromSID	Returns the username of a user, given the user's security identifier (SID).
GetUserSID	Returns the security identifier for a given user.
InsertDefaultSecurityClass	<i>For internal use.</i>

Method	Description
InsertDefaultSecurityClassWithAccessCode	<i>For internal use.</i>
IsApplicationAdministrator	Indicates whether the connected user is assigned to the Application Administrator role.
IsClassicHFMAApplication	Indicates whether an application is a Classic application.
IsConnectedUserAllowedToPerformTask	Indicates whether the connected user has rights to a task.
IsConnectedUserInRole	Indicates whether the connected user is assigned to a role.
IsValidWindowsUser	<i>This method is not supported as of Release 4.1.</i>
RemoveApplicationAdministrator	<i>Deprecated</i> - use <code>RemoveApplicationAdministrator2</code> .
RemoveApplicationAdministrator2	Removes a user from the Application Administrator role.
RemoveUser	<i>Deprecated</i> - use <code>RemoveUser2</code> .
RemoveUser2	Removes a user from an application.
RemoveUserFromRole	<i>Deprecated</i> - use <code>RemoveUserFromRole2</code> .
RemoveUserFromRole2	Removes a user from a role.
RenameSecurityClass	<i>For internal use.</i>
SetManySecurityClassRightsForUser	<i>Deprecated</i> - use <code>SetManySecurityClassRightsForUser2</code> .
SetManySecurityClassRightsForUser2	Sets a user's access and E-mail alerting rights for one or more security classes.
SetRolesForUser	Specifies one or more roles for a given user.
SetRulesMode	<i>For internal use.</i>
SetSecurityClassLabel	Changes the name of a security class in a Classic application.
SetSecurityClassLabelWithAccessCode	<i>For internal use.</i>
SetSecurityClassRightsForManyUsers	<i>Deprecated</i> - use <code>SetSecurityClassRightsForManyUsers2</code> .
SetSecurityClassRightsForManyUsers2	Sets one or more users' access and E-mail alerting rights to a given security class.
SetSecurityClassRightsForUser	<i>Deprecated</i> - use <code>SetSecurityClassRightsForUser2</code> .
SetSecurityClassRightsForUser2	Sets a user's access and E-mail alerting rights for a security class.
SetUserClassAccess	Sets security class access and E-mail alerting rights for one or more users.
TakeOwnership	<i>This method is not supported as of Release 4.1.</i>

For detailed descriptions of these methods, see [“HsvSecurityAccess Object Methods”](#) on page 454.

IHsvDataSecurity Interface Overview

The IHsvDataSecurity interface is used to return information about the connected user’s rights to process units. The interface also provides a method that refreshes access rights on application servers.

The IHsvDataSecurity interface is a child of the HsvSession object. Use HsvSession’s Security property to assign IHsvDataSecurity object references as shown in the following example:

```
Dim cIHsvDataSecurity As IHsvDataSecurity
Set cIHsvDataSecurity = cHsvSession.Security
```

The following table lists the IHsvDataSecurity interface’s methods.

Table 27 IHsvDataSecurity Interface Methods

Method	Description
GetCellLevelAccessRights	Returns the access rights that the connected user has to a cell.
GetProcessUnitAccessRights	Returns the access rights that the connected user has to a process unit.
GetProcessUnitAccessRightsAndState	Returns the access rights that the connected user has for a process unit, as well as the process unit’s current level.
GetProcessUnitAccessRightsEx	Returns the access rights that the connected user has to a submission phase, given either the member IDs of a cell in the phase or the phase ID.
GetProcessUnitAccessRightsAndStateEx	Returns the following information for a submission phase, given either the member IDs of a cell in the phase or the phase ID: <ul style="list-style-type: none">● The connected user’s access rights to the submission phase.● The review level of the submission phase.
RefreshAccessRightsCache	Refreshes the access rights on the application server.

For detailed descriptions of these methods, see [“IHsvDataSecurity Interface Methods”](#) on page 485.

HsvSystemInfo Type Library Overview

The HsvSystemInfo type library contains one object, the HsvSystemInfo object. Use the HsvSystemInfo object to get and set various types of system information such as application directories, server names, and task audit histories.

The HsvSystemInfo object is a child of the HsvSession object. Use HsvSession’s SystemInfo property to assign HsvSystemInfo object references as shown in the following example:

```
Dim cHsvSystemInfo As HsvSystemInfo
```

```
Set cHsvSystemInfo = cHsvSession.SystemInfo
```

The following table lists the HsvSystemInfo object's methods.

Table 28 HsvSystemInfo Object Methods

Method	Description
AddRefToHsxServer	<i>For internal use.</i>
AddTaskToAudit	Adds a given task for the user to the audit log.
AddTaskToRunningTasks	<i>For internal use.</i>
AddTaskToRunningTasksAndUpdatePOV	<i>For internal use.</i>
CheckAccess	Indicates whether the current user has access to the application.
ClearAuditTasks	Deletes the task audit history for all tasks that meet the specified criteria.
ClearAuditTasks2	Deletes the audit history for a given task that meets the specified criteria.
ClearRunningTask	<i>For internal use.</i>
DeleteUserParameter	Deletes a user parameter that has been created with <code>SetUserParameter</code> .
DisableNewConnections	Disables new Financial Management connections for the specified user and application server criteria.
EnableNewConnections	Enables new Financial Management connections for the specified user and application server criteria.
EnumActivityServers	Returns the names of the application servers for which there are task audit and data audit records.
EnumActivityUsers	Returns the usernames of all users who have performed at least one activity in the application.
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.
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.
EnumProhibitConnections	Returns information on the applications, application servers, and users for which connections have been disabled.
EnumRunningTasks	Returns information about the running tasks that meet the selection criteria.
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.
EnumRunningTasksPOV	Returns the dimension members, consolidation types, and other information for the running consolidations that meet the selection criteria.

Method	Description
ExtractTaskAudit	Extracts to a file the task audit records that meet the specified criteria. Criteria include date range, application server, user, and task.
GetActivityCodeDesc	Returns the description of a given type of activity.
GetActivityUserID	Returns the activity user ID for a given username.
GetApplicationDirectory	Returns the name of the application folder for an application.
GetApplicationName	Returns the name of the application to which the client is connected.
GetCalcRulestype	<i>For internal use.</i>
GetCOMDLLCalcRules	<i>For internal use.</i>
GetCurrentActivity	Returns information about the user's current activity.
GetExtractFileEncoding	Returns the type of file encoding for extracted files.
GetFormattedDateTime	Returns a string representation of the given double-byte date/time value using the language ID for the connected user.
GetFormattedResourceString	<i>For internal use.</i>
GetRunningTaskLogFilePathName	Returns the name and path of the log file for a given running task.
GetRunningTaskProgress	Returns information on the progress of a given running task.
GetRunningTasksCount	Returns the number of currently running tasks for the application.
GetRunningTaskStatus	Returns the status of a given task.
GetKillUsersStatus	<i>For internal use.</i>
GetKillUserStatus	Indicates whether an administrator has logged off the current user.
GetLanguageUserParameters	Gets the language in which member descriptions are displayed for the connected user.
GetLastModifiedDateForArtifact	<i>For internal use.</i>
GetModuleName	Returns the name of the module represented by a given module ID.
GetNumberFormattingUserParameters	Returns the double-byte Integers that identify the user's decimal and thousands separator characters.
GetResourceLanguageUserParameters	Returns the ID of the user's default language for resource strings such as error messages and other strings that are generated on the server.
GetResourceString	<i>For internal use.</i>
GetServerName	Returns the name of the server to which the client is connected.
GetUserName	Returns the username of the connected user.

Method	Description
GetUserParameter	Returns the value of a user parameter that has been set with <code>SetUserParameter</code> .
GetVBScriptCalcRules	Returns the rules that have been loaded into an application. The rules are returned as an array of bytes.
GetVBScriptMemberListRules	Returns an application's member lists as an array of bytes in the <code>.LST</code> file format.
GetWorkingDirectory	<i>For internal use.</i>
IsScheduledTaskReadyToRun	<i>For internal use.</i>
KeepRunningTaskStillAlive	<i>For internal use.</i>
KillUsers	Logs off users.
OutputSystemInfo	<i>For internal use.</i>
pHsvDSSecurityUnk	<i>For internal use.</i>
pHsvDSReportsUnk	<i>For internal use.</i>
ReleaseHsxServer	<i>For internal use.</i>
SetApplicationDirectory	<i>For internal use.</i>
SetCOMDLLCalcRules	<i>For internal use.</i>
SetCurrentActivity	Sets the activity for the current user.
SetCurrentModule	Sets the current module using a module name.
SetCurrentModuleEx	Sets the current module using a module ID.
SetExtractFileEncoding	Sets the file encoding type for extracted files.
SetLanguageUserParameters	Sets the language in which member descriptions are displayed for the connected user.
SetNumberFormattingUserParameters	Sets the user's decimal and thousands separator characters.
SetResourceLanguageForCurrentSession	Sets the language for a user's resource strings in the current session. The specified language does not persist beyond the current session.
SetResourceLanguageUserParameters	Sets the user's default language for resource strings such as error messages and other strings that are generated on the server.
SetUserParameter	Sets a parameter for the user. Use <code>SetUserParameter</code> to create and edit custom parameters for items such as personal preferences.
SetVBScriptCalcRules	Validates and loads a rules file; a flag determines whether <code>SetVBScriptCalcRules</code> loads after validation or validates without loading.

Method	Description
SetVBScriptMemberListRules	Validates and loads a member lists file; a flag determines whether SetVBScriptMemberListRules loads after validation or validates without loading.
StopRunningTask	Stops a given task.
UpdateRunningTaskLogFilePathName	<i>For internal use.</i>
UpdateRunningTaskPOV	<i>For internal use.</i>
UpdateRunningTaskProgress	<i>For internal use.</i>
UpdateRunningTaskProgressDetails	<i>For internal use.</i>
UpdateRunningTaskStatus	<i>For internal use.</i>
UpdateUserAppPreferences	<i>For internal use.</i>
WarnUsersForShutDown	<i>For internal use.</i>

For detailed descriptions of these methods, see [“HsvSystemInfo Object Methods” on page 491.](#)

HsvProcessFlow Type Library Overview

The HsvProcessFlow type library contains one object, the HsvProcessFlow object. Use the HsvProcessFlow object to execute Process Management-related features such as taking actions for and getting histories of process units and submission phases.

The HsvProcessFlow object is a child of the HsvSession object. Use HsvSession’s `ProcessFlow` property to assign HsvProcessFlow object references as shown in the following example:

```
Dim cHsvProcessFlow As HsvProcessFlow
Set cHsvProcessFlow = cHsvSession.ProcessFlow
```

The following table lists the methods of the HsvProcessFlow object.

Table 29 HsvProcessFlow Object Methods

Method	Description
Approve	Approves a process unit.
Approve2	Approves a process unit, and optionally attaches documents and approves all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.
ApproveEx	Approves a process unit, and optionally approves all other process units that consist of the specified Scenario, Year,

Method	Description
	Entity, and Value dimension members and the other base Period dimension members.
GetGroupPhaseFromCell	Returns the IDs of the submission group and submission phase to which a given cell is assigned.
GetHistory	Returns arrays containing the dates and times, users, actions, states, and comments of a process unit.
GetHistory2	Returns arrays containing the history of a process unit; the process unit's dates and times, users, actions, levels, comments, and names and paths of attached document are returned.
GetState	Returns the current state of a process unit.
GetPhasedSubmissionHistory	Returns arrays containing the history of a submission phase, given the member IDs of a cell in the phase.
GetPhasedSubmissionState	Returns the current review level of a given cell's submission phase.
GetPhasedSubmissionStateUsingPhaseID	Returns the review level of a submission phase, given either the member IDs of a cell in the phase or the phase ID.
PhasedSubmissionApprove	Approves a submission phase, given the member IDs of a cell in the phase.
PhasedSubmissionApprove2	Approves a submission phase and optionally attaches documents, given either the member IDs of a cell in the phase or the phase ID.
PhasedSubmissionApproveEx	Approves a submission phase, and optionally applies the action to submission phases for all base periods that intersect with the specified non-Period dimension members, given the member IDs of a cell in the phase.
PhasedSubmissionGetHistory2	Returns arrays containing the history of a submission phase, given the member IDs of a cell in the phase.
PhasedSubmissionGetHistory2UsingPhaseID	Returns arrays containing the history of a submission phase, given either the member IDs of a cell in the phase or the phase ID.
PhasedSubmissionProcessManagementChangeStateForMultipleEntities2	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.
PhasedSubmissionPromote	Promotes a submission phase to a specified review level, given the member IDs of a cell in the phase.
PhasedSubmissionPromote2	Promotes a submission phase and optionally attaches documents, given either the member IDs of a cell in the phase or the phase ID.

Method	Description
PhasedSubmissionPublish	Publishes a submission phase, given the member IDs of a cell in the phase.
PhasedSubmissionPublish2	Publishes a submission phase and optionally attaches documents, given either the member IDs of a cell in the phase or the phase ID.
PhasedSubmissionPublishEx	Publishes a submission phase, and optionally applies the action to submission phases for all base periods that intersect with the specified non-Period dimension members, given the member IDs of a cell in the phase.
PhasedSubmissionReject	Rejects a submission phase, given the member IDs of a cell in the phase.
PhasedSubmissionReject2	Rejects a submission phase and optionally attaches documents, given either the member IDs of a cell in the phase or the phase ID.
PhasedSubmissionSignOff	Signs off on a submission phase, given the member IDs of a cell in the phase.
PhasedSubmissionSignOff2	Signs off on a submission phase and optionally attaches documents, given either the member IDs of a cell in the phase or the phase ID.
PhasedSubmissionStart	Starts a submission phase, given the member IDs of a cell in the phase.
PhasedSubmissionStart2	Starts a submission phase and optionally attaches documents, given either the member IDs of a cell in the phase or the phase ID.
PhasedSubmissionStartEx	Starts a submission phase, and optionally applies the action to submission phases for all base periods that intersect with the specified non-Period dimension members, given the member IDs of a cell in the phase.
PhasedSubmissionSubmit	Submits a submission phase, given the member IDs of a cell in the phase.
PhasedSubmissionSubmit2	Submits a submission phase and optionally attaches documents, given either the member IDs of a cell in the phase or the phase ID.
ProcessManagementChangeStateForMultipleEntities	Applies a given process management action to process units for one or more Entity dimension members.
ProcessManagementChangeStateForMultipleEntities2	Applies a given process management action and attaches documents to process units for one or more Entity dimension members.
ProcessManagementChangeStateForMultipleEntitiesEx	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.

Method	Description
Promote	Promotes a process unit to a specified review level.
Promote2	Promotes a process unit to a specified review level, and provides the option of attaching one or more documents.
Publish	Publishes a process unit.
Publish2	Publishes a process unit, and optionally attaches documents and publishes all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.
PublishEx	Publishes a process unit, and optionally publishes all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.
Reject	Demotes a process unit from its current state to its previous state.
Reject2	Demotes a process unit from its current level to its previous level and optionally attaches documents.
SignOff	Signs off on a process unit.
SignOff2	Signs off on a process unit and attaches documents.
Start	Starts a process unit.
Start2	Starts a process unit, and optionally attaches documents and starts all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.
StartEx	Starts a process unit, and optionally starts all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.
Submit	Submits a process unit.
Submit2	Submits a process unit and provides the option to attach one or more documents.

For detailed descriptions of these methods, see [Chapter 14, “HsvProcessFlow Type Library.”](#)

HsvReports Type Library Overview

The HsvReports type library contains one object, the HsvReports object. Use the HsvReports object to execute Reporting-related features such as getting report information and maintaining reports.

The HsvReports object is a child of the HsvSession object. Use HsvSession's Reports property to assign HsvReports object references as shown in the following example:

```
Dim cHsvReports As HsvReports
Set cHsvReports = cHsvSession.Reports
```

The following table lists the methods of the HsvReports object.

Table 30 HsvReports Object Methods

Method	Description
CheckReportSecurityClass	Indicates whether the connected user has either Read or All access rights to a report's security class.
CheckReportSecurityClass2	Indicates whether the connected user has a given level of access rights to a report's security class.
CheckSecurityRole	Indicates whether the connected user is assigned to the security role that enables users to save reports of a given report type.
DeleteDocuments	Deletes documents from the application server
DeleteReport	Deletes a report from the application server.
DeleteReports	Deletes one or more reports from the application server.
EnumDocuments	Returns the names, descriptions, timestamps, and security class IDs of documents that meet the search criteria.
EnumDocumentsEx	Returns the names, descriptions, timestamps, security class IDs, privacy flags, folder content types, owners, file types, and document types of documents that meet the search criteria. You can also filter for public or private documents.
EnumReports	Returns the names, descriptions, and timestamps of reports on the application server.
GetDocument	Returns the definition of a document, as well as the document's security class ID and description. The document definition is returned as an array of bytes.
GetDocumentEx	Returns the definition of a document, as well as other properties such as the document's type, file type, and security class.
GetReport	Returns the definition of a report, as well as the report's security class ID and description. The report definition is returned as an array of bytes.
SaveDocument	Saves a document to the application server.
SaveDocumentEx	Saves a document to the application server and specifies the document's content type and privacy flag.
SetReport	Saves a report to the application server.

For detailed descriptions of these methods, see [Chapter 15, "HsvReports Type Library."](#)

HsvMDArrays Type Library Overview

The HsvMDArrays type library contains two objects that supplement the HsvData object by providing methods to manage arrays of cells: the HsvMDDataBuffer and HsvMDDataBufferLite. Both objects provide virtually identical functionality, but differ in how they cache subcubes:

- The HsvMDDataBuffer object caches the cells' subcubes to RAM. The object's methods are summarized in [“HsvMDDataBuffer Object Overview” on page 93](#).
- The HsvMDDataBufferLite object caches a minimum number of subcubes to RAM and the other subcubes to disk. The object's methods are summarized in [“HsvMDDataBufferLite Object Overview” on page 96](#).

The HsvMDArrays type library also contains the following objects and interfaces:

- The IHsvMDDataBufferLite interface can optionally be used to change the minimum number of subcubes cached to RAM by the HsvMDDataBufferLite object. For a summary, see [“IHsvMDDataBufferLite Interface” on page 99](#).
- The HsvTransactionData object provides access to the transaction data generated by statutory consolidations. The object's methods are summarized in [“HsvTransactionData Object Overview” on page 99](#).
- The HsvMDIndexList object contains an array of indexes that represent points-of-view in an HsvMDDataBuffer or HsvMDDataBufferLite object. For more information, see [“HsvMDIndexList Object Overview” on page 100](#)
- The HsvICTransactionsData object supplements the HsvICM object by providing access to intercompany transactions. For more information, see [“HsvICTransactionsData Object Overview” on page 100](#)

Tip: Older releases of Financial Management did not include the HsvMDDataBuffer object; those releases included the similar HsvMDCube object. In release 1.2, the HsvMDCube object was deprecated and the HsvMDDataBuffer object was added.

HsvMDDataBuffer Object Overview

The HsvMDDataBuffer object supplements the HsvData object by providing methods to manage arrays of cells, caching the cells' subcubes to RAM. Note that the cells in an HsvMDDataBuffer object are not stored in an application.

The HsvData object provides the following methods to move data between an application and an HsvMDDataBuffer object:

- `UpdateDataUsingMDDataBuffer` inserts an HsvMDDataBuffer object's cells into an application.
- `AddDataToMDDataBuffer` inserts an application's cell into an HsvMDDataBuffer object.

Create HsvMDDataBuffer object references with the following syntax. Note that you must include the `New` keyword, otherwise an error will occur:

```
Dim cHsvMDDataBuffer As HsvMDDataBuffer
Set cHsvMDDataBuffer = New HSVMDARRAYSLib.HsvMDDataBuffer
```

The following table summarizes the HsvMDDataBuffer object's methods.

Table 31 HsvMDDataBuffer Object Methods

Method	Description
BeginEnumeration	Locks an HsvMDDataBuffer object, and returns the number of subcubes contained by the object.
CreateDataIndexList	Creates an HsvMDIndexList object based upon the items in an HsvMDDataBuffer object.
CreateDataIndexListEx	<i>For internal use.</i>
EndEnumeration	Unlocks an HsvMDDataBuffer object that has been locked by <code>BeginEnumeration</code> .
EraseRecordFromPMBuffer	<i>For internal use.</i>
GetCheckLineItemDetailsForCaseInsensitiveDuplicates	Indicates whether case-insensitive duplicate checking is enabled for the HsvMDDataBuffer instance.
GetCubeIndexFromPOV	Returns the index of a subcube in an HsvMDDataBuffer object, given the IDs of the subcube's dimension members.
GetCubePOVFromIndex	Returns the IDs of a subcube's dimension members, given the subcube's index.
GetData	Returns a cell's data, given the member IDs that identify the cell.
GetDataAtIndex	Returns the data for a cell, given the indexes that identify the cell.
GetDescription	Returns a cell's description, given the member IDs that identify the cell.
GetDescriptionAtIndex	Returns the description for a cell, given the indexes that identify the cell.
GetLineItems	Returns the data and descriptions for a cell's line items, given the IDs that identify the cell's dimension members.
GetLineItemsAtIndex	Returns the data and descriptions for a cell's line items, given the indexes that identify the cell.
GetNumCellsForData	Returns the number of cells that are in a period of a subcube and that contain data.
GetNumCellsForDescriptions	Returns the number of cells that are in a period of a subcube and that contain descriptions.
GetNumCellsForLineItems	Returns the number of cells that are in a period of a subcube and that contain line items.
GetNumPeriodsInCubeForData	Returns the number of periods that are in a subcube and that contain cells with data.

Method	Description
GetNumPeriodsInCubeForDescriptions	Returns the number of periods that are in a subcube and that contain cells with descriptions.
GetNumPeriodsInCubeForLineItems	Returns the number of periods that are in a subcube and that contain cells with line items.
GetPeriodIndexFromPOVForData	Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain data.
GetPeriodIndexFromPOVForDescriptions	Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain descriptions.
GetPeriodIndexFromPOVForLineItems	Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain line items.
GetPeriodPOVFromIndexForData	Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain data.
GetPeriodPOVFromIndexForDescriptions	Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain descriptions.
GetPeriodPOVFromIndexForLineItems	Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain line items.
GetPMErrrorRecordCount	<i>For internal use.</i>
GetPMRecordCount	<i>For internal use.</i>
GetRecordFromPMBuffer	<i>For internal use.</i>
GetRecordFromPMErrrorBuffer	<i>For internal use.</i>
GetSortedNature	<i>For internal use.</i>
InsertDataAtBeginning	Inserts data for a cell at the top of an HsvMDDataBuffer object.
InsertDataAtEnd	Inserts data for a cell at the bottom of an HsvMDDataBuffer object.
InsertDescriptionAtBeginning	Inserts a cell description at the top of an HsvMDDataBuffer object.
InsertDescriptionAtEnd	Inserts a cell description at the bottom of an HsvMDDataBuffer object.
InsertLineItemsAtBeginning	Inserts line items for a cell at the top of an HsvMDDataBuffer object.
InsertLineItemsAtEnd	Inserts line items for a cell at the bottom of an HsvMDDataBuffer object.

Method	Description
InsertRecordIntoPMBuffer	<i>For internal use.</i>
InsertRecordIntoPMErrorsBuffer	<i>For internal use.</i>
RemoveAll	Clears all the cells from an HsvMDDataBuffer object.
SetCheckLineItemDetailsForCaseInsensitiveDuplicates	Specifies whether the system should check for case-insensitive duplicate line item descriptions.
SetData	Inserts a cell's data into an HsvMDDataBuffer object.
SetDescription	Inserts a cell's description into an HsvMDDataBuffer object.
SetGrowByAmount	Changes the amount of memory that the HsvMDDataBuffer object automatically allocates.
SetLineItems	Inserts a cell's line items into an HsvMDDataBuffer object.
SetSortedNature	<i>For internal use.</i>
Sort	<i>For internal use.</i>

HsvMDDataBufferLite Object Overview

The HsvMDDataBufferLite object supplements the HsvData object by providing methods to manage arrays of cells, caching a minimum number of subcubes to RAM and the other subcubes to disk. Note that the cells in an HsvMDDataBufferLite object are not stored in an application.

Note: By default, the HsvMDDataBufferLite object caches a minimum of one subcube to RAM. You can change the minimum number of subcubes that will be cached; for more information, see [“IHsvMDDataBufferLite Interface” on page 99](#).

The HsvData object provides the following methods to move data between an application and an HsvMDDataBufferLite object:

- `UpdateDataUsingMDDataBuffer` inserts an HsvMDDataBufferLite object's cells into an application.
- `AddDataToMDDataBuffer` inserts an application's cell into an HsvMDDataBufferLite object.

Create HsvMDDataBufferLite object references with the following syntax. Note that you must include the `New` keyword, otherwise an error will occur:

```
Dim cDataBufferLite As HsvMDDataBufferLite
Set cDataBufferLite = New HSVMDARRAYSLib.HsvMDDataBufferLite
The following table summarizes the HsvMDDataBufferLite object's methods.
```


Table 32 HsvMDDataBufferLite Object Methods

Method	Description
BeginEnumeration	Locks an HsvMDDataBufferLite object, and returns the number of subcubes contained by the object.
CreateDataIndexList	Creates an HsvMDIndexList object based upon the items in an HsvMDDataBufferLite object.
EndEnumeration	Unlocks an HsvMDDataBufferLite object that has been locked by <code>BeginEnumeration</code> .
EraseRecordFromPMBuffer	<i>For internal use.</i>
GetCheckLineItemDetailsForCaseInsensitiveDuplicates	Indicates whether case-insensitive duplicate checking is enabled for the HsvMDDataBufferLite instance.
GetCubeIndexFromPOV	Returns the index of a subcube in an HsvMDDataBufferLite object, given the IDs of the subcube's dimension members.
GetCubePOVFromIndex	Returns the IDs of a subcube's dimension members, given the subcube's index.
GetData	Returns a cell's data, given the member IDs that identify the cell.
GetDataAtIndex	Returns the data for a cell, given the indexes that identify the cell.
GetDescription	Returns a cell's description, given the member IDs that identify the cell.
GetDescriptionAtIndex	Returns the description for a cell, given the indexes that identify the cell.
GetLineItems	Returns the data and descriptions for a cell's line items, given the IDs that identify the cell's dimension members.
GetLineItemsAtIndex	Returns the data and descriptions for a cell's line items, given the indexes that identify the cell.
GetNumCellsForData	Returns the number of cells that are in a period of a subcube and that contain data.
GetNumCellsForDescriptions	Returns the number of cells that are in a period of a subcube and that contain descriptions.
GetNumCellsForLineItems	Returns the number of cells that are in a period of a subcube and that contain line items.
GetNumPeriodsInCubeForData	Returns the number of periods that are in a subcube and that contain cells with data.
GetNumPeriodsInCubeForDescriptions	Returns the number of periods that are in a subcube and that contain cells with descriptions.
GetNumPeriodsInCubeForLineItems	Returns the number of periods that are in a subcube and that contain cells with line items.

Method	Description
GetPeriodIndexFromPOVForData	Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain data.
GetPeriodIndexFromPOVForDescriptions	Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain descriptions.
GetPeriodIndexFromPOVForLineItems	Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain line items.
GetPeriodPOVFromIndexForData	Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain data.
GetPeriodPOVFromIndexForDescriptions	Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain descriptions.
GetPeriodPOVFromIndexForLineItems	Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain line items.
GetPMErrrorRecordCount	<i>For internal use.</i>
GetPMRecordCount	<i>For internal use.</i>
GetRecordFromPMBuffer	<i>For internal use.</i>
GetRecordFromPMErrrorBuffer	<i>For internal use.</i>
GetSortedNature	<i>For internal use.</i>
InsertDataAtEnd	Inserts data for a cell at the bottom of an HsvMDDataBufferLite object.
InsertDescriptionAtEnd	Inserts a cell description at the bottom of an HsvMDDataBufferLite object.
InsertLineItemsAtEnd	Inserts line items for a cell at the bottom of an HsvMDDataBufferLite object.
InsertRecordIntoPMBuffer	<i>For internal use.</i>
InsertRecordIntoPMErrrorBuffer	<i>For internal use.</i>
RemoveAll	Clears all the cells from an HsvMDDataBufferLite object.
SetCheckLineItemDetailsForCaseInsensitiveDuplicates	Specifies whether the system should check for case-insensitive duplicate line item descriptions.
SetData	Inserts a cell's data into an HsvMDDataBufferLite object.
SetDescription	Inserts a cell's description into an HsvMDDataBufferLite object.

Method	Description
SetGrowByAmount	Changes the amount of memory that the HsvMDDataBufferLite object automatically allocates.
SetLineItems	Inserts a cell's line items into an HsvMDDataBufferLite object.
SetSortedNature	<i>For internal use.</i>
Sort	<i>For internal use.</i>

IHsvMDDataBufferLite Interface

The IHsvMDDataBufferLite interface provides the `SetMinCubesInCache` method, which sets the number of minimum subcubes cached to RAM for the HsvMDDataBufferLite object. For details, see [“IHsvMDDataBufferLite Interface” on page 667](#).

HsvTransactionData Object Overview

The HsvTransactionData object's methods provide access to the transaction data generated by statutory consolidations. The HsvData object's `GetTransactionData` method populates an HsvTransactionData object with an array of data; the HsvTransactionData object methods set selection criteria for the data to be returned, and enumerate the array with which the object has been populated.

Create HsvTransactionData object references with the following syntax. Note that you must include the `New` keyword, otherwise an error will occur:

```
Dim cHsvTransData As HsvTransactionData
Set cHsvTransData = New HSVMDDARRAYSLib.HsvTransactionData
```

The following table summarized the HsvTransactionData object's methods.

Table 33 HsvTransactionData Object Methods

Method	Description
BeginDataEnum	Begins an enumeration of an HsvTransactionData object, and returns a count of the items with which <code>HsvData.GetTransactionData</code> has populated the object.
BeginQueryEnum	<i>For internal use.</i>
EndDataEnum	Ends an HsvTransactionData object enumeration. You should call <code>EndDataEnum</code> after you have finished working with an enumeration that was started with <code>BeginDataEnum</code> .
EndQueryEnum	<i>For internal use.</i>
GetFixedDimensionMembers	<i>For internal use.</i>
GetQueryItem	<i>For internal use.</i>

Method	Description
GetTransactionData	Returns source and destination data for a transaction, given the index of the transaction within the HsvTransactionData object's array of transactions. In addition to the data, the member IDs of the transaction's dimension members are returned.
Initialize	Specifies the Scenario and Year dimension members for the HsvTransactionData object's transaction data. You must call <code>Initialize</code> before calling the other HsvTransactionData object methods.
SetAccessRight	<i>For internal use.</i>
SetQueryItem	Specifies a transaction dimension member as a selection criterion for an HsvTransactionData object. To set multiple selection criteria for an HsvTransactionData object, make one <code>SetQueryItem</code> call per selection criterion.
SetTransactionData	<i>For internal use.</i>

HsvMDIndexList Object Overview

An HsvMDIndexList object contains an array of indexes that represent points-of-view in an HsvMDDataBuffer or HsvMDDataBufferLite object.

Object references are obtained with `CreateDataIndexList`, which is contained by both the HsvMDDataBuffer and HsvMDDataBufferLite objects.

The following table lists the HsvMDIndexList object's methods.

Table 34 HsvMDIndexList Object Methods

Method	Description
GetItem	Returns the member IDs of the dimension members for an index in an HsvMDIndexList object.
GetNumItems	Returns the number of indexes in an HsvMDIndexList object. Use this to loop with <code>GetItem</code> .

HsvICTransactionsData Object Overview

The HsvICTransactionsData object provides methods for working with intercompany transactions. An HsvICTransactionsData instance contains an array of intercompany transactions for a given scenario, year, and period, and enables you to access specific transactions and to process all of the array's transactions.

For details on using the object and its methods, see [“HsvICTransactionsData Object Methods” on page 674](#). The following table summarized the HsvICTransactionsData object's methods.

Table 35 HsvICTransactionsData Object Methods

Method	Description
AddAccountCustomCombination	<i>For internal use.</i>
AddICTransactionData	<i>For internal use.</i>

Method	Description
AddQueryDimensionMemberID	<i>For internal use.</i>
AddQueryField	<i>For internal use.</i>
AddQueryFieldItem	<i>For internal use.</i>
AddQueryOrderField	<i>For internal use.</i>
BeginDataEnum	Begins the enumeration of an HsvICTransactionsData instance's array of transactions, and returns a count of the transactions that the array contains.
BeginQueryFieldEnum	<i>For internal use.</i>
BeginQueryOrderEnum	<i>For internal use.</i>
EndDataEnum	Cleans up an HsvICTransactionsData instance; you should always call <code>EndDataEnum</code> after you finish working with an enumeration of transactions for which you have called <code>BeginDataEnum</code> .
EndQueryFieldEnum	<i>For internal use.</i>
EndQueryOrderEnum	<i>For internal use.</i>
GetAccessRights	Gets the user's read and write access rights to an intercompany transaction.
GetAccountCustomCombination	<i>For internal use.</i>
GetAccountCustomCombinationTotal	<i>For internal use.</i>
GetEntityPartnerOption	<i>For internal use.</i>
GetErrorStatus	Returns the HRESULT associated with a given intercompany transaction.
GetFilterOptions	<i>For internal use.</i>
GetFixedDimensionMembers	Returns the member IDs of the Scenario, Year, and Period dimension members for an HsvICTransactionsData instance's transactions.
GetICTransactionCell	Returns the member IDs of an intercompany transaction's Entity, Intercompany Partner, Account, and Custom dimension members.
GetICTransactionData	Returns an intercompany transaction's details, including the transaction's amounts, currency, dimension members, and so on.
GetMatchCode	<i>For internal use.</i>
GetNumTransactionsCached	<i>For internal use.</i>
GetPagingOption	<i>For internal use.</i>
GetPartnerAsEntityList	<i>For internal use.</i>
GetPartnerQueryDimensionMemberIDs	<i>For internal use.</i>
GetQueryDimensionMemberIDs	<i>For internal use.</i>

Method	Description
GetQueryFieldInformation	<i>For internal use.</i>
GetQueryFieldItem	<i>For internal use.</i>
GetQueryOrderField	<i>For internal use.</i>
GetTotalTransactions	<i>For internal use.</i>
GetTransGroupType	<i>For internal use.</i>
Initialize	Specifies the scenario, year, and period of the transactions that the HsvICTransactionsData instance will contain.
InitializeSequenceldMap	<i>For internal use.</i>
IsEntityInPartnerAsEntityList	<i>For internal use.</i>
IsICTransactionValid	<i>For internal use.</i>
RemoveQueryOrder	<i>For internal use.</i>
SetAccessRights	Sets the user's read and write access rights to an intercompany transaction.
SetEntityPartnerOption	<i>For internal use.</i>
SetErrorStatus	<i>For internal use.</i>
SetFilterOptions	<i>For internal use.</i>
SetICTransactionData	<i>For internal use.</i>
SetPagingOption	<i>For internal use.</i>
SetPartnerAsEntityList	<i>For internal use.</i>
SetPartnerQueryDimensionMemberIDs	<i>For internal use.</i>
SetQueryDimensionMemberIDs	<i>For internal use.</i>
SetTotalTransactions	<i>For internal use.</i>
SortByCell	<i>For internal use.</i>
UnInitialize	<i>For internal use.</i>

HsvDataCubes Type Library Overview

The HsvDataCubes type library contains the HsvCurrencyCube and HsvNodeCube objects:

- The HsvCurrencyCube object provides access to currency subcubes. The HsvCurrencyCube methods are listed in [Table 36 on page 103](#).
- The HsvNodeCube object provides access to node subcubes. The HsvNodeCube methods are listed in [Table 37 on page 103](#).

HsvCurrencyCube Object Overview

The HsvCurrencyCube object is used to access currency subcubes, which are described in [“About Subcubes” on page 43](#). The HsvCurrencyCube object is subordinate to the HsvData object: use HsvData’s [GetCurrencyCube](#) method to obtain HsvCurrencyCube object references.

The following table lists the HsvCurrencyCube object’s methods.

Table 36 HsvCurrencyCube Object Methods

Method	Description
BeginEnumerationOfStoredData	Returns a count of the subcube items for a Value dimension member, while also locking the subcube’s cells to prevent users from changing data.
EndEnumerationOfStoredData	Unlocks a subcube that was locked by <code>BeginEnumerationOfStoredData</code> .
GetCell	Returns the data in and transaction status of a cell in a subcube.
GetFixedDimensionMembers	Returns the member IDs of a subcube’s Scenario, Year, and Entity dimension members, as well as the member ID of the subcube’s input Value dimension member.
GetOneCellFromStoredItem	Returns cell information such as a cell’s data and member IDs. You identify the cell with the index number of the subcube item and the member IDs of the cell’s Period and View dimension members.
GetPOVFromStoredItem	Returns the member IDs of the Value, Account, Intercompany Partner, and Custom dimension members of a subcube item. You identify the item with its index number.

For detailed descriptions of these methods, see [“HsvCurrencyCube Object Methods” on page 688](#).

HsvNodeCube Object Overview

The HsvNodeCube object is used to access node subcubes, which are described in [“About Subcubes” on page 43](#). The HsvNodeCube object is subordinate to the HsvData object: use HsvData’s [GetNodeCube](#) method to obtain HsvNodeCube object references.

The following table lists the HsvNodeCube object’s methods.

Table 37 HsvNodeCube Object Methods

Method	Description
BeginEnumerationOfStoredData	Returns a count of the subcube items for a Value dimension member, while also locking the subcube’s cells to prevent users from changing data.
EndEnumerationOfStoredData	Unlocks a subcube that was locked by <code>BeginEnumerationOfStoredData</code> .
GetCell	Returns the data in and transaction status of a cell in a subcube.
GetFixedDimensionMembers	Returns the member IDs of a subcube’s Scenario, Year, and parent and child Entity dimension members.

Method	Description
GetOneCellFromStoredItem	Returns cell information such as a cell's data and member IDs. You identify the cell with the index number of the subcube item and the member IDs of the cell's Period and View dimension members.
GetPOVFromStoredItem	Returns the member IDs of the Value, Account, Intercompany Partner, and Custom dimension members of a subcube item. You identify the item with its index number.

For detailed descriptions of these methods, see [“HsvNodeCube Object Methods” on page 693](#).

HsvStarSchemaACM Type Library Overview

The HsvStarSchemaACM type library exposes Extended Analytics functionality, and contains the HsvStarSchemaACM object and the IHsvStarSchemaTemplates interface. For more information, see [“HsvStarSchemaACM Object Overview” on page 104](#) and [“IHsvStarSchemaTemplates Interface Overview” on page 105](#).

HsvStarSchemaACM Object Overview

The HsvStarSchemaACM object enables you to work with Extended Analytics extractions. For example, you can use this object to create, update, and delete star schemas. For information on setting HsvStarSchemaACM object references and details on the object's methods, see [“HsvStarSchemaACM Object Methods” on page 699](#)

The following table lists the HsvStarSchemaACM object's methods.

Table 38 HsvStarSchemaACM Object Methods

Method	Description
CreateStarSchema	Creates or updates a star schema by exporting data for all cells that intersect the specified dimension members.
DeleteStarSchema	Deletes a star schema from a given database.
EnumApplicationStarSchemas	<i>For internal use.</i>
EnumRegisteredDSNs	Returns an array of the Extended Analytics Data Source Names that have been registered on the application server.
GetAsynchronousTaskStatus	Returns status information for the asynchronous thread launched by <code>CreateStarSchema</code> .
GetExtractLogData	Returns a string that provides a log of the HsvStarSchemaACM instance's most recent call to <code>CreateStarSchema</code> .
GetPersistedSettings	<i>For internal use.</i>
QuitAsynchronousTask	Terminates the thread launched by <code>CreateStarSchema</code> .
SetDefaultTablePrefix	<i>For internal use.</i>

Method	Description
SetPersistedSettings	<i>For internal use.</i>
SetSession	Points to the HsvSession object that represents the connection to the application. You must call <code>SetSession</code> before using the other HsvStarSchemaACM methods.
TestSQLConnection	<i>For internal use.</i>

IHsvStarSchemaTemplates Interface Overview

The IHsvStarSchemaTemplates interface enables you to work with Extended Analytics templates. For information on setting IHsvStarSchemaTemplates object references and details on the interface's methods, see [“IHsvStarSchemaTemplates Interface” on page 706](#).

The following table lists the IHsvStarSchemaTemplates interface's methods.

Table 39 IHsvStarSchemaTemplates Interface Methods

Method	Description
DeleteTemplate	Deletes a given Extended Analytics template.
EnumTemplates	Returns the names of the application's Extended Analytics templates.
GetTemplate	Returns an XML string that contains a given Extended Analytics template's definition.
SetTemplate	Creates an Extended Analytics template, using an XML string that contains the template definition.

HsvICM Type Library

The HsvICM type library exposes intercompany transaction processing and administrative features, and contains the HsvICM object and the IHsvAdminICM interface. For more information, see [“HsvICM Object Overview” on page 105](#) and [“IHsvAdminICM Interface Overview” on page 107](#).

HsvICM Object Overview

The HsvICM object enables you to work with intercompany transactions. For example, you can create, match, and unmatched intercompany transactions. For information on setting HsvICM object references and details on the object's methods, see [“HsvICM Object Methods” on page 711](#).

The following table summarizes the HsvICM object's methods.

Table 40 HsvICM Object Methods

Method	Description
AllMVTTransPosted	Indicates whether all transactions for a given scenario, year, period, and entity that are in a matched state have been posted.

Method	Description
CheckReportSecurity	<i>For internal use.</i>
CreateICTransaction	Creates an intercompany transaction for a given cell.
DeleteICReasonCode	Deletes a reason code.
DoesCellSupportICTransactionDetail	Indicates whether a cell supports intercompany transactions.
GetColumnFilter	<i>For internal use.</i>
GetCurrencyInfo	Returns a currency's label, translation operator, and scale.
GetEntityCurrencyID	Returns the currency ID of an entity's default currency.
GetEntitiesContacts	Returns the usernames and security identifiers of the users to be alerted for a given scenario and intercompany transaction-related event for the specified entities.
GetICReasonCodeID	Gets the ID of a reason code.
GetICReasonCodeLabel	Returns a reason code from the code's internal ID.
GetICReasonCodes	Returns an application's reason codes and their corresponding IDs and descriptions.
GetICTransactions	Populates an HsvICTransactionsData object reference with intercompany transactions for the scenario, year, and period specified with the HsvICTransactionsData method Initialize .
GetMonitorICDetails	Returns counts of intercompany transactions that have various posting and matching statuses for the specified Scenario, Year, Period, and Entity dimension members.
GetMonitorICSummary	Returns the number of Entity dimension members that have various combinations of locking and process statuses for the given Scenario, Year, Period, and Entity dimension members.
GetMonitorICTransactions	Indicates whether one or more Entity dimension members have intercompany transactions for given Scenario, Year, and Period dimension members. For the entities that have intercompany transactions, <code>GetMonitorICTransactions</code> returns an array that consists of the entities' member IDs, process and locking statuses, and usernames and timestamps for the most recently modified transactions.
GetRowFilter	<i>For internal use.</i>
GetTransCurrencyID	Returns the currency ID of a given currency.
GetUnMatchTransactions	Populates an HsvICTransactionsData object reference with unmatched intercompany transactions for the scenario, year, and period specified with the HsvICTransactionsData method Initialize .
IsOneSideOfTransactionGroupWriteable	Indicates whether the user has write access to either the entity or Intercompany Partner cell of all transactions in a given range of an HsvICTransactionsData object reference's transactions.
MatchAutoAccounts	Matches intercompany transactions by account.

Method	Description
MatchAutoIDs	Matches intercompany transactions by Transaction ID or Reference ID.
NoTransMatchedOrPosted	Indicates whether matched or posted transactions exist for a given scenario, year, and period.
ProcessAllICTrans	Deletes, posts, unposts, or unmatches all transactions for a given scenario, year, and period.
ProcessICTransactions	Deletes, posts, unposts, or unmatches the specified intercompany transactions for a given scenario, year, and period.
SaveColumnFilter	<i>For internal use.</i>
SaveICReasonCode	Creates a reason code.
SaveICTransaction	Updates an existing transaction.
SaveRowFilter	<i>For internal use.</i>
SetReasonCodeToICTransactions	Sets the reason code for one or more intercompany transactions.
UnMatchICTransactions	Unmatches transactions to which the specified match codes have been assigned.

IHsvAdminICM Interface Overview

The IHsvAdminICM interface enables you to programmatically administer intercompany transactions. For example, you can use this interface to open and close periods, lock and unlock entities, and set and get period settings. For information on setting IHsvAdminICM object references and details on the interface's methods, see [“IHsvAdminICM Interface Methods” on page 732](#).

The following table summarizes the IHsvAdminICM interface's methods.

Table 41 IHsvAdminICM Interface Methods

Method	Description
CheckSecurityForICExtract	Indicates whether the connected user has security rights for extracting intercompany transactions.
CloseICPeriod	Closes a given period for intercompany transactions.
GetICEntitiesLockStatus	Indicates whether any entities are locked for a given scenario, year, and period.
GetLockStatusICEntities	<i>Deprecated</i> - use <code>GetICEntitiesLockStatus</code> .
GetLockStatusICEntity	Indicates whether an entity is locked or unlocked for a given scenario, period, and year.
GetSettingsICPeriod	Returns the period status, matching tolerance, and Match/Validate Before Post settings for a scenario, year, and period.

Method	Description
GetSettingsICPeriods	Returns the period status, matching tolerance, and Match/Validate Before Post settings of multiple periods for a scenario and year.
LoadICTransactions	Loads or scans intercompany transactions from arrays that specify the transactions' dimension members and data.
LockICEntity	Locks an entity for a given scenario, year, and period.
OpenICPeriod	Opens a period for a given scenario and year, and specifies the period's matching tolerance and Match/Validate Before Post settings.
OpenICPeriod2	Opens a period for a given scenario and year, and specifies the period's Match/Validate Before Post settings and absolute or percentage matching tolerance.
SavePeriodsSettings	<i>For internal use.</i>
UnlockICEntity	Unlocks an entity for a given scenario, year, and period.
UpdatePeriodSettings	Updates an open period's matching tolerance and Match/Validate Before Post settings for a given scenario and year.
UpdatePeriodSettings2	Updates an open period's Match/Validate Before Post settings and absolute or percentage matching tolerance for a given scenario and year.

Type Libraries for Loading and Extracting Information

Financial Management provides type libraries for loading and extracting various types of information. These type libraries are listed in the following table:

Table 42 Type Libraries for Loading and Extracting

Information Being Loaded/Extracted	Type Library
Security	HsvSecurityLoadACV
Metadata (extracts only)	HsvMetadataLoadACV
Rules and member lists	HsvRulesLoadACV
Data	HsvcDataLoad
Journals and journal templates	HsvJournalLoadACV

These type libraries are client tier Application Components. Use these type libraries to load information from and extract information to client computers.

The HsvSecurityLoadACV, HsvMetadataLoadACV, HsvcDataLoad, and HsvJournalLoadACV objects in these type libraries expose the IHsvLoadExtractOptions and IHsvLoadExtractOption interfaces, which are used to specify load and extract options. Note that these interfaces are *not* exposed by the HsvRulesLoadACV object.

The following topics introduce you to the objects for loading and extracting information:

- [“Applications and the Load and Extract Objects” on page 109](#)
- [“IHsvLoadExtractOptions Interface Overview” on page 109](#)
- [“IHsvLoadExtractOption Interface Overview” on page 110](#)
- [“HsvSecurityLoadACV Type Library Overview” on page 110](#)
- [“HsvMetadataLoadACV Type Library Overview” on page 111](#)
- [“HsvRulesLoadACV Type Library Overview” on page 112](#)
- [“HsvcDataLoad Type Library Overview” on page 113](#)
- [“HsvJournalLoadACV Type Library Overview” on page 114](#)

Applications and the Load and Extract Objects

To use the `HsvSecurityLoadACV`, `HsvMetadataLoadACV`, `HsvcDataLoad`, and `HsvJournalLoadACV` objects, you first open the application for which information is being loaded or extracted. You must then point to this application by calling `SetSession`. These objects all implement `SetSession`, which takes the `HsvSession` object returned by `OpenApplication`. After calling `SetSession`, you can then use the other properties and methods of these objects.

Note: Assign `HsvSecurityLoadACV`, `HsvMetadataLoadACV`, `HsvcDataLoad`, and `HsvJournalLoadACV` object references with the `Set` keyword.

IHsvLoadExtractOptions Interface Overview

Each load or extract option is represented by an instance of the `IHsvLoadExtractOption` interface. The `IHsvLoadExtractOptions` interface provides access to these `IHsvLoadExtractOption` instances. The `IHsvLoadExtractOptions` interface is a read-only collection of the `IHsvLoadExtractOption` instances available for a given type of load or extract operation.

Tip: For a more detailed introduction to the `IHsvLoadExtractOptions` and `IHsvLoadExtractOption` interfaces, see [“Load and Extract Option Interfaces” on page 743](#).

The `IHsvLoadExtractOptions` interface’s properties enable you to access and iterate through the available `IHsvLoadExtractOption` instances. The `IHsvLoadExtractOptions` interface is exposed in the `HsvSecurityLoadACV`, `HsvMetadataLoadACV`, `HsvcDataLoad`, and `HsvJournalLoadACV` objects.

The following table lists the properties of the `IHsvLoadExtractOptions` interface.

Table 43 IHsvLoadExtractOptions Interface Properties

Property	Description
Count	Returns a count of the IHsvLoadExtractOption instances available for setting load or extract options. In other words, this property returns a count of the available load or extract options.
Item	Returns object references to instances of the IHsvLoadExtractOption interface. In other words, this property provides access to the available load or extract options.

For detailed descriptions of these properties, see [“IHsvLoadExtractOptions Interface Properties” on page 744](#).

IHsvLoadExtractOption Interface Overview

Each load or extract option is represented by an instance of the IHsvLoadExtractOption interface. The IHsvLoadExtractOption interface’s properties get and set the values of load and extract options, and also get various types of option-related information. The IHsvLoadExtractOption interface is exposed in the HsvSecurityLoadACV, HsvMetadataLoadACV, HsvcDataLoad, and HsvJournalLoadACV objects.

Assign object references to instances of the IHsvLoadExtractOption interface with the IHsvLoadExtractOptions interface’s `Item` property. For more information, see [“Item” on page 744](#).

The following table lists the properties of the IHsvLoadExtractOption interface.

Table 44 IHsvLoadExtractOption Interface Properties

Property	Description
CurrentValue	Sets or returns the current value of a load or extract option.
DefaultValue	Returns the default value of a load or extract option.
MaxValue	For some load or extract options, <code>MaxValue</code> returns the option’s maximum valid value. Note that <code>MaxValue</code> is not supported by all load or extract options.
MinValue	For some load or extract options, <code>MinValue</code> returns the option’s minimum valid value. Note that <code>MinValue</code> is not supported by all load or extract options.
Name	Returns the name of a load or extract option. You can pass this name to <code>Item</code> .
OptionID	Returns the numeric ID of a load or extract option. You can pass this ID to <code>Item</code> .
ValidationList	Returns the valid values of certain load and extract options.

For detailed descriptions of these properties, see [“IHsvLoadExtractOption Interface Properties” on page 745](#).

HsvSecurityLoadACV Type Library Overview

The HsvSecurityLoadACV type library contains the HsvSecurityLoadACV object, which you use to load and extract security information. The HsvSecurityLoadACV type library also exposes

the `HsvLoadExtractOptions` and `HsvLoadExtractOption` interfaces, which you use to specify load and extract options.

For an overview of how to use this type library to load or extract metadata, see the following topics:

- [“Loading Security Information” on page 753](#)
- [“Extracting Security Information” on page 754](#)

Note: Assign `HsvSecurityLoadACV` object references with the `Set` keyword.

The following table lists the `HsvSecurityLoadACV` object’s properties, and [Table 46](#) lists the `HsvSecurityLoadACV` object’s methods.

Table 45 `HsvSecurityLoadACV` Object Properties

Property	Description
<code>ExtractOptions</code>	Returns an object reference to an instance of the <code>IHsvLoadExtractOptions</code> interface for an extraction operation. You must set this property before calling <code>Extract</code> .
<code>LoadOptions</code>	Returns an object reference to an instance of the <code>IHsvLoadExtractOptions</code> interface for a load operation. You must set this property before calling <code>Load</code> .

The above properties are also members of other type libraries, and are described in detail in [“Common Load and Extract Properties” on page 748](#).

The following table summarizes the `HsvSecurityLoadACV` object’s methods.

Table 46 `HsvSecurityLoadACV` Object Methods

Method	Description
<code>Extract</code>	Extracts security information to a text file on the client computer.
<code>Load</code>	Loads security information from a text file on a client computer.
<code>SetSession</code>	Points to the <code>HsvSession</code> object for the application in which security information is being loaded or extracted. Caution! You must call <code>SetSession</code> before using any of the <code>HsvSecurityLoadACV</code> object’s other properties or methods.

The `Extract` and `SetSession` methods are also members of other type libraries, and are described in [“Common Load and Extract Methods” on page 749](#). The `HsvSecurityLoadACV`’s `Load` method is slightly different than the `Load` method of other load- and extract-related type libraries; for details, see [“Load” on page 758](#).

HsvMetadataLoadACV Type Library Overview

The `HsvMetadataLoadACV` type library contains the `HsvMetadataLoadACV` object, which you use to extract metadata. The `HsvMetadataLoadACV` type library also exposes the `HsvLoadExtractOptions` and `HsvLoadExtractOption` interfaces, which you use to specify extract

options. For an overview of how to use this type library, see [“Extracting Metadata ” on page 760](#).

Note: Assign HsvMetadataLoadACV object references with the `Set` keyword.

The following table lists the HsvMetadataLoadACV object’s properties, and [Table 48](#) lists the HsvMetadataLoadACV object’s methods.

Table 47 HsvMetadataLoadACV Object Properties

Property	Description
ExtractOptions	Returns an object reference to an instance of the IHsvLoadExtractOptions interface for an extraction operation. You must set this property before extracting metadata with <code>Extract</code> .
LoadOptions	Returns an object reference to an instance of the IHsvLoadExtractOptions interface for a load operation. You <i>must</i> set this property before calling the <code>Load</code> method for any of the applicable objects.

The previously mentioned properties are also members of other type libraries, and are described in detail in [“Common Load and Extract Properties” on page 748](#).

The following table summarizes the HsvMetadataLoadACV object’s methods.

Table 48 HsvMetadataLoadACV Object Methods

Method	Description
Extract	Extracts metadata to a text file on the client computer.
Load	Loads metadata, data, or journals from a text file into an application, with the text file located on a client computer. For metadata, Load is supported only for Classic applications.
LoadWithAccessCode	<i>For internal use.</i>
SetSession	Points to the HsvSession object for the application in which metadata is being extracted. Caution! You must call <code>SetSession</code> before using any of the HsvMetadataLoadACV object’s other properties or methods.

The previously mentioned methods are also members of other type libraries, and are described in [“Common Load and Extract Methods” on page 749](#).

HsvRulesLoadACV Type Library Overview

The HsvRulesLoadACV type library contains one object, the HsvRulesLoadACV object. HsvRulesLoadACV is used to load and extract rules and member lists.

To assign HsvRulesLoadACV object references, use the `Set` keyword.

The following table lists the methods of the HsvRulesLoadACV object.

Table 49 HsvRulesLoadACV Object Methods

Method	Description
ExtractCalcRules	Extracts rules from an application into a text file on the client computer.
ExtractMemberListRules	Extracts member lists from an application into a text file on the client computer.
GetCalcRulesType	<i>For internal use.</i>
LoadCalcRules	Scans or loads a rules file from a client computer.
LoadCalcRules2	Scans or loads a rules file from a client computer, optionally validating whether the rules violate the referential integrity of any intercompany transactions.
LoadMemberListRules	Validates and loads a member lists file from a client computer. A flag determines whether <code>LoadMemberListRules</code> loads after validation or validates without loading.
ScriptableLoadCalcRules	<i>For internal use.</i>
ScriptableLoadMemberListRules	<i>For internal use.</i>
SetSession	Points to the <code>HsvSession</code> object for the application in which rules or member lists are being loaded or extracted. Caution! You must call <code>SetSession</code> before using any of the <code>HsvRulesLoadACV</code> object's other methods.
SetSessionAndResource	Points to the <code>HsvSession</code> object for the application in which rules or member lists are being loaded or extracted.

For more information on these methods, see [“HsvRulesLoadACV Object Methods” on page 769](#).

HsvcDataLoad Type Library Overview

The `HsvcDataLoad` type library contains the `HsvcDataLoad` object, which you use to load and extract data. The `HsvcDataLoad` type library also exposes the `HsvLoadExtractOptions` and `HsvLoadExtractOption` interfaces, which you use to specify load and extract options. For an overview of how to use this type library to load or extract data, see the following topics:

- [“Loading Data” on page 776](#)
- [“Extracting Data” on page 777](#)

Note: Assign `HsvcDataLoad` object references with the `Set` keyword.

The following table lists the `HsvcDataLoad` object's properties, and [Table 51](#) lists the `HsvcDataLoad` object's methods.

Table 50 HsvcDataLoad Object Properties

Property	Description
ExtractOptions	Returns an object reference to an instance of the IHsvLoadExtractOptions interface for an extraction operation. You must set this property before extracting data with <code>Extract</code> .
LoadOptions	Returns an object reference to an instance of the IHsvLoadExtractOptions interface for a load operation. You must set this property before loading data with <code>Load</code> .

The above properties are also members of other type libraries, and are described in detail in [“Common Load and Extract Properties” on page 748](#).

The following table summarizes the HsvcDataLoad object’s methods.

Table 51 HsvcDataLoad Object Methods

Method	Description
DMELoad	<i>For internal use.</i>
Extract	Extracts data to a text file on the client computer.
Load	Loads data from a text file on the client computer.
Load2	Loads data from a text file on a client computer and returns a flag indicating whether any errors were logged
SetFileForLoad	<i>For internal use.</i>
SetSession	Points to the HsvSession object for the application in which data is being loaded or extracted. Caution! You must call <code>SetSession</code> before using any of the HsvcDataLoad object’s other properties or methods.
StartLoad	<i>For internal use.</i>

`Extract`, `Load`, and `SetSession` are also members of other type libraries, and are described in [“Common Load and Extract Methods” on page 749](#). The other methods are described in [“HsvcDataLoad Object Methods” on page 783](#).

HsvJournalLoadACV Type Library Overview

The HsvJournalLoadACV type library contains the HsvJournalLoadACV object, which you use to load and extract journals and journal templates. The HsvJournalLoadACV type library also exposes the HsvLoadExtractOptions and HsvLoadExtractOption interfaces, which you use to specify load and extract options. For an overview of how to use this type library to load or extract journals and templates, see the following topics:

- [“Loading Journals” on page 785](#)
- [“Extracting Journals” on page 786](#)

Note: Assign HsvJournalLoadACV object references with the `Set` keyword.

The following table lists the HsvJournalLoadACV object’s properties, and [Table 53](#) lists the HsvJournalLoadACV object’s methods.

Table 52 HsvJournalLoadACV Object Properties

Property	Description
ExtractOptions	Returns an object reference to an instance of the IHsvLoadExtractOptions interface for an unfiltered journal extraction. You must set this property before extracting journals with <code>Extract</code> .
ExtractOptionsEx	Returns an object reference to an instance of the IHsvLoadExtractOptions interface for a filtered journal extraction. You must set this property before extracting journals with <code>ExtractEx</code> .
LoadOptions	Returns an object reference to an instance of the IHsvLoadExtractOptions interface for a load operation. You must set this property before loading journals with <code>Load</code> .

The above properties are also members of other type libraries, and are described in detail in “[Common Load and Extract Properties](#)” on page 748.

The following table summarizes the HsvJournalLoadACV object’s methods.

Table 53 HsvJournalLoadACV Object Methods

Method	Description
Extract	Extracts journals and journal templates to a text file on the client computer.
ExtractEx	Extracts journals from an application into a text file on the client computer, applying the filtering criteria set with <code>ExtractOptionsEx</code> .
Load	Loads journals and templates from a text file on the client computer.
SetSession	Points to the HsvSession object for the application in which data is being loaded or extracted. Caution! You must call <code>SetSession</code> before using any of the HsvJournalLoadACV object’s other properties or methods.

The above methods are also members of other type libraries, and are described in “[Common Load and Extract Methods](#)” on page 749.

HsvPOVSelection Type Library

The HsvPOVSelection type library exposes a control that provides a user interface for selecting dimension members. For summaries of the object’s methods and events, see [Table 54](#) on page 115 and [Table 55](#) on page 117.

The following table lists the HsvPOVSelection object’s methods. For detailed descriptions of these methods, see “[HsvPOVSelection Object Methods](#)” on page 796.

Table 54 HsvPOVSelection Object Methods

Method	Description
CheckAllItems	<i>For internal use.</i>

Method	Description
CheckAllItemsUsingSelectedList	Selects or clears all the currently displayed check boxes for a given dimension when the tab is configured for multi-select mode.
EnableDimension	Displays or hides the tab for a given dimension.
GetCheckedItems	Returns the member IDs of the selected members on a given dimension's tab when the tab is configured for multi-select mode.
GetListInfo	Returns the ID of the member list that is currently selected for a given dimension, as well as the member ID of the specified top member.
GetMember	Returns the member ID of the currently selected member for a given dimension when the tab is configured for single-select mode.
GetNumCheckedItems	Returns a count of the check boxes that are selected on a given dimension's tab.
GetOBPInfo	Indicates whether the control is set to display only active Entity dimension members.
Initialize	Displays the HsvPOVSelection control. Note: You must call <code>EnableDimension</code> before calling <code>Initialize</code> .
InitializeEx	<i>For internal use.</i>
InitializeHTTP	<i>For internal use.</i>
InitializeWithConnection	<i>For internal use.</i>
SelectDimension	Displays the tab for a given dimension.
SetCheckedItems	Selects or clears check boxes for the specified dimension members when the tab is configured for multi-select mode.
SetDimUnk	<i>For internal use.</i>
SetListInfo	Specifies the member list to display for a given dimension, as well as the top member of the list to display.
SetMember	Selects a member label on a given dimension's tab when the tab is configured for single-select mode.
SetOBPInfo	Specifies whether the HsvPOVSelection control displays only active Entity dimension members.
SetResourceAndLanguage	Sets the control to display in a given language.
ShowCheckBoxes	Configures a given dimension's tab for multi-select or single-select mode.

The following table lists the HsvPOVSelection control's events. For detailed descriptions of these methods, see [“HsvPOVSelection Object Events” on page 807](#).

Table 55 HsvPOVSelection Object Events

Method	Description
CheckBoxChanged	Fires when the user selects or clears a member's check box.
DimensionWasSelected	Fires when the tab for a dimension is selected by either the user or a call to SelectDimension .
MultiCheckBoxChanged	Fires when the user clicks either the button to select all check boxes or the button to clear them all.
SelectionChanged	Fires when the user selects a member's label.

HsvResourceManager Type Library Overview

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.

For information on setting HsvResourceManager object references and details on the object's methods, see "[HsvResourceManager Object Methods](#)" on page 812

The following table lists the HsvResourceManager object's methods.

Table 56 HsvResourceManager Object Methods

Method	Description
GetAvailableLanguages	Returns arrays that identify the languages for which resources are available.
GetCurrentVersion	Returns the version number of Financial Management that is displayed to end-users.
GetCurrentVersionInUserDisplayFormat	Returns the version number of Financial Management that is displayed to end-users.
GetFormattedDateTime	Converts a given timestamp to a formatted string.
GetFormattedDateTimeForLanguage	Returns a string representation of a double-byte date/time value.
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.
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.
GetFormattedResourceString	<i>For internal use.</i>
GetHelpDirectoryForLanguageID	Returns the name of the subdirectory that contains the Financial Management help system for a given language.

Method	Description
GetHFMLanguageIdFromUserLanguages	Returns the Financial Management ID for the language identified by a given language code.
GetLanguageCountryCodeFromLanguageId	Returns the two-letter language code for a given language ID.
GetLocaleIdFromLanguageId	Returns a Windows locale ID for the given Financial Management language ID.
GetResourceString	<i>For internal use.</i>
GetResourceStringFromHR	Returns the resource string for an HRESULT in the specified language.
GetUserDisplayDateTimeFormats	Returns an array of strings that describe the date and time formats into which Financial Management can convert timestamps.
GetUserLanguageFromHFMLanguageId	Returns the two-letter language code for the language represented by a given Financial Management language ID.
GetWindowsDateFormat	Returns the Windows date separator character and short date format for the computer.
GetWindowsDateFormatforLocale	Returns the Windows date format for the given locale.
Initialize	Specifies the tier for which resources are needed. Caution! You must call <code>Initialize</code> before calling any of the other <code>HsvResourceManager</code> methods.

3

Getting Started

In This Chapter

Logging a User on to an Application	119
Working with Metadata	120
Getting and Setting Cell Data	121
Getting and Setting User Preferences for Number Formatting	121
Error Handling	122
Loading and Extracting Application Information	122

This chapter provides links to examples of some basic tasks that you can perform.

Note: The Financial Management Software Development Toolkit contains some example Visual Basic 6 projects for this object model, including an example for getting started. The Software Development Toolkit is available on the Oracle® E-Delivery site.

Logging a User on to an Application

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

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

Note: [OpenApplication](#) returns an object reference to the HsvSession object, which provides properties for obtaining object references to several other objects. For more information, see “[HsvSession Object Properties](#)” on page 157.

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

Tip: The HsxClient UI object provides another way to log a user on. HsxClient exposes various Financial Management dialog boxes, including those used to enter logon credentials and open applications. For an example that shows how to programmatically use these dialog boxes, see the example for [HsxClientUI.OpenApplication](#).

Other Useful Methods for Logging On

- `HsxClient.EnumRegisteredClusterNames` returns the names of the clusters or application servers that have been registered for a client.
- `HsxServer.EnumDataSources` returns the names of the applications on a given cluster or application server.

Working with Metadata

The HsvMetadata type library consists of objects that provide access to metadata information such as the members in a given dimension, dimension member attributes, a member's label and member ID, and so on. These objects are introduced in [Chapter 8, "HsvMetadata Type Library."](#)

The following examples introduce you to the HsvMetadata type library's objects:

- The `IHsvTreeInfo` interface returns information on all Financial Management dimensions. For example, you can get the members of a dimension, a dimension's parent-child relationships, or the member ID of a member. Examples for the following methods introduce this interface:
 - You can get a member ID by passing the member's label to `GetItemID`. The example for `GetItemID` is a function that gets the ID of a member for a given dimension.

Note: To specify the dimension for which the `IHsvTreeInfo` interface will obtain information, use the HsvMetadata object's properties. If the dimension to be queried is not known at design-time, use the `Dimension` property, which is demonstrated in the `GetItemID` example.
 - `EnumAllMemberLabels` returns the labels of a dimension's members.
 - `EnumAncestors` returns a given member's ancestors.
- `HsvMetadata.IsCustomMemberValidForAccount` indicates whether a Custom dimension member is valid for an Account dimension member.
- `HsvMetadata.IsOrgByPeriodApplication` indicates whether the Organization by Period feature is enabled.
- Some of the HsvMetadata library objects that represent dimensions return attributes of dimension members. Here are a few examples:
 - `HsvAccounts.GetAccountType` returns the account type of an Account dimension member.
 - `HsvCustom.IsCalculated` indicates whether a Custom dimension member's data is calculated.
 - `HsvEntities.GetSecurityClassID` returns the ID of an Entity dimension member's security class.

Getting and Setting Cell Data

The `HsvData` object enables you to work with data. Examples for the following methods introduce this object:

- `GetCell` returns a cell's data and status.

Note: `GetCell` returns data as a `Double`. To return data as a string, use `GetTextCell`. `GetTextCell` also enables you to scale and specify the number of decimal places for the data.
- `GetCellLineItems` returns the data and descriptions of a cell's line items.

Note: `GetCellLineItems` returns line item data as a `Double`. To return the data as a string, use `GetTextCellLineItems`.
- `GetCells` returns the data and statuses of multiple cells.

Note: `GetCells` returns data as a `Double`. To return data as a string, use `GetTextCells`. `GetTextCells` also enables you to scale and specify the number of decimal places for the data.
- `SetCell` inserts data into a cell. You also can use `SetCell` to set a cell to `Null`.

Note: `SetCell` takes cell data as a `Double`. To pass cell data as a string that you can scale, use `SetTextCell`.
- `SetCellLineItems` inserts line items into a cell.

Note: `SetCellLineItems` takes line item data as a `Double`. To pass the data as a string, use `SetTextCellLineItems`.
- `SetCells` sets data for multiple cells.

Note: `SetCells` takes data as a `Double`. To pass data as a string that you can scale, use `SetTextCells`.

Getting and Setting User Preferences for Number Formatting

The `HsvSystemInfo` object provides the following methods for getting and setting the connected user's number formatting preferences:

- `GetNumberFormattingUserParameters` returns the user's decimal and thousands separator characters.
- `SetNumberFormattingUserParameters` sets the user's decimal and thousands separator characters.

Error Handling

To handle Financial Management errors, use the `HsvResourceManager` object, which is introduced in [“Error Handling and the HsvResourceManager Type Library” on page 811](#). 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 822](#).

Loading and Extracting Application Information

The object model provides objects for loading and extracting security, member lists, rules, data, journals, and metadata. For more information, see [Chapter 20, “Type Libraries for Loading and Extracting Information.”](#) That chapter includes the following examples:

- [“Example for Loading Security” on page 754](#)
- [“Example for Extracting Security” on page 755](#)
- [“Example for Loading Metadata” on page 760](#)
- [“Example for Extracting Metadata” on page 761](#)
- [“Example for Loading Data” on page 776](#)
- [“Example for Extracting Data” on page 777](#)
- [“Example for Loading Journals” on page 785](#)
- [“Example for Extracting Journals Without Filters” on page 787](#)
- [“Example for Filtered Journal Extractions” on page 788](#)

4

HsxClient Type Library

This chapter describes the members of the HsxClient type library. The HsxClient methods enable you to log on, open applications, create and delete applications, and to perform several other client-side operations.

To use the HsxClient type library, you must reference `HsxClient.dll` in your project. The HsxClient type library contains one object—the HsxClient object.

Note: You must reference the `HsxClient.dll` installed in the *<install directory>* `\FinancialManagement\Client` directory. Earlier releases of Financial Management installed `HsxClient.dll` in the *<install directory>* `\Common Files\Hyperion Shared\Bin` directory. If you have a project that references the `HsxClient.dll` in the old directory, you must update the project to reference the `HsxClient.dll` in the new directory.

HsxClient Object Methods

The HsxClient object's methods enable you to perform the following operations:

- Log on.
- Open applications.
- Register and unregister clusters or application servers for a client.
- Return the names of the clusters or application servers that have been registered for a client.
- Get and set an application's local storage folder.
- Get the username and domain of the connected user.
- Enable and disable new connections.
- Return usernames and other information for the users who currently are logged on.
- Log users off.

These methods are summarized in [Table 4 on page 50](#), and are described in detail in the following topics.

Note: Assign HsxClient object references with the `Set` keyword.

AuthenticateCSSToken

For internal use.

AuthenticateSecurityAgentCredentials

For internal use.

AuthenticateUserCredentials

For internal use.

AuthenticateUserOnCluster

For internal use.

AuthenticateUserOnClusterSSO

For internal use.

AuthenticateUserOnServer

For internal use.

CreateApplication

Deprecated. Use [CreateApplicationCAS](#).

CreateApplicationCAS

Creates a Classic application.

Before calling `CreateApplicationCAS`, you must use [SetLogonInfoSSO](#) to specify logon information for a user who belongs to the application server's or cluster's Creator group.

Note: The Creator group identifies the users who can create Financial Management applications; see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*. You can test whether the connected user is a member of the Creator group with [DoesUserHaveCreateApplicationRights](#).

Syntax

```
<HsxClient>.CreateApplicationCAS bstrClusterName, bstrProduct, bstrApp,  
bstrAppDesc, bstrDSN, varParam1, varParam2, varParam3, varParam4,  
varParam5, bstrProject, bstrAppWebServerUrl
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the application server cluster on which to create the application.
<i>bstrProduct</i>	String (ByVal). The product name. Enter the string "Financial Management" to create a Financial Management application.
<i>bstrApp</i>	String (ByVal). The name of the application.
<i>bstrAppDesc</i>	String (ByVal). A description of the application.
<i>bstrDSN</i>	String (ByVal). Future use. However, the argument is required—specify an empty string.
<i>varParam1</i>	Variant (ByVal). A binary array containing the contents of the application profile file (.per file) for the application. Create an application profile with Financial Management, then get the file's contents as a binary array (as shown in the example).
<i>varParam2</i>	Variant (ByVal). Future use. However, the argument is required—specify a null Variant.
<i>varParam3</i>	Variant (ByVal). Future use. However, the argument is required—specify a null Variant.
<i>varParam4</i>	Variant (ByVal). Future use. However, the argument is required—specify a null Variant.
<i>varParam5</i>	Variant (ByVal). Future use. However, the argument is required—specify a null Variant.
<i>bstrProject</i>	String (ByVal). The name of the Shared Services provisioning project. Tip: To get the names of the provisioning projects associated with an application server cluster, use EnumProvisioningProjects .
<i>bstrAppWebServerUrl</i>	String (ByVal). The URL of the virtual directory for Financial Management. The URL should include the protocol, Web server name and port, and virtual directory name.

Example

The following method creates an application. The first several lines use Visual Basic 6 methods to get the *varParam1* argument's array from the specified application profile.

```
Sub createApp(vFilename, sCluster As String, _  
    sApp As String, sProj As String, sVirtualDir As String)  
    Dim lFile As Long, lSize As Long, bytaAppCalData() As Byte  
    Dim vaAppCalData, v2, v3, v4, v5  
    Set cClient = New HsxClient  
    'Read the application profile and pass it as a binary array  
    lFile = FreeFile  
    lSize = FileLen(vFilename)  
    Open vFilename For Binary Access Read As #lFile  
    ReDim bytaAppCalData(lSize)  
    Get #lFile, , bytaAppCalData  
    Close #lFile  
    vaAppCalData = bytaAppCalData  
    'g_cClient is an HsxClient object reference for a logged-on user  
    g_cClient.CreateApplicationCAS sCluster, "Financial Management", _
```

```
sApp, "", "", vaAppCalData, v2, v3, v4, v5, sProj, _  
sVirtualDir  
End Sub
```

CreateApplicationCASWithAccessCode

For internal use.

CreateObjectOnCluster

Instantiates an object on a cluster or application server.

Tip: To instantiate a server-side object on a cluster or application server for an application that has been opened, use `HsvSession`'s [CreateObject](#) method.

Syntax

```
<HsxClient>.CreateObjectOnCluster(bstrClusterName, bstrClassIDOrProgID)
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the cluster or application server.
<i>bstrClassIDOrProgID</i>	String (ByVal). The name of the object to be instantiated.

Return Value

Object. `CreateObjectOnCluster` returns an instance of the object identified by the *bstrClassIDOrProgID* argument.

CreateObjectOnServer

Deprecated - use [CreateObjectOnCluster](#).

DeleteApplication

Deletes a Classic application.

Caution! The method will fail if executed against an application created with Oracle Hyperion EPM Architect, Fusion Edition.

Syntax

```
<HsxClient>.DeleteApplication bstrClusterName, bstrProduct, bstrApp
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the application server cluster.
<i>bstrProduct</i>	String (ByVal). The product name. Specify “Financial Management” to delete a Financial Management application.
<i>bstrApp</i>	String (ByVal). The name of the application to delete.

DeleteApplicationWithAccessCode

For internal use.

DeleteSystemErrors

Deletes records of system errors for a cluster.

Syntax

```
<HsxClient>.DeleteSystemErrors bstrClusterName, vbDeleteAll,
varabstrErrorReference
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the cluster.
<i>vbDeleteAll</i>	Boolean (ByVal). A flag that determines whether all system errors should be deleted. Pass TRUE to delete all system errors, FALSE to delete only those errors specified by the <i>varabstrErrorReference</i> argument.
<i>varabstrErrorReference</i>	String array (ByVal). The error reference numbers that identify the system errors to be deleted if the <i>vbDeleteAll</i> argument is set to FALSE. Tip: Error reference numbers are enclosed within braces ({ }), so the array items must include these braces.

DetermineWindowsLoggedOnUser

Returns the domain name and the username of the user who is logged onto Windows.

Syntax

```
<HsxClient>.DetermineWindowsLoggedOnUser pbstrDomainName, pbstrUserName
```

Argument	Description
<i>pbstrDomainName</i>	String. Returns the domain name.
<i>pbstrUserName</i>	String. Returns the username.

DisableNewConnections

Disables new Financial Management connections to a cluster for the specified application and application server criteria.

Tip: To disable new connections for a given user, use `HsvSystemInfo.DisableNewConnections`.

To enable new connections, use [EnableNewConnections](#).

Syntax

```
<HsxClient>.DisableNewConnections bstrClusterName, vbAllApplications,  
bstrAppName, vbAllServers, bstrServer
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the cluster.
<i>vbAllApplications</i>	Boolean (ByVal). Specifies whether to disable connections for all applications. Pass TRUE for all applications, FALSE to specify an application with the <i>bstrAppName</i> argument.
<i>bstrAppName</i>	String (ByVal). The name of the application for which to disable connections. This argument is used only if the <i>vbAllApplications</i> argument is set to FALSE.
<i>vbAllServers</i>	Boolean (ByVal). Specifies whether to disable connections for all application servers in the cluster. Pass TRUE for all application servers, FALSE to specify an application server with the <i>bstrServer</i> argument.
<i>bstrServer</i>	String (ByVal). The name of the application server for which to disable connections. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.

DoesUserHaveCreateApplicationRights

Indicates whether the connected user is a member of the Creator group for a given application server cluster.

Note: The Creator group identifies the users who can create Financial Management applications, and is specified with the Financial Management Configuration Utility. For instructions, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

Syntax

```
<HsxClient>.DoesUserHaveCreateApplicationRights bstrClusterName,  
pvbHasAccess
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the application server cluster.

Argument	Description
<code>pvbHasAccess</code>	Boolean. Indicates whether the connected user is a member of the Creator group. Returns TRUE if the user is a member, FALSE otherwise.

DoesUserHaveSystemAdminRights

Indicates whether the connected user is a member of the Administrator group for a given application server cluster.

Note: The Administrator group is specified with the Financial Management Configuration Utility. For instructions, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

Syntax

```
<HsxClient>.DoesUserHaveSystemAdminRights bstrClusterName, pvbHasAccess
```

Argument	Description
<code>bstrClusterName</code>	String (ByVal). The name of the application server cluster.
<code>pvbHasAccess</code>	Boolean. Indicates whether the connected user is a member of the Administrator group. Returns TRUE if the user is a member, FALSE otherwise.

EnableNewConnections

Enables new Financial Management connections to a cluster for the specified application and application server criteria.

Tip: To enable new connections for a given user, use `HsvSystemInfo.EnableNewConnections`.

To disable new connections, use `DisableNewConnections`.

Syntax

```
<HsxClient>.EnableNewConnections bstrClusterName, vbAllApplications,
bstrAppName, vbAllServers, bstrServer
```

Argument	Description
<code>bstrClusterName</code>	String (ByVal). The name of the cluster.
<code>vbAllApplications</code>	Boolean (ByVal). Specifies whether to enable connections for all applications. Pass TRUE for all applications, FALSE to specify an application with the <code>bstrAppName</code> argument.

Argument	Description
<i>bstrAppName</i>	String (ByVal). The name of the application for which to enable connections. This argument is used only if the <i>vbAllApplications</i> argument is set to FALSE.
<i>vbAllServers</i>	Boolean (ByVal). Specifies whether to enable connections for all application servers in the cluster. Pass TRUE for all application servers, FALSE to specify an application server with the <i>bstrServer</i> argument.
<i>bstrServer</i>	String (ByVal). The name of the application server for which to enable connections. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.

EnumProhibitConnections

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

Syntax

```
<HsxClient>.EnumProhibitConnections bstrClusterName, pvaravbAllApps,
pvarabstrAppNames, pvaravbAllServers, pvarabstrServerNames,
pvaravbAllUsers, pvaralActivityUserIDs, pvaralActivityUserNames
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the cluster.
<i>pvaravbAllApps</i>	Variant array. Indicates whether connections are disabled for all applications. Array items can contain 0 or -1: <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>pvarabstrAppNames</i> argument's array item are disabled. The array is returned as a Long subtype.
<i>pvarabstrAppNames</i>	Variant array. Returns the application names for which connections are disabled. Application names are returned only when the corresponding <i>pvaravbAllApps</i> argument's array item contains 0. <p>Note: If the corresponding <i>pvaravbAllApps</i> argument's array item contains -1, this array item contains the string "AllApps".</p> The array is returned as a String subtype.
<i>pvaravbAllServers</i>	Variant array. Indicates whether connections are disabled for all application servers. Array items can contain 0 or -1: <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>pvarabstrServerNames</i> argument's array item are disabled. The array is returned as a Long subtype.

Argument	Description
<i>pvarabstrServerNames</i>	<p>Variant array. Returns the application server names for which connections are disabled. Application server names are returned only when the corresponding <i>pvaravbAllServers</i> argument's array item contains 0.</p> <p>Note: If the corresponding <i>pvaravbAllServers</i> argument's array item contains -1, this array item contains the string "AllServers".</p> <p>The array is returned as a String subtype.</p>
<i>pvaravbAllUsers</i>	<p>Variant array. 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>pvaralActivityUserIDs</i> and <i>pvaralActivityUserNames</i> arguments' array items are disabled. <p>The array is returned as a Long subtype.</p>
<i>pvaralActivityUserIDs</i>	<p>Variant array. Returns the activity user IDs of the users for whom connections are disabled. Valid IDs are returned only when the corresponding <i>pvaravbAllUsers</i> argument's array item contains 0.</p> <p>Note: If the corresponding <i>pvaravbAllUsers</i> argument's array item contains -1, this array item contains -1.</p> <p>The array is returned as a Long subtype.</p>
<i>pvaralActivityUserNames</i>	<p>Variant array. Returns the usernames of the users for whom connections are disabled. Usernames are returned only when the corresponding <i>pvaravbAllUsers</i> argument's array item contains 0.</p> <p>Note: If the corresponding <i>pvaravbAllUsers</i> argument's array item contains -1, this array item contains an empty string.</p> <p>The array is returned as a String subtype.</p>

EnumProvisioningProjects

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

Syntax

```
<HsxClient>.EnumProvisioningProjects(bstrClusterName)
```

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

<i>bstrClusterName</i>	String (ByVal). The name of the cluster.
------------------------	--

Return Value

Variant. Returns an array containing the names of the provisioning projects.

EnumRegisteredClusterNames

Returns an array containing the names of the clusters or application servers that have been registered for the client.

Syntax

```
<HsxClient>.EnumRegisteredClusterNames()
```

Return Value

Variant array. Returns the cluster names. The array is returned as a String subtype.

Example

This example shows how to add a client's registered application servers to a combo box.

```
Dim vSrvs As Variant
'cClient is an HsxClient object reference
vSrvs = cClient.EnumRegisteredClusterNames
'Assign the array items to the combo box
For i = LBound(vSrvs) To UBound(vSrvs)
    'cmbServers is the comboBox
    cmbServers.AddItem vSrvs(i)
Next i
```

EnumRegisteredServerNames

Deprecated - use [EnumRegisteredClusterNames](#).

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

```
<HsxClient>.EnumUsersOnSystem bstrClusterName, vbAllApplications,  
bstrAppName, vbAllServers, bstrServer, vbAllUsers, lActivityUserID,  
pvaralActivitySessionIDs, pbstrAppNames, pbstrServerNames,  
pvaralActivityUserIDs, pbstrActivityUserNames, pvaralCurrentActivity,  
pvarabstrModuleNames, pvaradTimeStarted, pvarabstrDesc
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the cluster.
<i>vbAllApplications</i>	Boolean (ByVal). Specifies whether information for all applications is returned. Pass TRUE for all applications, FALSE to specify an application with the <i>bstrAppName</i> argument.
<i>bstrAppName</i>	String (ByVal). The name of the application server for which to return information. This argument is used only if the <i>vbAllApplications</i> argument is set to FALSE.
<i>vbAllServers</i>	Boolean (ByVal). Specifies whether 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.
<i>bstrServer</i>	String (ByVal). The name of the application server for which to return information. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <i>lActivityUserID</i> argument.
<i>lActivityUserID</i>	Long (ByVal). The activity user ID of the user for whom to return information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. To get a user's activity user ID, use <code>HsvSystemInfo.GetActivityUserID</code> .
<i>pvaralActivitySessionIDs</i>	Variant array. Returns the internal IDs of the user sessions. The array is returned as a Long subtype.
<i>pbstrAppNames</i>	Variant array. Returns the names of the applications to which users are logged on. The array is returned as a String subtype.
<i>pbstrServerNames</i>	Variant array. Returns the names of the application servers on which users are logged on. The array is returned as a String subtype.
<i>pvaralActivityUserIDs</i>	Variant array. Returns the activity user IDs of the users who are logged on. The array is returned as a Long subtype.
<i>pbstrActivityUserNames</i>	Variant array. Returns the usernames of the users who are logged on. The array is returned as a String subtype.
<i>pvaralCurrentActivity</i>	Variant array. Returns the IDs of the users' current activities. Valid values are represented by the <code>HFMConstants</code> type library constants listed in " User Activity Constants " on page 870. The array is returned as a Long subtype.
<i>pvarabstrModuleNames</i>	Variant array. Returns the names of the modules in which the users are active. The array is returned as a String subtype.

Argument	Description
<i>pvaradTimeStarted</i>	Variant array. Returns the timestamps of when the users accessed the system. Array items are formatted as Doubles that can be cast to the Date format. The array is returned as a Double subtype.
<i>pvarabstrDesc</i>	Variant array. <i>Future use.</i> An array is returned, but you can ignore it.

Example

The following function returns a two-dimensional array that contains the usernames and starting timestamps of all users logged on a given cluster.

```
Function getUsersTimes(sCluster As String) As Variant
Dim vaSessions, vaApps, vaServers, vaUserIDs, vaUserNames
Dim vaActivity, vaModules, vaTimes, vaDesc, vaRet()
'm_cHsxClient is an HsxClient object reference
m_cHsxClient.EnumUsersOnSystem sCluster, True, "", True, "", _
    True, 0, vaSessions, vaApps, vaServers, vaUserIDs, _
    vaUserNames, vaActivity, vaModules, vaTimes, vaDesc
ReDim vaRet(UBound(vaUserNames), 1)
For i = LBound(vaUserNames) To UBound(vaUserNames)
    vaRet(i, 0) = vaUserNames(i)
    vaRet(i, 1) = CDate(vaTimes(i))
Next i
getUsersTimes = vaRet
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 user 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

```
<HsxClient>.EnumUsersOnSystemEx bstrClusterName, vbAllApplications,
bstrAppName, vbAllServers, bstrServer, vbAllUsers, lActivityUserID,
lLanguageID, pvaralActivitySessionIDs, pbstrAppNames, pbstrServerNames,
pvaralActivityUserIDs, pbstrActivityUserNames, pvaralCurrentActivity,
pvarabstrModuleNames, pvaradTimeStarted, pvarabstrDesc
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the cluster.

Argument	Description
<i>vbAllApplications</i>	Boolean (ByVal). Specifies whether information for all applications is returned. Pass TRUE for all applications, FALSE to specify an application with the <i>bstrAppName</i> argument.
<i>bstrAppName</i>	String (ByVal). The name of the application server for which to return information. This argument is used only if the <i>vbAllApplications</i> argument is set to FALSE.
<i>vbAllServers</i>	Boolean (ByVal). Specifies whether 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.
<i>bstrServer</i>	String (ByVal). The name of the application server for which to return information. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <i>lActivityUserID</i> argument.
<i>lActivityUserID</i>	Long (ByVal). The activity user ID of the user for whom to return information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. To get a user's activity user ID, use <code>HsvSystemInfo.GetActivityUserID</code> .
<i>lLanguageID</i>	Long (ByVal). The ID of the language in which to return module names. Tip: You can use the <code>HsvResourceManager</code> method GetAvailableLanguages to obtain the IDs of the languages valid for a given release.
<i>pvaralActivitySessionIDs</i>	Variant array. Returns the internal IDs of the user sessions. The array is returned as a Long subtype.
<i>pbstrAppNames</i>	Variant array. Returns the names of the applications to which users are logged on. The array is returned as a String subtype.
<i>pbstrServerNames</i>	Variant array. Returns the names of the application servers on which users are logged on. The array is returned as a String subtype.
<i>pvaralActivityUserIDs</i>	Variant array. Returns the activity user IDs of the users who are logged on. The array is returned as a Long subtype.
<i>pbstrActivityUserNames</i>	Variant array. Returns the usernames of the users who are logged on. The array is returned as a String subtype.
<i>pvaralCurrentActivity</i>	Variant array. Returns the IDs of the users' current activities. Valid values are represented by the <code>HFMConstants</code> type library constants listed in "User Activity Constants" on page 870 . The array is returned as a Long subtype.
<i>pvarabstrModuleNames</i>	Variant array. Returns the names of the modules in which the users are active. The array is returned as a String subtype.
<i>pvaradTimeStarted</i>	Variant array. Returns the timestamps of when the users accessed the system. Array items are formatted as Doubles that can be cast to the Date format. The array is returned as a Double subtype.
<i>pvarabstrDesc</i>	Variant array. <i>Future use.</i> An array is returned, but you can ignore it.

EnumUsersOnSystemEx2

Returns the usernames and other information applicable to all users logged on to / logged out from 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 user information is returned in arrays that have a one-to-one correspondence.

Syntax

```
<IHsxClient>.EnumUsersOnSystemEx2 bstrClusterName, vbAllApplications,
bstrAppName, vbAllServers, bstrServer, vbAllUsers, lActivityUserID,
lLanguageID, pvaralActivitySessionIDs, pbstrAppNames, pbstrServerNames,
pvaralActivityUserIDs, pbstrActivityUserNames, pvaralCurrentActivity,
pvarabstrModuleNames, pvaradTimeStarted, pvarabstrDesc, pvaralSessionStatus
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the cluster.
<i>vbAllApplications</i>	Boolean (ByVal). Specifies whether information for all applications is returned. Pass TRUE for all applications, FALSE to specify an application with the <i>bstrAppName</i> argument.
<i>bstrAppName</i>	String (ByVal). The name of the application server for which to return information. This argument is used only if the <i>vbAllApplications</i> argument is set to FALSE.
<i>vbAllServers</i>	Boolean (ByVal). Specifies whether 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.
<i>bstrServer</i>	String (ByVal). The name of the application server for which to return information. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <i>lActivityUserID</i> argument
<i>lActivityUserID</i>	Long (ByVal). The activity user ID of the user for whom to return information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. To get a user's activity user ID, use <code>HsvSystemInfo.GetActivityUserID</code> .
<i>lLanguageID</i>	Long (ByVal). The ID of the language in which to return module names. Tip: You can use the <code>HsvResourceManager</code> method <code>GetAvailableLanguages</code> to obtain the IDs of the languages valid for a given release.
<i>pvaralActivitySessionIDs</i>	Variant array. Returns the internal IDs of the user sessions. The array is returned as a Long subtype.
<i>pbstrAppNames</i>	Variant array. Returns the names of the applications to which users are logged on. The array is returned as a String subtype.
<i>pbstrServerNames</i>	Variant array. Returns the names of the application servers on which users are logged on. The array is returned as a String subtype.
<i>pvaralActivityUserIDs</i>	Variant array. Returns the activity user IDs of the users who are logged on.

Argument	Description
	The array is returned as a Long subtype.
<i>pstrActivityUserNames</i>	Variant array. Returns the usernames of the users who are logged on. The array is returned as a String subtype.
<i>pvarCurrentActivity</i>	Variant array. Variant array. Returns the IDs of the users' current activities. Valid values are represented by the HFConstants type library constants listed in "User Activity Constants" on page 870 . The array is returned as a Long subtype.
<i>pvarabstrModuleNames</i>	Variant array. Returns the names of the modules in which the users are active. The array is returned as a String subtype.
<i>pvaradTimeStarted</i>	Variant array. Returns the timestamps of when the users accessed the system. Array items are formatted as Doubles that can be cast to the Date format. The array is returned as a Double subtype.
<i>pvarabstrDesc</i>	Variant array. Future use. An array is returned, but you can ignore it.
<i>pvaralSessionStatus</i>	Variant array. Returns an array of status IDs that identify the user's task status. Valid values are represented by the HFConstants type library constants listed in "Task Status Constants" on page 872 .

GetApplicationFolder

Returns the path of an application's local storage folder.

Syntax

```
<HsxClient>.GetApplicationFolder(bstrClusterName, bstrProduct, bstrApp)
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the application server cluster.
<i>bstrProduct</i>	String (ByVal). The name of the product. For Financial Management applications, enter the string "Financial Management".
<i>bstrApp</i>	String (ByVal). The name of the application.

Return Value

String. Returns the directory path of the local storage folder.

Example

This example assigns to the `sStoreFolder` variable the path of the local storage folder for an application named "SimpleEx."

```
sStoreFolder = cHsxClient.GetApplicationFolder("Gszabol", _  
"Financial Management", "SimpleEx")
```

GetClusterInfo

Returns the name of the cluster to which the specified application server is assigned.

Syntax

```
<HsxClient>.GetClusterInfo bstrServerToGetClusterInfo,  
vbUseAutomaticLoadBalancing, pbstrClusterName
```

Argument	Description
<i>bstrServerToGetClusterInfo</i>	String (ByVal). The name of the application server.
<i>vbUseAutomaticLoadBalancing</i>	Boolean (ByVal). Specifies whether the application server is registered for automatic load balancing. Pass TRUE if load balancing is used, FALSE otherwise.
<i>pbstrClusterName</i>	String. Returns the name of the cluster to which the application server is assigned.

GetHFMErrrorLogRecordSet

For internal use.

GetLogonInfo

Deprecated - use [GetLogonInfoSSO](#).

GetLogonInfoSSO

Gets the domain, username, and external authentication token for the connected user.

Note: An external authentication token cannot be retrieved until an application has been opened.

Syntax

```
<HsxClient>.GetLogonInfoSSO(pbstrDomain, pbstrUser)
```

Argument	Description
<i>pbstrDomain</i>	String. Returns the name of the user's domain.
<i>pbstrUser</i>	String. Returns the name of the user.

Return Value

String. Returns the external authentication token.

Example

GetLogonInfoSSO is used in the example for the HsvSecurityAccess method [GetUserInfoFromUniqueID2](#).

GetServer

Deprecated - use [GetServerOnCluster](#).

GetServerOnCluster

Returns an object reference to the HsxServer object that represents the specified cluster or server.

Syntax

```
<HsxClient>.GetServerOnCluster (bstrClusterName)
```

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

<i>bstrClusterName</i>	String (ByVal). The name of the cluster.
------------------------	--

Return Value

HsxServer object. Returns an object reference to the HsxServer object. For information on this object, see [Chapter 6, “HsxServer Type Library.”](#)

GetSSOTokenUsingWebSecurityAgentCredentials

For internal use.

GetWebSecurityAgentSettings

For internal use.

IsValidApplication

Indicates whether the specified application exists on the specified cluster.

Syntax

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

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

<i>bstrCluster</i>	String (ByVal). The cluster name.
--------------------	-----------------------------------

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

<i>bstrAppName</i>	String (ByVal). The application name.
--------------------	---------------------------------------

Return Value

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

KillUsers

Logs off users from a cluster. You can log off users from all application servers, or log off only users of given applications and application servers.

Tip: To log off given users or sessions, use `HsvSystemInfo.KillUsers`.

Syntax

```
<HsxClient>.KillUsers bstrClusterName, vbAllApplications, bstrAppName,  
vbAllServers, bstrServer
```

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

<i>bstrClusterName</i>	String (ByVal). The name of the cluster.
------------------------	--

<i>vbAllApplications</i>	Boolean (ByVal). Determines whether to log off users from all applications. Pass TRUE to log users off from all applications, FALSE to log users off the application specified in the <i>bstrAppName</i> argument.
--------------------------	--

<i>bstrAppName</i>	String (ByVal). The name of the application for which to log users off. This argument is used only if the <i>vbAllApplications</i> argument is set to FALSE.
--------------------	--

<i>vbAllServers</i>	Boolean (ByVal). 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
---------------------	---

<i>bstrServer</i>	String (ByVal). 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.
-------------------	--

OpenApplication

Opens a Financial Management application. `OpenApplication` returns object references to `HsxServer` and `HsvSession` objects.

You must call `SetLogonInfoSSO` before calling `OpenApplication`. `SetLogonInfoSSO` specifies the username, domain, and password or an SSO token; `OpenApplication` then authenticates this logon information.

Syntax

```
<HsxClient>.OpenApplication bstrClusterName, bstrProduct, bstrApp,  
ppIUnkServer, ppIUnkSession
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the application server cluster.
<i>bstrProduct</i>	String (ByVal). The product name. Enter "Financial Management" to open a Financial Management application.
<i>bstrApp</i>	String (ByVal). The name of the application.
<i>ppIUnkServer</i>	HsxServer object. Returns an object reference to the HsxServer object. For information on this object, see Chapter 6, "HsxServer Type Library."
<i>ppIUnkSession</i>	HsvSession object. Returns an object reference to an HsvSession object. For information on this object, see Chapter 7, "HsvSession Type Library."

Example

The following function logs a user on to an application and returns an HsvSession object reference. The function takes logon credentials that are passed to `SetLogonInfoSSO` and server and application names that are passed to `OpenApplication`.

```
Function openHfmApp(sDomain As String, sUser As String, _  
    sPass As String, sServer As String, sApp As String)  
    Dim cClient As HsxClient, cSession As HsvSession  
    Dim cServer As HsxServer  
    Set cClient = New HsxClient  
    cClient.SetLogonInfoSSO sDomain, sUser, "", sPass  
    cClient.OpenApplication sServer, "Financial Management", _  
        sApp, cServer, cSession  
    Set openHfmApp = cSession  
End Function
```

RegisterApplicationCAS

Registers a Classic application with Shared Services.

Caution! The method will fail if executed against an application created with Performance Management Architect.

Syntax

```
<HsxClient>.RegisterApplicationCAS bstrClusterName, bstrProjectName,  
bstrApp, bstrHFMMWebServerURL
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the application's server cluster.

Argument	Description
<i>bstrProjectName</i>	String (ByVal). The name of the Shared Services provisioning project for the application.
<i>bstrApp</i>	String (ByVal). The name of the application. Tip: To get the names of the provisioning projects associated with an application server cluster, use EnumProvisioningProjects .
<i>bstrHFMServerURL</i>	String (ByVal). The URL of the virtual directory for Financial Management. The URL should include the protocol, Web server name and port, and virtual directory name.

RegisterApplicationCASWithAccessCode

For internal use.

RegisterCluster

Registers a cluster or application server for a client.

Syntax

```
<HsxClient>.RegisterCluster bstrServerToGetClusterInfo,
vbUseAutomaticLoadBalancing, pbstrClusterName
```

Argument	Description
<i>bstrServerToGetClusterInfo</i>	String (ByVal). The machine name of the cluster or application server to be registered.
<i>vbUseAutomaticLoadBalancing</i>	Boolean (ByVal). Indicates whether automatic load balancing is used. Pass TRUE if load balancing is used, FALSE otherwise
<i>pbstrClusterName</i>	String. Returns the name of the registered cluster or application server.

Example

The following example registers an application server named ACME.

```
Dim sCluster As String
m_cHsxClient.RegisterCluster "ACME", False, sCluster
```

RegisterServer

Deprecated - use [RegisterCluster](#).

ScriptableEnumRegisteredClusterNames

For internal use.

ScriptableEnumRegisteredServerNames

For internal use.

ScriptableGetLogonInfoSSO

For internal use.

ScriptableOpenApplication

For internal use.

SetApplicationFolder

Sets an application's local storage folder.

Syntax

```
<HsxClient>.SetApplicationFolder bstrClusterName, bstrProduct, bstrApp,  
bstrFolder
```

Argument	Description
<i>bstrClusterName</i>	String (ByVal). The name of the cluster or application server.
<i>bstrProduct</i>	String (ByVal). The product name. Enter the string "Financial Management" to set a Financial Management application's local storage folder.
<i>bstrApp</i>	String (ByVal). The name of the application.
<i>bstrFolder</i>	String (ByVal). The path on the client computer to set as the local storage folder.

Example

The following example sets the local storage directory for an application named "AcmeConsol."

```
cHsxClient.SetApplicationFolder "88", "Financial Management", _  
"AcmeConsol", "C:\AcmeConsol\Files"
```

SetLogonInfo

Deprecated - use [SetLogonInfoSSO](#).

SetLogonInfoSSO

Sets logon information such as the username, password, domain or sets an SSO token. If all arguments are passed, then only the token is used. The logon information that SetLogonInfoSSO sets is authenticated when [OpenApplication](#) is called.

Syntax

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

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

<i>bstrDomain</i>	String (ByVal). The name of the user's domain.
-------------------	--

<i>bstrUser</i>	String (ByVal). The user's username.
-----------------	--------------------------------------

<i>bstrToken</i>	String (ByVal). The single sign-on token.
------------------	---

<i>bstrPassword</i>	String (ByVal). The user's password.
---------------------	--------------------------------------

Example

See the example for [OpenApplication](#).

UnregisterAllClusters

Unregisters all the clusters or application servers that are registered for the client.

Syntax

```
<HsxClient>.UnregisterAllClusters
```

UnregisterCluster

Unregisters the specified cluster or application server.

Syntax

```
<HsxClient>.UnregisterCluster bstrClusterName
```

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

<i>bstrClusterName</i>	String (ByVal). The name of the cluster or application server.
------------------------	--

Example

The following example unregisters a cluster named myCluster.

```
m_cHsxClient.UnregisterCluster "myCluster"
```


UnregisterServer

Deprecated - use [UnregisterCluster](#).

UpdateUserAppPreferences

For internal use.

WarnUsersForShutDown

For internal use.

In This Chapter

HsxClientUI Object Methods	147
DeleteApplication	148
GetServer	148
Initialize	148
Logon.....	148
OpenApplication	149
RegisterServer	150

This chapter describes the members of the HsxClientUI type library. The methods of this type library display the Financial Management dialog boxes used to log on, open applications, delete applications, and register and unregister clusters and application servers.

To use the HsxClientUI type library, you must reference `HsxClientUI.dll` in your project. The HsxClientUI type library contains one object—the HsxClientUI object.

Note: You must reference the `HsxClientUI.dll` installed in the *<install directory>* \Client directory. Earlier releases of Financial Management installed `HsxClientUI.dll` in the *<install directory>*\Common Files\Hyperion Shared\Bin directory. If you have a project that references the `HsxClientUI.dll` in the old directory, you must update the project to reference the `HsxClientUI.dll` in the new directory.

HsxClientUI Object Methods

The HsxClientUI object's methods display the following dialog boxes:

- Logon
- Register Cluster

These methods are summarized in [Table 5 on page 52](#), and are described in detail in the following topics.

Note: Set HsxClientUI object references with Visual Basic's `Set` keyword. After setting an object reference, you must call `Initialize`, passing an HsxClient object reference.

DeleteApplication

Displays the Delete Application dialog box.

Tip: To delete Classic Financial Management applications without displaying the Delete Application dialog box, use `HsxClient.DeleteApplication`.

GetServer

For internal use.

Initialize

Provides the `HsxClientUI` object with access to Financial Management's client layer.

Caution! You must call `Initialize` before calling any of the other `HsxClientUI` methods, otherwise an error will occur.

Syntax

```
<HsxClientUI>.Initialize pUnkHsxClient
```

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

<code>pUnkHsxClient</code>	HsxClient object (ByVal). An HsxClient object reference.
----------------------------	--

Example

This example declares and sets `HsxClientUI` and `HsxClient` object variables, then calls `Initialize`. After `Initialize` is called, you can then use the `m_cClientUI` object reference to call `HsxClientUI` methods.

```
Dim m_cClientUI As HsxClientUI, m_cClient As HsxClient
Set m_cClient = New HsxClient
Set m_cClientUI = New HsxClientUI
m_cClientUI.Initialize m_cClient
```

Logon

Displays the Logon dialog box.

The information that the user enters in the Logon dialog box is authenticated when an application is opened.

Tip: To set logon information without displaying the Logon dialog box, use `HsxClient.SetLogonInfo`.

Syntax

```
<HsxClientUI>.Logon(vbProvideOptionToUseWindowsLoggedOnUser)
```

Argument

Description

`vbProvideOptionToUseWindowsLoggedOnUser` Boolean (ByVal). The value passed is ignored by the system; however, you must pass either TRUE or FALSE.

Return Value

Boolean. Returns TRUE if the user clicks the Logon dialog box's Logon button, FALSE if the user clicks the Cancel button.

Caution! The return value does *not* indicate whether the user's logon information is valid, it only indicates whether the Logon or Cancel button has been clicked.

Example

Logon is used in the [Example](#) for `OpenApplication`.

OpenApplication

Displays the Open Application dialog box.

Before calling `OpenApplication`, you must set logon information by calling `Logon` or `HsxClient.SetLogonInfo`. The logon information that has been specified with the Logon dialog box or with `SetLogonInfoSSO` is authenticated after the user clicks the Open Application button.

Tip: To open applications without displaying the Open Application dialog box, use `HsxClient.OpenApplication`.

Syntax

```
<HsxClientUI>.OpenApplication(ppIUnkServer, ppIUnkSession, pbstrServer, pbstrProduct, pbstrApplication)
```

Argument

Description

`ppIUnkServer` HsxServer object. Returns an object reference to the HsxServer object for the application server that the user selects. For information on this object, see [Chapter 6, "HsxServer Type Library."](#)

`ppIUnkSession` HsvSession object. Returns an object reference to an HsvSession object. For information on this object, see [Chapter 7, "HsvSession Type Library."](#)

Argument	Description
<i>pBstrServer</i>	String. Returns the name of the application server that the user selects.
<i>pBstrProduct</i>	String. Returns the product name. "Financial Management" is returned for Financial Management applications.
<i>pBstrApplication</i>	String. Returns the name of the application that the user opens.

Return Value

Boolean. Returns TRUE if an application is opened, otherwise FALSE.

Example

This example uses the `Logon` method to display the Logon dialog box. If the user clicks the dialog box's **Logon** button, `Logon` returns TRUE, and `OpenApplication` — which is placed within an `If` structure — is called. The example assumes that the `m_cClientUI` object reference has been set with the code in the [Example](#) for `Initialize`.

```
Dim cServer As HsxServer, cSession As HsvSession
Dim sServer As String, sProd As String, sApp As String
Dim bLogon As Boolean
bLogon = m_cClientUI.Logon(True)
If bLogon = True Then
    m_cClientUI.OpenApplication cServer, cSession, _
        sServer, sProd, sApp
End If
```

RegisterServer

Displays the Register Cluster dialog box.

Tip: The `HsxClient` object's [RegisterCluster](#) and [UnregisterCluster](#) methods can also be used to register and unregister clusters and application servers.

Syntax

```
<HsxClientUI>.RegisterServer
```

Example

The following example displays the Register Cluster dialog box. The example assumes that the `m_cClientUI` object reference has been set with the code in the [Example](#) for `Initialize`.

```
m_cClientUI.RegisterServer
```

6

HsxServer Type Library

In This Chapter

HsxServer Object Methods	151
HsxServer Object Properties.....	156

This chapter describes the members of the HsxServer type library. Use the HsxServer methods to get application server-related information such as the names of an application server's applications and Data Source Names. HsxServer also provides a property that indicates whether external authentication has been enabled for a cluster.

To use the HsxServer type library, you must reference `HsxServer.exe` in your project. The HsxServer type library contains one object—the HsxServer object.

Note: You must reference the `HsxServer.exe` installed in the `<install directory>\Server` directory. Earlier releases of Financial Management installed `HsxServer.exe` in the `<install directory>\Common Files\Hyperion Shared\Bin` directory. If you have a project that references the `HsxServer.exe` in the old directory, you must update the project to reference the `HsxServer.exe` in the new directory.

There is one HsxServer object per application server. For more information, see “[Application Server Tier Type Libraries](#)” on page 41.

To assign HsxServer object references, use the HsxServer object reference returned by the HsxClient methods `OpenApplication` or `GetServerOnCluster` or the HsxClientUI method `OpenApplication`.

HsxServer Object Methods

The HsxServer object's methods enable you to get the following information for application servers:

- Names of Financial Management applications
- Names of registered Data Source Names
- Names and paths of the data link and system files

These methods are summarized in [Table 6 on page 53](#), and are described in detail in the following topics.

DeleteSystemErrors

For internal use.

EnumDataSources

Returns the names and descriptions of the applications on an application server.

Syntax

```
<HsxServer>.EnumDataSources pvarabstrProducts, pvarabstrApps,  
pvarabstrDescs, pvarabstrDSNs
```

Argument	Description
<i>pvarabstrProducts</i>	Variant array. Returns the product names for the applications. For Financial Management applications, the array items will consist of the string “Financial Management”. The array is returned as a String subtype.
<i>pvarabstrApps</i>	Variant array. Returns the names of the applications on the application server. The array is returned as a String subtype.
<i>pvarabstrDescs</i>	Variant array. Returns the descriptions of the applications. The array is returned as a String subtype.
<i>pvarabstrDSNs</i>	Variant array. For Financial Management, this returns as Empty. If you call <code>EnumDataSources</code> for other products, this returns an array of data sources for the applications. The array is returned as a String subtype.

Example

The following example inserts the names of all applications on an application server into a ComboBox.

```
Dim vaProds, vaApps, vaDescs, vaDSNs  
'cServer is an HsxServer object reference  
cServer.EnumDataSources vaProds, vaApps, vaDescs, vaDSNs  
For i = LBound(vaApps) To UBound(vaApps)  
    'cmbApps is the ComboBox  
    cmbApps.AddItem vaApps(i)  
Next i
```

EnumRegisteredDSNs

Returns an array of the registered Data Source Names on the application server.

Note: EnumRegisteredDSNs applies to the Data Source Names registered with Financial Management Server Administrator, *not* to ODBC Data Source Names.

Syntax

```
<HsxServer>.EnumRegisteredDSNs ()
```

Return Value

Variant array. Returns the names of the registered Data Source Names. The array is returned as a String subtype.

Example

This example tests whether the application server has a registered Data Source Name. The example applies UBound to the array returned by EnumRegisteredDSNs. If UBound returns 0 there is not a registered Data Source Name; in this case the user is warned and the procedure is terminated.

```
Dim vaDSNs, iDsnCount As Integer
vaDSNs = m_cHsxServer.EnumRegisteredDSNs ()
If UBound(vaDSNs) = 0 Then
    MsgBox "There is no DSN for the application server." & vbCr _
        & "Register a DSN before proceeding."
    Exit Sub
End If
```

GetClustersAndServers

Returns the names of the clusters and servers associated with an HsxServer object reference.

Syntax

```
<HsxServer>.GetClustersAndServers pvarabstrClusters, pvarabstrServers
```

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

<i>pvarabstrClusters</i>	Variant. Returns an array containing the cluster names.
--------------------------	---

<i>pvarabstrServers</i>	Variant. Returns an array of two-dimensional arrays containing the names of the clusters' servers. Each array of two-dimensional arrays has a one-to-one correspondence with the <i>varalPeriodIDs</i> argument's array.
-------------------------	--

Example

The following snippet prints to Visual Basic's Immediate window the names of the clusters and servers associated with an HsxServer object reference.

```
'g_cServer is an HsxServer object reference
g_cServer.GetClustersAndServers vaClusters, vaServers
For i = LBound(vaClusters) To UBound(vaClusters)
    Debug.Print "Cluster: " & vaClusters(i)
    For j = LBound(vaServers(i)) To UBound(vaServers(i))
```

```
        Debug.Print vbTab & vaServers(i)(j)
    Next j
Next i
```

GetDataSource

For internal use.

GetFileTransfer

For internal use.

GetHFMErrLogRecordSet

For internal use.

GetSystemDataLinkFile

Returns the name and path of the data link file for the application server.

Note: The data link file is defined with Financial Management Server Administrator.

Syntax

```
<HsxServer>.GetSystemDataLinkFile()
```

Return Value

String. Returns the data link file's name and path.

Example

This example assigns the name and path of the data link file to the `sDataLink` variable.

```
Dim sDataLink as String
sDataLink = m_cHsxServer.GetSystemDataLinkFile()
```

GetSystemFolder

Returns the name and path of the system file for the application server.

Note: The system file is defined with Financial Management Server Administrator.

Syntax

```
<HsxServer>.GetSystemFolder()
```

Return Value

String. Returns the name and path of the system folder.

Example

This example assigns the name and path of the data link file to the `sSysFolder` variable.

```
Dim sSysFolder As String
sSysFolder = m_CHsxServer.GetSystemFolder ()
```

GetXMLErrorFromDatabase

Returns the message for a system error, given the system error's reference number.

Syntax

```
<HsxServer>.GetXMLErrorFromDatabase (bstrErrorReference)
```

Argument	Description
<i>bstrErrorReference</i>	String (ByVal). The system error's reference number.

Return Value

String. Returns the system error message.

GetXMLErrorsListFromDatabase

Returns the reference numbers, log types, timestamps, application server names, and application names of system errors. 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

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

Argument	Description
<i>dStartTimeStamp</i>	Double (ByVal). The start of the date range to filter by, or 0 to omit filtering by timestamp.
<i>dEndTimeStamp</i>	Double (ByVal). The end of the date range to filter by, or 0 to omit filtering by timestamp.
<i>bstrServerName</i>	String (ByVal). The name of the application server to filter by, or an empty string to omit filtering by server.

Argument	Description
<i>bstrApplicationName</i>	String (ByVal). The name of the application to filter by or an empty string to omit filtering by server.
<i>pvarabstrReference</i>	Variant. Returns an array of strings containing the reference numbers of the system errors that match the search criteria.
<i>pvarabstrLogType</i>	Variant. Returns an array of strings containing the errors' log types. The valid return values are represented by the constants in “Log Severity Constants” on page 873 .
<i>pvarabstrTimeStamp</i>	Variant. Returns an array of strings containing the errors' timestamps.
<i>pvarabstrServerName</i>	Variant. Returns an array of strings containing the errors' application server names.
<i>pvarabstrApplicationName</i>	Variant. Returns an array of strings containing the errors' application names.

ScriptableEnumDataSources

For internal use.

ScriptableEnumRegisteredDSNs

For internal use.

HsxServer Object Properties

The HsxServer object contains one property, which is [CSSEnabled](#).

CSSEnabled

Indicates whether external authentication is enabled for the cluster. This property is a Boolean that returns TRUE if external authentication is enabled, FALSE otherwise.

7

HsvSession Type Library

In This Chapter

HsvSession Object Properties	157
HsvSession Object Methods	159

This chapter describes the members of the HsvSession type library. The HsvSession object's properties return object references for child objects such as HsvMetadata and HsvData. The HsvSession object also contains methods that enable you to instantiate custom server-side objects and to check whether application-related system information such as metadata has changed.

To use the HsvSession type library, you must reference `HsvSession.dll` in your project. The HsvSession type library contains one object—the HsvSession object.

HsvSession Object Properties

The HsvSession object's properties return object references to the child objects of the HsvSession object. These properties are summarized in [Table 7 on page 54](#), and are described in detail in the following topics.

Note: HsvSession object references are returned by the `OpenApplication` methods of the `HsxClient` and `HsxClientUI` objects.

Calculate

Returns an HsvCalculate object reference as shown in the following example:

```
Set cCalculate = m_cSession.Calculate
```

Data

Returns an HsvData object reference as shown in the following example:

```
Set cData = m_cSession.Data
```

ICM

Returns an object reference to the HsvICM object or the IHsvAdminICM interface. For more information, see [“HsvICM Object Methods” on page 711](#) and [“IHsvAdminICM Interface Methods” on page 732](#).

Journals

Returns an HsvJournals, IHsvJournalsEx, or IHsvJournalsReport object reference as shown in the following example:

```
Set cJournals = m_cSession.Journals  
Set cIJournalsEx = m_cSession.Journals  
Set cIJournalsReport = m_cSession.Journals
```

Metadata

Returns an HsvMetadata object reference as shown in the following example:

```
Set cMetadata = m_cSession.Metadata
```

ProcessFlow

Returns an HsvProcessFlow object reference as shown in the following example:

```
Set cProcFlow = m_cSession.ProcessFlow
```

Reports

Returns an HsvReports object reference as shown in the following example:

```
Set cHsvReports = cHsvSession.Reports
```

Security

Returns an HsvSecurityAccess or IHsvDataSecurity object reference as shown in the following example:

```
Set cHsvSecurityAccess = cHsvSession.Security  
Set cIHsvDataSecurity = cHsvSession.Security
```

SystemInfo

Returns an HsvSystemInfo object reference as shown in the following example:

```
Set cSysInfo= m_cSession.SystemInfo
```

HsvSession Object Methods

The HsvSession object's methods are summarized in [Table 8 on page 55](#), and are described in detail in the following topics.

CreateObject

Instantiates an object on the application server on which an application has been opened. Use `CreateObject` to instantiate server-side objects that you develop. For example, you can use `CreateObject` to instantiate server-side Application Components (ACMs).

Tip: To instantiate a server-side object without opening an application, use `HsvClient.CreateObjectOnServer`, which takes the application server name as an argument. For details, see [“CreateObjectOnServer” on page 126](#).

Syntax

```
<HsvSession>.CreateObject (bstrClassIDOrProgID)
```

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

<code>bstrClassIDOrProgID</code>	String (ByVal). The name of the object to be instantiated.
----------------------------------	--

Return Value

`Object.CreateObject` returns an instance of the object identified by the argument.

Example

`CreateObject` is used in the example for the `HsvStarSchemaACM` method [EnumRegisteredDSNs](#).

GetLicenseExpirationStatus

Deprecated.

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

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

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

<i>pvbMajorChange</i>	Boolean. Returns TRUE if the application's system information has changed, FALSE otherwise.
-----------------------	---

<i>pvbMinorChange</i>	Boolean. <i>For internal use.</i>
-----------------------	-----------------------------------

HasUserStatusChanged

Indicates whether the connected user has been logged off by an administrator.

Syntax

```
<HsvSession>.HasUserStatusChanged pvbKill, pvbWarn
```

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

<i>pvbKill</i>	Boolean. Indicates whether the user has been logged off. Returns TRUE if the user has been logged off, FALSE otherwise.
----------------	---

<i>pvbWarn</i>	Boolean. <i>For internal use.</i>
----------------	-----------------------------------

IsBusy

For internal use.

IsRunningTasks

For internal use.

LockMetadataLoadWithSystemChangeCheck

For internal use.

UnlockMetadataLoad

For internal use.



HsvMetadata Type Library

In This Chapter

About Member IDs	161
HsvMetadata Object Methods	162
HsvMetadata Object Properties	185
IHsvTreeInfo Interface Methods	188
HsvAccounts Object Methods	232
HsvCustom Object Methods	244
HsvEntities Object Methods	247
HsvICPs Object Methods	254
HsvPeriods Object Methods	256
HsvScenarios Object Methods	259
HsvValues Object Methods	268
HsvYears Object Methods	269
HsvCurrencies Object Methods	270

This chapter describes the members of the HsvMetadata type library. The properties and methods of this type library are used to get attributes of dimension members, to get information on dimension hierarchies and member lists, and to load and extract metadata.

To use the HsvMetadata type library, you must reference `HsvMetadata.dll` in your project.

The HsvMetadata type library contains several objects. The HsvMetadata object is parent of the other objects in this type library. For an overview of these objects, see “[HsvMetadata Type Library Overview](#)” on page 55.

About Member IDs

Many HsvMetadata methods take or return Long IDs that identify dimension members. These member IDs are used by many methods in the other type libraries. For example, several HsvData object methods take member IDs that identify the cells for which data is being get and set.

You get member IDs with various methods of the IHsvTreeInfo interface, which is implemented for all of the dimension-related objects. There are several methods that return member IDs, including the following methods:

- `GetItemID` takes a member’s label and returns the member’s ID.

- [EnumAllMemberIDs](#) returns an array containing the member IDs of all of a dimension's members.
- [EnumAllParentAndChildIDs](#) returns arrays containing the member IDs of a dimension's parent and child members. The arrays have a one-to-one correspondence that represents a dimension's parent-child hierarchy.
- [EnumMembers](#) returns arrays containing the parent and child members of a member list.

HsvMetadata Object Methods

The HsvMetadata object's methods return information on consolidation methods, load and extract metadata files on application servers, get the languages defined for applications, and return certain types of metadata-related information. These methods are summarized in [Table 11 on page 57](#).

Tip: The HsvMetadata object also contains several properties. For more information, see [“HsvMetadata Object Properties” on page 185](#).

Set HsvMetadata object references with the `Metadata` property of the HsvSession object as shown in the following example:

```
Dim cHsvMetadata as HsvMetadata
Set cHsvMetadata = m_cHsvSession.MetaData
```

EnumConsolidationMethodIDs

Returns IDs that identify an application's consolidation methods.

Tip: You can use these IDs with [GetConsolidationMethodInfo](#) and [GetConsolidationMethodDescription](#).

Syntax

```
<HsvMetadata>.EnumConsolidationMethodIDs pvaralIDs
```

Argument Description

pvaralIDs Variant array. Returns the IDs of the application's consolidation methods. The array is returned as a Long subtype.

Example

`EnumConsolidationMethodIDs` is used in the example for [GetConsolidationMethodInfo](#).

EnumExtractOptions

Returns a two-dimensional array of the metadata extract options that can be passed to [Extract](#). The array includes the options' names and default values. For some options, the array also identifies the valid range of values.

Syntax

```
<HsvMetadata>.EnumExtractOptions pvar2daOptions
```

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

<i>pvar2daOptions</i>	Variant array. Returns a two-dimensional safe array that represents the metadata extract options.
-----------------------	---

The first dimension identifies the options, and is indexed from 1 to 19. The indexes and corresponding extract options are listed in [Table 57 on page 165](#).

The second dimension provides information on options, and is indexed from 0 to 5:

- 0 = The option's index in the array of options passed to `Extract`. (Long subtype).
- 1 = The option's name. (String subtype).
- 2 = The option's default value. (The subtype varies).
- 3 = The option's minimum value, if any. (Long subtype).
- 4 = The option's maximum value, if any (Long subtype).
- 5 = A tab-delimited list of the option's valid values (String subtype). If the option is not restricted to a set of values, a blank String is returned.

For example, since the delimiter option is the twelfth item in the first dimension, array item (12, 2) stores the system's default delimiter.

Note: Options without minimum and maximum values will return **0** for items 3 and 4 of the second dimension.

Example

The following example defines a function named `getMetadataExtractDefaults` that returns an array of the default extract option values. This function loops through the array returned by `EnumExtractOptions`, assigning each option's default value to the `vaMetaSettings` array. `vaMetaSettings` is then assigned as the function's return value.

```
Function getMetadataExtractDefaults() As Variant
Dim vaOpts As Variant, vaMetaSettings(1 To 19) As Variant
'g_cMetadata is an HsvMetadata object reference
g_cMetadata.EnumExtractOptions vaOpts
'Assign the default values, which are stored in
'item # 2 of the second dimension of vaOpts.
For i = LBound(vaOpts) To UBound(vaOpts)
    vaMetaSettings(i) = vaOpts(i, 2)
Next i
getMetadataExtractDefaults = vaMetaSettings
End Function
```

Tip: This function is used in the [Example](#) for `Extract`.

EnumLanguages

Returns the numeric IDs and labels of the valid languages for an application. The IDs are returned in one array, and the labels in a second array.

Tip: If you need to get the ID of a language, loop through the array of labels returned by `EnumLanguages`, then get the corresponding ID from the array of IDs.

Syntax

```
<HsvMetadata>.EnumLanguages pvaralIDs, pvarabstrLabels
```

Argument	Description
<i>pvaralIDs</i>	Variant array. Returns the IDs of the application's languages. The array is returned as a Long subtype.
<i>pvarabstrLabels</i>	Variant array. Returns the labels of the application's languages. The array is returned as a String subtype.

Example

`EnumLanguages` is used in the [Example](#) for `GetConsolidationMethodInfo`.

EnumLoadOptions

Returns a two-dimensional array of the metadata load options that can be passed to [Load](#). The array includes the options' names and default values. For some options, the array also identifies the valid range of values.

Syntax

```
<HsvMetadata>.EnumLoadOptions pvar2daOptions
```

Argument	Description
<i>pvar2daOptions</i>	<p>Variant array. Returns a two-dimensional safe array that represents the metadata load options.</p> <p>The first dimension identifies the options, and is indexed from 1 to 29. The indexes and corresponding load options are listed in Table 58 on page 181.</p> <p>The second dimension provides information on options, and is indexed from 0 to 5:</p> <ul style="list-style-type: none">● 0 = The option's index in the array of options passed to <code>Load</code>. (Long subtype).● 1 = The option's name. (String subtype).● 2 = The option's default value. (The subtype varies).● 3 = The option's minimum value, if any. (Long subtype).● 4 = The option's maximum value, if any (Long subtype).● 5 = A tab-delimited list of the option's valid values (String subtype). If the option is not restricted to a set of values, a blank String is returned. <p>For example, since the delimiter option is the seventeenth item in the first dimension, array item <code>(17, 2)</code> stores the system's default delimiter.</p>

Argument **Description**

Note: Options without minimum and maximum values will return **0** for items 3 and 4 of the second dimension.

Example

See the example for [Load](#).

Extract

Extracts metadata into a text file. The file will be created on the application server.

Tip: You can extract files onto client PCs with the HsvMetadataLoadACV type library. This library also offers properties and methods that simplify handling of the metadata extract options. For more information, see “[Extracting Metadata](#) ” on page 760.

Syntax

```
<HsvMetadata>.Extract bstrServerFilename, bstrServerLogFilename,  
varavSettings
```

Argument **Description**

bstrServerFilename String (ByVal). The name and path of the metadata extract file. The path must exist on the application server.

bstrServerLogFilename String (ByVal). The name and path of the log file. The path must exist on the application server.

varavSettings Variant array (ByVal). The metadata extract options. The array is 1-based and contains 19 items. For details on valid indexes and values, see [Table 57 on page 165](#).

Tip: Use [EnumExtractOptions](#) to return information about the valid extract options.

The following table describes the metadata extract options. Note that the listed indexes apply to the array passed to `Extract` and to the first dimension of the array returned by `EnumExtractOptions`.

Table 57 Metadata Extract Options

Index	Extract Option Information
1	Option: Currencies Usage: Specifies whether currencies are extracted. Pass to Extract: Boolean – TRUE to extract currencies, otherwise FALSE.
2	Option: Scenarios Usage: Specifies whether scenarios are extracted. Pass to Extract: Boolean – TRUE to extract scenarios, otherwise FALSE.

Index	Extract Option Information
3	Option: Years <i>For internal use.</i>
4	Option: Periods <i>For internal use.</i>
5	Option: Views <i>For internal use.</i>
6	Option: Entities Usage: Specifies whether entities are extracted. Pass to Extract: Boolean – TRUE to extract entities, otherwise FALSE.
7	Option: Accounts Usage: Specifies whether accounts are extracted. Pass to Extract: Boolean – TRUE to extract accounts, otherwise FALSE.
8	Option: Custom1 Usage: Specifies whether Custom 1 dimension members are extracted. Pass to Extract: Boolean – TRUE to extract Custom 1 members, otherwise FALSE.
9	Option: Custom2 Usage: Specifies whether Custom 2 dimension members are extracted. Pass to Extract: Boolean – TRUE to extract Custom 2 members, otherwise FALSE.
10	Option: Custom3 Usage: Specifies whether Custom 3 dimension members are extracted. Pass to Extract: Boolean – TRUE to extract Custom 3 members, otherwise FALSE.
11	Option: Custom4 Usage: Specifies whether Custom 4 dimension members are extracted. Pass to Extract: Boolean – TRUE to extract Custom 4 members, otherwise FALSE.
12	Option: Delimiter Usage: Specifies an extract file's delimiter. Pass to Extract: String – a valid delimiter character. <code>EnumExtractOptions</code> returns the valid delimiters.
13	Option: AppSettings Usage: Specifies whether application settings are extracted. Pass to Extract: Boolean – TRUE to extract application settings, otherwise FALSE.
14	Option: FileFormat Usage: Specifies whether the metadata extract file is in an ASCII text or an XML format. Pass to Extract: Long – 0 to extract into a text file, 1 to extract into an XML file.

Index	Extract Option Information
15	<p>Option: ConsolMethods</p> <p>Usage: Specifies whether consolidation methods are extracted.</p> <p>Pass to Extract: Boolean – TRUE to extract consolidation methods, otherwise FALSE.</p>
16	<p>Option: ExtractSystemMembers</p> <p>Usage: Specifies whether system members are extracted.</p> <p>Pass to Extract: Boolean – TRUE to extract system members, otherwise FALSE.</p> <p>Note: If this option is set to TRUE, you must also set to TRUE the options for the system members to be extracted.</p>
17	<p>Option: SystemAccounts</p> <p>Usage: Specifies whether system-generated Account dimension members are extracted.</p> <p>Pass to Extract: Boolean – TRUE to extract system accounts, otherwise FALSE. If this option is set to TRUE you must also set the ExtractSystemMembers option to TRUE.</p>
18	<p>Option: Values</p> <p>Usage: Specifies whether Value dimension members are extracted.</p> <p>Pass to Extract: Boolean – TRUE to extract Value dimension members, otherwise FALSE. These are system members, so if this option is set to TRUE you must also set the ExtractSystemMembers option to TRUE.</p>
19	<p>Option: ICPS</p> <p>Usage: Specifies whether Intercompany Partner dimension members are extracted.</p> <p>Pass to Extract: Boolean – TRUE to extract Intercompany Partner dimension members, otherwise FALSE. These are system members, so if this option is set to TRUE you must also set the ExtractSystemMembers option to TRUE.</p>

Example

The following example extracts metadata into an XML file. The custom function `getMetadataExtractDefaults` assigns the default extract options to the `vaSettings` array; for the definition of this custom function, see the example for [EnumExtractOptions](#). The file format option — item 14 of the `vaSettings` array — is set to XML, then `Extract` is called.

```
Dim vaOpts As Variant, vaSettings() As Variant
vaSettings = getMetadataExtractDefaults()
vaSettings(14) = 1
'g_cMetadata is an HsvMetadata object reference
g_cMetadata.Extract "c:\acme\myAppExt.xml", _
    "c:\acme\myappExt.log", vaSettings
```

GetApplicationAttribute

Returns the raw value of a given application setting attribute.

Tip: To obtain a user-readable string representing the attribute's value, pass the raw value to [TranslateApplicationAttributeForDisplay](#).

Syntax

```
<HsvMetadata>.GetApplicationAttribute(sAttributeID)
```

Argument Description

sAttributeID Integer (ByVal). 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 840](#).

Return Value

VARIANT. Returns the raw value of the attribute.

Example

The following function returns the user-readable string that represents the value of a given application setting attribute.

```
Function getAppAttrString(iAttrId As Long) As String
Dim cMetadata As HsvMetadata, vRet, sRet As String
'g_cSession is an HsvSession object reference
Set cMetadata = g_cSession.Metadata
vRet = cMetadata.GetApplicationAttribute(iAttrId)
sRet = g_cMetadata.TranslateApplicationAttributeForDisplay(iAttrId, vRet)
getAppAttrString = sRet
End Function
```

GetApplicationCurrency

Returns the label of the application's default currency.

Syntax

```
<HsvMetadata>.GetApplicationCurrency pbstrAppCurr
```

Argument Description

pbstrAppCurr String. Returns the label of the default currency.

GetApplicationSettingsTimeStamp

Returns a timestamp that indicates when the application settings were last updated.

Syntax

```
<HsvMetadata>.GetApplicationSettingsTimeStamp pdTimeStamp
```

Argument Description

pdTimeStamp Double. Returns the timestamp showing when the settings were updated.

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

The timestamp can be converted to a Date format; for example, in Visual Basic you can convert with CDate.

Example

The following function returns the application settings timestamp, converted to a Date format.

```
Function getAppSetStamp() As Date
Dim dTime As Double
'g_cMetadata is a previously set HsvMetadata object
g_cMetadata.GetApplicationSettingsTimeStamp dTime
getAppSetStamp = CDate(dTime)
End Function
```

GetByIndexValidationAccount

Returns the member ID of a validation account, given the index of the account.

Note: You can also obtain the member ID of the primary validation account with [GetValidationAccount](#).

Syntax

```
<HsvMetadata>.GetByIndexValidationAccount lIndex, plAccountID
```

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

lIndex Long (ByVal). The index of the validation account.

An application has settings for the primary Validation Account and a series of validation accounts; the names of these settings take the form of **Validation Account *n***. The index is one-based and corresponds to *n*. For example, to obtain the member ID of Validation Account 2, pass 2.

plAccountID Long. Returns the member ID. If there is no validation account corresponding to the specified index, -1 is returned.

Example

The following function returns the label of the validation account for the specified index.

```
Function getValAcctName(lIndex As Long) As String
Dim lId As Long, cMetadata As HsvMetadata, cTreeInfo As IHsvTreeInfo
Dim sLabel As String
'g_cSession represents an HsvSession instance
Set cMetadata = g_cSession.Metadata
Set cTreeInfo = cMetadata.Accounts
cMetadata.GetByIndexValidationAccount lIndex, lId
cTreeInfo.GetLabel lId, sLabel
getValAcctName = sLabel
End Function
```

GetCellLevelAccountType

Returns the account type of a cell, given the member IDs of the cell's Account and Custom dimension members. A cell's account type can differ from that of the cell's Account dimension member in cases where the SwitchSignForFlow attribute or the SwitchSignForFlow attribute, or both, is enabled for one or more of the cell's Custom dimension members.

Tip: `HsvAccounts.GetAccountType` returns an Account dimension member's account type. For details, see [“GetAccountType” on page 232](#).

Syntax

```
<HsvMetadata>.GetCellLevelAccountType lAccount, lCustom1, lCustom2, lCustom3, lCustom4, psAccountType
```

Argument	Description
<i>lAccount</i>	Long (ByVal). The member ID of the Account dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the Custom 4 dimension member.
<i>psAccountType</i>	Integer. Returns the cell's account type. For a list of constants that represent the valid return values, see “Account Type Constants” on page 826 .

GetConsolidationMethodDescription

Returns a consolidation method's description for the specified language.

Syntax

```
<HsvMetadata>.GetConsolidationMethodDescription lMethodID, lLangID, pbstrDescription
```

Argument	Description
<i>lMethodID</i>	Long (ByVal). The ID of the consolidation method. Tip: You can get consolidation method IDs with EnumConsolidationMethodIDs .
<i>lLangID</i>	Long (ByVal). The ID of the language. Tip: For a list of constants that represent supported languages, see Table 101 on page 833 .
<i>pbstrDescription</i>	String. The consolidation method's description for the language identified by the <i>lLangID</i> argument.

Example

`GetConsolidationMethodDescription` is used in the [Example](#) for `GetConsolidationMethodInfo`.

GetConsolidationMethodInfo

Returns the attributes of a consolidation method, given the method's ID.

Note: To get a consolidation method's description, use [GetConsolidationMethodDescription](#).

Syntax

```
<HsvMetadata>.GetConsolidationMethodInfo lConsolMethodID, pbstrLabel,
pbUsedByCalcRoutine, pbIsHoldingMethod, psToPercentControlOp,
pdToPercentControl, pdPercentConsol, psControl
```

Argument	Description
<i>lConsolMethodID</i>	Long (ByVal). The consolidation method's ID. Tip: You can get consolidation method IDs with EnumConsolidationMethodIDs .
<i>pbstrLabel</i>	String. The method's label.
<i>pbUsedByCalcRoutine</i>	Boolean. Returns the method's <code>UsedByCalcRoutine</code> attribute.
<i>pbIsHoldingMethod</i>	Boolean. Returns the method's <code>IsHoldingMethod</code> attribute.
<i>psToPercentControlOp</i>	Integer. Returns the method's <code>ToPercentControlComp</code> attribute. Constants that represent the valid return values are listed in Table 94 on page 830 .
<i>pdToPercentControl</i>	Double. Returns the method's <code>ToPercentControl</code> attribute.
<i>pdPercentConsol</i>	Double. Returns the method's <code>PercentConsol</code> attribute. Note: There are predefined values that can be used for this attribute. Constants that represent these predefined values are listed in Table 95 on page 830 .
<i>psControl</i>	Integer. Returns the method's <code>Control</code> attribute. Constants that represent the valid return values are listed in Table 93 on page 830 .

Example

This example inserts a consolidation method's description for the user's language into a text box. `EnumConsolidationMethodIDs` gets the IDs of the application's consolidation methods. A loop searches these IDs for that of the consolidation method named "M5," then `GetConsolidationMethodInfo` returns information for this method. The user's language is determined by `HsvSystemInfo.GetLanguageParameters`, and the application's valid languages are returned by `EnumLanguages`. A loop searches these languages for the user's language. The ID of this language is passed to `GetConsolidationMethodDescription`, and the description returned is inserted into a text box.

```
Dim vaIDs As Variant, sLabel As String, bCalc As Boolean
```

```

Dim bHold As Boolean, iPctControl As Integer, dConsol As Double
Dim iControl As Integer, cSysInfo As HsvSystemInfo
Dim lUserLang As Long, sDesc As String, vaLangIDs As Variant
Dim vaLangDescs As Variant, dPctConsol As Double
m_cMetadata.EnumConsolidationMethodIDs vaIDs
For i = LBound(vaIDs) To UBound(vaIDs)
    m_cMetadata.GetConsolidationMethodInfo vaIDs(i), sLabel, _
        bCalc, bHold, iPctControl, dConsol, dPctConsol, iControl
    If sLabel = "M5" Then
        Set cSysInfo = m_cSession.SystemInfo
        cSysInfo.GetLanguageUserParameters lUserLang
        m_cMetadata.EnumLanguages vaLangIDs, vaLangDescs
        For j = LBound(vaLangIDs) To UBound(vaLangIDs)
            If vaLangIDs(j) = lUserLang Then
                m_cMetadata.GetConsolidationMethodDescription _
                    vaIDs(i), vaLangIDs(j), sDesc
                txtMethDescription.Text = sDesc
            Exit Sub
        End If
    Next j
End If
Next i
End Sub

```

GetConsolidationMethodsTimeStamp

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

Syntax

```
<HsvMetadata>.GetConsolidationMethodsTimeStamp pdTimeStamp
```

Argument Description

pdTimeStamp Double. Returns the timestamp showing when the consolidation methods were updated.
The timestamp can be converted to a Date format; for example, in Visual Basic you can convert with CDate.

Example

The following function returns the consolidation methods timestamp, converted to a Date format.

```

Function getConsolMethStamp() As Date
Dim dTime As Double
'g_cMetadata is a previously set HsvMetadata object
g_cMetadata.GetConsolidationMethodsTimeStamp dTime
getConsolMethStamp = CDate(dTime)
End Function

```

GetCurrencyTimeStamp

Returns a timestamp that indicates when the application's currencies were last updated.

Syntax

```
<HsvMetadata>.GetCurrencyTimeStamp pdTimeStamp
```

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

<i>pdTimeStamp</i>	Double. Returns the timestamp showing when the currencies were updated. The timestamp can be converted to a Date format; for example, in Visual Basic you can convert with CDate.
--------------------	--

Example

The following function returns the currencies timestamp, converted to a Date format.

```
Function getCurrencyStamp() As Date
Dim dTime As Double
'g_cMetadata is a previously set HsvMetadata object
g_cMetadata.GetCurrencyTimeStamp dTime
getCurrencyStamp = CDate(dTime)
End Function
```

GetCurrencyValueIDForEntityValueCombination

Returns the currency value ID for the specified entity-value combination.

Syntax

```
<HsvMetadata>.GetCurrencyValueIDForEntityValueCombination
(lEntity, lParent, lValue)
```

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

<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension.
----------------	--

<i>lParent</i>	Long (ByVal). The member ID of the Parent dimension.
----------------	--

<i>lValue</i>	Long (ByVal). The member ID of the Value dimension.
---------------	---

Return Value

Long. Returns the currency value ID.

GetCurrencyValueIDsForEntityValueCombinations

Returns the currency value IDs for the entity value combinations.

Syntax

```
<HsvMetadata>.GetCurrencyValueIDsForEntityValueCombinations_  
(varalEntities, varalParents, varalValues)
```

Argument Description

varalEntities Long Array (ByVal). The member IDs of the Entity dimension.

varalParents Long Array (ByVal). The member IDs of the Parent dimension

varalValues Long Array (ByVal). The member IDs of the Value dimension.

Return Value

Variant. Returns the currency value IDs.

GetDefaultValueOfActiveStatusAccount

Returns the default value for an application's active status account. In other words, this method returns the DefaultValueForActive application setting's value.

Syntax

```
<HsvMetadata>.GetDefaultValueOfActiveStatusAccount pdValue
```

Argument Description

pdValue Double. Returns the active status account's default value, which is either 0 or 1.

GetFdmAppName

Returns the value of the FDM Application Name setting for the application.

Syntax

```
<HsvMetadata>.GetFdmAppName()
```

Return Value

String

Returns the setting's value.

GetFrequencyID

The frequency (YTD, MTD, QTD, and so forth) for which to return an internal numeric ID.

Syntax

```
<HsvMetadata>.GetFrequencyID bstrFrequencyLabel, plFrequencyID
```

Argument	Description
<i>bstrFrequencyLabel</i>	String (ByVal). The app for which to return a frequency (YTD, MTD, QTD, and so forth).
<i>plFrequencyID</i>	Long (ByVal). Returns a numeric value for the frequency input for <i>bstrFrequencyLabel</i> .

GetICPEntitiesAggregationWeight

For internal use.

GetSupportSubmissionPhaseForAccountFlag

Indicates whether phased submissions are enabled for the Account dimension.

Syntax

```
<HsvMetadata>.GetSupportSubmissionPhaseForAccountFlag  
pbSupportSubmissionPhaseForAccount
```

Argument	Description
<i>pbSupportSubmissionPhaseForAccount</i>	Boolean. Returns TRUE if phased submissions are enabled for the dimension.

Example

GetSupportSubmissionPhaseForAccountFlag is used in the example for [GetUseSubmissionPhaseFlag](#).

GetSupportSubmissionPhaseForCustom1Flag

Indicates whether phased submissions are enabled for the Custom1 dimension.

Syntax

```
<HsvMetadata>.GetSupportSubmissionPhaseForCustom1Flag  
pbSupportSubmissionPhaseForCustom1
```

Argument	Description
<i>pbSupportSubmissionPhaseForCustom1</i>	Boolean. Returns TRUE if phased submissions are enabled for the dimension.

Example

GetSupportSubmissionPhaseForCustom1Flag is used in the example for [GetUseSubmissionPhaseFlag](#).

GetSupportSubmissionPhaseForCustom2Flag

Syntax

Indicates whether phased submissions are enabled for the Custom2 dimension.

```
<HsvMetadata>.GetSupportSubmissionPhaseForCustom2Flag  
pbSupportSubmissionPhaseForCustom2
```

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

<i>pbSupportSubmissionPhaseForCustom2</i>	Boolean. Returns TRUE if phased submissions are enabled for the dimension.
---	--

Example

GetSupportSubmissionPhaseForCustom2Flag is used in the example for [GetUseSubmissionPhaseFlag](#).

GetSupportSubmissionPhaseForCustom3Flag

Indicates whether phased submissions are enabled for the Custom3 dimension.

Syntax

```
<HsvMetadata>.GetSupportSubmissionPhaseForCustom3Flag  
pbSupportSubmissionPhaseForCustom3
```

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

<i>pbSupportSubmissionPhaseForCustom3</i>	Boolean. Returns TRUE if phased submissions are enabled for the dimension.
---	--

Example

GetSupportSubmissionPhaseForCustom3Flag is used in the example for [GetUseSubmissionPhaseFlag](#).

GetSupportSubmissionPhaseForCustom4Flag

Indicates whether phased submissions are enabled for the Custom4 dimension.

Syntax

```
<HsvMetadata>.GetSupportSubmissionPhaseForCustom4Flag  
pbSupportSubmissionPhaseForCustom4
```

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

<i>pbSupportSubmissionPhaseForCustom4</i>	Boolean. Returns TRUE if phased submissions are enabled for the dimension.
---	--

Example

GetSupportSubmissionPhaseForCustom4Flag is used in the example for [GetUseSubmissionPhaseFlag](#).

GetSupportSubmissionPhaseForICPFlag

Indicates whether phased submissions are enabled for the Intercompany Partner dimension.

Syntax

```
<HsvMetadata>.GetSupportSubmissionPhaseForICPFlag  
pbSupportSubmissionPhaseForICP
```

Argument

Description

pbSupportSubmissionPhaseForICP Boolean. Returns TRUE if phased submissions are enabled for the dimension.

Example

GetSupportSubmissionPhaseForICPFlag is used in the example for [GetUseSubmissionPhaseFlag](#).

GetUseSubmissionPhaseFlag

Indicates whether phased submissions are enabled for the application.

Syntax

```
<HsvMetadata>.GetUseSubmissionPhaseFlag vbUseSubmissionPhaseFlag
```

Argument

Description

vbUseSubmissionPhaseFlag Boolean. Returns TRUE if phased submissions are enabled.

Example

The following example shows the usage of various HsvMetadata type library methods for phased submissions. If phased submissions are enabled, the example tests whether the application uses the Account, Intercompany Partner, and Custom dimensions for phased submissions. If a dimension is used, the example obtains the ID of the submission group for the passed dimension member. The results are printed to Visual Basic's Immediate window.

```
Sub printSubmissionPhaseProps(lAcct As Long, lIcp As Long, lC1 As Long, _  
    lC2 As Long, lC3 As Long, lC4 As Long)  
Dim cMetadata As HsvMetadata, cAcct As HsvAccounts, cIcp As HsvICPs  
Dim cC1 As HsvCustom, cC2 As HsvCustom, cC3 As HsvCustom, cC4 As HsvCustom  
Dim lGroup As Long, bRet As Boolean  
'g_cSession is a global that stores the HsvSession instance  
Set cMetadata = g_cSession.Metadata  
Set cAcct = cMetadata.Accounts
```

```

Set cIcp = cMetadata.ICPs
Set cC1 = cMetadata.Custom1
Set cC2 = cMetadata.Custom2
Set cC3 = cMetadata.Custom3
Set cC4 = cMetadata.Custom4
cMetadata.GetUseSubmissionPhaseFlag bRet
If bRet = True Then
    Debug.Print "Phased submissions enabled."
    cMetadata.GetSupportSubmissionPhaseForAccountFlag (bRet)
    If bRet = True Then
        Debug.Print "Account dimension used:"
        cAcct.GetSubmissionGroup lAcct, lGroup
        Debug.Print "    Account's Group = " + CStr(lGroup)
    End If
    cMetadata.GetSupportSubmissionPhaseForICPFlag (bRet)
    If bRet = True Then
        Debug.Print "ICP dimension used:"
        cIcp.GetSubmissionGroup lIcp, lGroup
        Debug.Print "    ICP Group = " + CStr(lGroup)
    End If
    cMetadata.GetSupportSubmissionPhaseForCustom1Flag (bRet)
    If bRet = True Then
        Debug.Print "Custom1 dimension used:"
        cC1.GetSubmissionGroup lC1, lGroup
        Debug.Print "    Custom1 Group = " + CStr(lGroup)
    End If
    cMetadata.GetSupportSubmissionPhaseForCustom2Flag (bRet)
    If bRet = True Then
        Debug.Print "Custom2 dimension used:"
        cC2.GetSubmissionGroup lC2, lGroup
        Debug.Print "    Custom2 Group = " + CStr(lGroup)
    End If
    cMetadata.GetSupportSubmissionPhaseForCustom3Flag (bRet)
    If bRet = True Then
        Debug.Print "Custom3 dimension used:"
        cC3.GetSubmissionGroup lC3, lGroup
        Debug.Print "    Custom3 Group = " + CStr(lGroup)
    End If
    cMetadata.GetSupportSubmissionPhaseForCustom4Flag (bRet)
    If bRet = True Then
        Debug.Print "Custom4 dimension used:"
        cC4.GetSubmissionGroup lC4, lGroup
        Debug.Print "    Custom4 Group = " + CStr(lGroup)
    End If
Else
    Debug.Print "Phased submissions disabled."
End If
End Sub

```

GetValidationAccount

Returns the member ID of an application's Validation Account setting.

Note: To return the member ID of a Validation Account *n* property, use [GetByIndexValidationAccount](#).

Syntax

```
<HsvMetadata>.GetValidationAccount plAccountID
```

Argument Description

plAccountID Long. Returns the account's member ID. If the application does not have a validation account, -1 is returned.

Example

The following example tests the value returned by `GetValidationAccount`. If it is greater than -1, `IHsvTreeInfo.GetLabel` gets the validation account's label, which is then placed into a text box. If `GetValidationAccount` returns -1, then the text box indicates that the application lacks a validation account.

```
Dim lAcctID As Long, sAcctName As String
m_cMetadata.GetValidationAccount lAcctID
If lAcctID > -1 Then
    m_cTreeInfo.GetLabel lAcctID, sAcctName
    txtValAcct.Text = sAcctName
Else
    txtValAcct.Text = "No validation account."
End If
```

IsCustomMemberValidForAccount

Returns a Boolean that indicates whether a Custom dimension member is valid for an account.

Tip: The valid Custom dimensions for an account are defined by the account's `Custom1TopMember`, `Custom2TopMember`, `Custom3TopMember`, and `Custom4TopMember` attributes.

Syntax

```
<HsvMetadata>.IsCustomMemberValidForAccount lDimID, lCustomID, lAccountID,
pvbIsValid
```

Argument Description

lDimID Long (ByVal). The number that identifies the Custom dimension to be tested. Use one of the Custom dimension constants listed in ["Dimension ID Constants" on page 834](#).

lCustomID Long (ByVal). The member ID of the Custom dimension member.

lAccountID Long (ByVal). The member ID of the account.

pvbIsValid Boolean. Returns TRUE if the Custom dimension member is valid for the account, otherwise FALSE.

Example

This example tests whether the Custom 1 dimension member named Golf is valid for the AdminExpenses account. `GetItemID` gets the IDs of the Golf and AdminExpenses members,

and the example then passes these IDs to `IsCustomMemberValidForAccount`. If Golf is a valid Custom 1 member for the AdminExpenses account, then `IsCustomMemberValidForAccount` sets `bIsValid` to `TRUE`, and any code placed within the `If` structure would be executed.

```
Dim cTreeInfo As IHsvTreeInfo, lCust1ID As Long, lAcct As Long
Dim bIsValid As Boolean
'g_cMetadata is an HsvMetadata object reference
Set cTreeInfo = g_cMetadata.Accounts
lAcct = cTreeInfo.GetItemID("AdminExpenses")
Set cTreeInfo = g_cMetadata.Custom1
lCust1ID = cTreeInfo.GetItemID("Golf")
g_cMetadata.IsCustomMemberValidForAccount DIMENSIONCUSTOM1, _
    lCust1ID, lAcct, bIsValid
If bIsValid = True Then
    ...
End If
```

IsOrgByPeriodApplication

Indicates whether the `OrgByPeriodApplication` application setting is on or off for the application to which the user is connected. In other words, this method indicates whether the Organization by Period feature is enabled.

Syntax

```
<HsvMetadata>.IsOrgByPeriodApplication pvarbIsEnabled
```

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

<i>pvarbIsEnabled</i>	Boolean. Returns <code>TRUE</code> if the <code>OrgByPeriodApplication</code> application setting is on, <code>FALSE</code> if it is off.
-----------------------	---

Example

The following example tests the `OrgByPeriodApplication` application setting. If the setting is on, any code placed within the `If` structure would be executed.

```
Dim bOrgEnabled As Boolean
'g_cMetadata is an HsvMetadata object reference
g_cMetadata.IsOrgByPeriodApplication bOrgEnabled
If bOrgEnabled = True Then
    ... 'Insert code here
End If
```

Load

Loads metadata into a Classic application, using a load file on the application server.

Caution! The method will fail if executed against an application created with Performance Management Architect.

You can load files from client PCs with the `HsvMetadataLoadACV` type library. This library also offers properties and methods that simplify handling of the metadata load options. See [“Loading Metadata” on page 759](#).

Syntax

```
<HsvMetadata>.Load bstrServerFilename, bstrServerLogFilename, varavSettings
```

Argument	Description
<i>bstrServerFilename</i>	String (ByVal). The name and path of the metadata load file. This file must exist on the application server. Tip: The load file format has changed since the original release of Financial Management. If the load file is in an obsolete file format, error number 80040526 (hexadecimal) is thrown.
<i>bstrServerLogFilename</i>	String (ByVal). The name and path of the log file. The path must exist on the application server.
<i>varavSettings</i>	Variant array (ByVal). The load options for the metadata load operation. The array is 1-based and contains 29 items. For details on indexes and valid values, see Table 58 on page 181 . Tip: Use EnumLoadOptions to return information about the valid load options.

The following table describes the metadata load options. Note that the listed indexes apply to the array passed to `Load` and to the first dimension of the array returned by `EnumLoadOptions`.

Caution! If you set one or more of the “Clear” options to `TRUE`, the application’s data will be erased. These are the options indexed from 9 through 16.

Table 58 Metadata Load Options

Index	Load Option Information
1	Option: Currencies Usage: Specifies whether currencies are loaded. Pass to Load: Boolean – <code>TRUE</code> to load currencies, otherwise <code>FALSE</code> .
2	Option: Scenarios Usage: Specifies whether scenarios are loaded. Pass to Load: Boolean – <code>TRUE</code> to load scenarios, otherwise <code>FALSE</code> .
3	Option: Entities Usage: Specifies whether entities are loaded. Pass to Load: Boolean – <code>TRUE</code> to load entities, otherwise <code>FALSE</code> .
4	Option: Accounts Usage: Specifies whether accounts are loaded. Pass to Load: Boolean – <code>TRUE</code> to load accounts, otherwise <code>FALSE</code> .

Index	Load Option Information
5	<p>Option: Custom1</p> <p>Usage: Specifies whether Custom 1 dimension members are loaded.</p> <p>Pass to Load: Boolean – TRUE to load Custom 1 members, otherwise FALSE.</p>
6	<p>Option: Custom2</p> <p>Usage: Specifies whether Custom 2 dimension members are loaded.</p> <p>Pass to Load: Boolean – TRUE to load Custom 2 members, otherwise FALSE.</p>
7	<p>Option: Custom3</p> <p>Usage: Specifies whether Custom 3 dimension members are loaded.</p> <p>Pass to Load: Boolean – TRUE to load Custom 3 members, otherwise FALSE.</p>
8	<p>Option: Custom4</p> <p>Usage: Specifies whether Custom 4 dimension members are loaded.</p> <p>Pass to Load: Boolean – TRUE to load Custom 4 members, otherwise FALSE.</p>
9	<p>Option: ClearCurrencies</p> <p>Usage: Specifies whether previously loaded currencies are deleted before metadata is loaded.</p> <p>Pass to Load: Boolean – TRUE to delete previously loaded currencies, otherwise FALSE. Setting this to TRUE will erase all of the application's data.</p>
10	<p>Option: ClearScenarios</p> <p>Usage: Specifies whether previously loaded scenarios are deleted before metadata is loaded.</p> <p>Pass to Load: Boolean – TRUE to delete previously loaded scenarios, otherwise FALSE. Setting this to TRUE will erase all of the application's data.</p>
11	<p>Option: ClearEntities</p> <p>Usage: Specifies whether previously loaded entities are deleted before metadata is loaded.</p> <p>Pass to Load: Boolean – TRUE to delete previously loaded entities, otherwise FALSE. Setting this to TRUE will erase all of the application's data.</p>
12	<p>Option: ClearAccounts</p> <p>Usage: Specifies whether previously loaded accounts are deleted before metadata is loaded.</p> <p>Pass to Load: Boolean – TRUE to delete previously loaded accounts, otherwise FALSE. Setting this to TRUE will erase all of the application's data.</p>
13	<p>Option: ClearCustom1</p> <p>Usage: Specifies whether previously loaded Custom 1 dimension members are deleted before metadata is loaded.</p> <p>Pass to Load: Boolean – TRUE to delete previously loaded Custom 1 members, otherwise FALSE. Setting this to TRUE will erase all of the application's data.</p>
14	<p>Option: ClearCustom2</p> <p>Usage: Specifies whether previously loaded Custom 2 dimension members are deleted before metadata is loaded.</p>

Index	Load Option Information
	<p>Pass to Load: Boolean – TRUE to delete previously loaded Custom 2 members, otherwise FALSE. Setting this to TRUE will erase all of the application’s data.</p>
15	<p>Option: ClearCustom3</p> <p>Usage: Specifies whether previously loaded Custom 3 dimension members are deleted before metadata is loaded.</p> <p>Pass to Load: Boolean – TRUE to delete previously loaded Custom 3 members, otherwise FALSE. Setting this to TRUE will erase all of the application’s data.</p>
16	<p>Option: ClearCustom4</p> <p>Usage: Specifies whether previously loaded Custom 4 dimension members are deleted before metadata is loaded.</p> <p>Pass to Load: Boolean – TRUE to delete previously loaded Custom 4 members, otherwise FALSE. Setting this to TRUE will erase all of the application’s data.</p>
17	<p>Option: Delimiter</p> <p>Usage: Specifies a load file’s delimiter.</p> <p>Pass to Load: String – a valid delimiter character. Note that <code>EnumLoadOptions</code> returns the valid delimiters.</p>
18	<p>Option: Prescan</p> <p>Usage: Specifies whether a load file is loaded or is merely scanned for syntax accuracy when <code>Load</code> is called.</p> <p>Pass to Load: Boolean – TRUE to scan without loading, FALSE to load the metadata. By default, this option is set to FALSE.</p>
19	<p>Option: AppSettings</p> <p>Usage: Specifies whether application settings are loaded.</p> <p>Pass to Load: Boolean – TRUE to load application settings, otherwise FALSE.</p>
20	<p>Option: FileFormat</p> <p>Usage: Specifies whether the metadata load file is in an ASCII text or an XML format.</p> <p>Pass to Load: Long – 0 to load a text file, 1 to load an XML file.</p>
21	<p>Option: UseReplaceMode</p> <p>Usage: Specifies whether the metadata replaces or is merged with existing metadata.</p> <p>Pass to Load: Boolean – TRUE to replace existing metadata, FALSE to merge with existing metadata.</p>
22	<p>Option: ConsolMethods</p> <p>Usage: Specifies whether consolidation methods are loaded.</p> <p>Pass to Load: Boolean – TRUE to load consolidation methods, otherwise FALSE.</p>
23	<p>Option: ClearConsolMethods</p> <p>Usage: Specifies whether existing consolidation methods are cleared before metadata is loaded.</p> <p>Pass to Load: Boolean – TRUE to delete previously loaded consolidation methods, otherwise FALSE.</p>
24	<p>Option: LoadSystemMembers</p> <p>Usage: Specifies whether system members can be loaded.</p>

Index	Load Option Information
	<p>Pass to Load: Boolean – TRUE to load system members, otherwise FALSE.</p> <p>Note: If this option is set to TRUE, you must also set to TRUE the options for the system members to be loaded.</p>
25	<p>Option: SystemAccounts</p> <p>Usage: Specifies whether to load system accounts.</p> <p>Pass to Load: Boolean – TRUE to load system accounts, otherwise FALSE.</p>
26	<p>Option: Values</p> <p>Usage: Specifies whether Value dimension members are loaded.</p> <p>Pass to Load: Boolean – TRUE to load Value dimension members, otherwise FALSE. These are system members, so if this option is set to TRUE you must also set the LoadSystemMembers option to TRUE.</p>
27	<p>Option: ICs</p> <p>Usage: Specifies whether Intercompany Partner dimension members are loaded.</p> <p>Pass to Load: Boolean – TRUE to load Intercompany Partner dimension members, otherwise FALSE. These are system members, so if this option is set to TRUE you must also set the LoadSystemMembers option to TRUE.</p>
28	<i>For internal use.</i>
29	<p>Option: CheckIntegrity</p> <p>Usage: Specifies whether to validate the integrity of the metadata file against the metadata in the current application.</p> <p>Note: If integrity errors occur, they are noted in the log file and no portion of the file is loaded into the application.</p> <p>Pass to Load: Boolean – TRUE to check integrity, otherwise FALSE.</p>

Example

The following method loads metadata from an XML file. [EnumLoadOptions](#) gets the default load options, and then the method sets the file format option to XML.

```

Sub LoadHfmMetaXml(sLoadFile As String, sLogFile As String)
Dim vaOpts As Variant, vaSettings() As Variant
'g_cMetadata is an HsvMetadata object reference
g_cMetadata.EnumLoadOptions vaOpts
'Assign the default values, which are stored in
'item # 2 of the second dimension of vaOpts.
ReDim vaSettings(LBound(vaOpts) To UBound(vaOpts))
For i = LBound(vaOpts) To UBound(vaOpts)
    vaSettings(i) = vaOpts(i, 2)
Next i
vaSettings(20) = 1
g_cMetadata.Load sLoadFile, sLogFile, vaSettings
End Sub

```


LoadWithAccessCode

For internal use.

TranslateApplicationAttributeForDisplay

Returns a user-readable string that represents the value of a given application setting attribute. You must pass the raw value of the attribute.

Syntax

```
<HsvMetadata>.TranslateApplicationAttributeForDisplay(sAttributeID,  
varAttribValue)
```

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

<i>sAttributeID</i>	Integer (ByVal). 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 840 .
---------------------	--

<i>varAttribValue</i>	Variant (ByVal). The raw value of the attribute. To obtain an attribute’s raw value, use GetApplicationAttribute .
-----------------------	--

Return Value

String. The string that represents the attribute value.

Example

`TranslateApplicationAttributeForDisplay` is used in the example for [GetApplicationAttribute](#).

HsvMetadata Object Properties

Use the HsvMetadata properties to assign object references for HsvMetadata object’s child objects. These properties are summarized in [Table 10 on page 57](#), and are described in detail in the following topics.

Tip: For an example that shows how to assign an HsvMetadata object reference, see [“HsvMetadata Object Methods” on page 162](#).

Accounts

The `Accounts` property sets object references to the HsvAccounts object and to the IHsvTreeInfo interface. IHsvTreeInfo object references are set for use with the Account dimension.

Example

Accounts is used in the example for `HsvAccounts`. [GetAccountType](#).

Currencies

The `Currencies` property sets object references to the `HsvCurrencies` object.

Example

Currencies is used in the example for `HsvCurrencies`. [EnumCurrencies](#).

Custom1

The `Custom1` property sets object references to the `HsvCustom` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the `Custom1` dimension.

Example

Custom1 is used in the example for `HsvCustom`. [GetSecurityClassID](#).

Custom2

The `Custom2` property sets object references to the `HsvCustom` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the `Custom2` dimension.

Custom3

The `Custom3` property sets object references to the `HsvCustom` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the `Custom3` dimension.

Custom4

The `Custom4` property sets object references to the `HsvCustom` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the `Custom4` dimension.

Dimension

The `Dimension` property sets object references to the `IHsvTreeInfo` interface.

`Dimension` takes an `Integer` parameter. This parameter is a dimension ID, which is an ID that specifies the dimension for which the `IHsvTreeInfo` interface object reference is being set. The `HFMConstants` type library contains constants that represent the dimension IDs and that can be passed to `Dimension`. For a listing of these constants, see [“Dimension ID Constants” on page 834](#).

C# requires you to access the `Dimension` property with the accessor method `get_Dimension`.

Tip: The `Dimension` property can be used to define a useful custom function that takes in a dimension ID and a member name and returns the member's ID. For code samples, see [GetItemID](#).

Entities

The `Entities` property sets object references to the `HsvEntities` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the Entity dimension.

Example

`Entities` is used in the example for `HsvEntities`. [GetDefaultValueID](#).

ICPs

The `ICPs` property sets object references to the `HsvICPs` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the Intercompany Partner dimension.

Example

The following example sets an `IHsvTreeInfo` object reference:

```
Dim cTreeInfo As IHsvTreeInfo, vaLabels As Variant
'g_cMetadata is a previously set HsvMetadata object reference
Set cTreeInfo = g_cMetadata.ICPs
```

Periods

The `Periods` property sets object references to the `HsvPeriods` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the Period dimension.

Example

`Periods` is used in the example for `HsvPeriods`. [GetFrequency](#).

Scenarios

The `Scenarios` property sets object references to the `HsvScenarios` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the Scenario dimension.

Example

`Scenarios` is used in the example for `HsvScenarios`. [GetMissingDataZeroViewForAdjValues](#).

Values

The `Values` property sets object references to the `HsvValues` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the `Value` dimension.

Views

The `Views` property sets object references to the `HsvViews` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the `View` dimension.

Years

The `Years` property sets object references to the `HsvYears` object and to the `IHsvTreeInfo` interface. `IHsvTreeInfo` object references are set for use with the `Year` dimension.

Example

`Years` is used in the example for `HsvYears`. [GetYearRange](#).

IHsvTreeInfo Interface Methods

The `IHsvTreeInfo` interface is implemented by all of the dimension objects in the `HsvMetadata` type library; for more information, see [“HsvMetadata Type Library Overview” on page 55](#). The `IHsvTreeInfo` interface’s methods get IDs and labels of dimension members, and also get information about dimension hierarchies. These methods are summarized in [Table 19 on page 65](#), and are described in detail in the following topics.

To assign an `IHsvTreeInfo` object reference, use the `HsvMetadata` object property that corresponds to the applicable dimension. The following example uses the `Accounts` property to assign an `IHsvTreeInfo` object reference for the `Account` dimension:

```
Dim cMetadata as HsvMetadata, cTreeInfo as IHsvTreeInfo
Set cMetadata = m_cSession.Metadata
Set cTreeInfo = cMetadata.Accounts
```

Tip: You can also use the `HsvMetadata.Dimension` property to assign `IHsvTreeInfo` object references. For more information, see [“Dimension” on page 186](#).

EnumAllMemberIDs

Returns an array of the member IDs for all of a dimension’s members.

Syntax

```
<IHsvTreeInfo>.EnumAllMemberIDs pvaralMemberIDs
```

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

pvaralMemberIDs Variant array. Returns the dimension's member IDs. The array is returned as a Long subtype.

Note: System-generated members are included in the return value.

Example

EnumAllMemberIDs is used in the [Example](#) for `HsvAccounts.GetNumDecimalPlaces`.

EnumAllMemberLabels

Returns the labels of all the members of a dimension.

Syntax

```
<IHsvTreeInfo>.EnumAllMemberLabels pvaralMemberLabels
```

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

pvaralMemberLabels Variant array. Returns the dimension's member labels. The array is returned as a String subtype.

Note: System-generated members are included in the return value.

Example

The following example adds the names of an application's Value dimension members to a ComboBox control. The example sets an `IHsvTreeInfo` object reference for the Value dimension, then uses `EnumAllMemberLabels` to assign the Value dimension labels to the `vaMemNames` variable. The example then populates the combo box by looping through the array assigned to `vaMemNames`.

```
Dim cTreeInfo As IHsvTreeInfo, vaMemNames
'g_cMetadata is an HsvMetadata object reference
Set cTreeInfo = g_cMetadata.Values
cTreeInfo.EnumAllMemberLabels vaMemNames
For i = LBound(vaMemNames) To UBound(vaMemNames)
    'cmbValues is the ComboBox
    cmbValues.AddItem vaMemNames(i)
Next i
```

EnumAllParentAndChildIDs

Returns arrays that contain the member IDs of a dimension's parent and child members. The arrays are returned in the method's arguments; the first argument returns the parent IDs and the second argument returns the child IDs.

The elements in the two arrays have a one-to-one correspondence. For example, the member identified by the parent array's *third* element is the parent of the member identified by the child array's *third* element.

Syntax

```
<IHsvTreeInfo>.EnumAllParentAndChildIDs pvaralParentIDs, pvaralChildIDs
```

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

<i>pvaralParentIDs</i>	Variant array. Returns the IDs of the parent members. The array is returned as a Long subtype.
------------------------	--

<i>pvaralChildIDs</i>	Variant array. Returns the IDs of the child members. The array is returned as a Long subtype.
-----------------------	---

Note: System-generated members are included in the return value.

To better understand the relationship between the two arrays, see [Understanding the Arguments' Parent-Child Arrays](#). This topic describes the `EnumAllParentAndChildLabels` method, which returns a similar set of arrays, the only difference being that labels are returned instead of IDs.

Example

See the example for `EnumAllParentAndChildLabels`. The same logic applies to `EnumAllParentAndChildIDs`; the only difference is that `EnumAllParentAndChildLabels` returns member labels instead of IDs.

EnumAllParentAndChildLabels

Returns arrays that contain member labels and that represent the parent-child relationships of a dimension's members. The arrays are returned in the method's arguments; the first argument returns the parent labels and the second argument returns the child labels.

The elements in the two arrays have a one-to-one correspondence. For example, the member identified by the parent array's *first* element is the parent of the member identified by the child array's *first* element.

Syntax

```
<IHsvTreeInfo>.EnumAllParentAndChildLabels pvarabstrParents,  
pvarabstrChildren
```

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

<i>pvarabstrParents</i>	Variant array. Returns the labels of the parent members. The array is returned as a String subtype.
-------------------------	---

<i>pvarabstrChildren</i>	Variant array. Returns the labels of the child members. The array is returned as a String subtype.
--------------------------	--

Note: System-generated members are included in the return value.

Understanding the Arguments' Parent-Child Arrays

To understand how the two arrays relate to each other, consider the HIERARCHIES sections of metadata load files, which are documented in the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*. HIERARCHIES sections consist of comma-delimited records, where the first field contains the parent member's label and the second field contains the child member's label. Consider the following example of a HIERARCHIES section:

```

!HIERARCHIES=Entity
, [None]
, Regional
Regional, UnitedStates
UnitedStates, California
California, Sunnyvale
California, FosterCity

```

This HIERARCHIES section lists six parent-child combinations for the Entity dimension. The following table enumerates the elements of the arrays that `EnumAllParentAndChildLabels` would return for this set of entities:

Index #	Parent array element	Child array element
0		[None]
1		Regional
2	Regional	UnitedStates
3	UnitedStates	California
4	California	Sunnyvale
5	California	FosterCity

When an entity has no child, the corresponding parent array element will be an empty string; this is why the table shows no parent array element for the [None] and Regional entities.

Example

The following example displays the names of the parent entities for the entity specified in a combo box control. The example sets an `IHsvTreeInfo` object reference for the Entity dimension, then calls `EnumAllParentAndChildLabels` to populate the `vaParents` and `vaChildren` variables with arrays. The example then loops through the `vaChildren` array; when an array element contains the name of the entity specified in the `comboEnts` combo box, the entity's parent is concatenated to the `sParents` variable.

```

Dim cIHsvTreeInfo As IHsvTreeInfo, sParents As String
Dim vaParents, vaChildren
Set cIHsvTreeInfo = m_cMetadata.Dimension(DIMENSIONENTITY)
cIHsvTreeInfo.EnumAllParentAndChildLabels vaParents, vaChildren
For i = LBound(vaChildren) To UBound(vaChildren)
    If vaChildren(i) = comboEnts.Text Then
        sParents = sParents & vbCrLf & vaParents(i)
    End If
Next i
MsgBox comboEnts.Text & " parents: " & sParents

```

EnumAncestors

Returns an array containing the member IDs of a given member's ancestors.

Syntax

```
<IHsvTreeInfo>.EnumAncestors lItemID, bIgnoreDups, pvaralIDs
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the member.
<i>bIgnoreDups</i>	Boolean (ByVal). Specifies whether duplicate member IDs are returned. Pass TRUE to filter out duplicates, FALSE to return duplicates.
<i>pvaralIDs</i>	Variant array. Returns the array of the ancestors' member IDs. The array is returned as a Long subtype.

Example

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

```
Function getAncestors(lDim As Integer, sMem As String) As Variant
Dim lMember As Long, cTreeInfo As IHsvTreeInfo
Dim vaIDs, saLabels() As String
'g_cMetadata is an HsvMetadata object reference
Set cTreeInfo = g_cMetadata.Dimension(lDim)
lMember = cTreeInfo.GetItemID(sMem)
cTreeInfo.EnumAncestors lMember, False, vaIDs
'if the member has no ancestors, the return value is null
If IsEmpty(vaIDs) = True Then
    getAncestors = Null
    Exit Function
Else
    ReDim saLabels(UBound(vaIDs))
    For i = LBound(vaIDs) To UBound(vaIDs)
        cTreeInfo.GetLabel vaIDs(i), saLabels(i)
    Next i
    getAncestors = saLabels
End If
End Function
```

EnumBaseMemberIDs

Returns an array containing the member IDs of a given parent's base-level members. You can also use `EnumBaseMemberIDs` to get all of a dimension's base-level members.

Syntax

```
<IHsvTreeInfo>.EnumBaseMemberIDs lParentID, bIgnoreDuplicates,
pvaralBaseMemberIDs
```

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

<i>lParentID</i>	Long (ByVal). Identifies whether member IDs will be returned for a node or for the entire dimension: <ul style="list-style-type: none">● To return a node's base-level member IDs, pass the member ID of the node's parent.
------------------	---

Argument	Description
	<ul style="list-style-type: none"> To return all of a dimension's base-level members, pass -1.
<i>ignoreDuplicatas</i>	Boolean (ByVal). Specifies whether duplicate member IDs are returned. Pass TRUE to filter out duplicates, FALSE to return duplicates.
<i>pvaralBaseMemberIDs</i>	Variant array. Returns the array of the base-level members' IDs. The array is returned as a Long subtype.

Example

The following example prints the labels of the Regional entity's base-level members to the Immediate window. The example loops through the array returned by `EnumBaseMemberIDs`, passing the member IDs to `GetLabel`.

```
Dim cMetadata As HsvMetadata, lPar As Long
Dim cTreeInfo As IHsvTreeInfo, vaIDs
Dim sLabel As String
Set cMetadata = m_cSession.Metadata
Set cTreeInfo = cMetadata.Entities
lPar = cTreeInfo.GetItemID("Regional")
cTreeInfo.EnumBaseMemberIDs lPar, True, vaIDs
'Exit the sub if vaIDs is empty.
If IsEmpty(vaIDs) = True Then Exit Sub
For i = LBound(vaIDs) To UBound(vaIDs)
    cTreeInfo.GetLabel vaIDs(i), sLabel
    Debug.Print sLabel
Next i
```

EnumDefaultAncestors

Returns the member IDs of a given member's default ancestors. If the top-level member of the dimension hierarchy is passed, `EnumDefaultAncestors` returns an empty Variant.

Tip: You can return the labels of a member's default ancestors with [EnumDefaultAncestorsLabels](#).

Syntax

```
<IHsvTreeInfo>.EnumDefaultAncestors lMemberID, pvaralAncestorsIDs
```

Argument	Description
<i>lMemberID</i>	Long (ByVal). The member ID of the dimension member.
<i>pvaralAncestorsIDs</i>	Variant. Returns an array of the default ancestors' member IDs, or an empty Variant if a top-level member is passed to the <i>lMemberID</i> argument. If an array is returned, it has a subtype of Long.

EnumDefaultAncestorsLabels

Returns the labels of a given member's default ancestors. If the top-level member of the dimension hierarchy is passed, `EnumDefaultAncestorsLabels` returns an empty Variant.

Tip: You can return the member IDs of a member's default ancestors with [EnumDefaultAncestors](#).

Syntax

```
<IHsvTreeInfo>.EnumDefaultAncestorsLabels lMemberID,  
pvarabstrAncestorsLabels
```

Argument	Description
<i>lMemberID</i>	Long (ByVal). The member ID of the dimension member.
<i>pvarabstrAncestorsLabels</i>	Variant. Returns an array of the default ancestors' labels, or an empty Variant if a top-level member is passed to the <i>lMemberID</i> argument. If an array is returned, it has a subtype of String.

Example

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

```
Function getEntityDefAncestors(sMemberName As String) As Variant  
Dim cTreeInfo As IHsvTreeInfo, lMemID As Long  
Dim vaParLabels As Variant  
'm_cMetadata is an HsvMetadata object reference  
Set cTreeInfo = m_cMetadata.Entities  
lMemID = cTreeInfo.GetItemID(sMemberName)  
cTreeInfo.EnumDefaultAncestorsLabels lMemID, vaParLabels  
getEntityDefAncestors = vaParLabels  
End Function
```

EnumDescendants

Returns an array containing the member IDs of a given member's descendants.

Syntax

```
<IHsvTreeInfo>.EnumDescendants lItemID, bIgnoreDups, pvaralIDs
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the member.
<i>bIgnoreDups</i>	Boolean (ByVal). Specifies whether duplicate member IDs are returned. Pass TRUE to filter out duplicates, FALSE to return duplicates.

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

<i>pvarAllIDs</i>	Variant array. Returns the array of the descendants' member IDs. The array is returned as a Long subtype.
-------------------	---

Example

The following example prints the labels of the UnitedStates entity's descendant members to the Immediate window. The example loops through the array returned by EnumDescendants, passing the member IDs to GetLabel.

```
Dim cMetadata As HsvMetadata, lMember As Long
Dim cTreeInfo As IHsvTreeInfo, vaIDs
Dim sLabel As String
Set cMetadata = m_cSession.Metadata
Set cTreeInfo = cMetadata.Entities
lMember = cTreeInfo.GetItemID("UnitedStates")
cTreeInfo.EnumDescendants lMember, FALSE, vaIDs
'Exit the sub if there are no descendants
If IsEmpty(vaIDs) = True Then Exit Sub
For i = LBound(vaIDs) To UBound(vaIDs)
    cTreeInfo.GetLabel vaIDs(i), sLabel
    Debug.Print sLabel
Next i
```

EnumIDsOfChildren

Returns either the top members of a dimension hierarchy or the child members of a parent member.

Syntax

```
<IHsvTreeInfo>.EnumIDsOfChildren(lListTopMemberID, lItemID,
pvarChildIDArray)
```

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

<i>lListTopMemberID</i>	Long (ByVal). Pass the HFMCconstants type library constant TREE_ROOT to this argument.
-------------------------	--

<i>lItemID</i>	Long (ByVal). The value you pass depends upon whether you want to return the top members of a hierarchy or the children of a parent: <ul style="list-style-type: none">● To return the top members of a hierarchy, pass -1.● To return the children of a member, pass the parent's member ID.
----------------	--

<i>pvarChildIDArray</i>	Variant array. Returns the member IDs of either the dimension hierarchy's top members or of a parent's child members, depending upon the value passed to the <i>lItemID</i> .
-------------------------	---

Return Value

Integer. Indicates the success of the function call; returns 0 for success or -1 for an error.

Example

This example prints the member labels of the Entity dimension's top members to the Immediate window. Note how -1 is passed to both the *lListTopMemberID* and *lItemID* arguments.

```
Dim cMetadata As HsvMetadata, cTreeInfo As IHsvTreeInfo
Dim vaChildIDs, sLabel As String, iSuccess As Integer
Set cMetadata = m_cSession.Metadata
Set cTreeInfo = cMetadata.Entities
iSuccess = cTreeInfo.EnumIDsOfChildren(TREE_ROOT, -1, _
vaChildIDs)
If iSuccess = 0 Then
  For i = LBound(vaChildIDs) To UBound(vaChildIDs)
    cTreeInfo.GetLabel CLng(vaChildIDs(i)), sLabel
    Debug.Print sLabel
  Next i
End If
```

Tip: If you replaced the line above that calls `EnumIDsOfChildren` with the following lines, then the example would print the children of the California entity:

```
lPar = cTreeInfo.GetItemID("California")
iSuccess = cTreeInfo.EnumIDsOfChildren(TREE_ROOT, lPar, _
vaChildIDs)
```

EnumIDsOfChildren

Returns children and the parent item of the requested children.

Syntax

```
<HsvTreeInfo>.EnumIDsOfChildren(lListTopMemberID,
lItemID, pvarChildIDArray)
```

Argument	Description
<i>lListTopMemberID</i>	Long (ByVal). Pass the HFMConstants type library constant <code>TREE_ROOT</code> to this argument.
<i>lItemID</i>	Long (ByVal). The value you pass depends upon whether you want to return the top members of a hierarchy or the children of a parent: <ul style="list-style-type: none">● To return the top members of a hierarchy, pass -1.● To return the children of a member and the member itself, pass the parent's member ID.
<i>pvarChildIDArray</i>	Variant array. Returns the member IDs of either the dimension hierarchy's top members or of a parent's child members and the parent itself, depending upon the value passed to the <i>lItemID</i> .

Return Value

Integer. Indicates the success of the function call; returns 0 for success or -1 for an error.

Example

This example prints the member labels of the Entity dimension's children and the entity itself to the Immediate window.

```
Dim cMetadata As HsvMetadata, cTreeInfo As IHsvTreeInfo
Dim vaChildIDs, sLabel As String, iSuccess As Integer
Set cMetadata = m_cSession.Metadata
Set cTreeInfo = cMetadata.Entities
    lPar = cTreeInfo.GetItemID("California")
    iSuccess = cTreeInfo.EnumIDsOfChildren(TREE_ROOT, lPar, _
    vaChildIDs)
If iSuccess = 0 Then
    For i = LBound(vaChildIDs) To UBound(vaChildIDs)
        cTreeInfo.GetLabel CLng(vaChildIDs(i)), sLabel
        Debug.Print sLabel
    Next i
End If
```

The entity dimension's ID is the last item in the array.

EnumMemberLists

Gets the names of the member lists for a dimension.

Syntax

```
<IHsvTreeInfo>.EnumMemberLists pvarabstrListNames
```

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

<i>pvarabstrListNames</i>	Variant array. Returns the names of the dimension's member lists. The array is returned as a String subtype.
---------------------------	--

Example

This example puts the labels of the Period dimension's member lists into a combo box control. The `Dimension` property sets an `IHsvTreeInfo` object reference for the Period dimension, and the member list names returned by `EnumMemberLists` are placed in a combo box named `comboLists`.

```
Dim cIHsvTreeInfo As IHsvTreeInfo, vaLists
Set cIHsvTreeInfo = m_cMetadata.Dimension(DIMENSIONPERIOD)
cIHsvTreeInfo.EnumMemberLists vaLists
For i = LBound(vaLists) To UBound(vaLists)
    comboLists.AddItem vaLists(i)
Next i
```

EnumMembers

Returns an array containing the member IDs of the dimension members in either the default dimension hierarchy or in a member list. The value passed to the *lListID* argument determines whether the returned IDs are from the default hierarchy or from a list.

Tip: To return members of a dynamic member list, use [EnumMembers2](#).

For the Entity dimension, `EnumMembers` returns child entity member IDs in the `pvarallItemIDs` argument and the corresponding parent member IDs in the `pvaralParentIDs` argument. For the other dimensions, `EnumMembers` returns members in the `pvarallItemIDs` argument, and the `pvaralParentIDs` argument is left uninitialized.

Tip: If you return members from the default dimension hierarchy, you can also use the `lListTopMemberID` argument to return only members of a given node.

Syntax

```
<IHsvTreeInfo>.EnumMembers lListID, lListTopMemberID, pvarallItemIDs,  
pvaralParentIDs
```

Argument	Description
<i>lListID</i>	Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the <code>HFMConstants</code> type library constant <code>MEMBER_LIST_ALL_HIERARCHY</code> to return the members in the default hierarchy, or a valid list ID to return the members in a member list. Tip: You can get member list IDs with <code>GetMemberListID</code> . For more information, see “GetMemberListID” on page 221 .
<i>lListTopMemberID</i>	Long (ByVal). The usage of this argument depends on what you pass to the <i>lListID</i> argument: <ul style="list-style-type: none">● If you pass <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, you can return either member IDs for all of the dimension’s members by passing the <code>TREE_ROOT</code> constant, or member IDs for a node’s members by passing the member ID of the node’s parent.● If you pass anything other than <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, the <i>lListTopMemberID</i> argument is ignored. Because this argument is not optional, you must still pass a valid Long.
<i>pvarallItemIDs</i>	Variant array. For the Entity dimension, this argument returns the member IDs of the child entities. For the other dimensions, this argument returns the member IDs of the dimension members. The array is returned as a Long subtype.
<i>pvaralParentIDs</i>	Variant array. For the Entity dimension, this argument returns the member IDs of the parents of the members returned in the <i>pvarallItemIDs</i> argument. The array is returned as a Long subtype. For the other dimensions, an empty Variant is returned.

Example

The following example prints out a node of the Entity dimension to the Immediate window. The node’s parent member is Europe; note how `MEMBER_LIST_ALL_HIERARCHY` is passed as the *lListID* argument and how Europe’s member ID is passed as the *lListTopMemberID* argument.

```
Dim cTreeInfo As IHsvTreeInfo  
Dim vaChildIDs, vaParIDs, lEnt As Long  
Dim sLabel As String, sParLabel As String  
Set cTreeInfo = m_cMetadata.Entities
```

```

lEnt = cTreeInfo.GetItemID("Europe")
cTreeInfo.EnumMembers MEMBER_LIST_ALL_HIERARCHY, lEnt, _
vaChildIDs, vaParIDs
For i = LBound(vaChildIDs) To UBound(vaChildIDs)
    cTreeInfo.GetLabel CLng(vaParIDs(i)), sParLabel
    cTreeInfo.GetLabel CLng(vaChildIDs(i)), sLabel
    Debug.Print sParLabel & " " & sLabel
Next i

```

EnumMembers2

Returns an array containing the member IDs of the dimension members in either the default dimension hierarchy, a static member list, or a dynamic member list for given Scenario, Year, Period, and Entity dimension members. The value passed to the *lListID* argument determines whether the returned IDs are from the default hierarchy or from a member list.

For the Entity dimension, `EnumMembers2` returns child entity member IDs in the *pvaralItemIDs* argument and the corresponding parent member IDs in the *pvaralParentIDs* argument. For the other dimensions, `EnumMembers2` returns members in the *pvaralItemIDs* argument, and the *pvaralParentIDs* argument is left uninitialized.

Note: You can also return members from the default dimension hierarchy or from a static member list with `EnumMembers`, which does not take member IDs of Scenario, Year, and Period dimension members.

Syntax

```

<IHsvTreeInfo>.EnumMembers2 lListID, lListTopMemberID, lScenario, lYear,
lPeriod, lEntity, pvaralItemIDs, pvaralParentIDs

```

Argument	Description
<i>lListID</i>	<p>Long (ByVal). Identifies either a member list or the default dimension hierarchy. Pass a valid list ID to return the members in a member list, or the HFMConstants type library constant <code>MEMBER_LIST_ALL_HIERARCHY</code> to return the members in the default hierarchy.</p> <p>Tip: You can get member list IDs with GetMemberListID.</p>
<i>lListTopMemberID</i>	<p>Long (ByVal). The usage of this argument depends on what you pass to the <i>lListID</i> argument:</p> <ul style="list-style-type: none"> ● If you pass <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, you can return either member IDs for all of the dimension's members by passing the <code>TREE_ROOT</code> constant, or member IDs for a node's members by passing the member ID of the node's parent. ● If you pass anything other than <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, the <i>lListTopMemberID</i> argument is ignored. Because this argument is not optional, you must still pass a valid Long.
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member for a dynamic member list.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member for a dynamic member list.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member for a dynamic member list.

Argument	Description
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member for a dynamic member list. Note: This argument was added to <code>EnumMembers2</code> in Release 4.0. If you have code that requires <code>EnumMembers2</code> to function as it did prior to Release 4.0, pass <code>-1</code> .
<i>pvarAllItemIDs</i>	Variant array. For the Entity dimension, this argument returns the member IDs of the child entities. For the other dimensions, this argument returns the member IDs of the dimension members. The array is returned as a Long subtype.
<i>pvarAllParentIDs</i>	Variant array. For the Entity dimension, this argument returns the member IDs of the parents of the members returned in the <i>pvarAllItemIDs</i> argument. The array is returned as a Long subtype. For the other dimensions, an empty Variant is returned.

Example

The following function returns an array containing the members in an Entity dimension dynamic member list for a given scenario, year, and period. In the array, the parent and child entities are delimited by periods.

```
Function getDynamicList(lScen As Long, lYear As Long, _
    lPer As Long) As Variant
    Dim cIHsvTreeInfo As IHsvTreeInfo, lListID As Long
    Dim vaChildIDs, vaParIDs, vaRet()
    Dim sLabel As String, sParLabel As String
    'm_cHsvMetadata is an HsvMetadata object
    Set cIHsvTreeInfo = g_cMetadata.Entities
    cIHsvTreeInfo.GetMemberListID "Dynamic", lListID
    cIHsvTreeInfo.EnumMembers2 lListID, -1, lScen, lYear, lPer, _
        MEMBERNOTUSED, vaChildIDs, vaParIDs
    ReDim vaRet(UBound(vaChildIDs))
    For i = LBound(vaChildIDs) To UBound(vaChildIDs)
        cIHsvTreeInfo.GetLabel CLng(vaParIDs(i)), sParLabel
        cIHsvTreeInfo.GetLabel CLng(vaChildIDs(i)), sLabel
        vaRet(i) = sParLabel & "." & sLabel
    Next i
    getDynamicList = vaRet
End Function
```

EnumMembersWithAttribValue

Enumerates the attribute IDs and corresponding labels for a given attribute. For example, a user can enumerate all of the entities with `ATTRIB_ENTITY_IS_ICP = true`.

Syntax

```
<HsvTreeInfo>.EnumMembersWithAttribValue attribId, varAttribValue, flags,
pvaralIDs, pvarabstrLabels
```

Argument	Description
<i>attribId</i>	Integer (ByVal). Attribute ID of member.

Argument	Description
<i>varAttribValue</i>	Variant (ByVal). The raw value of the attribute. To obtain an attribute's raw value, use GetApplicationAttribute .
<i>flags</i>	Long (ByVal). Used to refine the search. Example: ATTRFILTER_FLAG_INCLUDE_SYS_MEMBERS- returns members that include system members If 0 is passed, system members will not be returned although they match the searched value <i>varAttribValue</i> . ATTRFILTER_FLAG_CASE_SENSITIVE- performs case-sensitive search of <i>varAttributeValue</i> .
<i>pvaralIDs</i>	Variant array. Returns the array of the member IDs. The array is returned as a Long subtype.
<i>pvarabstrLabels</i>	Variant array. Returns an array of member labels. The array is returned as a String subtype

EnumParents

Returns an array containing the member IDs of a given member's parent members.

Syntax

```
<IHsvTreeInfo>.EnumParents lItemID, pvaralIDs
```

Argument Description

<i>lItemID</i>	Long (ByVal). The member ID of the member.
<i>pvaralIDs</i>	Variant array. Returns the array of the parents' member IDs. The array is returned as a Long subtype.

Example

The following example prints the labels of the California entity's parent members to the Immediate window. The example loops through the array returned by `EnumParents`, passing the member IDs to `GetLabel`.

```
Dim lMember As Long, cTreeInfo As IHsvTreeInfo, vaIDs
Dim sLabel As String
Set cTreeInfo = m_cMetadata.Entities
lMember = cTreeInfo.GetItemID("California")
cTreeInfo.EnumParents lMember, vaIDs
For i = LBound(vaIDs) To UBound(vaIDs)
    cTreeInfo.GetLabel vaIDs(i), sLabel
    Debug.Print sLabel
Next i
```

EnumSortedIDsOfChildren

Returns an array containing the member IDs an Intercompany Partner dimension member's children, with the members sorted according to the specified sorting criteria.

Note: For dimensions other than Intercompany Partner, `EnumSortedIDsOfChildren` behaves like `EnumIDsOfChildren` in that it returns children but does not sort them.

Syntax

```
<IHsvTreeInfo>.EnumSortedIDsOfChildren(lListTopMemberID, lItemID,  
lSortOptions, pvarChildIDArray)
```

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

<i>lListTopMemberID</i>	Long (ByVal). Pass the <code>HFMConstants</code> type library constant <code>TREE_ROOT</code> to this argument.
-------------------------	---

<i>lItemID</i>	Long (ByVal). The member ID of the member for which to return children.
----------------	---

<i>lSortOptions</i>	Long (ByVal). The member information by which to sort. You can pass any combination of the <code>HFMConstants</code> type library constants that are listed in “ Metadata Information Constants ” on page 867 and that include the word “SORT.”
---------------------	---

Note: By default, `EnumSortedMembers` sorts in ascending order, so there is no constant for ascending order.

You can specify multiple sorting options by using the `Or` operator with these constants.

<i>pvarChildIDArray</i>	Variant. Returns an array of the children’s member IDs, with the array sorted as specified by the <i>lSortOptions</i> argument.
-------------------------	---

Return Value

Integer. Indicates whether the call succeeded. Returns 0 upon success, or a non-zero error number upon failure.

EnumSortedIDsOfChildren

Returns sorted children and the parent item of the specified children.

Syntax

```
<HsvTreeInfo>.EnumSortedIDsOfChildren(lListTopMemberID, lItemID,  
lSortOptions, pvarChildIDArray)
```

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

<i>lListTopMemberID</i>	Long (ByVal). Pass the <code>HFMConstants</code> type library constant <code>TREE_ROOT</code> to this argument.
-------------------------	---

<i>lItemID</i>	Long (ByVal). The member ID of the member for which to return children.
----------------	---

<i>lSortOptions</i>	Long (ByVal). The member information by which to sort. You can pass any combination of the <code>HFMConstants</code> type library constants that are listed in “ Metadata Information Constants ” on page 867 and that include the word “SORT.”
---------------------	---

Note: By default, `EnumSortedMembers` sorts in ascending order, so there is no constant for ascending order.

You can specify multiple sorting options by using the `Or` operator with these constants.

Argument	Description
<i>pvarChildIDArray</i>	Variant. Returns an array of the children's member IDs, and the parent itself, with the array sorted as specified by the <i>lSortOptions</i> argument.

Return Value

Integer. Indicates whether the call succeeded. Returns 0 upon success, or a non-zero error number upon failure.

EnumSortedMembers

Returns an array containing the member IDs of the Intercompany Partner dimension members in a member list, with the members sorted according to the specified sorting criteria.

Note: For dimensions other than Intercompany Partner, `EnumSortedMembers` behaves like `EnumMembers` in that it returns members on the specified member list but does not sort them.

Syntax

```
<IHsvTreeInfo>.EnumSortedMembers lListID, lListTopMemberID, lSortOptions,
pvaralItemIDs, pvaralParentIDs
```

Argument	Description
<i>lListID</i>	Long (ByVal). The ID of the member list. Tip: You can get member list IDs with GetMemberListID .
<i>lListTopMemberID</i>	Long (ByVal). Specifies whether to return all members on the list or only a given member and its descendants. Valid values are described below: <ul style="list-style-type: none"> ● To return all members, pass the HFMCconstants type library constant <code>MEMBERNOTUSED</code>. ● To return only a given member and its descendants, pass the member ID of the member.
<i>lSortOptions</i>	Long (ByVal). The member information by which to sort. You can pass any combination of the HFMCconstants type library constants that are listed in “Metadata Information Constants” on page 867 and that include the word “SORT.” Note: By default, <code>EnumSortedMembers</code> sorts in ascending order, so there is no constant for ascending order. You can specify multiple sorting options by using the <code>Or</code> operator with these constants.
<i>pvaralItemIDs</i>	Variant. Returns an array of member IDs that are sorted as specified by the <i>lSortOptions</i> argument.
<i>pvaralParentIDs</i>	Variant. Returns an array of member IDs of the parents of the members returned by the <i>pvaralItemIDs</i> argument.

Example

The following example prints to Visual Basic's Immediate window the labels of the [ICP Entities] Intercompany Partner member and its descendants on the [Hierarchy] member list. The labels are sorted in ascending order.

```
Dim cTreeInfo As IHsvTreeInfo, lTopId As Long
Dim lListId As Long, sLabel As String, vaChildren As Variant
Dim vaParents As Variant
'g_cMetadata is a previously set HsvMetadata object reference
Set cTreeInfo = g_cMetadata.ICPs
lTopId = cTreeInfo.GetItemID("[ICP Entities]")
cTreeInfo.GetMemberListID "[Hierarchy]", lListId
cTreeInfo.EnumSortedMembers lListId, lTopId,
WEBOM_METADATA_INFO_SORTBY_LABEL, _
    vaChildren, vaParents
For i = LBound(vaChildren) To UBound(vaChildren)
    cTreeInfo.GetLabel vaChildren(i), sLabel
    Debug.Print sLabel
Next i
```

Find

Returns the member IDs of those members of a member list with labels that match a search string. Each time a member matching the search criteria is found, the member ID of the member and the member ID of its parent are returned.

The *pbFoundMatch* argument returns a Boolean that indicates the success of the find operation, thus allowing you to loop until all occurrences are found, as shown in the example.

Syntax

```
<IHsvTreeInfo>.Find lListID, lTopMemberID, bstrSearchText, bForward, plPos,
plMemberID, plParentID, pbFoundMatch
```

Argument	Description
<i>lListID</i>	Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the HFMConstants type library constant <code>MEMBER_LIST_ALL_HIERARCHY</code> to search the default hierarchy, or a valid list ID to search a member list. You can get member list IDs with GetMemberListID .
<i>lTopMemberID</i>	Long (ByVal). The usage of this argument depends on what you pass to the <i>lListID</i> argument: <ul style="list-style-type: none">● If you pass <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, you can either search the entire dimension hierarchy by passing the HFMConstants type library constant <code>TREE_ROOT</code>, or search a node's members by passing the member ID of the node's parent.● If you pass anything other than <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, the <i>lListTopMemberID</i> argument is ignored. Note that since this argument is not optional, you must still pass a valid Long.
<i>bstrSearchText</i>	String (ByVal). The search string. Pass all or part of the labels of the members for which the search is being conducted. If you pass a partial label, <code>Find</code> will search for all members that <i>begin</i> with the passed string.

Argument	Description
<i>bForward</i>	Boolean (ByVal). Indicates whether the search will go forwards or backwards in the dimension hierarchy or member list. Pass TRUE to search forwards, FALSE to search backwards.
<i>pIPos</i>	Long. Sets the starting position of the search within the dimension hierarchy or member list, and returns a number that should be passed to the next call to <code>Find</code> . Use this argument as follows: <ol style="list-style-type: none"> To start a search for a string, pass -1. For each subsequent search for the string, pass the number returned in the previous call to <code>Find</code> (as shown in the example).
<i>pIMemberID</i>	Long. If a member label is found that matches the search string, this argument returns the member ID of the member.
<i>pIParentID</i>	Long. If a member label is found that matches the search string, this argument returns the member ID of the member's parent.
<i>pbFoundMatch</i>	Boolean. Returns TRUE if a member label is found that matches the search string, otherwise FALSE.

Example

The following subroutine prints to Visual Basic's Immediate window the labels of all members of a given dimension and member list that partially match the specified string. The example also prints labels of parent members.

```

Sub printListMatch(iDim As Integer, lListId As Long, _
    sDesc As String)
Dim lPos As Long, cTreeInfo As IHsvTreeInfo, lMem As Long
Dim lPar As Long, bRet As Boolean, sMemLabel As String
Dim sParLabel As String
'g_cMetadata is an HsvMetadata object reference
Set cTreeInfo = g_cMetadata.Dimension(iDim)
lPos = -1
bRet = True
Do
    cTreeInfo.Find lListId, TREE_ROOT, sDesc, True, lPos, _
        lMem, lPar, bRet
    If bRet = True Then
        cTreeInfo.GetLabel lMem, sMemLabel
        cTreeInfo.GetLabel lPar, sParLabel
        Debug.Print sMemLabel & " " & sParLabel
    End If
Loop Until bRet = False
End Sub

```

FindByDesc

Returns the member IDs of those members of a member list with descriptions that match a search string in a given language. Each time a member matching the search criteria is found, the member ID of the member and the member ID of its parent are returned.

The *pbFoundMatch* argument returns a Boolean that indicates the success of the find operation, thus allowing you to loop until all occurrences are found, as shown in the example.

Syntax

```
<IHsvTreeInfo>.FindByDesc lListID, lTopMemberID, bstrSearchText, bForward,  
lLanguageID, plPos, plMemberID, plParentID, pbFoundMatch
```

Argument	Description
<i>lListID</i>	Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the HFMConstants type library constant <code>MEMBER_LIST_ALL_HIERARCHY</code> to search the default hierarchy, or a valid list ID to search a member list. You can get member list IDs with GetMemberListID .
<i>lTopMemberID</i>	Long (ByVal). The usage of this argument depends on what you pass to the <i>lListID</i> argument: <ul style="list-style-type: none">• If you pass <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, you can either search the entire dimension hierarchy by passing the HFMConstants type library constant <code>TREE_ROOT</code>, or search a node's members by passing the member ID of the node's parent.• If you pass anything other than <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, the <i>lListTopMemberID</i> argument is ignored. Note that since this argument is not optional, you must still pass a valid Long.
<i>bstrSearchText</i>	String (ByVal). The search string. Pass all or part of the descriptions for which the search is being conducted. If you pass a partial description, the method searches for all descriptions that <i>begin</i> with the passed string.
<i>bForward</i>	Boolean (ByVal). Indicates whether the search will go forwards or backwards in the dimension hierarchy or member list. Pass <code>TRUE</code> to search forwards, <code>FALSE</code> to search backwards.
<i>lLanguageID</i>	Long (ByVal). The ID of the language. You can get this ID with the EnumLanguages method of the HsvMetadata object.
<i>plPos</i>	Long. Sets the starting position of the search within the dimension hierarchy or member list, and returns a number that should be passed to the next call to <code>Find</code> . Use this argument as follows: <ol style="list-style-type: none">1. To start a search for a string, pass -1.2. For each subsequent search for the string, pass the number returned in the previous call to <code>FindByDesc</code> (as shown in the example).
<i>plMemberID</i>	Long. If a description is found that matches the search string, this argument returns the member ID of the member.
<i>plParentID</i>	Long. If a description is found that matches the search string, this argument returns the member ID of the member's parent.
<i>pbFoundMatch</i>	Boolean. Returns <code>TRUE</code> if a member label is found that matches the search string, otherwise <code>FALSE</code> .

Example

The following subroutine prints to Visual Basic's Immediate window the labels of all members of a given dimension and member list that have descriptions partially matching the specified string. The example also prints labels of parent members.

```
Sub printListDescMatch(iDim As Integer, lListId As Long, _  
    sDesc As String, lLangId As Long)  
    Dim lPos As Long, cTreeInfo As IHsvTreeInfo, lMem As Long  
    Dim lPar As Long, bRet As Boolean, sMemLabel As String
```

```

Dim sParLabel As String
'g_cMetadata is an HsvMetadata object reference
Set cTreeInfo = g_cMetadata.Dimension(iDim)
lPos = -1
bRet = True
Do
    cTreeInfo.FindByDesc lListId, TREE_ROOT, sDesc, True, lLangId, _
        lPos, lMem, lPar, bRet
    If bRet = True Then
        cTreeInfo.GetLabel lMem, sMemLabel
        cTreeInfo.GetLabel lPar, sParLabel
        Debug.Print sMemLabel & " " & sParLabel
    End If
Loop Until bRet = False
End Sub

```

FindMatchingMembersFromHierarchy

Returns the member IDs of the members with labels that match a search string.

Syntax

```

<IHsvTreeInfo>.FindMatchingMembersFromHierarchy bstrSearchText,
lTopMemberID, varbExactMatch, pvaravaralPaths

```

Argument	Description
<i>bstrSearchText</i>	String (ByVal). The search string. Pass all or part of the labels of the members for which the search is being conducted. If you pass a partial label, the method searches for all members that <i>begin</i> with the passed string.
<i>lTopMemberID</i>	Long (ByVal). The member ID of the top member in the hierarchy from which to begin searching. To search the entire dimension hierarchy, use the HFMConstants type library constant <code>TREE_ROOT</code> .
<i>varbExactMatch</i>	Boolean (ByVal). Specifies whether to search for an exact or partial match. Pass <code>TRUE</code> for an exact match search, <code>FALSE</code> for a partial match search.
<i>pvaravaralPaths</i>	VARIANT. Returns an array of arrays containing the member IDs of the matching members. Each array contains one item for each member in the path from the specified top member to the matching member. For example, suppose that a search returns two members: Canada (with no parent) and UnitedStates.California. The indexes to the array of arrays would be similar to those in the following list: <ul style="list-style-type: none"> ● (0) (0) = UnitedStates ● (0) (1) = California ● (1) (0) = Canada

Example

The following subroutine prints to Visual Basic's Immediate window the labels of all members of a given dimension with labels that partially match the specified string.

```

Sub printPartMatchingLabels(iDim As Integer, sSearch As String)
Dim cTreeInfo As IHsvTreeInfo, vaMems, sLabel As String

```

```

'Set the IHsvTreeInfo interface to the specified dimension.
'g_cMetadata is an HsvMetadata object reference.
Set cTreeInfo = g_cMetadata.Dimension(iDim)
cTreeInfo.FindMatchingMembersFromHierarchy sSearch, TREE_ROOT, False, _
    vaMems
'Loop through the array of arrays
For i = LBound(vaMems) To UBound(vaMems)
    'Loop through the items in each array
    For j = LBound(vaMems(i)) To UBound(vaMems(i))
        cTreeInfo.GetLabel vaMems(i)(j), sLabel
        Debug.Print sLabel
    Next j
    Debug.Print vbCrLf
Next i
End Sub

```

FindMatchingMembersFromHierarchyByDesc

Returns the member IDs of the members with descriptions in a given language that match a search string.

Syntax

```

<IHsvTreeInfo>.FindMatchingMembersFromHierarchyByDesc bstrSearchText,
lTopMemberID, varbExactMatch, lLanguageID, pvaravaralPaths

```

Argument	Description
<i>bstrSearchText</i>	String (ByVal). The search string. Pass all or part of the descriptions for which the search is being conducted. If you pass a partial description, the method searches for all descriptions that <i>begin</i> with the passed string.
<i>lTopMemberID</i>	Long (ByVal). The member ID of the top member in the hierarchy from which to begin searching. To search the entire dimension hierarchy, use the HFMCConstants type library constant <code>TREE_ROOT</code> .
<i>varbExactMatch</i>	Boolean (ByVal). Specifies whether to search for an exact or partial match. Pass <code>TRUE</code> for an exact match search, <code>FALSE</code> for a partial match search.
<i>lLanguageID</i>	Long (ByVal). The ID of the language. You can get this ID with the EnumLanguages method of the HsvMetadata object.
<i>pvaravaralPaths</i>	VARIANT. Returns an array of arrays containing the member IDs of the matching members. Each array contains one item for each member in the path from the specified top member to the matching member. For example, suppose that a search returns two members: Canada (with no parent) and UnitedStates.California. The indexes to the array of arrays would be similar to those in the following list: <ul style="list-style-type: none"> ● (0) (0) = UnitedStates ● (0) (1) = California ● (1) (0) = Canada

Example

The following subroutine prints to Visual Basic's Immediate window the labels of all members of a given dimension with descriptions that partially match the specified string.

```
Sub printMatchingDescs(iDim As Integer, langId As Long, _
    sSearch As String)
Dim cTreeInfo As IHsvTreeInfo, vaMems, sLabel As String
'Set the IHsvTreeInfo interface to the specified dimension.
'g_cMetadata is an HsvMetadata object reference.
Set cTreeInfo = g_cMetadata.Dimension(iDim)
cTreeInfo.FindMatchingMembersFromHierarchyByDesc sSearch, TREE_ROOT, _
    False, langId, vaMems
'Loop through the array of arrays
For i = LBound(vaMems) To UBound(vaMems)
    'Loop through the items in each array
    For j = LBound(vaMems(i)) To UBound(vaMems(i))
        cTreeInfo.GetLabel vaMems(i)(j), sLabel
        Debug.Print sLabel
    Next j
    Debug.Print vbCrLf
Next i
End Sub
```

FindMatchingMembersFromHierarchyWildcard

Returns the member IDs of the members with labels or descriptions that match a search string; the search string can include wildcard characters. Description searches are for descriptions in a given language.

Syntax

```
<IHsvTreeInfo>.FindMatchingMembersFromHierarchyWildcard bstrSearchText,
lTopMemberID, lLanguageID, pvaravaralPaths
```

Argument	Description
<i>bstrSearchText</i>	String (ByVal). String (ByVal). The search string. You can use asterisks (*) as wildcard characters. The following list describes the rules for wildcard searching: <ul style="list-style-type: none">● You can use a wildcard at the beginning of the search string.● You can use a wildcard at the end of the search string.● You can use wildcards at both the beginning and end of the search string. <p>Note: Asterisks placed anywhere other than the beginning or end are treated as literal characters, not as wildcards.</p> <ul style="list-style-type: none">● You can omit wildcards to perform an exact match search.
<i>lTopMemberID</i>	Long (ByVal). The member ID of the top member in the hierarchy from which to begin searching. To search the entire dimension hierarchy, use the HFMConstants type library constant <code>TREE_ROOT</code> .
<i>lLanguageID</i>	Long (ByVal). Specifies whether to search labels or descriptions. For description searches, this argument also specifies the language of the descriptions to be searched. Pass one of the following values: <ul style="list-style-type: none">● To search for labels, pass the HFMConstants type library constant <code>HFM_NO_LANGUAGE</code>.

Argument	Description
	<ul style="list-style-type: none"> To search for descriptions, pass the ID of the language in which to search. You can get this ID with the EnumLanguages method of the HsvMetadata object.
<i>varavaralPaths</i>	<p>Variant. Returns an array of arrays containing the member IDs of the matching members. Each array contains one item for each member in the path from the specified top member to the matching member.</p> <p>For example, suppose that a search returns two members: Canada (with no parent) and UnitedStates.California. The indexes to the array of arrays would be similar to those in the following list:</p> <ul style="list-style-type: none"> (0) (0) = UnitedStates (0) (1) = California (1) (0) = Canada

Example

The following subroutine prints to Visual Basic's Immediate window the labels of all members of a given dimension with labels that match the specified string.

```
Sub printWildMatchingLabels(iDim As Integer, sSearch As String)
Dim cTreeInfo As IHsvTreeInfo, vaMems, sLabel As String
'Set the IHsvTreeInfo interface to the specified dimension.
'g_cMetadata is an HsvMetadata object reference.
Set cTreeInfo = g_cMetadata.Dimension(iDim)
cTreeInfo.FindMatchingMembersFromHierarchyWildcard sSearch, TREE_ROOT, _
    HFM_NO_LANGUAGE, vaMems
'Loop through the array of arrays
For i = LBound(vaMems) To UBound(vaMems)
    'Loop through the items in each array
    For j = LBound(vaMems(i)) To UBound(vaMems(i))
        cTreeInfo.GetLabel vaMems(i)(j), sLabel
        Debug.Print sLabel
    Next j
    Debug.Print vbCrLf
Next i
End Sub
```

GetAllPathsToMember

Returns an array of strings that represent the possible paths in a dimension's hierarchy to a given member.

Syntax

```
<IHsvTreeInfo>.GetAllPathsToMember lMemberID, varabstrPaths
```

Argument	Description
<i>lMemberID</i>	Long (ByVal). The member ID of the member for which you want to return the hierarchical paths.
<i>varabstrPaths</i>	Variant. Returns an array of strings that represent the hierarchical paths to the member. The array contains an item for each path to the member.

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

The members in these strings are delimited by backslashes (\), as shown in the following examples for a member named "Connecticut":

```
\Regional\UnitedStates\Connecticut
```

```
\Management\Imbler\Connecticut
```

Example

The following function takes an entity's name and returns the possible paths to the entity.

```
Function getEntityHierarchy(sEntity As String) As Variant
Dim cTreeInfo As IHsvTreeInfo, vaPaths As Variant, lId As Long
Set cTreeInfo = m_cMetadata.Entities
lId = cTreeInfo.GetItemID(sEntity)
cTreeInfo.GetAllPathsToMember lId, vaPaths
getEntityHierarchy = vaPaths
End Function
```

GetAttributeValue

Returns the value of a given member's metadata attribute.

Syntax

```
<IHsvTreeInfo>.GetAttributeValue lItemID, iAttribute, vValue
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the dimension member.
----------------	--

<i>iAttribute</i>	Integer (ByVal). The metadata attribute for which to return the value. Valid values are represented by the HFMConstants type library constants listed in the following topics:
-------------------	--

- ["Scenario Attribute Constants" on page 837](#)
- ["Entity Attribute Constants" on page 831](#)
- ["Account Attribute Constants" on page 824](#)
- ["Custom Dimension Attributes" on page 828](#)
- ["Intercompany Partner Attribute Constants" on page 836](#)

<i>vValue</i>	Variant. Returns the attribute's value.
---------------	---

Example

The following function returns the UserDefined1 attribute of a given Scenario dimension member.

Note: You can also get this attribute with `HsvScenarios.GetUserDefined1`.

```
Function getScenUserDef1(lId As Long) As Variant
Dim cIHsvTreeInfo As IHsvTreeInfo, vRet As Variant
```

```

Set cIHsvTreeInfo = m_cMetadata.Scenarios
cIHsvTreeInfo.GetAttributeValue lId, ATTRIB_SCENARIO_USERDEF1, _
    vRet
getScenUserDef1 = vRet
End Function

```

GetDefaultHierarchyPosition

Returns the default position of a member within a dimension's hierarchy.

Note: A dimension's hierarchy is represented by the system-generated [Hierarchy] member list.

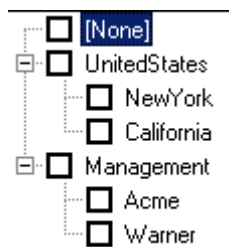
Syntax

```
<IHsvTreeInfo>.GetDefaultHierarchyPosition lMemberID, plPos
```

Argument Description

lMemberID Long (ByVal). The member ID of the member.

plPos Long. Returns the position of the member in the fully expanded hierarchy. This is a zero-based value. For example, in the following hierarchy, California would return 3 and Management would return 4.



For a member that appears more than once in the hierarchy, `GetDefaultHierarchyPosition` returns the position where the member is located under its default parent.

Tip: To get a member's default parent, you can use [GetDefaultParent](#) or [GetDefaultParentLabel](#).

GetDefaultItemIDHierarchy

Returns an array containing the path of a dimension member and its ancestors; the array contains the member IDs of the members in this path. You can return the entire path or just a portion of the path.

For example, suppose an Entity dimension member named SanFrancisco has ancestors named California, UnitedStates, and Regional, respectively. You could use `GetDefaultItemIDHierarchy` to return the member IDs of the entire path, or to return the member IDs of only a portion of the path, such as the portion from UnitedStates down.

Syntax

```
<IHsvTreeInfo>.GetDefaultItemIDHierarchy(lListID, lListTopMemberID,  
lItemID, lParentID, pvarItemIDHierarchy)
```

Argument	Description
<i>lListID</i>	Long (ByVal). Pass the HFMCConstants type library constant <code>MEMBER_LIST_ALL_HIERARCHY</code> to use the default dimension hierarchy. Caution! If you pass a value other than <code>MEMBER_LIST_ALL_HIERARCHY</code> , the <i>lListTopMemberID</i> and <i>lParentID</i> arguments are ignored, and only the member ID passed in the <i>lItemID</i> argument is returned.
<i>lListTopMemberID</i>	Long (ByVal). The value you pass depends upon whether you want to return the entire path of ancestors or only a portion of the path: <ul style="list-style-type: none">● To return the entire path of ancestors, pass the HFMCConstants type library constant <code>TREE_ROOT</code>.● To return a portion of the path, pass the member ID of the topmost member that you want to return.
<i>lItemID</i>	Long (ByVal). The member ID of the member for which you want to return the path.
<i>lParentID</i>	Long (ByVal). The member ID of the parent of the <i>lItemID</i> member. Tip: For Entity dimension members with more than one parent, pass the HFMCConstants type library constant <code>MEMBERNOTUSED</code> to have Financial Management automatically use the entity's first parent.
<i>pvarItemIDHierarchy</i>	Variant array. Returns the member IDs for the path of ancestors. Note that the <i>lItemID</i> argument's member ID is included at the bottom of the path. The array is returned as a Long subtype.

Return Value

Integer. Indicates the success of the function call; returns 0 for success or -1 for an error.

Example

The following example returns the member IDs of the SanFrancisco entity's ancestors, filtering out all members in the hierarchy above UnitedStates.

```
Dim cTreeIn As IHsvTreeInfo, lEnt As Long  
Dim lTop As Long, lPar As Long, vaIDs, sLabel As String  
Dim iRet As Integer  
Set cTreeIn = m_cMetadata.Entities  
lTop = cTreeIn.GetItemID("UnitedStates")  
lEnt = cTreeIn.GetItemID("SanFrancisco")  
lPar = cTreeIn.GetItemID("California")  
cTreeIn.GetDefaultItemIDHierarchy MEMBER_LIST_ALL_HIERARCHY, _  
lTop, lEnt, lPar, vaIDs  
For i = LBound(vaIDs) To UBound(vaIDs)  
    cTreeIn.GetLabel vaIDs(i), sLabel  
    Debug.Print sLabel  
Next i
```

GetDefaultMemberID

Returns the member ID of a dimension's default member.

Syntax

```
<IHsvTreeInfo>.GetDefaultMemberID plDefaultMemberID
```

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

<i>plDefaultMemberID</i>	Long. Returns the member ID of the default member.
--------------------------	--

Example

The following function returns the label of a given dimension's default member. The HsvMetadata's [Dimension](#) property sets the IHsvTreeInfo object reference for the specified dimension and [GetLabel](#) obtains the label for the member ID returned by `GetDefaultMemberID`.

```
Function getDefaultMemLabel(lDim As Long) As String
Dim cTreeInfo As IHsvTreeInfo, lId As Long, sRet As String
'g_cMetadata is an HsvMetadata object reference
Set cTreeInfo = g_cMetadata.Dimension(lDim)
cTreeInfo.GetDefaultMemberID lId
cTreeInfo.GetLabel lId, sRet
getDefaultMemLabel = sRet
End Function
```

GetDefaultParent

Returns the member ID of a given dimension member's default parent.

Tip: You can get the label of a member's default parent with [GetDefaultParentLabel](#).

Syntax

```
<IHsvTreeInfo>.GetDefaultParent lMemberID, plParentID
```

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

<i>lMemberID</i>	Long (ByVal). The member ID of the dimension member.
------------------	--

<i>plParentID</i>	Long. Returns the member ID of the default parent. If the <i>lMemberID</i> argument identifies a top-level member of the dimension's hierarchy, -1 is returned.
-------------------	---

GetDefaultParentLabel

Returns the label of a given dimension member's default parent.

Tip: You can get the ID of a member's default parent with [GetDefaultParent](#).

Syntax

```
<IHsvTreeInfo>.GetDefaultParentLabel lMemberID, pbstrParentLabel
```

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

<i>lMemberID</i>	Long (ByVal). The member ID of the dimension member.
------------------	--

<i>pbstrParentLabel</i>	String. Returns the label of the default parent. If the <i>lMemberID</i> argument identifies a top-level member of the dimension's hierarchy, the string "#root" is returned.
-------------------------	---

Example

The following function returns the label of a given Entity dimension member's default parent.

```
Function getEntityDefParent(sMemberName As String) As String
Dim cTreeInfo As IHsvTreeInfo, lMemID As Long
Dim sParLabel As String
'm_cMetadata is an HsvMetadata object reference
Set cTreeInfo = m_cMetadata.Entities
lMemID = cTreeInfo.GetItemID(sMemberName)
cTreeInfo.GetDefaultParentLabel lMemID, sParLabel
getEntityDefParent = sParLabel
End Function
```

GetDescription

Gets the description of a dimension member. Since Financial Management supports multi-language descriptions of members, `GetDescription` takes an argument that specifies the language in which you want the description.

Syntax

```
<IHsvTreeInfo>.GetDescription(lItemID, lLanguageID, pbstrDesc)
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the member.
----------------	--

<i>lLanguageID</i>	Long (ByVal). The ID number of the language. You can get this ID with the EnumLanguages method of the <code>HsvMetadata</code> object.
--------------------	--

<i>pbstrDesc</i>	String. Returns the description of the member.
------------------	--

Return Value

Integer. Indicates the success of the function call; returns 0 for success or -1 for an error.

Example

This example places the French description of an entity into a text box. The example assigns the label of the entity specified in the `ComboEnts` combo box to the `sEnt` variable. The example

then uses `EnumLanguages` to find the language ID of the French language, and `GetItemID` to get the member ID of the entity assigned to `sEnt`. `GetDescription` is then called for this member, and the description is placed in the `txtEntDesc` text box.

```
Dim cIHsvTreeInfo as IHsvTreeInfo, lMemID As Long
Dim sEnt As String, lLangID As Long, sDesc As String
Dim vaIDs, vaLabels, iLBounds As Integer, iLanguages As Integer
sEnt = comboEnts.Text
m_cMetadata.EnumLanguages vaIDs, vaLabels
iLBounds = LBound(vaLabels)
iLanguages = UBound(vaLabels)
Do Until iLBounds > iLanguages
    If vaLabels(iLBounds) = "French" Then
        lLangID = vaIDs(iLBounds)
    End If
    iLBounds = iLBounds + 1
Loop
Set cIHsvTreeInfo = m_cMetadata.Entities
lMemID = cIHsvTreeInfo.GetItemID(sEnt)
cIHsvTreeInfo.GetDescription lMemID, lLangID, sDesc
txtEntDesc.Text = sDesc
```

GetDisplayInfo

Returns the label and description of a dimension member. You can call `GetDisplayInfo` for a member in an application's default dimension hierarchy or a member in a member list. If you call `GetDisplayInfo` for the default dimension hierarchy, a Boolean indicating whether the member has child members is also returned.

Syntax

```
<IHsvTreeInfo>.GetDisplayInfo lListID, lItemID, lParentID, pbHasChildren,
pbstrLabel, pbstrDesc
```

Argument	Description
<i>lListID</i>	Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the <code>HFMConstants</code> type library constant <code>MEMBER_LIST_ALL_HIERARCHY</code> to return a member from the default hierarchy, or pass a valid list ID to return a member from a member list. Tip: You can get member list IDs with <code>GetMemberListID</code> . For more information, see “GetMemberListID” on page 221 .
<i>lItemID</i>	Long (ByVal). The member ID of the dimension member for which you want to return information.
<i>lParentID</i>	Long (ByVal). For hierarchical dimensions such as the Entity dimension, pass the member ID of the parent of the <i>lItemID</i> argument's dimension member. For non-hierarchical dimensions such as the Scenario dimension, pass the <code>HFMConstants</code> type library constant <code>MEMBERNOTUSED</code> .
<i>pbHasChildren</i>	Boolean. If you pass <code>MEMBER_LIST_ALL_HIERARCHY</code> to the <i>lListID</i> argument, <i>pbHasChildren</i> returns <code>TRUE</code> if the <i>lItemID</i> argument's member has children, otherwise <code>FALSE</code> .

Argument	Description
	If you pass anything other than <code>MEMBER_LIST_ALL_HIERARCHY</code> to the <i>lListID</i> argument, <i>pbHasChildren</i> always returns <code>FALSE</code> , regardless of whether the <i>lItemID</i> argument's member has children.
<i>pbstrLabel</i>	String. Returns the label of the <i>lItemID</i> argument's member. For hierarchical dimensions, the label of the <i>lParentID</i> argument's member is prefixed to the label of the <i>lItemID</i> argument's member, with a period delimiting the parent and child labels.
<i>pbstrDesc</i>	String. Returns the description of the <i>lItemID</i> argument's member. The description is returned in the default language.

Example

The following example prints the name and description of the California entity to the Immediate window.

```
Dim cTreeInfo As IHsvTreeInfo, lEnt As Long, lPar As Long
Dim bHasChildren As Boolean, sName As String, sDesc As String
Set cTreeInfo = m_cMetadata.Entities
lEnt = cTreeInfo.GetItemID("California")
lPar = cTreeInfo.GetItemID("UnitedStates")
cTreeInfo.GetDisplayInfo MEMBER_LIST_ALL_HIERARCHY, lEnt, _
lPar, bHasChildren, sName, sDesc
Debug.Print sName & " " & sDesc
```

GetDisplayInfoForSeveralItems

Returns arrays containing the labels and descriptions of dimension members. You can call `GetDisplayInfoForSeveralItems` for members in an application's default dimension hierarchy or members in a member list. If you call this method for the default dimension hierarchy, an array of Booleans indicating whether the members have child members is also returned.

The arrays have a one-to-one correspondence. For example, the third elements in the returned arrays contain the label and description of the member identified by the third element in the *varalItemIDs* argument's array.

Syntax

```
<IHsvTreeInfo>.GetDisplayInfoForSeveralItems lListID, varalItemIDs,
varalParentIDs, pvaravbHasChildren, pvarabstrLabels, pvarabstrDescs
```

Argument	Description
<i>lListID</i>	Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the <code>HFMConstants</code> type library constant <code>MEMBER_LIST_ALL_HIERARCHY</code> to return the members in the default hierarchy, or pass a valid list ID to return the members in a member list. Tip: You can get member list IDs with <code>GetMemberListID</code> . For more information, see “GetMemberListID” on page 221 .

Argument	Description
<i>varallItemIDs</i>	Variant array (ByVal). The member IDs of the dimension members for which you want to return information.
<i>varalParentIDs</i>	<p>Variant array (ByVal). For hierarchical dimensions such as the Entity dimension, pass the member IDs of the parents of the <i>varallItemIDs</i> argument's dimension members.</p> <p>For non-hierarchical dimensions such as the Scenario dimension, pass an array in which each element is set to the HFMConstants type library constant <code>MEMBERNOTUSED</code>.</p> <p>Caution! This array must contain the same number of elements as the array passed to the <i>varallItemIDs</i> argument.</p>
<i>pvaravbHasChildren</i>	<p>Variant array. If you pass <code>MEMBER_LIST_ALL_HIERARCHY</code> to the <i>lListID</i> argument, each array element returns <code>TRUE</code> if the corresponding <i>varallItemIDs</i> array element has children, otherwise <code>FALSE</code>.</p> <p>If you pass anything other than <code>MEMBER_LIST_ALL_HIERARCHY</code> to the <i>lListID</i> argument, each array element returns <code>FALSE</code>, regardless of whether the corresponding <i>varallItemIDs</i> array element has children.</p> <p>The array is returned as a Boolean subtype.</p>
<i>pvarabstrLabels</i>	<p>Variant array. Returns the labels of the members. For hierarchical dimensions, the labels of the <i>varalParentIDs</i> array's members are prefixed to the labels of the <i>varallItemIDs</i> array's member, with periods delimiting the parent and child labels.</p> <p>The array is returned as a String subtype.</p>
<i>pvarabstrDescs</i>	<p>Variant array. Returns the descriptions of the members. The descriptions are returned in the default language.</p> <p>The array is returned as a String subtype.</p>

Example

The following example prints the labels and descriptions of all of the Entity dimension's members to the Immediate window. `EnumAllParentAndChildIDs` gets the member IDs of the dimension's parent and child members, and these arrays are passed to `GetDisplayInfoForSeveralItems`.

```
Dim cTreeInfo As IHsvTreeInfo, vaParIDs, vaChildIDs
Dim vaHasKids, vaLabels, vaDescs
Set cTreeInfo = m_cMetadata.Entities
cTreeInfo.EnumAllParentAndChildIDs vaParIDs, vaChildIDs
cTreeInfo.GetDisplayInfoForSeveralItems _
MEMBER_LIST_ALL_HIERARCHY, vaChildIDs, vaParIDs, _
vaHasKids, vaLabels, vaDescs
For i = LBound(vaChildIDs) To UBound(vaChildIDs)
    Debug.Print vaLabels(i) & " " & vaDescs(i)
Next i
```

GetItemGeneration

Returns the generation level of a given member in the dimension hierarchy. Generation levels are ordered incrementally from the dimension name (level 1) down to the leaf members.

If a member is in multiple branches of the hierarchy, `GetItemGeneration` calculates the generation level within the branch that contains the parent identified by the member's `DefaultParent` attribute. For example, in the following diagram the `NewYork` entity is in two branches of the `Entity` dimension. If the `NewYork`'s `DefaultParent` is `UnitedStates`; `GetItemGeneration` returns 4; if the `DefaultParent` is `MainDivision`, `GetItemGeneration` returns 3:

Syntax

```
<IHsvTreeInfo>.GetItemGeneration (MemberID, plGeneration)
```

Argument	Description
<i>MemberID</i>	Long (ByVal). The member ID of the dimension member.
<i>plGeneration</i>	Long. Returns the number that represents the member's generation level within the dimension hierarchy.

GetItemID

Returns the member ID of a dimension member. The member's label is passed and its ID is returned.

Syntax

```
<IHsvTreeInfo>.GetItemID (bstrLabel)
```

Argument	Description
<i>bstrLabel</i>	String (ByVal). The label of the dimension member.

Return Value

Long. Returns the member ID.

Examples

The following Visual Basic 6 example defines a custom function named `GetMemberID` that returns the member ID of a dimension member. The `GetMemberID` function has the following arguments:

- The *IDimID* argument takes the ID of the applicable dimension, which is passed to the `Dimension` property to set an `IHsvTreeInfo` object reference for the applicable dimension. (For a listing of valid dimension IDs, see [“Dimension ID Constants” on page 834.](#))
- The *sMemLabel* argument takes the label of the dimension member, which is passed to `GetItemID`.

The member ID returned by `GetItemID` is set as the `GetMemberID` function's return value.

```
Function GetMemberID(lDimID As Long, sMemLabel As String) As Long
Dim cTreeInfo As IHsvTreeInfo
'g_cMetadata is an HsvMetadata object reference.
Set cTreeInfo = g_cMetadata.Dimension(lDimID)
```

```
GetMemberID = cTreeInfo.GetItemID(sMemLabel)
End Function
```

Following is a C# example that implements the custom GetMemberID function.

```
public int getMemberId(short shDimId, string sLabel)
{
    //gets a dimension member ID from a member label
    HSVMETADATALib.HsvMetadata cMetadata =
(HSVMETADATALib.HsvMetadata)g_cSession.Metadata;
    //HSVMETADATALib.IHsvTreeInfo cTreeInfo = cMetadata.Dimension(lDimId);
    //use accessor method for Dimension
    HSVMETADATALib.IHsvTreeInfo cTreeInfo = (HSVMETADATALib.IHsvTreeInfo)
    cMetadata.get_Dimension(shDimId);
    int iId = cTreeInfo.GetItemID(sLabel);
    return iId;
}
```

Tip: The GetMemberID function defined in this example is called by many other examples throughout this book, such as the [Example](#) for GetCell.

GetItemIDQL

For internal use.

GetItemLevel

Returns the level of a given member in the dimension hierarchy.

Note: Levels are ordered incrementally from the leaf members (level 0) up to the root of the dimension hierarchy.

Syntax

```
<IHsvTreeInfo>.GetItemLevel lMemberID, plLevel
```

Argument Description

lMemberID Long (ByVal). The member ID of the dimension member.

plLevel Long. Returns the number that represents the member's level within the dimension hierarchy.

Example

The following function indicates whether a given Period dimension member is a leaf member.

```
Function isPeriodLeaf(sMemberName As String) As Boolean
Dim cTreeInfo As IHsvTreeInfo, lMemID As Long, lLevel As Long
'm_cMetadata is an HsvMetadata object reference
Set cTreeInfo = m_cMetadata.Periods
lMemID = cTreeInfo.GetItemID(sMemberName)
```

```

cTreeInfo.GetItemLevel lMemID, lLevel
If lLevel = 0 Then
    isPeriodLeaf = True
Else
    isPeriodLeaf = False
End If
End Function

```

GetLabel

Gets the label of a dimension member. `GetLabel` is useful when working with methods such as `GetDefaultValueID` that return member IDs.

Syntax

```
<IHsvTreeInfo>.GetLabel(lItemID, pbstrLabel)
```

Argument Description

lItemID Long (ByVal). The member ID of the dimension member for which you want to get the label.

pbstrLabel String. Returns the label of the dimension member identified by the *lItemID* argument.

Return Value

Integer. Indicates the success of the function call; returns 0 for success or -1 for an error.

Example

`GetLabel` is used in the [Example](#) for `GetDefaultValueID`.

GetMemberListID

Returns the numeric ID of a member list.

Syntax

```
<IHsvTreeInfo>.GetMemberListID bstrListName, plListID
```

Argument Description

bstrListName String (ByVal). The name of the member list.

plListID Long. Returns the ID of the member list specified in the *bstrListName* argument.

Example

This example gets the ID of a member list named First Quarter. The `Dimension` property sets an `IHsvTreeInfo` object reference for the Period dimension, and `GetListIDFromName` assigns the ID of the member list named First Quarter to the `lID` variable.

```
Dim cIHsvTreeInfo As IHsvTreeInfo, lID As Long
```

```
Set cIHsvTreeInfo = m_cMetadata.Dimension(DIMENSIONPERIOD)
cIHsvTreeInfo.GetMemberListID "FirstQuarter", lID
```

GetMemberListName

Returns the name of a member list for which you have the ID.

Syntax

```
<IHsvTreeInfo>.GetMemberListName lListID, pbstrListName
```

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

<i>lListID</i>	Long (ByVal). The ID of the member list. Get this ID from a method that returns member list IDs.
----------------	--

<i>pbstrListName</i>	String. Returns the name of the member list identified by the <i>lListID</i> argument.
----------------------	--

Example

`GetMemberListName` is used in the example for the `HsvPOVSelection` method [GetListInfo](#).

GetNumBaseMembers

Returns a count of the base-level members beneath a dimension member.

Syntax

```
<IHsvTreeInfo>.GetNumBaseMembers lParentID, bIgnoreDuplicates,
plNumBaseMembers
```

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

<i>lParentID</i>	Long (ByVal). The member ID of the member for which you want to return the count.
------------------	---

<i>bIgnoreDuplicates</i>	Boolean (ByVal). Specifies whether duplicate member IDs are included in the count. Pass TRUE to filter out duplicates, FALSE to count duplicates.
--------------------------	---

<i>plNumBaseMembers</i>	Long. Returns the count of base-level members.
-------------------------	--

Example

The following example compares the number of base members for entities named Regional and Management. If the respective counts returned by `GetNumBaseMembers` are different, any code placed within the `If` structure would be executed.

```
Dim cTreeInfo As IHsvTreeInfo, lPar As Long, lRegCount As Long
Dim lManagCount As Long
Set cTreeInfo = m_cMetadata.Entities
lPar = cTreeInfo.GetItemID("Regional")
cTreeInfo.GetNumBaseMembers lPar, True, lRegCount
lPar = cTreeInfo.GetItemID("Management")
```

```

cTreeInfo.GetNumBaseMembers lPar, True, lManagCount
If lRegCount <> lManagCount Then
    ...
End If

```

GetNumChildren

Returns the number of children that are one level beneath a dimension member in a dimension hierarchy.

Syntax

```
<IHsvTreeInfo>.GetNumChildren(lItemID, plNum)
```

Argument Description

<i>lItemID</i>	Long (ByVal). The member ID of the member.
<i>plNum</i>	Long. Returns the number of children beneath the dimension member.

Return Value

Integer. Indicates the success of the function call; returns 0 for success or -1 for an error.

Example

This example tests whether the entity specified in a combo box named `comboEnts` has children. If the entity has children, then any code placed within the `If` structure would be executed.

```

Dim cIHsvTreeInfo As IHsvTreeInfo, sEnt As String
Dim lNumChild As Long, lEnt As Long
sEnt = comboEnts.Text
Set cIHsvTreeInfo = m_cMetadata.Entities
lEnt = cIHsvTreeInfo.GetItemID(sEnt)
cIHsvTreeInfo.GetNumChildren lEnt, lNumChild
If lNumChild > 0 Then
    MsgBox Str(lNumChild)
End If

```

GetNumDescendants

Returns the number of descendants beneath a member in a dimension's hierarchy, given the member's ID.

Syntax

```
<IHsvTreeInfo>.GetNumDescendants lParentID, plNum
```

Argument Description

<i>lParentID</i>	Long (ByVal). The member's ID.
------------------	--------------------------------

Argument Description

plNum Long. Returns the number of descendants beneath the dimension member.

Example

This example creates a function that takes a Scenario dimension member's label and returns a count of the member's descendants. The label is passed to `GetItemID`, which returns the member's ID. This ID is passed to `GetNumDescendants`, which returns the number of descendants that is assigned as the function's return value.

```
Function getEntDescendants(sParLabel) As Long
Dim cTreeInfo As IHsvTreeInfo, lParID As Long, lNumDesc As Long
Set cTreeInfo = m_cMetadata.Entities
lParID = cTreeInfo.GetItemID(sParLabel)
cTreeInfo.GetNumDescendants lParID, lNumDesc
getEntDescendants = lNumDesc
End Function
```

GetNumMembers

Returns the number of members in a dimension.

Syntax

```
<IHsvTreeInfo>.GetNumMembers plNumMembers
```

Argument Description

plNumMembers Long. Returns the number of members in the dimension to which the `IHsvTreeInfo` object reference has been initialized.

Example

The following example prints the number of members in the Account, Entity, and Scenario dimensions to Visual Basic's Immediate window. For each of these dimensions the `IHsvTreeInfo` object reference is set, `GetNumMembers` is called, and then `GetNumMembers`' return value is passed to `Debug.Print`.

```
Dim cTreeInfo As IHsvTreeInfo, lNumMems As Long
'Initialize object reference to Account dimension
Set cTreeInfo = m_cMetadata.Accounts
cTreeInfo.GetNumMembers lNumMems
Debug.Print "Account dimension members: " & CStr(lNumMems)
'Initialize object reference to Entity dimension
Set cTreeInfo = m_cMetadata.Entities
cTreeInfo.GetNumMembers lNumMems
Debug.Print "Entity dimension members: " & CStr(lNumMems)
'Initialize object reference to Scenario dimension
Set cTreeInfo = m_cMetadata.Scenarios
cTreeInfo.GetNumMembers lNumMems
Debug.Print "Scenario dimension members: " & CStr(lNumMems)
```


GetNumParents

Returns the number of parents for a dimension member, given the member's ID.

Syntax

```
<IHsvTreeInfo>.GetNumParents lMemberID, plNum
```

Argument Description

lMemberID Long (ByVal). The ID of the dimension member.

plNum Long. Returns the number of parents for the member.

Example

This example creates a function that takes an Entity dimension member's label and returns a count of the member's parents. `GetItemID` gets the member's ID, which is passed to `GetNumParents`. The count of members returned by `GetNumParents` is assigned as the function's return value.

```
Function getEntityParentCount(sChildLabel As String) As Long
Dim cTreeInfo As IHsvTreeInfo, lChildID As Long
Dim lNumParents As Long
Set cTreeInfo = m_cMetadata.Entities
lChildID = cTreeInfo.GetItemID(sChildLabel)
cTreeInfo.GetNumParents lChildID, lNumParents
getEntityParentCount = lNumParents
End Function
```

GetQualifiedLabel

For internal use.

GetTreeCapabilities

Returns various properties of a dimension.

Syntax

```
<IHsvTreeInfo>.GetTreeCapabilities pbReadWrite, pbMultiLevels,
pbMultiParents, pbDataVariesByParent
```

Argument	Description
<i>pbReadWrite</i>	Boolean. Returns TRUE if the dimension can be edited with Metadata Manager, otherwise FALSE.
<i>pbMultiLevels</i>	Boolean. Returns TRUE if the dimension allows parent-child relationships, otherwise FALSE.
<i>pbMultiParents</i>	Boolean. Returns TRUE if the dimension's members can have multiple parents, otherwise FALSE.

Argument	Description
<i>pbDataVariesByParent</i>	Boolean. Returns TRUE if the dimension's members data varies by parent, otherwise FALSE. This argument should return TRUE only for the Entity dimension.

Example

This example assigns the tree capability properties of the Account dimension to the Boolean variables used as `GetTreeCapabilities`' arguments.

```
Dim cIHsvTreeInfo As IHsvTreeInfo
Dim bRead As Boolean, bTreeView As Boolean
Dim bMultiPar As Boolean, bDataPar As Boolean
Set cIHsvTreeInfo = m_cMetadata.Dimension(DIMENSIONACCOUNT)
cIHsvTreeInfo.GetTreeCapabilities bRead, bTreeView, _
bMultiPar, bDataPar
```

GetTreeName

Returns the name of the dimension to which an `IHsvTreeInfo` object reference has been set.

Syntax

```
<IHsvTreeInfo>.GetTreeName()
```

Return Value

String. The name of the dimension.

Example

This example tests whether an `IHsvTreeInfo` object reference declared in another procedure has been set for the Entity dimension. If it has not, an error message is displayed, and then the procedure is exited.

```
Dim sDim As String
sDim = m_cIHsvTreeInfo.GetTreeName()
If sDim <> "Entity" Then
    MsgBox "Error: The IHsvTreeInfo object reference " & vbCrLf _
        & "is not set to the Entity dimension."
    Exit Sub
End If
```

GetTreeTimeStamp

Returns a timestamp that indicates when the dimension was last updated.

Syntax

```
<IHsvTreeInfo>.GetTreeTimeStamp pdTimeStamp
```

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

<i>pdTimeStamp</i>	Double. Returns the timestamp showing when the dimension was updated. The timestamp can be converted to a Date format; for example, in Visual Basic you can convert with CDate.
--------------------	--

Example

The following function returns an array of Dates that indicate when all of the dimensions were last updated. The HFMConstants type library constants “[Dimension ID Constants](#)” on page 834 are used for the *daDates* variable declaration, the loop, and the index of the array.

```
Function getDimStamps() As Date()  
Dim cTreeInfo As IHsvTreeInfo, dTime As Double  
Dim daDates(DIMENSION_UBOUND) As Date  
For i = DIMENSION_LBOUND To DIMENSION_UBOUND  
    'g_cMetadata is a previously set HsvMetadata instance  
    Set cTreeInfo = g_cMetadata.Dimension(i)  
    cTreeInfo.GetTreeTimeStamp dTime  
    daDates(i) = CDate(dTime)  
Next i  
getDimStamps = daDates  
End Function
```

HasChildren

Returns a Boolean that indicates whether a dimension member has child members.

Syntax

```
<IHsvTreeInfo>.HasChildren(lItemID, pbHasChildren)
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the member.
----------------	--

<i>pbHasChildren</i>	Boolean. Returns TRUE if the dimension member has child members, otherwise FALSE.
----------------------	---

Return Value

Integer. Indicates the success of the function call; returns 0 for success or -1 for an error.

Example

The following example tests whether the entity selected in a combo box named *comboEnts* has children. If it does, then any code placed within the *If* structure would be executed.

```
Dim cIHsvTreeInfo As IHsvTreeInfo  
Dim lEntID As Long, bHasKids As Boolean  
Set cIHsvTreeInfo = m_cMetadata.Entities  
lEntID = cIHsvTreeInfo.GetItemID(comboEnts.Text)  
cIHsvTreeInfo.HasChildren lEntID, bHasKids  
If bHasKids = True Then
```

```
...  
End If
```

IsMemberABaseOf

Indicates whether a member is in a base-level position beneath another member in a dimension hierarchy.

Syntax

```
<IHsvTreeInfo>.IsMemberABaseOf lMemberID, lParentID, pvarbIsBaseOf
```

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

<i>lMemberID</i>	Long (ByVal). The member ID of the possible base-level dimension member.
------------------	--

<i>lParentID</i>	Long (ByVal). The member ID of the possible higher-level dimension member.
------------------	--

<i>pvarbIsBaseOf</i>	Boolean. Returns TRUE if the member identified by the <i>lMemberID</i> argument is a base-level member beneath the member identified by the <i>lParentID</i> argument, FALSE otherwise.
----------------------	---

Example

This example creates a function that takes two dimension member labels and returns a Boolean indicating whether the first member is a base-level member beneath the second member. `GetItemID` returns the member IDs of the dimension members passed to the function. These IDs are passed to `IsMemberABaseOf`, and the Boolean returned is assigned as the example function's return value.

```
Function isBaseEntityFromLabel(sMem, sPar) As Boolean  
Dim cTreeInfo As IHsvTreeInfo, lMemID As Long  
Dim lParID As Long, bReturn As Boolean  
Set cTreeInfo = m_cMetadata.Entities  
lMemID = cTreeInfo.GetItemID(sMem)  
lParID = cTreeInfo.GetItemID(sPar)  
cTreeInfo.IsMemberABaseOf lMemID, lParID, bReturn  
isBaseEntityFromLabel = bReturn  
End Function
```

IsMemberAChildOf

Indicates whether one member is a child of another member.

Syntax

```
<IHsvTreeInfo>.IsMemberAChildOf lMemberID, lParentID, pbIsChildOf
```

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

<i>lMemberID</i>	Long (ByVal). The member ID of the possible child.
------------------	--

<i>lParentID</i>	Long (ByVal). The member ID of the possible parent.
------------------	---

Argument Description

pblsChildOf Boolean. Returns TRUE if the *IMemberID* member is a child of the *IParentID* member, otherwise FALSE.

Example

The following example tests whether the Connecticut entity is a child of the Regional entity, and prints the Boolean returned by `IsMemberAChildOf` to the Immediate window.

```
Dim lItem As Long, cTreeInfo As IHsvTreeInfo, lPar As Long
Dim bRet As Boolean
Set cTreeInfo = m_cMetadata.Entities
lItem = cTreeInfo.GetItemID("Connecticut")
lPar = cTreeInfo.GetItemID("Regional")
cTreeInfo.IsMemberAChildOf lItem, lPar, bRet
Debug.Print bRet
```

IsMemberADescendantOf

Indicates whether one member is a descendant beneath another member in a dimension hierarchy.

Syntax

```
<IHsvTreeInfo>.IsMemberADescendantOf lMemberID, lParentID,
pvarbIsDescendantOf
```

Argument Description

IMemberID Long (ByVal). The member ID of the possible descendant.

IParentID Long (ByVal). The member ID of the possible higher-level dimension member.

pvarbIsDescendantOf Boolean. Returns TRUE if the member identified by the *IMemberID* argument is a descendant beneath the member identified by the *IParentID* argument, FALSE otherwise.

Example

This example creates a function that takes two dimension member labels and returns a Boolean indicating whether the first member is a descendant beneath the second member. `GetItemID` returns the member IDs of the dimension members passed to the function. These IDs are passed to `IsMemberADescendantOf`, and the Boolean returned is assigned as the example function's return value.

```
Function isDescEntityFromLabel(sMem, sPar) As Boolean
Dim cTreeInfo As IHsvTreeInfo, lMemID As Long
Dim lParID As Long, bReturn As Boolean
Set cTreeInfo = m_cMetadata.Entities
lMemID = cTreeInfo.GetItemID(sMem)
lParID = cTreeInfo.GetItemID(sPar)
cTreeInfo.IsMemberADescendantOf lMemID, lParID, bReturn
isDescEntityFromLabel = bReturn
End Function
```

SortMembersBasedOnList

Filters and sorts member IDs, using the members in a member list as the filtering and sorting criteria. Given an array of member IDs and an ID of a member list, `SortMembersBasedOnList` filters out the IDs of those members that are not in the member list, then sorts the remaining IDs according to the order in which the members are defined in the list.

Tip: You can also filter and sort against a dimension's default hierarchy instead of a member list.

Syntax

```
<IHsvTreeInfo>.SortMembersBasedOnList lListID, lListTopMemberID,  
bIgnoreDuplicates, varalInputItemIDs, varalInputParentIDs, pvaralItemIDs,  
pvaralParentIDs
```

Argument	Description
<i>lListID</i>	Long (ByVal). Identifies whether a member list or the default dimension hierarchy will be used to filter and sort. Pass a valid list ID to use a member list, or pass the <code>HFMConstants</code> type library constant <code>MEMBER_LIST_ALL_HIERARCHY</code> to use the default hierarchy. Tip: You can get member list IDs with <code>GetMemberListID</code> . For more information, see “GetMemberListID” on page 221 .
<i>lListTopMemberID</i>	Long (ByVal). The usage of this argument depends on what you pass to the <i>lListID</i> argument: <ul style="list-style-type: none">● If you pass <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, this determines whether you sort and filter by using all of the dimension's members or by the members of a node. Pass the <code>TREE_ROOT</code> constant to use all members, or the member ID of the node's parent to use a node.● If you pass anything other than <code>MEMBER_LIST_ALL_HIERARCHY</code> to <i>lListID</i>, the <i>lListTopMemberID</i> argument is ignored. Note that since this argument is not optional, you must still pass a valid Long.
<i>bIgnoreDuplicates</i>	Boolean (ByVal). Specifies whether duplicate member IDs are returned. Pass <code>TRUE</code> to filter out duplicates, <code>FALSE</code> to return duplicates.
<i>varalInputItemIDs</i>	Long array (ByVal). The member IDs to be sorted and filtered.
<i>varalInputParentIDs</i>	Long array (ByVal). The member IDs of the parent members passed in the <i>varalInputItemIDs</i> argument. The items in these two arrays must have a one-to-one correspondence.
<i>pvaralItemIDs</i>	Variant array. Returns the filtered and sorted member IDs. The array is returned as a Long subtype.
<i>pvaralParentIDs</i>	Variant array. Returns the IDs of the parents of the members returned in the <i>pvaralItemIDs</i> argument. The array is returned as a Long subtype.

TranslateAttributeValueForDisplay

Returns a String representation of a dimension member's attribute. However, in cases where the attribute value is itself a dimension member, the member ID of the attribute value is returned instead of the member's label.

Syntax

```
<IHsvTreeInfo>.TranslateAttributeValueForDisplay sAttrib, vValue,  
bstrValue, plDimID
```

Argument Description

<i>sAttrib</i>	Integer (ByVal). The ID of the attribute. These IDs are represented by the following groups of constants in the HFMConstants type library: <ul style="list-style-type: none">● “Account Attribute Constants” on page 824● “Custom Dimension Attributes” on page 828● “Entity Attribute Constants” on page 831● “Scenario Attribute Constants” on page 837● “Currency Attribute Constants” on page 831● “Consolidation Method Attribute Constants” on page 829
<i>vValue</i>	Variant (ByVal). The attribute value to be converted to a String representation. For example, to get the String representation of an account type, you can pass the numeric value returned by <code>HsvAccounts.GetAccountType</code> .
<i>bstrValue</i>	String. Returns one of the following values: <ul style="list-style-type: none">● If the attribute value is not a dimension member, this argument returns the String representation of the attribute value.● If the attribute value is a dimension member, this argument returns a blank string. In this case the member ID of the attribute value is returned in the <i>plDimID</i> argument. For example, a blank string would be returned if <code>TranslateAttributeValueForDisplay</code> is called to obtain an account's <code>Custom1TopMember</code> attribute and the attribute value is a valid <code>Custom 1</code> dimension member.
<i>plDimID</i>	Long. Returns one of the following values: <ul style="list-style-type: none">● If the attribute value is not a dimension member, this argument returns -1.● If the attribute value is a dimension member, this argument returns the member ID. In this case you can get the member's label with GetItemID.

Example

The following function takes an account name and returns the String representation of the account's type.

```
Function getAccountTypeString(sLabel As String) As String  
Dim cAccounts As HsvAccounts, cTreeInfo As IHsvTreeInfo  
Dim lAcctId As Long, iType As Integer, sVal As String  
Dim lRetId As Long  
Set cAccounts = m_cMetadata.Accounts  
Set cTreeInfo = m_cMetadata.Accounts  
lAcctId = cTreeInfo.GetItemID(sLabel)  
cAccounts.GetAccountType lAcctId, iType  
cTreeInfo.TranslateAttributeValueForDisplay _ ATTRIB_ACCOUNT_TYPE, iType,  
sVal, lRetId  
getAccountTypeString = sVal  
End Function
```

HsvAccounts Object Methods

The HsvAccounts object's methods return attributes of Account dimension members. These methods are summarized in [Table 12 on page 59](#), and are described in detail in the following topics.

Assign HsvAccounts object references with the Accounts property of the HsvMetadata object as shown in the following example:

```
Dim cMetadata as HsvMetadata, cAccounts as HsvAccounts
Set cMetadata = m_cSession.Metadata
Set cAccounts = cMetadata.Accounts
```

GetAccountType

Returns the number that identifies an account's type.

Tip: `HsvMetadata.GetCellLevelAccountType` returns a cell's account type, which in some cases can differ from the account type of the cell's account. For details, see [“GetCellLevelAccountType” on page 170](#).

Syntax

```
<HsvAccounts>.GetAccountType lItemID, psAccountType
```

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

<i>ItemID</i>	Long (ByVal). The member ID of the account.
---------------	---

<i>psAccountType</i>	Integer. Returns the account type of the account. For a list of constants that represent the valid return values, see “Account Type Constants” on page 826 .
----------------------	--

Example

The following function indicates whether a given account has an account type of CurrencyRate.

```
Function isCurrencyRateType(sMem As String) As Boolean
Dim cAccounts As HsvAccounts, cTreeInfo As IHsvTreeInfo
Dim lAccount As Long, iAcctType As Integer
'g_cMetadata is an HsvMetadata object reference
Set cAccounts = g_cMetadata.Accounts
Set cTreeInfo = g_cMetadata.Accounts
lAccount = cTreeInfo.GetItemID(sMem)
cAccounts.GetAccountType lAccount, iAcctType
If iAcctType = ACCOUNTTYPE_CURRENCYRATE Then
    isCurrencyRateType = True
Else
    isCurrencyRateType = False
End If
End Function
```


GetCalcAttribute

Returns the value of a given Account dimension member's CalcAttribute attribute.

Syntax

```
<HsvAccounts>.GetCalcAttribute lItemID, pbstrUDAttr
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the Account dimension member.
----------------	--

<i>pbstrUDAttr</i>	String. Returns the attribute's value.
--------------------	--

GetICPTopMember

Returns the member ID of the Intercompany Partner dimension member that has been assigned as an Account dimension member's ICPTopMember attribute.

Syntax

```
<HsvAccounts>.GetICPTopMember lItemID, plICPTopMember
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the account.
----------------	---

<i>plICPTopMember</i>	Long. Returns the member ID of the Intercompany Member assigned as the account's ICPTopMember attribute.
-----------------------	--

Example

The following is a function that takes an Account member's label and returns the label of the Intercompany Partner member assigned as the account's ICPTopMember attribute.

```
Function getTopIcpLabel(sAcctName As String) As String
Dim cAccounts As HsvAccounts, lTopID As Long, lAcctID As Long
Dim cTreeInfo As IHsvTreeInfo, sLabel As String
Set cAccounts = m_cMetadata.Accounts
Set cTreeInfo = m_cMetadata.Accounts
lAcctID = cTreeInfo.GetItemID(sAcctName)
cAccounts.GetICPTopMember lAcctID, lTopID
cTreeInfo.GetLabel lTopID, sLabel
getTopIcpLabel = sLabel
End Function
```

GetIsICP

Indicates the value assigned to the IsICP attribute of an Account dimension member.

Syntax

```
<HsvAccounts>.GetIsICP lItemID, psIsICP
```

Argument Description

<i>ItemID</i>	Long (ByVal). The member ID of the account.
<i>psIsICP</i>	Integer. Indicates the value of the account's IsICP attribute. Valid values are as follows: <ul style="list-style-type: none">● -1 if the IsICP value is Y.● 0 if the IsICP value is N.● 1 if the IsICP value is R.

Example

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

```
Function getIsIcpLabel(sLabel As String) As Integer
Dim cAccounts As HsvAccounts, iRet As Integer, lAcctID As Long
Dim cTreeInfo As IHsvTreeInfo
Set cAccounts = m_cMetadata.Accounts
Set cTreeInfo = m_cMetadata.Accounts
lAcctID = cTreeInfo.GetItemID(sLabel)
cAccounts.GetIsICP lAcctID, iRet
getIsIcpLabel = iRet
End Function
```

GetNumDecimalPlaces

Returns the maximum number of digits to the right of the decimal point that an account supports. In other words, `GetNumDecimalPlaces` returns the value to which an account's `NumDecimalPlaces` attribute has been set.

Syntax

```
<HsvAccounts>.GetNumDecimalPlaces lItemID, psNumDecimalPlaces
```

Argument Description

<i>ItemID</i>	Long (ByVal). The member ID of the account.
<i>psNumDecimalPlaces</i>	Integer. Returns the account's maximum number of decimal places.

Example

This example inserts the labels and the number of the supported decimal places of an application's accounts into a Microsoft Excel 97 spreadsheet. The member IDs of the accounts are obtained with `IHsvTreeInfo.EnumAllMemberIDs`. A two-dimensional array containing the accounts' labels and their corresponding `NumDecimalPlaces` values is created, using `GetNumDecimalPlaces` and `IHsvTreeInfo.GetLabel`. This array is then inserted into Microsoft Excel.

```
Dim cAccounts As HsvAccounts, cTreeInfo As IHsvTreeInfo
Dim vaAcctIDs, sLabel As String, iNumDecs As Integer
```

```

Dim lHiBound As Long, vaAcctDecimals(), lRows As Long
Dim xlApp As Excel.Application, wb As Excel.Workbook
Set cAccounts = m_cMetadata.Accounts
Set cTreeInfo = m_cMetadata.Accounts
cTreeInfo.EnumAllMemberIDs vaAcctIDs
lHiBound = UBound(vaAcctIDs)
ReDim vaAcctDecimals(lHiBound, lHiBound)
For i = LBound(vaAcctIDs) To lHiBound
    cAccounts.GetNumDecimalPlaces vaAcctIDs(i), iNumDecs
    cTreeInfo.GetLabel vaAcctIDs(i), sLabel
    vaAcctDecimals(i, 0) = sLabel
    vaAcctDecimals(i, 1) = iNumDecs
Next i
'Open Excel - assumes Excel is referenced in Project>References.
Set xlApp = CreateObject("Excel.Application")
xlApp.Visible = True
Set wb = xlApp.Workbooks.Add()
lRows = UBound(vaAcctDecimals)
'Loop through the arrays and put the data into Excel.
For i = 0 To lRows
    xlApp.Range("A" & Trim$(Str(i + 1))).Value = _
        vaAcctDecimals(i, 0)
    xlApp.Range("B" & Trim$(Str(i + 1))).Value = _
        vaAcctDecimals(i, 1)
Next i

```

GetPlugAccount

Returns the member ID of an account's plug account. In other words, `GetPlugAccount` returns the member ID of the account that has been defined as an account's `PlugAcct` attribute.

Syntax

```
<HsvAccounts>.GetPlugAccount lAccountID, plPlugAccountID
```

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

<i>lAccountID</i>	Long (ByVal). The member ID of the account for which you want to return the plug account.
-------------------	---

<i>plPlugAccountID</i>	Long. Returns the member ID of the plug account.
------------------------	--

Example

The following example creates a function that takes an account label and returns the label of the account's Plug account.

```

Function GetPlugFromLabel(sAcctLabel As String) As String
Dim cAccounts As HsvAccounts, cTreeInfo As IHsvTreeInfo
Dim lAcctID As Long, lPlugID As Long, sPlugLabel As String
Set cAccounts = m_cMetadata.Accounts
Set cTreeInfo = m_cMetadata.Accounts
lAcctID = cTreeInfo.GetItemID(sAcctLabel)
cAccounts.GetPlugAccount lAcctID, lPlugID
cTreeInfo.GetLabel lPlugID, sPlugLabel
GetPlugFromLabel = sPlugLabel

```

End Function

GetSecurityClassID

Returns the ID of the security class that has been assigned to an account, given the account's member ID.

Syntax

```
<HsvAccounts>.GetSecurityClassID lItemID, plSecurityClassID
```

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

<i>lItemID</i>	Long (ByVal). The account's member ID.
----------------	--

<i>plSecurityClassID</i>	Long. Returns the ID of the account's security class, or SECURITY_CLASS_NONE if a security class has not been assigned to the account.
--------------------------	--

Tip: To get the label of the security class, pass the ID returned by this argument to the HsvSecurityAccess object's GetSecurityClassLabel method.

Example

The following example creates a function that takes an account's label and returns the label of the account's security class, or a blank string if a security class has not been assigned to the account. IHsvTreeInfo.GetItemID returns the account's member ID, which is then passed to GetSecurityClassID. If GetSecurityClassID indicates that the account has no security class, a blank string is assigned as the function's return value; otherwise, HsvSecurityAccess.GetSecurityClassLabel gets the security class's label, which is then assigned as the function's return value.

```
Function getAcctSecLabel(sAcctLabel As String) As String
    Dim cAccounts As HsvAccounts, cTreeInfo As IHsvTreeInfo
    Dim cSecurityAccess As HsvSecurityAccess
    Dim lAcctID As Long, lSecID As Long, sSecLabel As String
    Set cAccounts = m_cMetadata.Accounts
    Set cTreeInfo = m_cMetadata.Accounts
    lAcctID = cTreeInfo.GetItemID(sAcctLabel)
    cAccounts.GetSecurityClassID lAcctID, lSecID
    If lSecID < 0 Then
        getAcctSecLabel = ""
    Else
        Set cSecurityAccess = m_cSession.Security
        cSecurityAccess.GetSecurityClassLabel lSecID, sSecLabel
        getAcctSecLabel = sSecLabel
    End If
End Function
```

GetSubmissionGroup

Returns the value of an account's Submission Group property.

Syntax

```
<HsvAccounts>.GetSubmissionGroup lItemID, plSubmissionGroup
```

Argument	Description
<i>lItemID</i>	Long. The member ID of the account.
<i>plSubmissionGroup</i>	Long. Returns the Submission Group value.

Example

`GetSubmissionGroup` is used in the example for [GetUseSubmissionPhaseFlag](#).

GetTopMemberOfValidCustom1Hierarchy

Returns the member ID of the top member of an account's Custom 1 dimension hierarchy. In other words, this method returns the member ID of the Custom 1 member that has been defined as an account's `Custom1TopMember` attribute.

Syntax

```
<HsvAccounts>.GetTopMemberOfValidCustom1Hierarchy lItemID, plTopMember
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the account.
<i>plTopMember</i>	Long. Returns the member ID of the top Custom 1 dimension member.

Example

The following example creates a function that takes an account's label and returns the label of its top Custom 1 dimension member.

```
Function GetCust1FromAcct(sAcct As String) As String
Dim cAccounts As HsvAccounts, cTreeInfo As IHsvTreeInfo
Dim lAcctID As Long, lCust1ID As Long, sLabel As String
Set cAccounts = m_cMetadata.Accounts
Set cTreeInfo = m_cMetadata.Accounts
lAcctID = cTreeInfo.GetItemID(sAcct)
cAccounts.GetTopMemberOfValidCustom1Hierarchy lAcctID, lCust1ID
Set cTreeInfo = m_cMetadata.Custom1
cTreeInfo.GetLabel lCust1ID, sLabel
GetCust1FromAcct = sLabel
End Function
```

GetTopMemberOfValidCustom2Hierarchy

Returns the member ID of the top member of an account's Custom 2 dimension hierarchy. In other words, this method returns the member ID of the Custom 2 member that has been defined as an account's `Custom2TopMember` attribute.

Syntax

```
<HsvAccounts>.GetTopMemberOfValidCustom2Hierarchy lItemID, plTopMember
```

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

<i>ItemID</i>	Long (ByVal). The member ID of the account.
---------------	---

<i>plTopMember</i>	Long. Returns the member ID of the top Custom 2 dimension member.
--------------------	---

Example

See the [Example](#) for `GetTopMemberOfValidCustom1Hierarchy`, replacing that method with `GetTopMemberOfValidCustom2Hierarchy`.

GetTopMemberOfValidCustom3Hierarchy

Returns the member ID of the top member of an account's Custom 3 dimension hierarchy. In other words, this method returns the member ID of the Custom 3 member that has been defined as an account's `Custom3TopMember` attribute.

Syntax

```
<HsvAccounts>.GetTopMemberOfValidCustom3Hierarchy lItemID, plTopMember
```

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

<i>ItemID</i>	Long (ByVal). The member ID of the account.
---------------	---

<i>plTopMember</i>	Long. Returns the member ID of the top Custom 3 dimension member.
--------------------	---

Example

See the [Example](#) for `GetTopMemberOfValidCustom1Hierarchy`, replacing that method with `GetTopMemberOfValidCustom3Hierarchy`.

GetTopMemberOfValidCustom4Hierarchy

Returns the member ID of the top member of an account's Custom 4 dimension hierarchy. In other words, this method returns the member ID of the Custom 4 member that has been defined as an account's `Custom4TopMember` attribute.

Syntax

```
<HsvAccounts>.GetTopMemberOfValidCustom4Hierarchy lItemID, plTopMember
```

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

<i>ItemID</i>	Long (ByVal). The member ID of the account.
---------------	---

<i>plTopMember</i>	Long. Returns the member ID of the top Custom 4 dimension member.
--------------------	---

Example

See the [Example](#) for `GetTopMemberOfValidCustom1Hierarchy`, replacing that method with `GetTopMemberOfValidCustom4Hierarchy`.

GetUserDefined1

Returns an account's `UserDefined1` attribute, given the account's member ID.

Syntax

```
<HsvAccounts>.GetUserDefined1 lItemID, pbstrUDAttr
```

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

<i>lItemID</i>	Long (ByVal). The account's member ID.
----------------	--

<i>pbstrUDAttr</i>	String. Returns the <code>UserDefined1</code> attribute.
--------------------	--

GetUserDefined2

Returns an account's `UserDefined2` attribute, given the account's member ID.

Syntax

```
<HsvAccounts>.GetUserDefined2 lItemID, pbstrUDAttr
```

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

<i>lItemID</i>	Long (ByVal). The account's member ID.
----------------	--

<i>pbstrUDAttr</i>	String. Returns the <code>UserDefined2</code> attribute.
--------------------	--

GetUserDefined3

Returns an account's `UserDefined3` attribute, given the account's member ID.

Syntax

```
<HsvAccounts>.GetUserDefined3 lItemID, pbstrUDAttr
```

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

<i>lItemID</i>	Long (ByVal). The account's member ID.
----------------	--

<i>pbstrUDAttr</i>	String. Returns the <code>UserDefined3</code> attribute.
--------------------	--

GetXBRLTags

Returns an account's XBRLTags attribute, given the account's member ID.

Syntax

```
<HsvAccounts>.GetXBRLTags lItemID, pbstrXBRLTags
```

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

<i>lItemID</i>	Long (ByVal). The account's member ID.
----------------	--

<i>pbstrXBRLTags</i>	String. Returns the XBRLTags attribute.
----------------------	---

IsCalculated

Returns a Boolean that indicates whether an account's data is calculated by Financial Management or is manually entered. In other words, this method returns the value to which an account's IsCalculated attribute has been set.

Syntax

```
<HsvAccounts>.IsCalculated lItemID, pbIsCalculated
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the account.
----------------	---

<i>pbIsCalculated</i>	Boolean. Returns TRUE if the account's data is calculated, FALSE if the data is manually entered.
-----------------------	---

Example

The following example tests whether the account specified in a combo box control is a calculated account. If so, any code placed in the If structure would be executed.

```
Dim cAccounts As HsvAccounts, cTreeInfo As IHsvTreeInfo
Dim lAccount as Long, sAcctLabel As String, bIsCalc As Boolean
Set cAccounts = m_cMetadata.Accounts
Set cTreeInfo = m_cMetadata.Accounts
sAcctLabel = comboAcct.Text
lAccount = cTreeInfo.GetItemID(sAcctLabel)
cAccounts.IsCalculated lAccount, bIsCalc
If bIsCalc = True Then
    ...
End If
```

IsConsolidated

Returns a Boolean that indicates whether an account's data is consolidated to parent entities. In other words, this method returns the value to which an account's IsConsolidated attribute has been set.

Syntax

```
<HsvAccounts>.IsConsolidated lItemID, pbIsConsolidated
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the account.
----------------	---

<i>pbIsConsolidated</i>	Boolean. Returns TRUE if the account is consolidated to parents, FALSE if it is ignored during consolidation.
-------------------------	---

IsCustom1AggregationEnabled

Returns the value to which an account's EnableCustom1Aggr attribute has been set.

Syntax

```
<HsvAccounts>.IsCustom1AggregationEnabled lItemID, pbIsEnabled
```

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

<i>lItemID</i>	Long. The member ID of the account.
----------------	-------------------------------------

<i>pbIsEnabled</i>	Boolean. Returns TRUE if the EnableCustom1Aggr attribute is enabled, FALSE if it is disabled.
--------------------	---

Example

The following example tests whether the EnableCustom1Aggr attribute of the account specified in the `comboAcct` combo box is enabled. If this attribute is enabled, any code placed within the `If` structure would be executed.

```
Dim cAccounts As HsvAccounts, cTreeInfo As IHsvTreeInfo
Dim lAcctID As Long, bIsEnabled As Boolean
Dim sAcctLabel As String
Set cAccounts = m_cMetadata.Accounts
Set cTreeInfo = m_cMetadata.Accounts
sAcctLabel = comboAcct.Text
lAcctID = cTreeInfo.GetItemID(sAcctLabel)
cAccounts.IsCustom1AggregationEnabled lAcctID, bIsEnabled
If bIsEnabled = True Then
    ...
End If
```

IsCustom2AggregationEnabled

Returns the value to which an account's EnableCustom2Aggr attribute has been set.

Syntax

```
<HsvAccounts>.IsCustom2AggregationEnabled lItemID, pbIsEnabled
```

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

<i>ItemID</i>	Long. The member ID of the account.
---------------	-------------------------------------

<i>pbIsEnabled</i>	Boolean. Returns TRUE if the EnableCustom2Aggr attribute is enabled, FALSE if it is disabled.
--------------------	---

Example

See the [Example](#) for `IsCustom1AggregationEnabled`, replacing that method with `IsCustom2AggregationEnabled`.

IsCustom3AggregationEnabled

Returns the value to which an account's `EnableCustom3Aggr` attribute has been set.

Syntax

```
<HsvAccounts>.IsCustom3AggregationEnabled lItemID, pbIsEnabled
```

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

<i>ItemID</i>	Long. The member ID of the account.
---------------	-------------------------------------

<i>pbIsEnabled</i>	Boolean. Returns TRUE if the EnableCustom3Aggr attribute is enabled, FALSE if it is disabled.
--------------------	---

Example

See the [Example](#) for `IsCustom1AggregationEnabled`, replacing that method with `IsCustom3AggregationEnabled`.

IsCustom4AggregationEnabled

Returns the value to which an account's `EnableCustom4Aggr` attribute has been set.

Syntax

```
<HsvAccounts>.IsCustom4AggregationEnabled lItemID, pbIsEnabled
```

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

<i>ItemID</i>	Long. The member ID of the account.
---------------	-------------------------------------

<i>pbIsEnabled</i>	Boolean. Returns TRUE if the EnableCustom4Aggr attribute is enabled, FALSE if it is disabled.
--------------------	---

Example

See the [Example](#) for `IsCustom1AggregationEnabled`, replacing that method with `IsCustom4AggregationEnabled`.

IsICP

Returns a Boolean that indicates whether the account is used in intercompany transactions. In other words, this method returns the value to which an account's IsICP attribute has been set.

Syntax

```
<HsvAccounts>.IsICP lItemID, pbIsICP
```

Argument Description

lItemID Long (ByVal). The member ID of the account.

pbIsICP Boolean. Returns TRUE if the account is used in intercompany transactions, FALSE otherwise.

IsICPRestricted

Indicates whether an Account dimension member is restricted from having Intercompany Partner transactions with itself. In other words, this method indicates whether an Account dimension member's IsICP attribute has been set to a value of R.

Syntax

```
<HsvAccounts>.IsICPRestricted lItemID, pbIsICPRestricted
```

Argument Description

lItemID Long (ByVal). The member ID of the account.

pbIsICPRestricted Boolean. Returns TRUE if the account's IsICP attribute has been set to R, FALSE otherwise.

UsesLineItems

Returns a Boolean that indicates whether an account allows line items.

Syntax

```
<HsvAccounts>.UsesLineItems lItemID, pvarbUsesLineItems
```

Argument Description

lItemID Long (ByVal). The member ID of the account.

pvarbUsesLineItems Boolean. Returns TRUE if the account allows line items, FALSE otherwise.

Example

UsesLineItems is used in the example for HsvData.[GetCellLineItems](#).

HsvCustom Object Methods

The HsvCustom object applies to the four Custom dimensions. These methods are summarized in [Table 13 on page 61](#), and are described in detail in the following topics.

Use the HsvMetadata object's Custom1, Custom2, Custom3, and Custom4 properties to assign HsvCustom object references. The following example uses the Custom2 property to assign an HsvCustom object reference for the Custom 2 dimension:

```
Dim cMetadata as HsvMetadata, cCustom as HsvCustom
Set cMetadata = m_cSession.MetaData
Set cCustom = cMetadata.Custom2
```

GetAggregationWeight

For internal use.

GetSecurityClassID

Returns the ID of the security class assigned to a Custom 1, 2, 3, or 4 dimension member.

Syntax

```
<HsvCustom>.GetSecurityClassID lItemID, plSecurityClassID
```

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

<i>lItemID</i>	Long (ByVal). The ID of the Custom dimension member.
----------------	--

<i>plSecurityClassID</i>	Long. Returns the ID of the Custom dimension member's security class. If no security class has been assigned to the member, SECURITY_CLASS_NONE is returned.
--------------------------	--

Example

The following function takes the name of a Custom 1 dimension member and returns the name of the member's security class, or a blank string if no security class has been assigned.

```
Function getCustom1SecClassName(sCustom1Label As String) As _
String
Dim cCustom1 As HsvCustom, cTreeInfo As IHsvTreeInfo
Dim lCustId As Long, lSecId As Long, sSecLabel As String
Set cCustom1 = m_cMetadata.Custom1
Set cTreeInfo = m_cMetadata.Custom1
lCustId = cTreeInfo.GetItemID(sCustom1Label)
cCustom1.GetSecurityClassID lCustId, lSecId
If lSecId = SECURITY_CLASS_NONE Then
    getCustom1SecClassName = ""
Else
    m_cSecurityAccess.GetSecurityClassLabel lSecId, sSecLabel
    getCustom1SecClassName = sSecLabel
End If
End Function
```

GetSubmissionGroup

Returns the value of a Custom dimension member's Submission Group property.

Syntax

```
<HsvCustom>.GetSubmissionGroup lItemID, plSubmissionGroup
```

Argument	Description
<i>lItemID</i>	Long. The member ID of the Custom dimension member.
<i>plSubmissionGroup</i>	Long. Returns the Submission Group value.

Example

GetSubmissionGroup is used in the example for [GetUseSubmissionPhaseFlag](#).

GetUserDefined1

Returns the value assigned to the UserDefined1 attribute of a Custom 1, 2, 3, or 4 dimension member.

Syntax

```
<HsvCustom>.GetUserDefined1 lItemID, pbstrUDAttr
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the Custom dimension member.
<i>pbstrUDAttr</i>	String. Returns the value assigned to the UserDefined1 attribute.

GetUserDefined2

Returns the value assigned to the UserDefined2 attribute of a Custom 1, 2, 3, or 4 dimension member.

Syntax

```
<HsvCustom>.GetUserDefined2 lItemID, pbstrUDAttr
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the Custom dimension member.
<i>pbstrUDAttr</i>	String. Returns the value assigned to the UserDefined2 attribute.

GetUserDefined3

Returns the value assigned to the UserDefined3 attribute of a Custom 1, 2, 3, or 4 dimension member.

Syntax

```
<HsvCustom>.GetUserDefined3 lItemID, pbstrUDAttr
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the Custom dimension member.
----------------	---

<i>pbstrUDAttr</i>	String. Returns the value assigned to the UserDefined3 attribute.
--------------------	---

IsCalculated

Returns a Boolean that indicates whether a Custom dimension member's data is calculated by Financial Management or is manually entered.

Syntax

```
<HsvCustom>.IsCalculated lItemID, pbIsCalculated
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the Custom dimension member.
----------------	---

<i>pbIsCalculated</i>	Boolean. Returns TRUE if the Custom dimension member's data is calculated, FALSE if it is manually entered.
-----------------------	---

Example

The following function indicates whether a given Custom dimension member is calculated. The second argument takes the ID of the Custom dimension; dimension IDs are represented by the HFMCConstants type library constants listed in [“Dimension ID Constants” on page 834](#).

```
Function isCustomMemCalculated(sMem As String, iDim As Integer) _  
    As Boolean  
    Dim cTreeInfo As IHsvTreeInfo, cCustom As HsvCustom  
    Dim bIsCalc As Boolean  
    'g_cMetadata is an hsvmetadata object reference  
    Set cCustom = g_cMetadata.Dimension(iDim)  
    Set cTreeInfo = g_cMetadata.Dimension(iDim)  
    lCust = cTreeInfo.GetItemID(sMem)  
    cCustom.IsCalculated lCust, bIsCalc  
    If bIsCalc = True Then  
        isCustomMemCalculated = True  
    Else  
        isCustomMemCalculated = False  
    End If  
End Function
```

IsSwitchSignEnabledForFlow

Indicates whether a Custom dimension member's SwitchSignForFlow attribute is enabled.

Syntax

```
<HsvCustom>.IsSwitchSignEnabledForFlow lMemberID, pbFlag
```

Argument Description

lMemberID Long (ByVal). The member ID of the Custom dimension member.

pbFlag Boolean. Indicates whether the SwitchSignForFlow attribute is enabled. Returns TRUE if it is enabled, FALSE otherwise.

IsSwitchTypeEnabledForFlow

Indicates whether a Custom dimension member's SwitchTypeForFlow attribute is enabled.

Syntax

```
<HsvCustom>.IsSwitchTypeEnabledForFlow lMemberID, pbFlag
```

Argument Description

lMemberID Long (ByVal). The member ID of the Custom dimension member.

pbFlag Boolean. Indicates whether the SwitchTypeForFlow attribute is enabled. Returns TRUE if it is enabled, FALSE otherwise.

HsvEntities Object Methods

The HsvEntities object's methods return attributes of Entity dimension members. These methods are summarized in [Table 14 on page 62](#), and are described in detail in the following topics.

Assign HsvEntities object references with the Entities property of the HsvMetadata object as shown in the following example:

```
Dim cMetadata as HsvMetadata, cEntities as HsvEntities
Set cMetadata = m_cSession.Metadata
Set cEntities = cMetadata.Entities
```

GetAllowAdjustments

Returns a Boolean that indicates whether an entity allows journal entries. In other words, this method returns the value to which an entity's AllowAdjs attribute has been set.

Syntax

```
<HsvEntities>.GetAllowAdjustments lItemID, pbOK
```

Argument Description

<i>ItemID</i>	Long (ByVal). The member ID of the entity.
<i>pbOK</i>	Boolean. Returns TRUE is the entity allows journal entries, FALSE if it does not.

GetAllowAdjustmentsFromChildren

Returns a Boolean that indicates whether an entity allows journal postings from its children. In other words, this method returns the value to which an entity's `AllowAdjFromChildren` attribute has been set.

Syntax

```
<HsvEntities>.GetAllowAdjustmentsFromChildren lItemID, pbOK
```

Argument Description

<i>ItemID</i>	Long (ByVal). The member ID of the entity.
<i>pbOK</i>	Boolean. Returns TRUE is the entity allows journal postings from its children, FALSE if it does not.

GetDefaultValueID

Returns the member ID of the Value dimension member that represents an entity's default currency. In other words, this method returns the member ID of the Value dimension member that has been specified for an entity's `DefCurrency` attribute.

Syntax

```
<HsvEntities>.GetDefaultValueID lItemID, plValueID
```

Argument Description

<i>ItemID</i>	Long (ByVal). The member ID of the entity.
<i>plValueID</i>	Long. Returns the member ID of the Value dimension member that represents the entity's default currency.

Example

The following example uses `GetDefaultValueID` to get the default currency of the entity in the `comboEnts` combo box control. The example then uses `GetLabel` to get the name of the currency, and inserts this name into the `lblDefCurr` label control.

```
Dim cEntities As HsvEntities, cTreeInfo As IHsvTreeInfo
Dim sEntLabel As String, lValId As Long, sValLabel As String
Dim lEnt As Long
Set cEntities = m_cMetadata.Entities
sEntLabel = comboEnts.Text
'Get the ID of the entity in the combo box
Set cTreeInfo = m_cMetadata.Entities
```



```

lEnt = cTreeInfo.GetItemID(sEntLabel)
cEntities.GetDefaultValueID lEnt, lValId
'Get the label of the ID returned by GetDefaultValueID
Set cTreeInfo = m_cMetadata.Values
cTreeInfo.GetLabel lValId, sValLabel
'Insert the currency name in the label control
lblDefCurr.Caption = sValLabel

```

GetHoldingCompany

Returns the member ID of an entity's HoldingCompany attribute.

Syntax

```
<HsvEntities>.GetHoldingCompany lItemID, plHoldingCompanyEntityID
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the entity for which you want to get the HoldingCompany attribute.
<i>plHoldingCompanyEntityID</i>	Long. Returns the member ID of the entity specified as the HoldingCompany attribute, or -1 if the entity does not have a holding company.

Example

The following example creates a function that takes an entity's label and returns either the label of the entity's holding company or a blank string if the entity has no holding company. The member ID of the entity is obtained with `IHsvTreeInfo.GetItemID`. This ID is passed to `GetHoldingCompany`. If `GetHoldingCompany` returns -1, a blank string is assigned as `GetHoldCoLabel`'s return value. Otherwise, the ID returned by `GetHoldingCompany` is passed to `IHsvTreeInfo.GetLabel`, and the label returned is assigned as `GetHoldCoLabel`'s return value.

```

Private Function GetHoldCoLabel(sEnt As String) As String
Dim lEntID As Long, cEntities As HsvEntities, lHoldCoID As Long
Dim cTreeInfo As IHsvTreeInfo, sHoldCoLabel As String
Set cEntities = m_cMetadata.Entities
Set cTreeInfo = m_cMetadata.Entities
lEntID = cTreeInfo.GetItemID(sEnt)
cEntities.GetHoldingCompany lEntID, lHoldCoID
If lHoldCoID = -1 Then
    GetHoldCoLabel = ""
Else
    cTreeInfo.GetLabel lHoldCoID, sHoldCoLabel
    GetHoldCoLabel = sHoldCoLabel
End If
End Function

```

GetSecurityAsPartnerID

Returns the value assigned to the SecurityAsPartner attribute of an Entity dimension member.

Syntax

```
<HsvEntities>.GetSecurityAsPartnerID lItemID, plSecurityAsPartnerID
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the entity.
----------------	--

<i>plSecurityAsPartnerID</i>	Long. Returns the value assigned to the SecurityAsPartner attribute.
------------------------------	--

GetSecurityClassID

Returns the ID number of an entity's security class. In other words, this method returns the ID of the security class that has been defined for an entity's SecurityClass attribute.

Caution! This method is different than the identically named `GetSecurityClassID` method of the `HsvSecurityAccess` object. The `HsvSecurityAccess` method takes a security class name and returns the corresponding ID; for more information, see [“GetSecurityClassID” on page 472](#).

Syntax

```
<HsvEntities>.GetSecurityClassID lItemID, plSecurityClassID
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the entity.
----------------	--

<i>plSecurityClassID</i>	Long. Returns the ID number of the security class, or <code>SECURITY_CLASS_NONE</code> if a security class has not been assigned to the entity.
--------------------------	---

Tip: Get the label for the returned ID by passing the ID to `HsvSecurityAccess.GetSecurityClassLabel`. For more information, see [“GetSecurityClassLabel” on page 473](#).

Example

The following function takes an Entity member's label and returns the label of its security class.

```
Function getEntSecClass(sMem As String) As String
Dim lSecID As Long, lEntId As Long, cEntities As HsvEntities
Dim cTreeInfo As IHsvTreeInfo, cSecurity As HsvSecurityAccess
Dim sSecClass As String
'g_cMetadata is an HsvMetadata object reference
Set cEntities = g_cMetadata.Entities
Set cTreeInfo = g_cMetadata.Entities
'g_cSession is an HsvSession object reference
Set cSecurity = g_cSession.Security
lEntId = cTreeInfo.GetItemID(sMem)
cEntities.GetSecurityClassID lEntId, lSecID
'if the entity has a security class, return its label
If lSecID > -1 Then
    cSecurity.GetSecurityClassLabel lSecID, sSecClass
```

```

        getEntSecClass = sSecClass
    'if the entity has no security class, return a blank string
Else
    getEntSecClass = ""
End If
End Function

```

GetUserDefined1

Returns the value that has been defined for an entity's UserDefined1 attribute.

Syntax

```
<HsvEntities>.GetUserDefined1 lItemID, pbstrUserDefined
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the entity.
<i>pbstrUserDefined</i>	String. Returns the value that has been defined for the UserDefined1 attribute.

GetUserDefined2

Returns the value that has been defined for an entity's UserDefined2 attribute.

Syntax

```
<HsvEntities>.GetUserDefined2 lItemID, pbstrUserDefined
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the entity.
<i>pbstrUserDefined</i>	String. Returns the value that has been defined for the UserDefined2 attribute.

GetUserDefined3

Returns the value that has been defined for an entity's UserDefined3 attribute.

Syntax

```
<HsvEntities>.GetUserDefined3 lItemID, pbstrUserDefined
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the entity.
<i>pbstrUserDefined</i>	String. Returns the value that has been defined for the UserDefined3 attribute.

IsChild

Returns a Boolean that indicates whether an entity is a child of another entity.

Syntax

```
<HsvEntities>.IsChild lParentID, lEntityID, pbIsChild
```

Argument Description

lParentID Long (ByVal). The member ID of the possible parent entity.

lEntityID Long (ByVal). The member ID of the possible child entity.

pbIsChild Boolean. Returns TRUE if the *lEntityID* entity is a child of the *lParentID* entity, FALSE otherwise.

Example

The following example uses `IsChild` to see if the entity specified in the `comboParent` combo box is a parent of the entity specified in the `comboChild` combo box. If there is no parent-child relationship, the user is warned and the procedure is exited.

```
Dim bIsChild As Boolean, cEntities As HsvEntities
Dim cTreeInfo As IHsvTreeInfo, lParID As Long, lChildID As Long
Set cEntities = m_cMetadata.Entities
Set cTreeInfo = m_cMetadata.Entities
lParID = cTreeInfo.GetItemID(comboParent.Text)
lChildID = cTreeInfo.GetItemID(comboChild.Text)
cEntities.IsChild lParID, lChildID, bIsChild
If bIsChild = False Then
    MsgBox "The entities don't have a parent-child relationship!"
    Exit Sub
End If
```

IsDescendant

Returns a Boolean that indicates whether an entity is a descendant of another entity.

Syntax

```
<HsvEntities>.IsDescendant lParentID, lEntityID, pbIsDescendant
```

Argument Description

lParentID Long (ByVal). The member ID of the possible higher-level entity.

lEntityID Long (ByVal). The member ID of the possible descendant entity.

pbIsDescendant Boolean. Returns TRUE if the *lEntityID* argument's entity is a descendant of the *lParentID* argument's entity, FALSE if it is not a descendant.

Example

The following example creates a function that takes the labels of an entity and its possible descendant entity and returns a Boolean indicating whether there is an ancestor – descendant relationship.

```
Function IsEntityDescendant(sParent As String, sIsDescend _  
As String) As Boolean  
Dim cEntities As HsvEntities, cTreeInfo As IHsvTreeInfo  
Dim lParID As Long, lDescID As Long, bIsDesc As Boolean  
Set cEntities = m_cMetadata.Entities  
Set cTreeInfo = m_cMetadata.Entities  
lParID = cTreeInfo.GetItemID(sParent)  
lDescID = cTreeInfo.GetItemID(sIsDescend)  
cEntities.IsDescendant lParID, lDescID, bIsDesc  
IsEntityDescendant = bIsDesc  
End Function
```

IsICP

Returns a Boolean that indicates whether an entity is an intercompany entity. In other words, this method returns the value that has been defined for an entity's IsICP attribute.

Syntax

```
<HsvEntities>.IsICP lItemID, pbIsICP
```

Argument Description

lItemID Long (ByVal). The member ID of the entity.

pbIsICP Boolean. Returns TRUE if the entity is an intercompany entity, FALSE if it is not.

Example

The following example tests whether an entity specified in a combo box is an intercompany entity. If it is, any code placed within the If structure would be executed.

```
Dim cEntities As HsvEntities, cTreeInfo As IHsvTreeInfo  
Dim sEntity As String, lEntityID As Long, bIsICP As Boolean  
Set cEntities = m_cMetadata.Entities  
Set cTreeInfo = m_cMetadata.Entities  
sEntity = comboEnt.Text  
lEntityID = cTreeInfo.GetItemID(sEntity)  
cEntities.IsICP lEntityID, bIsICP  
If bIsICP = True Then  
    ...  
End If
```

IsOrgByPeriodFilteringOn

Returns a Boolean that indicates whether an entity has organization by period filtering enabled.

Syntax

```
<HsvEntities>.IsOrgByPeriodFilteringOn pvarbFilteringOn, plScenario,  
plYearID, plPeriodID
```

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

<i>pvarbFilteringOn</i>	Boolean. Returns TRUE if the entity has organization by period filtering enabled, otherwise FALSE.
-------------------------	--

<i>plScenario</i>	Long. Returns the ID of the subcube's Scenario dimension member.
-------------------	--

<i>plYearID</i>	Long. Returns the ID of the subcube's Year dimension member.
-----------------	--

<i>plPeriodID</i>	Long. Returns the ID of the subcube's Period dimension member.
-------------------	--

IsSecurityAsPartnerEnabled

Indicates whether metadata security using entities' SecurityAsPartner attributes is enabled.

By default, metadata security is determined by the security classes assigned to dimension members. However, for the Entity dimension you can use the Web object model to enable and disable an alternate form of security that uses entities' SecurityAsPartner attributes, and not their SecurityClass attributes, to determine access rights. This security is applied with the HFMwMetadata object's `EnableOrDisableUseSecurityAsPartner` method; for more information, see the *Oracle Hyperion Financial Management, Fusion Edition Web Developer's Guide*.

Note: To get the member ID of an Entity dimension member's SecurityAsPartner attribute, use [GetSecurityAsPartnerID](#).

Syntax

```
<HsvEntities>.IsSecurityAsPartnerEnabled pvarbEnabled
```

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

<i>pvarbEnabled</i>	Boolean. Indicates whether SecurityAsPartner security is enabled. Returns TRUE if enabled, FALSE if disabled.
---------------------	---

HsvICPs Object Methods

The HsvICPs object's methods return attributes of Intercompany Partner dimension members. These methods are described in detail in the following topics.

Assign HsvICPs object references with the ICP's property of the HsvMetadata object as shown in the following example:

```
Dim cMetadata as HsvMetadata, cICPs as HsvICPs  
Set cMetadata = m_cSession.Metadata  
Set cICPs = cMetadata.ICPs
```

GetSecurityClassID

Returns the ID of the security class that has been assigned to an Intercompany Partner, given the Intercompany Partner's member ID.

Syntax

```
<HsvICPs>.GetSecurityClassID lItemID, plSecurityClassID
```

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

<i>lItemID</i>	Long (ByVal). The Intercompany Partner's member ID.
----------------	---

<i>plSecurityClassID</i>	Long. Returns the ID of the Intercompany Partner's security class, or SECURITY_CLASS_NONE if a security class has not been assigned to the Intercompany Partner.
--------------------------	--

Tip: To get the label of the security class, pass the ID returned by this argument to the HsvSecurityAccess object's GetSecurityClassLabel method.

Example

The following example creates a function that takes an Intercompany Partner's label and returns the label of the Intercompany Partner's security class, or a blank string if a security class has not been assigned to the Intercompany Partner. IHsvTreeInfo.GetItemID returns the Intercompany Partner's member ID, which is then passed to GetSecurityClassID. If GetSecurityClassID indicates that the Intercompany Partner has no security class, a blank string is assigned as the function's return value; otherwise, HsvSecurityAccess.GetSecurityClassLabel gets the security class's label, which is then assigned as the function's return value.

```
Function getICPsecLabel(sICPLabel As String) As String
    Dim cICPs As HsvICPs, cTreeInfo As IHsvTreeInfo
    Dim cSecurityAccess As HsvSecurityAccess
    Dim lICPID As Long, lSecID As Long, sSecLabel As String
    Set cICPs = m_cMetadata.ICPs
    Set cTreeInfo = m_cMetadata.ICPs
    lICPID = cTreeInfo.GetItemID(sICPLabel)
    cICPs.GetSecurityClassID lICPID, lSecID
    If lSecID < 0 Then
        getICPsecLabel = ""
    Else
        Set cSecurityAccess = m_cSession.Security
        cSecurityAccess.GetSecurityClassLabel lSecID, sSecLabel
        getICPsecLabel = sSecLabel
    End If
End Function
```

GetSubmissionGroup

Returns the value of an Intercompany Partner dimension member's Submission Group property.

Syntax

```
<HsvICPs>.GetSubmissionGroup lItemID, plSubmissionGroup
```

Argument	Description
<i>lItemID</i>	Long. The member ID of the Intercompany Partner member.
<i>plSubmissionGroup</i>	Long. Returns the Submission Group value.

Example

GetSubmissionGroup is used in the example for [GetUseSubmissionPhaseFlag](#).

HsvPeriods Object Methods

Use the HsvPeriods object's methods to return the following Period dimension-related information:

- The ID of an application's base frequency.
- The ordinal position of a period within a frequency's periods.
- The number of periods in a frequency.
- The member ID of a period at a given ordinal position within a frequency.
- The member IDs of a frequency's Period dimension members.

The HsvPeriods object's methods are summarized in [Table 16 on page 63](#), and are described in detail in the following topics.

Note: Assign HsvPeriods object references with the `Periods` property of the HsvMetadata object as shown in the following example:

```
Dim cMetadata as HsvMetadata, cPeriods as HsvPeriods
Set cMetadata = m_cSession.Metadata
Set cPeriods = cMetadata.Periods
```

GetBaseFrequency

Returns the internal ID of the application's *base frequency*, which is the lowest-level frequency in the application. For example, if an application has frequencies of year-to-date, quarter-to-date, and periodic, the periodic frequency would be the base frequency.

Syntax

```
<HsvPeriods>.GetBaseFrequency plBaseFrequency
```


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

<i>plBaseFrequency</i>	Long. Returns the ID of the application's base frequency.
------------------------	---

Example

`GetBaseFrequency` is used in the [Example](#) for `GetPeriodsInFrequency`.

GetFrequency

Returns the ordinal position of a period within an application's base frequency, and also returns the ID of the base frequency.

Syntax

```
<HsvPeriods>.GetFrequency lPeriod, plFreq, plPeriodNumInFreq
```

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

<i>lPeriod</i>	Long (ByVal). The member ID of the period.
----------------	--

<i>plFreq</i>	Long. Returns the internal ID number of the application's base frequency.
---------------	---

<i>plPeriodNumInFreq</i>	Long. Returns the ordinal position of the period within the base frequency. This is a zero-based value, with 0 equalling the first period, 1 equalling the second period, and so on.
--------------------------	--

Example

The following example tests whether July is the first period in the base frequency. If so, then `GetFrequency`'s *plPeriodNumInFreq* argument returns 0, and any code placed within the `If` structure would be executed.

```
Dim cTreeInfo As IHsvTreeInfo, cPeriods As HsvPeriods
Dim lID As Long, lFreq As Long, lPerPosition As Long
Set cTreeInfo = m_cMetadata.Periods
Set cPeriods = m_cMetadata.Periods
lID = cTreeInfo.GetItemID("July")
cPeriods.GetFrequency lID, lFreq, lPerPosition
If lPerPosition = 0 Then
    ...
End If
```

GetNumPeriodsInFrequency

Returns the number of periods in a frequency.

Syntax

```
<HsvPeriods>.GetNumPeriodsInFrequency lFrequency, plNumPeriods
```

Argument	Description
<i>IFrequency</i>	Long (ByVal). The ID of the frequency. To use the application's base frequency, pass the ID returned by <code>GetBaseFrequency</code> .
<i>plNumPeriods</i>	Long. Returns the number of periods in the frequency.

Example

The following example prints the number of periods in the application's base frequency to Visual Basic's Immediate window. `GetBaseFrequency` returns the ID of the application's base frequency, and this ID is passed to `GetNumPeriodsInFrequency`.

```
Dim cPeriods As HsvPeriods, lFreqID As Long, lNumPeriods As Long
Set cPeriods = m_cMetadata.Periods
cPeriods.GetBaseFrequency lFreqID
cPeriods.GetNumPeriodsInFrequency lFreqID, lNumPeriods
Debug.Print CStr(lNumPeriods) & " periods in the base frequency."
```

GetPeriodFromFrequency

Returns the member ID of the period that is at a given ordinal position within a frequency.

Syntax

```
<HsvPeriods>.GetPeriodFromFrequency lFreq, lPeriodNumInFreq, plPeriod
```

Argument	Description
<i>lFreq</i>	Long (ByVal). The ID of the frequency. To use the application's base frequency, pass the ID returned by <code>GetBaseFrequency</code> .
<i>lPeriodNumInFreq</i>	Long (ByVal). The period's ordinal position within the frequency. This is zero-based, so pass 0 to get the member ID of the first period, 1 to get the member ID of the second period, and so on.
<i>plPeriod</i>	Long. The member ID of the period.

Example

The following example creates a function named `GetOrdinalPerName` that takes the ordinal position of a period and returns the period's label. In this function, `GetBaseFrequency` returns the ID of the application's default frequency. This ID is passed to `GetPeriodFromFrequency` along with the ordinal position. The member ID returned by `GetPeriodFromFrequency` is passed to `IHsvTreeInfo.GetLabel`, and the returned period label is set as `GetOrdinalPerName`'s return value.

```
Function GetOrdinalPerName(lPerPos As Long) As String
Dim cTreeInfo As IHsvTreeInfo, cPeriods As HsvPeriods
Dim lFreqID As Long, lPerID As Long, sPerName As String
Set cTreeInfo = m_cMetadata.Periods
Set cPeriods = m_cMetadata.Periods
cPeriods.GetBaseFrequency lFreqID
cPeriods.GetPeriodFromFrequency lFreqID, lPerPos, lPerID
```

```
cTreeInfo.GetLabel lPerID, sPerName
GetOrdinalPerName = sPerName
End Function
```

GetPeriodsInFrequency

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

Syntax

```
<HsvPeriods>.GetPeriodsInFrequency lFrequency, pvaralPeriodIDs
```

Argument Description

lFrequency Long (ByVal). The ID of the frequency. To use the application's base frequency, pass the ID returned by `GetBaseFrequency`.

pvaralPeriodIDs Variant array. Returns the member IDs of the periods in the frequency. The array is returned as a Long subtype.

Example

The following example creates a function named `CheckPeriodsInYear` that returns a count of the periods in an application's base frequency. `GetBaseFrequency` gets the ID of the base frequency; this ID is then passed to `GetPeriodsInFrequency`. The returned array of period IDs is passed to `UBound`; since this is a zero-based array, 1 is added to the number returned by `UBound` to calculate the count of periods, and the sum is set as `CheckPeriodsInYear`'s return value.

```
Function CheckPeriodsInYear() As Long
Dim cTreeInfo As IHsvTreeInfo, cPeriods As HsvPeriods
Dim lFreqID As Long, vaPeriods
Set cTreeInfo = m_cMetadata.Periods
Set cPeriods = m_cMetadata.Periods
cPeriods.GetBaseFrequency lFreqID
cPeriods.GetPeriodsInFrequency lFreqID, vaPeriods
CheckPeriodsInYear = UBound(vaPeriods) + 1
End Function
```

HsvScenarios Object Methods

The `HsvScenarios` object's methods return attributes of Scenario dimension members. These methods are summarized in [Table 17 on page 63](#), and are described in detail in the following topics.

Assign `HsvScenarios` object references with the `Scenarios` property of the `HsvMetadata` object as shown in the following example:

```
Dim cMetadata as HsvMetadata, cScenarios as HsvScenarios
Set cMetadata = m_cSession.MetaData
Set cScenarios = cMetadata.Scenarios
```

GetPhasedSubmissionStartYear

Returns the Phased Submission start year for the scenario.

Syntax

```
<HsvScenarios>.GetPhasedSubmissionStartYear lItemID, plStartYear
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the scenario.
----------------	--

<i>plStartYear</i>	Long. Returns the start year.
--------------------	-------------------------------

EnumPhasedSubmissionStartYears

Returns and array of Scenario IDs and their corresponding Phased Submission start years.

Syntax

```
<HsvScenarios>.EnumPhasedSubmissionStartYears pvaralScenarioIds,  
pvaralStartYears
```

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

<i>pvaralScenarioIds</i>	Variant array. Returns an array of Scenario IDs. The array is returned as a Long subtype.
--------------------------	---

<i>pvaralStartYears</i>	Variant array. Returns an array of Phased Submission Start Years. The array is returned as a Long subtype.
-------------------------	--

GetDefaultFrequency

Returns the ID number of a Scenario dimension member's default frequency.

Syntax

```
<HsvScenarios>.GetDefaultFrequency lItemID, plFrequency
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the Scenario dimension member.
----------------	---

<i>plFrequency</i>	Long. Returns the default frequency's ID number.
--------------------	--

GetDefaultView

Returns the number that identifies the default view of a Scenario dimension member.

Syntax

```
<HsvScenarios>.GetDefaultView lItemID, plView
```

Argument Description

ItemID Long (ByVal). The member ID of the Scenario dimension member.

pView Long. Returns the number that identifies the scenario's default view.

Some View dimension members are hard-coded into Financial Management. These members are represented by the HFM type library constants listed in [“View Dimension Member Constants” on page 833](#).

Tip: To get the label for the returned ID, pass the ID to `IHsvTreeInfo.GetLabel` (for the View dimension). For more information, see [“GetLabel” on page 221](#).

Example

The following example populates a label control with the default view of the scenario selected in a Combo box control. Note how the ID returned by `GetDefaultView` is passed to `GetLabel`.

```
Dim cScenarios As HsvScenarios, cTreeInfo As IHsvTreeInfo
Dim lScen As Long, sCat As String, lDefView As Long
Dim sDefView As String
Set cScenarios = m_cMetadata.Scenarios
sCat = comboCat.Text
'Set the cTreeInfo variable to the Scenario dimension
Set cTreeInfo = m_cMetadata.Scenarios
lScen = cTreeInfo.GetItemID(sCat)
cScenarios.GetDefaultView lScen, lDefView
'Set the cTreeInfo variable to the View dimension
Set cTreeInfo = m_cMetadata.Views
cTreeInfo.GetLabel lDefView, sDefView
lblDefView.Caption = sDefView
```

GetMaximumReviewLevel

Returns the value assigned to the `MaximumReviewLevel` attribute of a Scenario dimension member.

Syntax

```
<HsvScenarios>.GetMaximumReviewLevel lItemID, psMaximumReviewLevel
```

Argument Description

ItemID Long (ByVal). The member ID of the scenario.

psMaximumReviewLevel Integer. Returns the value assigned to the `MaximumReviewLevel` attribute.

Example

The following function takes a Scenario member's name and returns the value assigned to the member's `MaximumReviewLevel` attribute.

```
Function getMaxLevelFromLabel(sScenario As String) As Integer
Dim cScenario As HsvScenarios, cTreeInfo As IHsvTreeInfo
Dim lScenID As Long, iMax As Integer
```

```

Set cScenario = m_cMetadata.Scenarios
Set cTreeInfo = m_cMetadata.Scenarios
lScenID = cTreeInfo.GetItemID(sScenario)
cScenario.GetMaximumReviewLevel lScenID, iMax
getMaxLevelFromLabel = iMax
End Function

```

GetMissingDataZeroViewForAdjValues

Returns the member ID of the View dimension member that has been assigned as the ZeroViewForAdj attribute of a given Scenario dimension member.

Syntax

```
<HsvScenarios>.GetMissingDataZeroViewForAdjValues lItemID, plView
```

Argument Description

lItemID Long (ByVal). The member ID of the Scenario dimension member.

plView Long. Returns the member ID of the View dimension member assigned as the ZeroViewForAdj attribute.

Example

The following function returns the label of the View dimension member assigned as a given Scenario member's ZeroViewForAdj attribute.

```

Function getLabelZeroViewAdj(sScenMemName As String) As String
Dim cTreeInfo As IHsvTreeInfo, lScenId As Long
Dim cScenarios As HsvScenarios, lViewID As Long
Dim sViewLabel As String
'm_cMetadata is an HsvMetadata object reference
Set cTreeInfo = m_cMetadata.Scenarios
Set cScenarios = m_cMetadata.Scenarios
lScenId = cTreeInfo.GetItemID(sScenMemName)
cScenarios.GetMissingDataZeroViewForAdjValues lScenId, lViewID
Set cTreeInfo = m_cMetadata.Views
cTreeInfo.GetLabel lViewID, sViewLabel
getLabelZeroViewAdj = sViewLabel
End Function

```

GetMissingDataZeroViewForNonAdjValues

Returns the member ID of the View dimension member that has been assigned as the ZeroViewForNonadj attribute of a given Scenario dimension member.

Syntax

```
<HsvScenarios>.GetMissingDataZeroViewForNonAdjValues lItemID, plView
```

Argument Description

<i>ItemID</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>pView</i>	Long. Returns the member ID of the View dimension member assigned as the ZeroViewForNonadj attribute.

Example

The following function returns the label of the View dimension member assigned as a given Scenario member's ZeroViewForNonadj attribute.

```
Function getLabelZeroViewNonadj(sScenMemName As String) As _
    String
Dim cTreeInfo As IHsvTreeInfo, lScenId As Long
Dim cScenarios As HsvScenarios, lViewID As Long
Dim sViewLabel As String
'm_cMetadata is an HsvMetadata object reference
Set cTreeInfo = m_cMetadata.Scenarios
Set cScenarios = m_cMetadata.Scenarios
lScenId = cTreeInfo.GetItemID(sScenMemName)
cScenarios.GetMissingDataZeroViewForNonAdjValues lScenId, _
    lViewID
Set cTreeInfo = m_cMetadata.Views
cTreeInfo.GetLabel lViewID, sViewLabel
getLabelZeroViewNonadj = sViewLabel
End Function
```

GetSecurityClassID

Returns the ID of the security class that has been assigned to a scenario, given the scenario's member ID.

Syntax

```
<HsvScenarios>.GetSecurityClassID lItemID, plSecurityClassID
```

Argument Description

<i>ItemID</i>	Long (ByVal). The scenario's member ID.
<i>plSecurityClassID</i>	Long. Returns the ID of the scenario's security class, or -1 if a security class has not been assigned to the scenario.

Tip: To get the label of the security class, pass the ID returned by this argument to the HsvSecurityAccess object's GetSecurityClassLabel method.

Example

The following example creates a function that takes a scenario's label and returns the label of the scenario's security class, or a blank string if a security class has not been assigned to the scenario. `IHsvTreeInfo.GetItemID` returns the scenario's member ID, which is then passed to `GetSecurityClassID`. If `GetSecurityClassID` indicates that the scenario has no security class, a blank string is assigned as the function's return value; otherwise,

`HsvSecurityAccess.GetSecurityClassLabel` gets the security class's label, which is then assigned as the function's return value.

```
Function getScenSecLabel(sScenLabel As String) As String
Dim cScenarios As HsvScenarios, cTreeInfo As IHsvTreeInfo
Dim cSecurityAccess As HsvSecurityAccess
Dim lScenID As Long, lSecID As Long, sSecLabel As String
Set cScenarios = m_cMetadata.Scenarios
Set cTreeInfo = m_cMetadata.Scenarios
lScenID = cTreeInfo.GetItemID(sScenLabel)
cScenarios.GetSecurityClassID lScenID, lSecID
If lSecID < 0 Then
    getScenSecLabel = ""
Else
    Set cSecurityAccess = m_cSession.Security
    cSecurityAccess.GetSecurityClassLabel lSecID, sSecLabel
    getScenSecLabel = sSecLabel
End If
End Function
```

GetUserDefined1

Returns the text stored in the `UserDefined1` attribute of a Scenario dimension member.

Syntax

```
<HsvScenarios>.GetUserDefined1 lItemID, pbstrUserDefined
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>pbstrUserDefined</i>	String. Returns the <code>UserDefined1</code> attribute's text.

Example

This example tests whether the `UserDefined1` attribute of a Scenario dimension member consists of the letter Y. The Scenario dimension member's label is specified in a combo box named `comboCat`. `GetItemID` returns this member's ID, which is then passed to `GetUserDefined1`. If `GetUserDefined1` returns the letter Y, then any code placed within the `If` structure would be executed.

```
Dim cScenarios As HsvScenarios, cTreeInfo As IHsvTreeInfo
Dim lScen As Long, sCatLabel As String, sUser As String
Set cScenarios = m_cMetadata.Scenarios
Set cTreeInfo = m_cMetadata.Scenarios
sCatLabel = comboCat.Text
lScen = cTreeInfo.GetItemID(sCatLabel)
cScenarios.GetUserDefined1 lScen, sUser
If Trim(sUser) = "Y" Then
    ...
End If
```


GetUserDefined2

Returns the text stored in the UserDefined2 attribute of a Scenario dimension member.

Syntax

```
<HsvScenarios>.GetUserDefined2 lItemID, pbstrUserDefined
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the Scenario dimension member.
----------------	---

<i>pbstrUserDefined</i>	String. Returns the UserDefined2 attribute's text.
-------------------------	--

Example

See the [Example](#) for GetUserDefined1, and replace GetUserDefined1 with GetUserDefined2.

GetUserDefined3

Returns the text stored in the UserDefined3 attribute of a Scenario dimension member.

Syntax

```
<HsvScenarios>.GetUserDefined3 lItemID, pbstrUserDefined
```

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

<i>lItemID</i>	Long (ByVal). The member ID of the Scenario dimension member.
----------------	---

<i>pbstrUserDefined</i>	String. Returns the UserDefined3 attribute's text.
-------------------------	--

Example

See the [Example](#) for GetUserDefined1, and replace GetUserDefined1 with GetUserDefined3.

IsConsolidateYTD

Indicates whether a given Scenario dimension member supports year-to-date consolidation.

Note: A Scenario member's ConsolidateYTD attribute determines whether the member supports year-to-date consolidation

Syntax

```
<HsvScenarios>.IsConsolidateYTD lItemID, pbConsolidateYTD
```

Argument	Description
<i>ItemID</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>pbConsolidateYTD</i>	Boolean. Indicates whether the member supports year-to-date consolidation. Returns TRUE if the member supports year-to-date consolidation, FALSE if it does not.

Example

The following function indicates whether a given Scenario dimension member supports year-to-date consolidation.

```
Function isScenYTD(sMemberName As String) As Boolean
Dim lMemId As Long, bYTD As Boolean
Dim cTreeInfo As IHsvTreeInfo, cScenario As HsvScenarios
'm_cMetadata is an HsvMetadata object reference
Set cTreeInfo = m_cMetadata.Scenarios
Set cScenario = m_cMetadata.Scenarios
lMemId = cTreeInfo.GetItemID(sMemberName)
cScenario.IsConsolidateYTD lMemId, bYTD
isScenYTD = bYTD
End Function
```

IsPhasedSubmissionEnabled

Indicates whether Phased Submissions is enabled for a given Scenario and Year.

Syntax

```
<HsvScenarios>.IsPhasedSubmissionEnabled lScenarioId, lYear, pvbEnabled
```

Argument	Description
<i>lScenarioId</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>pvbEnabled</i>	Boolean. Returns True if Phased Submissions is enabled, False otherwise.

SupportsAccountAllocations

For internal use.

SupportsEmailAlerting

Indicates whether E-mail alerting is enabled for a given Scenario dimension member.

Syntax

```
<HsvScenarios>.SupportsEmailAlerting lItemID, pvbSupportsAlerting
```

<i>Argument</i>	Description
<i>ItemID</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>pvbSupportsAlerting</i>	Boolean. Indicates whether the member supports E-mail alerting. Returns TRUE if the alerting is enabled, otherwise FALSE.

SupportsProcessFlow

Indicates whether Process Management is enabled for a given Scenario dimension member.

Syntax

```
<HsvScenarios>.SupportsProcessFlow lItemID, pvbSupports
```

Argument Description

<i>ItemID</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>pvbSupports</i>	Boolean. Returns TRUE if the scenario enables the Process Management feature, otherwise FALSE.

Example

The following example tests whether an application's Scenario dimension members enable the Process Management feature. `EnumAllMemberIDs` gets the member IDs of the Scenario dimension members. In the example's loop, each ID is passed to `SupportsProcessFlow`, and each member's label and the Boolean returned by `SupportsProcessFlow` are printed to the Immediate window.

```
Dim vaIDs, cTreeInfo As IHsvTreeInfo, bRet As Boolean
Dim cScenarios As HsvScenarios, sLabel As String
Set cTreeInfo = m_cMetadata.Scenarios
Set cScenarios = m_cMetadata.Scenarios
cTreeInfo.EnumAllMemberIDs vaIDs
For i = LBound(vaIDs) To UBound(vaIDs)
    cScenarios.SupportsProcessFlow vaIDs(i), bRet
    cTreeInfo.GetLabel vaIDs(i), sLabel
    Debug.Print sLabel & ": " & bRet
Next i
```

SupportsTargetSetting

For internal use.

SupportsTimePeriodAllocations

For internal use.

UsesLineItems

Indicates whether a Scenario dimension member supports line items. In other words, this method indicates the value assigned to a Scenario dimension member's UsesLineItems attribute.

Syntax

```
<HsvScenarios>.UsesLineItems lItemID, pvarbUsesLineItems
```

Argument	Description
<i>lItemID</i>	Long (ByVal). The member ID of the scenario.
<i>pvarbUsesLineItems</i>	Boolean. Returns TRUE if the scenario supports line items, FALSE otherwise.

HsvValues Object Methods

The HsvValues object's methods return information about an application's Value dimension members. These methods are summarized in [“HsvValues Object Overview” on page 64](#), and are described in detail in the following topics.

Assign HsvValues object references with the [Values](#) property of the HsvMetadata object as shown in the following example:

```
Dim cValues As HsvValues  
'g_cMetadata is a previously set HsvMetadata object reference  
Set cValues = g_cMetadata.Values
```

GetCurrencyIDFromValueID

Returns a currency ID for a given Value dimension member ID.

Note: A currency's ID differs from that of the member ID of the Value dimension member that Financial Management creates for the currency. Value member IDs are obtained with IHsvTreeInfo methods such as [GetItemID](#) and [EnumMembers](#), while currency IDs are obtained with HsvCurrencies methods such as [GetCurrencyID](#) and [EnumCurrencies](#).

Syntax

```
<HsvValues>.GetCurrencyIDFromValueID lValue, plCurrencyID
```

Argument	Description
<i>lValue</i>	Long (ByVal). The member ID of the Value dimension member.
<i>plCurrencyID</i>	Long. Returns the currency ID of the currency that corresponds to the specified Value dimension member.

GetValueIDFromCurrencyID

Returns the member ID of a Value dimension member that corresponds to a given currency.

Note: A currency's ID differs from that of the member ID of the Value dimension member that Financial Management creates for the currency. Value member IDs are obtained with IHsvTreeInfo methods such as [GetItemID](#) and [EnumMembers](#), while currency IDs are obtained with HsvCurrencies methods such as [GetCurrencyID](#) and [EnumCurrencies](#).

Syntax

```
<HsvValues>.GetValueIDFromCurrencyID lCurrencyID, plValue
```

Argument Description

lCurrencyID Long (ByVal). The currency's ID.

plValue Long. Returns the member ID of the Value dimension member that corresponds to the specified currency.

HsvYears Object Methods

The HsvYears object provides the [GetYearRange](#) method, which returns an application's valid range of years.

Assign HsvYears object references with the *Years* property of the HsvMetadata object as shown in the following example:

```
Dim cMetadata as HsvMetadata, cYears As HsvYears
Set cMetadata = m_cSession.Metadata
Set cYears = cMetadata.Years
```

GetYearRange

Indicates the range of years that is valid for an application. [GetYearRange](#) returns the first and last years in the range of valid years.

Syntax

```
<HsvYears>.GetYearRange plStartYear, plEndYear
```

Argument Description

plStartYear Long. Returns the first year.

plEndYear Long. Returns the last year.

Example

This example creates a function that indicates whether a year is valid for an application. The function takes the year, then compares the year to the years returned by [GetYearRange](#). If the

year is less than the first valid year or greater than the last valid year, the year is not valid for the application and the function returns FALSE. Otherwise, the function returns TRUE.

```
Private Function IsYearInRange(lYYYY As Long) As Boolean
Dim cYears As HsvYears, lStart As Long, lEnd As Long
Set cYears = m_cMetadata.Years
cYears.GetYearRange lStart, lEnd
If lYYYY < lStart Or lYYYY > lEnd Then
    IsYearInRange = False
Else
    IsYearInRange = True
End If
End Function
```

HsvCurrencies Object Methods

The HsvCurrencies object's methods return information about an application's currencies. These methods are summarized in "[HsvCurrencies Object Overview](#)" on page 68, and are described in detail in the following topics.

A currency is identified by a *currency ID*. A currency's ID differs from that of the member ID of the Value dimension member that Financial Management creates for the currency. Currency IDs are obtained with HsvCurrencies methods such as [GetCurrencyID](#) and [EnumCurrencies](#), while Value member IDs are obtained with IHsvTreeInfo methods such as [GetItemID](#) and [EnumMembers](#).

Tip: The HsvValues object provides methods for obtaining a currency ID from a Value dimension member ID and a member ID from a currency ID. For more information, see "[HsvValues Object Methods](#)" on page 268.

Assign HsvCurrencies object references with the [Currencies](#) property of the HsvMetadata object as shown in the following example:

```
Dim cCurrencies As HsvCurrencies
'g_cMetadata is a previously set HsvMetadata object reference
Set cCurrencies = g_cMetadata.Currencies
```

EnumCurrencies

Returns arrays containing the currency IDs and labels of the application's currencies. The arrays have a one-to-one correspondence.

Tip: To return only currencies for which the DisplayInICT attribute is enabled, use [EnumCurrencies2](#).

Syntax

```
<HsvCurrencies>.EnumCurrencies pvaralCurrencyIDs, pvarabstrCurrencyLabels
```

Argument	Description
<i>pvaralCurrencyIDs</i>	Variant. Returns an array of currency IDs. The array is returned as a Long subtype.
<i>pvarabstrCurrencyLabels</i>	Variant. Returns an array of currency labels. The array is returned as a String subtype.

Example

The following example prints the labels of an application's currencies to Visual Basic's Immediate window.

```
Dim cCurrencies As HsvCurrencies, vaIDs As Variant, vaLabels As Variant
'g_cMetadata is an HsvMetadata object reference
Set cCurrencies = g_cMetadata.Currencies
cCurrencies.EnumCurrencies vaIDs, vaLabels
For i = LBound(vaLabels) To UBound(vaLabels)
    Debug.Print vaLabels(i)
Next i
```

EnumCurrencies2

Returns arrays containing the currency IDs and labels 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.

Syntax

```
<HsvCurrencies>.EnumCurrencies2 vbICTOnly, pvaralIDs, pvarabstrLabels
```

Argument	Description
<i>vbICTOnly</i>	Boolean (ByVal). Specifies whether to filter currencies. Pass TRUE to return only currencies for which the DisplayInICT attribute is enabled, FALSE to return all currencies.
<i>pvaralIDs</i>	Variant. Returns an array of currency IDs. The array is returned as a Long subtype.
<i>pvarabstrLabels</i>	Variant. Returns an array of currency labels. The array is returned as a String subtype.

Example

The following example prints the labels of the currencies for which the DisplayInICT attribute is enabled to Visual Basic's Immediate window.

```
Dim cCurrencies As HsvCurrencies, vaIDs As Variant, vaLabels As Variant
'g_cMetadata is an HsvMetadata object reference
Set cCurrencies = g_cMetadata.Currencies
cCurrencies.EnumCurrencies2 True, vaIDs, vaLabels
For i = LBound(vaLabels) To UBound(vaLabels)
    Debug.Print vaLabels(i)
Next i
End Sub
```

GetCurrencyDescription

Returns a currency's description in a given language. The language must be valid.

Tip: `HsvMetadata.EnumLanguages` returns labels and IDs of the valid languages for the application.

Syntax

```
<HsvCurrencies>.GetCurrencyDescription(lCurrencyID, lLanguageID)
```

Argument Description

lCurrencyID Long (ByVal). The ID of the currency.

You can get a currency ID with [GetCurrencyID](#) and [EnumCurrencies](#).

lLanguageID Long (ByVal). The ID of the language. You can get language IDs with `HsvMetadata.EnumLanguages`.

Return Value

String. Returns the currency's description.

Example

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

GetCurrencyID

Returns the currency ID of a given currency.

Syntax

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

Argument Description

bstrLabel String (ByVal). The currency's label.

Return Value

Long. Returns the currency ID.

Example

The following function returns the label of a currency for a given language. The currency ID obtained by `GetCurrencyID` is passed to [GetCurrencyDescription](#) along with the language ID obtained by `HsvMetadata.EnumLanguages`.

```
Function getCurrencyDesc(sCurr As String, sLang As String) As String  
Dim cCurrencies As HsvCurrencies, lCurrID As Long
```



```

Dim vaLangIDs As Variant, vaLangDescs As Variant
'g_cMetadata is a previously set HsvMetadata object reference
Set cCurrencies = g_cMetadata.Currencies
g_cMetadata.EnumLanguages vaLangIDs, vaLangDescs
For i = LBound(vaLangDescs) To UBound(vaLangDescs)
    If vaLangDescs(i) = sLang Then
        lCurrID = cCurrencies.GetCurrencyID(sCurr)
        getCurrencyDesc = cCurrencies.getCurrencyDescription(lCurrID, _
            vaLangIDs(i))
        Exit Function
    End If
Next i
End Function

```

GetCurrencyLabel

Returns a currency label, given a currency ID.

Syntax

```
<HsvCurrencies>.GetCurrencyLabel (lCurrencyID)
```

Argument Description

lCurrencyID Long (ByVal). The currency ID.

Return Value

String. Returns the currency's label.

GetCurrencyTranslationOperator

Returns the conversion operator for a given currency.

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

Syntax

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

Argument Description

lCurrencyID Long (ByVal). The currency's ID.

Return Value

String. Returns the conversion operator.

Example

The following function takes a currency's label and returns its conversion operator. The currency's ID is obtained with [GetCurrencyID](#).

```
Function getCurrencyOp(sCurr As String) As String
Dim cCurrencies As HsvCurrencies, lCurrID As Long
'g_cMetadata is a previously set HsvMetadata object reference
Set cCurrencies = g_cMetadata.Currencies
lCurrID = cCurrencies.GetCurrencyID(sCurr)
getCurrencyOp = cCurrencies.GetCurrencyTranslationOperator(lCurrID)
End Function
```

GetScale

Returns the scale of a given currency.

Syntax

```
<HsvCurrencies>.GetScale(lCurrencyID)
```

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

<i>lCurrencyID</i>	Long (ByVal). The currency's ID.
--------------------	----------------------------------

Return Value

Integer. Returns the currency's scale.

Example

The following function takes a currency's label and returns its scale. The currency's ID is obtained with [GetCurrencyID](#).

```
Function getCurrencyScale(sCurr As String) As Integer
Dim cCurrencies As HsvCurrencies, lCurrID As Long
'g_cMetadata is a previously set HsvMetadata object reference
Set cCurrencies = g_cMetadata.Currencies
lCurrID = cCurrencies.GetCurrencyID(sCurr)
getCurrencyScale = cCurrencies.GetScale(lCurrID)
End Function
```

9

HsvData Type Library

In This Chapter

About Cell Statuses.....	275
Update Mode Constants	276
HsvData Object Methods.....	276

This chapter describes the members of the HsvData type library. The methods of this type library are used to get and set data in cells. The chapter also explains the cell status codes that some HsvData methods return, as well as constants that represent data update modes.

To use the HsvData type library, you must reference `HsvData.dll` in your project. The HsvData type library contains the HsvData object.

Note: The HsvData type library also contains the `IHsvDataValidation` interface and `HsvDataPipe` object, but both are for internal use.

About Cell Statuses

Several HsvData 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 842](#).
- 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 843](#).
- 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 842](#).

The applicable HsvData 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 Visual Basic, 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 returned by an `HsvData` method has been assigned to the `lStat` variable:

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

Update Mode Constants

The following constants represent update modes. The update mode determines how existing data in the application cells is handled during operations such as loading data and copying data.

Table 59 Enum `HSV_DATA_UPDATE_MODE` Constants

Constant	Description
<code>HSV_DATA_UPDATE_ACCUMULATE</code>	Accumulate mode: if a cell contains data, the data in the cell to be copied or loaded is added to the existing data.
<code>HSV_DATA_UPDATE_MERGE</code>	Merge mode: if a cell contains data, and the corresponding cell to be loaded or copied also contains data, the existing data is replaced. However, if the cell to be loaded or copied does not contain data, then the existing data is preserved.
<code>HSV_DATA_UPDATE_REPLACE</code>	Replace mode: existing data in all specified cells is cleared, then the data is loaded or copied. However, if the connected user does not have full access rights to all specified cells, none of the data is changed.
<code>HSV_DATA_UPDATE_REPLACEWITHSECURITY</code>	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 or loaded. Cells to which the user lacks full access are ignored.

HsvData Object Methods

The `HsvData` object contains several methods. The methods enable you to perform numerous data-related operations, including the following operations:

- Return, set, and clear data for one or more cells
- Return, set, and clear cell text for one or more cells
- Return and set a cell's line items
- Return a cell's translation, metadata, and calculation statuses

These methods are summarized in [Table 21 on page 69](#), and are described in detail in the following topics.

Assign `HsvData` object references with the `Data` property of the `HsvSession` object. For an example, see [“HsvData Type Library Overview” on page 69](#).

Tip: Many of the `HsvData` object's methods use dimension member IDs to identify cells. For information on how to get member IDs, see [“About Member IDs” on page 161](#).

AddDataToMDDataBuffer

Adds an application's cell to an HsvMDDataBuffer or HsvMDDataBufferLite object. Any data, description, or line items for the cell will be added.

Syntax

```
<HsvData>.AddDataToMDDataBuffer lScenario, lYear, lPeriod, lView, lEntity,  
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,  
varbIncludeDerivedData, pIUnkDataBuffer
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>varbIncludeDerivedData</i>	Boolean (ByVal). Determines whether derived data is included. Pass TRUE to include derived data, FALSE otherwise.
<i>pIUnkDataBuffer</i>	HsvMDDataBuffer or HsvMDDataBufferLite object (ByVal). The object reference for the object to which the cell is being added.

Example

The following example adds cells to an HsvMDDataBuffer object. The member IDs for all dimension members other than the Account dimension are passed from another procedure. `IHsvTreeInfo.EnumAllMemberIDs` gets the member IDs of the application's accounts, and the first `For...Next` loop uses `HsvMetadata.IsCustomMemberValidForAccount` to identify the accounts that have valid intersections for the Custom dimension members passed from the other procedure. Cells for the valid intersections are then added to an HsvMDDataBuffer object with `AddDataToMDDataBuffer`.

```
Dim cHsvMDDataBuffer As HsvMDDataBuffer, iMemsCount As Integer
```

```

Dim vaMemIDs, bCust1Valid As Boolean, bCust2Valid As Boolean
Dim bCust3Valid As Boolean, bCust4Valid As Boolean
Set cHsvMDBuffer = New HSVMDARRAYSLib.HsvMDDataBuffer
'Get the IDs of all the Account dimension members.
m_cIHsvTreeInfo.EnumAllMemberIDs vaMemIDs
'Add the cells for accounts that are valid for the
'custom dimensions to an HsvMDDataBuffer object.
For iMemsCount = LBound(vaMemIDs) To UBound(vaMemIDs)
    m_cHsvMetadata.IsCustomMemberValidForAccount 8, m_lCust1, _
    vaMemIDs(iMemsCount), bCust1Valid
    m_cHsvMetadata.IsCustomMemberValidForAccount 9, m_lCust2, _
    vaMemIDs(iMemsCount), bCust2Valid
    m_cHsvMetadata.IsCustomMemberValidForAccount 10, m_lCust3, _
    vaMemIDs(iMemsCount), bCust3Valid
    m_cHsvMetadata.IsCustomMemberValidForAccount 11, m_lCust4, _
    vaMemIDs(iMemsCount), bCust4Valid
    If bCust1Valid = True And bCust2Valid = True And _
    bCust3Valid = True And bCust4Valid = True Then
        m_cHsvData.AddDataToMDDDataBuffer m_lScen, m_lYear, m_lPer, _
        m_lView, m_lEnt, m_lPar, m_lVal, _
        vaMemIDs(iMemsCount), m_lICP, m_lCust1, m_lCust2, _
        m_lCust3, m_lCust4, FALSE, cHsvMDBuffer
    End If
Next iMemsCount

```

AttachDocumentToCell

Attaches a previously loaded document to a cell.

Tip: To detach a document to a cell, use [DetachDocumentFromCell](#).

Syntax

```

<HsvData>.AttachDocumentToCell lScenario, lYear, lPeriod, lEntity, lParent,
lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
bstrDocPath, bstrDocFile

```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.

Argument Description

<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>bstrDocPath</i>	String (ByVal). The path in which the document has been loaded. Folders in the path are delimited by backslashes (\).
<i>bstrDocFile</i>	String (ByVal). The name of the file to attach. Caution! This file must be loaded in the folder specified by the <i>bstrDocPath</i> argument, otherwise an error will be thrown.

ClearAllData

Deletes all data from an application.

Note: To delete an application's data, the user must be assigned an Administrator role for the application.

Syntax

```
<HsvData>.ClearAllData
```

Example

This example clears the data in the application that was opened when the `m_cHsvSession` object variable (for the `HsvSession` object) was set.

```
Dim cHsvData as HsvData  
Set cHsvData = m_cHsvSession.Data  
cHsvData.ClearAllData
```

Tip: `HsvClient.OpenApplication` returns an `HsvSession` object reference. For more information, see [“OpenApplication” on page 140](#).

ClearAllDescriptionsInSubCube

Removes the cell text from all the cells in a subcube. The subcube is identified by `ClearAllDescriptionsInSubCube`'s arguments. (For information on subcubes, see [“About Subcubes” on page 43](#).)

Syntax

```
<HsvData>.ClearAllDescriptionsInSubCube lScenario, lYear, lPeriod,  
lEntity, lParent, lValue
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.

Example

The following example clears the cell text in all cells contained by a subcube. The subcube is identified by the member IDs passed in the calls to the user-defined `GetMemberID` function; for details on `GetMemberID`, see the [Examples](#) for `GetItemID`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
m_cData.ClearAllDescriptionsInSubCube lScen, lYear, lPer, _
lEnt, lPar, lVal
```

ClearDataAuditItems

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

Syntax

```
<HsvData>.ClearDataAuditItems dTimeToClearBefore
```

Argument Description

<i>dTimeToClearBefore</i>	Double (ByVal). The timestamp that represents the date and time. This must be a Double that can be cast to a Date format.
---------------------------	---

Example

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

```
Dim cData As HsvData
'm_cHsvSession is an HsvSession object reference
Set cData = m_cHsvSession.Data
cData.ClearDataAuditItems CDbl(Now)
```


ClearInputData

Deletes data from a subcube's cells.

Note: For a description of subcubes, see [“About Subcubes” on page 43](#).

The Boolean *vbClearInputValueID* and *vbClearNoneValueID* arguments enable you to delete data from the subcube cells for currency-related Value dimension members and for the [None] Value dimension member. For details on the different types of Value members, see [Currency-Related Value Dimension Members](#).

The *varlaAccountSubsetIncludeList* and *vbUseAccountSubsetIncludeList* arguments enable you to delete data for only cells that intersect specified accounts. Conversely, the *varlaAccountSubsetExcludeList* and *vbUseAccountSubsetExcludeList* arguments enable you to delete data for all cells other than those that intersect specified accounts.

Tip: To clear subcube cells for all accounts, set the *vbUseAccountSubsetIncludeList* and *vbUseAccountSubsetExcludeList* arguments to FALSE.

Syntax

```
<HsvData>.ClearInputData lScenario, lYear, lEntity, lParent,  
vbClearInputValueID, vbClearNoneValueID, vbClearNodeLevelValueID,  
varlaPeriodSubset, varlaAccountSubsetIncludeList,  
vbUseAccountSubsetIncludeList, varlaAccountSubsetExcludeList,  
vbUseAccountSubsetExcludeList, pvbDataExistedPriorToClear
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the subcube's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the subcube's Year dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the subcube's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>vbClearInputValueID</i>	Boolean (ByVal). Specifies whether cells for the subcube's currency-related Value dimension members are to be cleared. Pass TRUE to clear these cells, otherwise FALSE.
<i>vbClearNoneValueID</i>	Boolean (ByVal). Specifies whether cells for the subcube's [None] Value dimension member are to be cleared. Pass TRUE to clear these cells, otherwise FALSE. Note: [None] Value dimension members are used with system accounts and currency rate accounts.
<i>vbClearNodeLevelValueID</i>	Boolean (ByVal). This argument is obsolete; you must pass a Boolean, but the value passed has no effect.
<i>varlaPeriodSubset</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.

Argument	Description
<i>varlaAccountSubsetIncludeList</i>	Long array (ByVal). If you set the <i>vbUseAccountSubsetIncludeList</i> argument to TRUE, only subcube cells that intersect the accounts specified are cleared. Pass the member IDs of the Account dimension members for the cells to be cleared. Note: This argument is ignored if you set <i>vbUseAccountSubsetIncludeList</i> to FALSE.
<i>vbUseAccountSubsetIncludeList</i>	Boolean (ByVal). Specifies whether the <i>varlaAccountSubsetIncludeList</i> argument is used or ignored. Pass TRUE to use it, FALSE to ignore it.
<i>varlaAccountSubsetExcludeList</i>	Long array (ByVal). If you set the <i>vbUseAccountSubsetExcludeList</i> argument to TRUE, all subcube cells other than those that intersect the accounts specified are cleared. Pass the member IDs of the Account dimension members for the cells that should not be cleared. Note: This argument is ignored if you set <i>vbUseAccountSubsetExcludeList</i> to FALSE.
<i>vbUseAccountSubsetExcludeList</i>	Boolean (ByVal). Specifies whether the <i>varlaAccountSubsetExcludeList</i> argument is used or ignored. Pass TRUE to use it, FALSE to ignore it.
<i>pvbDataExistedPriorToClear</i>	Boolean. Indicates whether any of the specified cells contained data before <i>ClearInputData</i> was called. Returns TRUE if one or more of the cells contained data, FALSE otherwise.

Currency-Related Value Dimension Members

To illustrate the Value dimension members that are affected by *ClearInputData*'s *vbClearInputValueID* argument and *CopyInputDataForMultipleEntities*' *vbCopyEntityCurrencyValueID* argument, the following figure shows an application's Value dimension members listed in hierarchical order. The members from <Parent Curr Total> down to USD are the currency-related members affected by these arguments.

Figure 1 Types of Value Dimension Members

[None]
[Contribution Total]
[Contribution Adjs]
[Contribution]
[Elimination]
[Proportion]
[Parent Total]
[Parent Adjs]
[Parent]
<Parent Curr Total>
<Parent Curr Adjs>
<Parent Currency>
<Entity Curr Total>
<Entity Curr Adjs>
<Entity Currency>
EURO Total
EURO Adjs
EURO
USD Total
USD Adjs
USD

ClearInvalidData

Scans for or deletes invalid records.

Syntax

```
<HsvData>.ClearInvalidData vbScanOnly, lOptions, bstrLogFileName
```

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

<i>vbScanOnly</i>	Boolean (ByVal). A flag indicating whether to scan for or delete invalid records. Pass TRUE to scan, FALSE to delete.
-------------------	---

<i>lOptions</i>	Long (ByVal). This argument's value is ignored in the current release. You must pass a valid Long.
-----------------	--

<i>bstrLogFileName</i>	String (ByVal). The name and path of the log file for the operation.
------------------------	--

CopyInputData

Deprecated – use [CopyInputDataForMultipleEntities](#).

CopyInputDataForMultipleEntities

Copies data for one or more entities from one set of cells to another set of cells.

`CopyInputDataForMultipleEntities` enables you to multiply the source cells' amounts by a factor before inserting the amounts in the target cells, and to copy the source cells' cell text descriptions to the target cells.

Syntax

```
<HsvData>.CopyInputDataForMultipleEntities lView, lSourceScenario,  
lDestScenario, lSourceYear, lDestYear, varlaSourcePeriodSubset,  
varlaDestPeriodSubset, varlaEntities, varlaParents,  
vbCopyEntitiyCurrencyValueID, vbCopyNoneValueID, varlaAccountSubset,  
dFactor, vbCopyCellText, vbCopyDerived, vbIgnoreWarnings, lEnumUpdateMode,  
bstrLogFileName, vbEnableDetailedLogging
```

<i>Argument</i>	Description
<i>lView</i>	Long (ByVal). The member ID of the source and target cells' View dimension member.
<i>lSourceScenario</i>	Long (ByVal). The member ID of the Scenario dimension member for the source cells.
<i>lDestScenario</i>	Long (ByVal). The member ID of the Scenario dimension member for the target cells.
<i>lSourceYear</i>	Long (ByVal). The member ID of the Year dimension member for the source cells.
<i>lDestYear</i>	Long (ByVal). The member ID of the Year dimension member for the target cells.
<i>varlaSourcePeriodSubset</i>	Long array (ByVal). The member IDs of the Period dimension members for the source cells.
<i>varlaDestPeriodSubset</i>	Long array (ByVal). The member IDs of the Period dimension members for the target cells.
<i>varlaEntities</i>	Long array (ByVal). The member IDs of the source and target cells' Entity dimension members.
<i>varlaParents</i>	Long array (ByVal). The member IDs of the parents of the <i>varlaEntities</i> argument's entities.
<i>vbCopyEntitiyCurrencyValueID</i>	Boolean (ByVal). 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, otherwise FALSE. Tip: For an explanation of the different types of Value dimension members, see Currency-Related Value Dimension Members .
<i>vbCopyNoneValueID</i>	Boolean (ByVal). 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, otherwise FALSE.
<i>varlaAccountSubset</i>	Long array (ByVal). The member IDs of the source and target cells' Account dimension members.
<i>dFactor</i>	Double (ByVal). 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.

<i>vbCopyCellText</i>	Boolean (ByVal). Specifies whether the source cells' cell text descriptions are to be copied to the target cells. Pass TRUE to include the cell text descriptions, otherwise FALSE.
<i>vbCopyDerived</i>	Boolean (ByVal). Specifies whether derived source data is to be copied to the destination cells as stored input. Pass TRUE to copy derived source data, otherwise FALSE.
<i>vbIgnoreWarnings</i>	Boolean (ByVal). Specifies whether data will be copied if the system generates a warning-level error. Pass TRUE to copy data if warnings occur, otherwise FALSE. Note: Warnings will be included in the log file regardless of whether you specify TRUE or FALSE.
<i>lEnumUpdateMode</i>	Long (ByVal). Determines the update mode. For more information and valid values, see "Update Mode Constants" on page 276 .
<i>bstrLogFile</i>	String (ByVal). The name and path of the log file in which the system will include information regarding the copy data operation.
<i>vbEnableDetailedLogging</i>	Boolean (ByVal). Specifies whether the log file includes details such as the names of the source and destination dimension members. Pass TRUE to include such details, FALSE otherwise.

Example

The following example creates a subroutine that copies data from one year's cells to another. The subroutine takes IDs of the cells' dimension members. Note that the scenarios and years passed to this subroutine are used by `CopyInputDataForMultipleEntities` as both sources and targets, and that the `lEnumUpdateMode` argument sets `CopyInputDataForMultipleEntities` to replace mode.

```
Sub CopyCellsYearReplace(lView, lScen, lSrcYear, _
    lDestYear, laPer, laEnts, laPars, laAccts)
    Dim cHsvData As HsvData
    Set cHsvData = m_cHsvSession.Data
    cHsvData.CopyInputDataForMultipleEntities lView, lScen, lScen, _
        lSrcYear, lDestYear, laPer, laPer, laEnts, laPars, True, _
        True, laAccts, 1, True, True, True, _
        HSV_DATA_UPDATE_REPLACE, "c:\temp\CopyLog.txt", False
End Sub
```

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

```
<HsvData>.DeleteLineItemDetails lScenario, lYear, lEntity, lAccount, lICP,
    lCustom1, lCustom2, lCustom3, lCustom4, varabstrDetail
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member for the intersecting cells.
------------------	--

Argument	Description
<i>IYear</i>	Long (ByVal). The member ID of the Year dimension member for the intersecting cells.
<i>IEntity</i>	Long (ByVal). The member ID of the Entity dimension member for the intersecting cells.
<i>IAccount</i>	Long (ByVal). The member ID of the Account dimension member for the intersecting cells. Tip: Use <code>HsvAccounts.UsesLineItems</code> to check whether an account supports line items.
<i>IICP</i>	Long (ByVal). The member ID of the Intercompany Partner dimension member for the intersecting cells.
<i>ICustom1</i>	Long (ByVal). The member ID of the Custom 1 dimension member for the intersecting cells.
<i>ICustom2</i>	Long (ByVal). The member ID of the Custom 2 dimension member for the intersecting cells.
<i>ICustom3</i>	Long (ByVal). The member ID of the Custom 3 dimension member for the intersecting cells.
<i>ICustom4</i>	Long (ByVal). The member ID of the Custom 4 dimension member for the intersecting cells.
<i>varabstrDetail</i>	String array (ByVal). An array of 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.

Example

The following example deletes two line items. The example assumes the variables passed to `DeleteLineItemDetails` have been set to the member IDs for the intersecting cells.

```
Dim saDesc(1) As String
saDesc(0) = "Chairs"
saDesc(1) = "Desks"
m_chsvData.DeleteLineItemDetails lScen, lYr, lEnt, lAcct, _
lIcp, lCust1, lCust2, lCust3, lCust4, saDesc
```

DetachDocumentFromCell

Detaches a document from a cell.

Tip: To attach a document to a cell, use [AttachDocumentToCell](#). To get the names and paths of the documents attached to a cell, use [GetAttachedDocumentsToCell](#).

Syntax

```
<HsvData>.DetachDocumentFromCell lScenario, lYear, lPeriod, lEntity,
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
bstrDocPath, bstrDocFile
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.

Argument	Description
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>bstrDocPath</i>	String (ByVal). The path in which the document has been loaded. Folders in the path are delimited by backslashes (\).
<i>bstrDocFile</i>	String (ByVal). The name of the file to detach. Caution! This file must be loaded in the folder specified by the <i>bstrDocPath</i> argument, otherwise an error will be thrown.

Example

The following subroutine detaches all documents that are attached to a given cell. `GetAttachedDocumentsToCell` returns the names and paths of the documents to be detached.

```
Sub detachAllDocsFromCell(lScen As Long, lYear As Long, lPer As Long, _
    lEnt As Long, lPar As Long, lVal As Long, lAcct As Long, lIcp As _
    Long, lCust1 As Long, lCust2 As Long, lCust3 As Long, lCust4 As Long)
Dim vaPaths As Variant, vaFileNames As Variant
'g_cData is a previously set HsvData object reference
g_cData.GetAttachedDocumentsToCell lScen, lYear, lPer, lEnt, lPar, _
    lVal, lAcct, lIcp, lCust1, lCust2, lCust3, lCust4, vaPaths, vaFileNames
For i = LBound(vaPaths) To UBound(vaPaths)
    g_cData.DetachDocumentFromCell lScen, lYear, lPer, lEnt, lPar, lVal, _
        lAcct, lIcp, lCust1, lCust2, lCust3, lCust4, vaPaths(i), _
        vaFileNames(i)
Next i
End Sub
```

DMELoad

For internal use.

DoCellDescriptionsExist

Indicates whether one or more of the cells for a combination of Scenario, Year, and Entity dimension members contain cell text descriptions.

Syntax

```
<HsvData>.DoCellDescriptionsExist lScenario, lYear, lEntity, lParent, pvbExist
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the cells' Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cells' Year dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cells' Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>pvbExist</i>	Boolean. Returns TRUE if one or more of the cells contain cell text descriptions, otherwise FALSE.

DoesCellDescriptionExist

Indicates whether a cell contains cell text. The method's last argument takes a Boolean variable that indicates whether the cell specified by the other arguments contains cell text.

Note: Cell text is the text entered when a user right-clicks a Data Explorer cell and selects the Cell Text option.

Syntax

```
<HsvData>.DoesCellDescriptionExist lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, pvarbExists
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.

Argument Description

<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pvarbExists</i>	Boolean. Returns TRUE if the cell contains cell text, otherwise FALSE.

Example

This example shows how to call `DoesCellDescriptionExist` and then test the *pvarbExists* argument. If `DoesCellDescriptionExist` returns FALSE, any code placed in the `If` structure would be executed. The example assumes that the variables for the member ID arguments have been previously set in another procedure.

```
m_chsvData.DoesCellDescriptionExist lScen, lYr, lPer, lEnt, _  
lPar, lVal, lAcct, lIcp, lCus1, lCus2, lCus3, lCus4, bExists  
If bExists = False Then  
    ...  
End If
```

DoesDataExist

Indicates whether any data exists in the cells intersected by the Scenario, Year, and parent and child Entity dimension members specified in the arguments.

`DoesDataExist` returns FALSE if the specified *parent entity's* cells do not contain data, regardless of whether the child entity's cells contain data.

Syntax

```
<HsvData>.DoesDataExist lScenario, lYear, lEntity, lParent, pvbExist
```

Argument Description

<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>pvbExist</i>	Boolean. Returns TRUE if any of the cells contain data, otherwise FALSE.

Example

This example shows how to call `DoesDataExist` and then test the *pvbExist* argument. If `DoesDataExist` returns FALSE, any code placed in the `If` structure would be executed. The

example assumes that the variables for the member ID arguments have been previously set in another procedure.

```
m_chsvData.DoesDataExist lScen, lYr, lEnt, lPar, bExists  
If bExists = False Then  
...  
End If
```

DoesSparseDataExist

Indicates whether sparse data exists for the specified cell. A dimension that may not contain data for every combination of dimension members is considered sparse.

`DoesSparseDataExist` returns `FALSE` if the specified *parent entity*'s cells do not contain sparse data, regardless of whether the child entity's cells contain sparse data.

Syntax

```
<HsvData>.DoesSparseDataExist lScenario, lYear, lPeriod, lEntity, lParent,  
lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,  
pvbDataExists
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Parent dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pvbDataExists</i>	Boolean (ByVal). Boolean. Returns <code>TRUE</code> if the cell contains sparse data, otherwise <code>FALSE</code> .

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.

The *lStartRecord* and *lEndRecord* arguments specify the starting and ending indexes of the range of records, and the *plTotalNumRecords* 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 *plTotalNumRecords* to loop through the remaining records.

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

To get the history of a given cell, 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

```
<HsvData>.EnumDataAuditItems dStartTime, dEndTime, vbAllServers,  
bstrServer, vbAllUsers, bstrUserName, lStartRecord, lEndRecord,  
pvaralScenarios, pvaralYear, pvaralPeriod, pvaralEntity, pvaralParent,  
pvaralValue, pvaralICP, pvaralAccount, pvaralCustom1, pvaralCustom2,  
pvaralCustom3, pvaralCustom4, pvarabstrServers, pvarabstrUserNames,  
pvaradTimeModified, pvaralActivityCode, pvaradValues, pvaralNoData,  
plTotalNumRecords
```

Argument	Description
<i>dStartTime</i>	Double (ByVal). 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.
<i>dEndTime</i>	Double (ByVal). 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.
<i>vbAllServers</i>	Boolean (ByVal). 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.
<i>bstrServer</i>	String (ByVal). 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.
<i>vbAllUsers</i>	Boolean (ByVal). 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. If you pass FALSE, use the <i>bstrUserName</i> argument to specify the username by which to filter.

Argument	Description
<i>bstrUserName</i>	String (ByVal). The user name of the user by which to filter. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE.
<i>IStartRecord</i>	Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.
<i>IEndRecord</i>	Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.
<i>pvaralScenarios</i>	Variant array. Returns the member IDs of the Scenario dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvaralYear</i>	Variant array. Returns the member IDs of the Year dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvaralPeriod</i>	Variant array. Returns the member IDs of the Period dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvaralEntity</i>	Variant array. Returns the member IDs of the Entity dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvaralParent</i>	Variant array. Returns the member IDs of the parents of the Entity dimension members returned by the <i>pvaralEntity</i> argument. The array items are returned as a Long subtype.
<i>pvaralValue</i>	Variant array. Returns the member IDs of the Value dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvaralICP</i>	Variant array. Returns the member IDs of the Intercompany Partner dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvaralAccount</i>	Variant array. Returns the member IDs of the Account dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvaralCustom1</i>	Variant array. Returns the member IDs of the Custom 1 dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvaralCustom2</i>	Variant array. Returns the member IDs of the Custom 2 dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvaralCustom3</i>	Variant array. Returns the member IDs of the Custom 3 dimension members for the data changes' cells.

Argument	Description
	The array items are returned as a Long subtype.
<i>pvaralCustom4</i>	Variant array. Returns the member IDs of the Custom 4 dimension members for the data changes' cells. The array items are returned as a Long subtype.
<i>pvarabstrServers</i>	Variant array. Returns the names of the application servers on which the data changes were made. The array items are returned as a String subtype.
<i>pvarabstrUserNames</i>	Variant array. Returns the usernames of the users who made the data changes. The array items are returned as a String subtype.
<i>pvaradTimeModified</i>	Variant array. Returns the timestamps of the data changes. These are returned as Double values that can be cast to the Date format. The array items are returned as a Double subtype.
<i>pvaralActionCode</i>	Variant array. Returns the IDs of the user activities that caused the data changes. Valid values are represented by the HFConstants type library constants listed in "User Activity Constants" on page 870 . The array items are returned as a Long subtype.
<i>pvaradValues</i>	Variant array. Returns the cell values that the data changes resulted in. The array items are returned as a Double subtype.
<i>pvaralNoData</i>	Variant array. 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. The array items are returned as a Long subtype.
<i>pITotalNumRecords</i>	Long. Returns the total number of audit records in the database that meet the filtering criteria.

EnumDataAuditItems2

For internal use.

EnumEntitiesWithDataForScenarioYear

Returns a variant array of Entity ID's that have data for a given scenario and year.

Note: Does not account for Calculated data.

Syntax

```
<HsvData>.EnumEntitiesWithDataForScenarioYear lScenario, lYear,
pvaralEntityIDs
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>pvarEntityIDs</i>	Variant. The member IDs for the entities that have data for the specified scenario and year.

EnumExtractOptions

Returns a two-dimensional array of the data extract options that can be passed to [Extract](#). The array includes the options' names and default values. For some options, the array also identifies the valid range of values.

Syntax

```
<HsvData>.EnumExtractOptions pvar2davOptions
```

Argument	Description
<i>pvar2davOptions</i>	Variant array. Returns a two-dimensional safe array that represents the data extract options. The first dimension identifies the options, and is indexed from 1 to 20. The indexes and corresponding extract options are listed in Table 60 on page 296 . The second dimension provides information on options, and is indexed from 0 to 5: <ul style="list-style-type: none"> ● 0 = The option's index in the array of options passed to Extract. (Long subtype). ● 1 = The option's name. (String subtype). ● 2 = The option's default value. (The subtype varies). ● 3 = The option's minimum value, if any. (Long subtype). ● 4 = The option's maximum value, if any (Long subtype). ● 5 = A tab-delimited list of the option's valid values (String subtype). If the option is not restricted to a set of values, a blank String is returned. For example, since the delimiter option is the second item in the first dimension, array item (2, 2) stores the system's default delimiter. Note: Options without minimum and maximum values will return 0 for items 3 and 4 of the second dimension.

Example

The following example defines a function named `getDataExtractDefaults` that returns an array of the default extract option values. This function loops through the array returned by `EnumExtractOptions`, assigning each option's default value to the `vaDataSettings` array. `vaDataSettings` is then assigned as the function's return value.

```
Function getDataExtractDefaults() As Variant
Dim vaOpts, vaDataSettings(1 To 20)
m_chsvData.EnumExtractOptions vaOpts
'Assign the default values, which are stored in the
'item # 2 of the second dimension of vaOpts.
For i = LBound(vaOpts) To UBound(vaOpts)
```

```

    vaDataSettings(i) = vaOpts(i, 2)
Next i
getDataExtractDefaults = vaDataSettings
End Function

```

Tip: This function is used in the [Example](#) for `Extract`.

EnumLoadOptions

Returns a two-dimensional array of the data load options that can be passed to `Load`. The array includes the options' names and default values. For some options, the array also identifies the valid range of values.

Syntax

```
<HsvData>.EnumLoadOptions pvar2davOptions
```

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

<i>pvar2davOptions</i>	Variant array. Returns a two-dimensional safe array that represents the data load options.
------------------------	--

The first dimension identifies the options, which are listed in [Table 61 on page 354](#).

The second dimension provides information on options, and is indexed from 0 to 5:

- 0 = The option's index in the array of options passed to `Load`. (Long subtype).
- 1 = The option's name. (String subtype).
- 2 = The option's default value. (The subtype varies).
- 3 = The option's minimum value, if any. (Long subtype).
- 4 = The option's maximum value, if any (Long subtype).
- 5 = A tab-delimited list of the option's valid values (String subtype). If the option is not restricted to a set of values, a blank String is returned.

For example, since the delimiter option is the first item in the first dimension, array item (1, 2) stores the system's default delimiter.

Note: Options without minimum and maximum values will return **0** for items 3 and 4 of the second dimension.

Example

`EnumLoadOptions` is used in the example for [Load](#).

Extract

Extracts data into a text file. The file will be created on the application server.

Tip: You can extract files onto client PCs with the `HsvcDataLoad` type library. This library also offers properties and methods that simplify handling of the data extract options. For more information, see [“Extracting Data” on page 777](#).

Syntax

```
<HsvData>.Extract bstrServerFileName, bstrLogFileName, varavSettings
```

Argument	Description
<i>bstrServerFileName</i>	String (ByVal). The name and path of the data extract file. The path must exist on the application server.
<i>bstrLogFileName</i>	String (ByVal). The name and path of the log file. The path must exist on the application server.
<i>varavSettings</i>	Variant array (ByVal). The extract options for the data extraction operation. The array is 1-based and contains 20 items. For details on valid indexes and values, see Table 60 . Tip: Use EnumExtractOptions to return information about the valid extract options. In this array you must specify member IDs for the Scenario, Year, Period, Entity, and Account extract options – these options are set in array items 7 through 12. You can use the defaults returned by EnumExtractOptions for the other options.

The following table describes the data extract options. Some of the valid values are represented by constants of the HsvcDataLoad type library. To use these constants, you must reference HsvcDataLoad in your project; for information on this library, see [“HsvcDataLoad Type Library” on page 775](#).

The listed indexes apply to the array passed to `Extract` and to the first dimension of the array returned by `EnumExtractOptions`.

Table 60 Data Extract Options

Index	Extract Option Information
1	Option: View Usage: Specifies the View dimension member for which data is being extracted. Pass to Extract: One of the following HsvcDataLoad type library constants: <ul style="list-style-type: none">● HSV_DATA_VIEW_PERIODIC. Data for the Periodic view will be extracted.● HSV_DATA_VIEW_YTD. Data for the year-to-date view will be extracted.● HSV_DATA_VIEW_SCENARIO. Data for the scenario’s default View dimension member will be extracted. Tip: These constants correspond to the Data View option buttons in the Extract Data workspace frame.
2	Option: Delimiter Usage: Specifies an extract file’s delimiter. Pass to Extract: A valid delimiter character (String subtype).
3	Option: Log file name. <i>For internal use.</i>
4	Option: Append to Log File Usage: Specifies whether log data is appended to or overwrites the existing log file. Pass to Extract: TRUE to append, FALSE to overwrite.

Index	Extract Option Information
5	<p>Option: Append to Extract File</p> <p><i>For internal use.</i></p>
6	<p>Option: Extract Calculated</p> <p>Usage: Specifies whether to extract calculated data.</p> <p>Pass to Extract: TRUE to extract calculated data, FALSE otherwise.</p>
7	<p>Option: Scenario</p> <p>Usage: Specifies the Scenario dimension member for which data is being extracted.</p> <p>Pass to Extract: A Long containing the Scenario member's ID.</p>
8	<p>Option: Year</p> <p>Usage: Specifies the Year dimension member for which data is being extracted.</p> <p>Pass to Extract: A Long containing the Year member's ID.</p>
9	<p>Option: Period Subset</p> <p>Usage: Specifies the Period dimension members for which data is being extracted.</p> <p>Pass to Extract: A Long array containing Period member IDs.</p>
10	<p>Option: Entity Subset</p> <p>Usage: Specifies the child Entity dimension members for which data is being extracted.</p> <p>Pass to Extract: A Long array containing Entity member IDs.</p>
11	<p>Option: Parent Subset</p> <p>Usage: Specifies the parent Entity dimension members for which data is being extracted.</p> <p>Pass to Extract: A Long array containing Entity dimension IDs.</p>
12	<p>Option: Account Subset</p> <p>Usage: Specifies the Account dimension members for which data is being extracted.</p> <p>Pass to Extract: A Long array containing Account member IDs.</p>
13	<p>Option: ICP Subset</p> <p><i>For internal use.</i></p>
14	<p>Option: Value Subset</p> <p><i>For internal use.</i></p>
15	<p>Option: Custom 1 Subset</p> <p><i>For internal use.</i></p>
16	<p>Option: Custom 2 Subset</p> <p><i>For internal use.</i></p>
17	<p>Option: Custom 3 Subset</p> <p><i>For internal use.</i></p>

Index	Extract Option Information
18	Option: Custom 4 Subset <i>For internal use.</i>
19	Option: Extract All Data <i>For internal use.</i>
20	Option: Unicode <i>For internal use.</i>

Example

The following example extracts data for the dimension members listed below:

- Scenario = Actual
- Year = 2000
- Periods = July, August, September
- Account = Sales
- Child entity = Connecticut
- Parent entity = UnitedStates

The custom function `getDataExtractDefaults` assigns the default extract options to the `vaSettings` array. The IDs of the dimension members listed above are obtained with the custom function `GetMemberID`, then assigned to the `vaSettings` array; this overrides the previously assigned default values for `vaSettings` items 7 through 12. `vaSettings` is then passed to `Extract`.

Tip: For details on the `getDataExtractDefaults` custom function, see the [Example](#) for `EnumExtractOptions`. For details on the `GetMemberID` custom function, see the [Examples](#) for `IHsvTreeInfo.GetItemID`.

```
Dim vaSettings, lScen As Long, lYear As Long, laPer(2) As Long
Dim laAcct(0) As Long, laPar(0) As Long, laEnt(0) As Long
'Get the default extract options
vaSettings = getDataExtractDefaults()
'Set the scenario and year
vaSettings(7) = GetMemberID(DIMENSIONSCENARIO, "Actual")
vaSettings(8) = GetMemberID(DIMENSIONYEAR, "2000")
'Set the periods
laPer(0) = GetMemberID(DIMENSIONPERIOD, "July")
laPer(1) = GetMemberID(DIMENSIONPERIOD, "August")
laPer(2) = GetMemberID(DIMENSIONPERIOD, "September")
vaSettings(9) = laPer
'Set the child and parent entities
laEnt(0) = GetMemberID(DIMENSIONENTITY, "Connecticut")
vaSettings(10) = laEnt
laPar(0) = GetMemberID(DIMENSIONENTITY, "UnitedStates")
vaSettings(11) = laPar
```

```
'Set the account
laAcct(0) = GetMemberID(DIMENSIONACCOUNT, "Sales")
vaSettings(12) = laAcct
'Extract the data
m_cHsvData.Extract "c:\Acme\myAppExt.dat", _
"c:\Acme\myAppExt.log", vaSettings
```

ExtractDataAuditItems

Extracts to a file the data audit information that meets the specified filtering criteria. The filtering criteria include dimension members, date range, application server, and username.

Syntax

```
<HsvData>.ExtractDataAuditItems bstrServerFileName, bstrLogFileName,
bstrDelimitChar, dStartTime, dEndTime, varabstrMemberFilter, vbAllServers,
bstrServer, vbAllUsers, bstrUserName
```

Argument	Description
<i>bstrServerFileName</i>	String (ByVal). The name and path of the file into which to extract audit information.
<i>bstrLogFileName</i>	String (ByVal). The name and path of the log file for the extraction.
<i>bstrDelimitChar</i>	String (ByVal). The delimiter character for the extract file.
<i>dStartTime</i>	Double (ByVal). 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.
<i>dEndTime</i>	Double (ByVal). 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.
<i>varabstrMemberFilter</i>	String array (ByVal). 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 834 .

Caution! Do not use the constant that represents parent Entity members as an index, because parents cannot be used as filters.

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:

```
vaFilter(DIMENSIONACCOUNT) = "Sales"
vaFilter(DIMENSIONCUSTOM1) = ""
vaFilter(DIMENSIONCUSTOM2) = ""
vaFilter(DIMENSIONCUSTOM3) = ""
vaFilter(DIMENSIONCUSTOM4) = ""
vaFilter(DIMENSIONENTITY) = ""
vaFilter(DIMENSIONICP) = ""
vaFilter(DIMENSIONPERIOD) = ""
vaFilter(DIMENSIONSCENARIO) = "Actual"
```

Argument	Description
	<code>vaFilter(DIMENSIONVALUE) = ""</code> <code>vaFilter(DIMENSIONVIEW) = ""</code> <code>vaFilter(DIMENSIONYEAR) = ""</code>
<i>vbAllServers</i>	Boolean (ByVal). 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.
<i>bstrServer</i>	String (ByVal). 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.
<i>vbAllUsers</i>	Boolean (ByVal). 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. If you pass FALSE, use the <i>bstrUserName</i> argument to specify the username by which to filter.
<i>bstrUserName</i>	String (ByVal). The username by which to filter. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE.

ExtractDrillableRegions

Retrieves all ERPI URL definitions.

Syntax

```
<HsvData>.ExtractDrillableRegions bstrRegionsFilename, bstrLogFileName,
pvbErrors, pvbWarnings
```

Argument	Description
<i>bstrRegionsFilename</i>	String (ByVal). The file path to the definition file on the Financial Management application server.
<i>bstrLogFileName</i>	String (ByVal). The name and path of the log file. The path must exist on the application server.
<i>pvbErrors</i>	Boolean. Returns TRUE if errors occurred during the load task. See the log file for details.
<i>pvbWarnings</i>	Boolean. Returns TRUE if warnings occurred during the load task.

Examples

This function is intended to be used by *HFMawbAgent*. The following example will extract all existing definitions and save it to `C:\temp\~RD416e.tmp`. Progress messages and errors are saved in the log file `C:\temp\~RD9bc2.tmp`. If errors or warnings occur, *vbError* or *vbWarnings* will be set to `VARIANT_TRUE`.

```
ExtractDrillableRegions(_T("C:\temp\~RD416e.tmp"), T("C:\temp\
~RD9bc2.tmp"), &vbError, &vbWarnings);
```

ExtractDrillableRegionsByURLNames

Retrieves only the URLs that appear in the input array *varabstrURLNames*.

Syntax

```
<HsvData>.ExtractDrillableRegionsByURLNames varabstrURLNames,  
bstrRegionsFilename, bstrLogFileName, pvbErrors, pvbWarnings
```

Argument	Description
<i>varabstrURLNames</i>	Variant (ByVal). Array of URL names.
<i>bstrRegionsFilename</i>	String (ByVal). The file path to the definition file on Financial Management application server.
<i>bstrLogFileName</i>	String (ByVal). The name and path of the log file. The path must exist on the application server.
<i>pvbErrors</i>	Boolean. Returns TRUE if errors occurred during the load task. See the log file for details.
<i>pvbWarnings</i>	Boolean. Returns TRUE if warnings occurred during the load task.

Example

This function is intended to be used by *HFMAwbAgent*. The following example extracts existing URL definitions for “ERPI_USSales,” “ERPI_EUSales,” and “ERPI_Salary” and save it to C:\temp\~RD4a6e.tmp. Progress messages and errors are going to be saved in the log file C:\temp\~RD9bc3.tmp. If any errors or warnings occur, *vbError* or *vbWarnings* will be set to VARIANT_TRUE.

```
ExtractDrillableRegionsByURLNames(varabstrURLNames, _T("C:\\temp\  
~RD4a6e.tmp"), _T("C:\\temp\\~RD9bc3.tmp"), &vbError, &vbWarnings);
```

The input parameter *varabstrURLName* should be an array of BSTR, containing “ERPI_USSales,” “ERPI_EUSales,” and “ERPI_Salary.”

FilterMembersThatHaveData

Returns a list of members for a specified dimension for which data exists for the specified POVs.

Syntax

```
<HsvData>.FilterMembersThatHaveData varalScenario, varalYear, varalPeriod,  
varalEntity, varalParent, varalValue, varalAccount, varalICP, varalCustom1,  
varalCustom2, varalCustom3, varalCustom4, lDimID,  
vbConsiderBaseMembersOnly, varalMembersToFilter, pvaralMembers
```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.

Argument	Description
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>lDimID</i>	Long (ByVal). The dimension for which you want a list of members. You can use the following dimensions: Account, ICP, Custom 1, Custom 2, Custom 3, or Custom 4. For valid values see "Dimension ID Constants" on page 834 .
<i>vbConsiderBaseMembers Only</i>	Boolean (ByVal). Determines if base members only are tested.
<i>varalMembersToFilter</i>	Variant Array (ByVal). The dimension member IDs for which you want to determine if data exists. For example, you can pass in an array of account members for which you want to determine if data exists.
<i>pvaralMembers</i>	Variant. Returns the filtered list of member IDs that contain data for any of the specified POVs.

FilterMembersThatHaveData2

For a specified dimension, returns the filtered indexes of member IDs that contain data for any of the specified POVs.

Syntax

```
<HsvData>.FilterMembersThatHaveData2 varalScenario, varalYear,
varalPeriod, varalEntity, varalParent, varalValue, varalAccount, varalICP,
varalCustom1, varalCustom2, varalCustom3, varalCustom4, lDimID,
vbConsiderBaseMembersOnly, varalMembersToFilter,
pvaralFilteredMemberIndexes
```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.

Argument	Description
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>IDimID</i>	Long (ByVal). The dimension for which you want a list of members. You can use the following dimensions: Account, ICP, Custom 1, Custom 2, Custom 3, or Custom 4. For valid values see " Dimension ID Constants " on page 834.
<i>vbConsiderBaseMembersOnly</i>	Boolean (ByVal). Determines if base members only are tested.
<i>varalMembersToFilter</i>	Variant Array (ByVal). The dimension member IDs for which you want to determine if data exists. For example, you can pass in an array of account members for which you want to determine if data exists.
<i>pvaralFilteredMemberIndexes</i>	Variant. Returns the filtered indexes of member IDs that contain data for any of the specified POVs. The indexes are the <i>varalMembersToFilter</i> array item IDs which allow you to relate the data returned to the filtered members requested. You can use the index to get the dimension ID from the <i>varalMembersToFilter</i> array.

FormatNumberToText

Takes a number and returns it as a String. `FormatNumberToText` scales the number and puts a fixed number of digits to the right of the decimal point, adding or removing digits as needed.

By default, `FormatNumberToText` formats the returned number as follows:

- `FormatNumberToText` scales by applying the `Scale` attribute of the applicable Value dimension member. This member is specified by either the `lValue` argument or the `lEntity` and `lParent` arguments; see the following table for details.
- `FormatNumberToText` specifies the number of digits to the right of the decimal point by applying the `NumDecimalPlaces` attribute of the account identified by the `lAccount` argument.

`FormatNumberToText` also enables you to override this default formatting.

Syntax

```
<HsvData>.FormatNumberToText(lEntity, lParent, lValue, lAccount, dNumber, lStatus, vbUseDefaultNumDecimals, sAlternateNumDecimals, vbUseDefaultScale, sAlternateScale, vbUseTheUsersFormattingParameters)
```

Argument	Description
<i>lEntity</i>	<p>Long (ByVal). If the <i>lValue</i> argument specifies the <Entity Currency>, <Entity Curr Adjs>, or <Entity Curr Total> Value dimension member, <code>FormatNumberToText</code> scales the return value by using the <code>Scale</code> attribute of the entity's default currency.</p> <p>Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the <i>lParent</i> or <i>lValue</i> argument.</p>
<i>lParent</i>	<p>Long (ByVal). The member ID of the parent of the Entity dimension member identified by the <i>lEntity</i> argument. If the <i>lValue</i> argument specifies the <Parent Currency>, <Parent Curr Adjs>, or <Parent Curr Total> Value dimension member, <code>FormatNumberToText</code> scales the return value by using the <code>Scale</code> attribute of this parent entity's default currency.</p> <p>Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the <i>lEntity</i> or <i>lValue</i> argument.</p>
<i>lValue</i>	<p>Long (ByVal). The member ID of the Value dimension member for the currency whose <code>Scale</code> attribute is to be applied to the return value.</p> <p>If you pass the member ID of the <Entity Currency>, <Entity Curr Adjs>, <Entity Curr Total>, <Parent Currency>, <Parent Curr Adjs>, or <Parent Curr Total> Value dimension member, the scaling is determined by the <i>lEntity</i> and <i>lParent</i> arguments as described above.</p>
<i>lAccount</i>	<p>Long (ByVal). The member ID of the Account dimension member whose <code>NumDecimalPlaces</code> attribute is to be applied to the return value.</p>
<i>dNumber</i>	<p>Double (ByVal). The number to be converted and formatted.</p>
<i>lStatus</i>	<p>Long (ByVal). The cell status; the status determines whether a 0 passed to the <i>dNumber</i> argument is returned as the string 0 or as the string <code>NoData</code>. The following list provides guidelines for this argument:</p> <ul style="list-style-type: none">● If you pass cell data returned by another method to the <i>dNumber</i> argument, it is recommended that you also pass the status returned by that method to the <i>lStatus</i> argument.● If you want to format a number that is not returned from a cell, pass 0. <p>Caution! If you pass cell data to <i>dNumber</i> but also pass a manually-specified status to <i>lStatus</i>, a security breach may occur. This could occur if the cell's status indicates that the user does not have read access but you pass in 0 instead of the status.</p>
<i>vbUseDefaultNumDecimals</i>	<p>Boolean (ByVal). Specify <code>TRUE</code> to apply the <code>NumDecimalPlaces</code> attribute of the account identified by the <i>lAccount</i> argument, otherwise <code>FALSE</code>. If you specify <code>FALSE</code>, the number of decimal places is determined by the <i>sAlternateNumDecimals</i> argument.</p>
<i>sAlternateNumDecimals</i>	<p>Integer (ByVal). If you set the <i>vbUseDefaultNumDecimals</i> argument to <code>FALSE</code>, this argument specifies the number of decimal places to be applied to the return value.</p> <p>To use the application's default number of decimals, pass the <code>HFMConstants</code> type library constant <code>DEFAULT_NUM_DECIMALS</code>, described in "Number Defaults Constants" on page 881.</p> <p>If you set the <i>vbUseDefaultNumDecimals</i> argument to <code>TRUE</code>, this argument is ignored.</p>

Argument	Description
<i>vbUseDefaultScale</i>	Boolean (ByVal). Specify TRUE to apply the Scale attribute of the currency identified by the <i>lValue</i> argument, otherwise FALSE. If you specify FALSE, the <i>sAlternateScale</i> argument specifies the scaling to be applied.
<i>sAlternateScale</i>	Integer (ByVal). If you set the <i>vbUseDefaultScale</i> argument to FALSE, this argument indicates the degree of scaling to apply to the return value. To leave the return value unscaled, pass 0. To scale the return value, each whole-number increment over 0 scales by a tenth. In other words, passing 1 scales by a tenth, passing 2 scales by a hundredth, and so on. To use the application's default scaling, pass the HFMConstants type library constant <code>DEFAULT_SCALE</code> , described in "Number Defaults Constants" on page 881 . If you set the <i>vbUseDefaultScale</i> argument to TRUE, this argument is ignored.
<i>vbUseTheUsersFormatting Parameters</i>	Boolean (ByVal). Determines whether the user's preferred decimal and thousands separator characters should be applied to the return value. Pass TRUE to apply the user's preferred characters, FALSE to apply the system defaults.

Return Value

String. The String equivalent of the number passed as the *dNumber* argument, formatted as specified by the arguments.

Example

This example creates a function that takes a number and returns the number formatted as a String, with only one decimal place applied. This function also takes the dimension member IDs for `FormatNumberToText`.

```
Function ReturnOneDecimal(lEnt As Long, lPar As Long, _
    lVal As Long, lAcct As Long, dNum As Double) As String
ReturnOneDecimal = m_chsvData.FormatNumberToText(lEnt, lPar, _
    lVal, lAcct, dNum, 1, False, 1, True, 0, False)
End Function
```

FormatNumberToText2

Takes a number and returns it as a String, with the option to remove trailing zeroes.

`FormatNumberToText2` scales the number and puts a fixed number of digits to the right of the decimal point, adding or removing digits as needed. The method also returns the number of decimal places and the scaling applied to the formatted number.

By default, `FormatNumberToText2` formats the returned number as follows:

- `FormatNumberToText2` scales by applying the Scale attribute of the applicable Value dimension member. This member is specified by either the *lValue* argument or the *lEntity* and *lParent* arguments; see the following table for details.
- `FormatNumberToText2` specifies the number of digits to the right of the decimal point by applying the NumDecimalPlaces attribute of the account identified by the *lAccount* argument.

`FormatNumberToText2` also enables you to override this default formatting.

Syntax

```
<HsvData>.FormatNumberToText2(lEntity, lParent, lValue, lAccount, dNumber, lStatus, vbUseDefaultNumDecimals, sAlternateNumDecimals, vbUseDefaultScale, sAlternateScale, vbUseTheUsersFormattingParameters, vbRemoveTrailingZeros, nNumDecimals, nScale)
```

Argument	Description
<i>lEntity</i>	<p>Long (ByVal). If the <i>lValue</i> argument specifies the <Entity Currency>, <Entity Curr Adjs>, or <Entity Curr Total> Value dimension member, <code>FormatNumberToText2</code> scales the return value by using the <code>Scale</code> attribute of the entity's default currency.</p> <p>Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the <i>lParent</i> or <i>lValue</i> argument.</p>
<i>lParent</i>	<p>Long (ByVal). The member ID of the parent of the Entity dimension member identified by the <i>lEntity</i> argument. If the <i>lValue</i> argument specifies the <Parent Currency>, <Parent Curr Adjs>, or <Parent Curr Total> Value dimension member, <code>FormatNumberToText2</code> scales the return value by using the <code>Scale</code> attribute of this parent entity's default currency.</p> <p>Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the <i>lEntity</i> or <i>lValue</i> argument.</p>
<i>lValue</i>	<p>Long (ByVal). The member ID of the Value dimension member for the currency whose <code>Scale</code> attribute is to be applied to the return value.</p> <p>If you pass the member ID of the <Entity Currency>, <Entity Curr Adjs>, <Entity Curr Total>, <Parent Currency>, <Parent Curr Adjs>, or <Parent Curr Total> Value dimension member, the scaling is determined by the <i>lEntity</i> and <i>lParent</i> arguments as described above.</p>
<i>lAccount</i>	<p>Long (ByVal). The member ID of the Account dimension member whose <code>NumDecimalPlaces</code> attribute is to be applied to the return value.</p>
<i>dNumber</i>	<p>Double (ByVal). The number to be converted and formatted.</p>
<i>lStatus</i>	<p>Long (ByVal). The cell status; the status determines whether a 0 passed to the <i>dNumber</i> argument is returned as the string 0 or as the string <code>NoData</code>. The following list provides guidelines for this argument:</p> <ul style="list-style-type: none">● If you pass cell data returned by another method to the <i>dNumber</i> argument, it is recommended that you also pass the status returned by that method to the <i>lStatus</i> argument.● If you want to format a number that is not returned from a cell, pass 0. <p>Caution! If you pass cell data to <i>dNumber</i> but also pass a manually-specified status to <i>lStatus</i>, a security breach may occur. This could occur if the cell's status indicates that the user does not have read access but you pass in 0 instead of the status.</p>
<i>vbUseDefaultNumDecimals</i>	<p>Boolean (ByVal). Specify <code>TRUE</code> to apply the <code>NumDecimalPlaces</code> attribute of the account identified by the <i>lAccount</i> argument, otherwise <code>FALSE</code>. If you specify <code>FALSE</code>, the number of decimal places is determined by the <i>sAlternateNumDecimals</i> argument.</p>
<i>sAlternateNumDecimals</i>	<p>Integer (ByVal). If you set the <i>vbUseDefaultNumDecimals</i> argument to <code>FALSE</code>, this argument specifies the number of decimal places to be applied to the return value.</p> <p>To use the application's default number of decimals, pass the <code>HFMConstants</code> type library constant <code>DEFAULT_NUM_DECIMALS</code>, described in "Number Defaults Constants" on page 881.</p> <p>If you set the <i>vbUseDefaultNumDecimals</i> argument to <code>TRUE</code>, this argument is ignored.</p>

Argument	Description
<i>vbUseDefaultScale</i>	Boolean (ByVal). Specify TRUE to apply the Scale attribute of the currency identified by the <i>IValue</i> argument, otherwise FALSE. If you specify FALSE, the <i>sAlternateScale</i> argument specifies the scaling to be applied.
<i>sAlternateScale</i>	Integer (ByVal). If you set the <i>vbUseDefaultScale</i> argument to FALSE, this argument indicates the degree of scaling to apply to the return value. To leave the return value unscaled, pass 0. To scale the return value, each whole-number increment over 0 scales by a tenth. In other words, passing 1 scales by a tenth, passing 2 scales by a hundredth, and so on. To use the application's default scaling, pass the HFMConstants type library constant <code>DEFAULT_SCALE</code> , described in "Number Defaults Constants" on page 881 . If you set the <i>vbUseDefaultScale</i> argument to TRUE, this argument is ignored.
<i>vbUseTheUsersFormatting Parameters</i>	Boolean (ByVal). Determines whether the user's preferred decimal and thousands separator characters should be applied to the return value. Pass TRUE to apply the user's preferred characters, FALSE to apply the system defaults.
<i>vbRemoveTrailingZeros</i>	Boolean (ByVal). A flag that specifies whether to remove trailing zeroes. Pass TRUE to remove trailing zeroes, FALSE otherwise.
<i>nNumDecimals</i>	Integer. Returns the number of decimal places applied to the return value.
<i>nScale</i>	Integer. Returns the degree of scaling applied to the return value.

Return Value

String. The String equivalent of the number passed as the *dNumber* argument, formatted as specified by the arguments.

FormatStatusToText

Returns a String code that represents the numeric calculation status passed as the argument. Use `FormatStatusToText` to get a meaningful description of the calculation statuses returned by methods such as `GetCalcStatus` and `GetCell`.

Syntax

```
<HsvData>.FormatStatusToText (lStatus)
```

Argument Description

IStatus Long (ByVal). A valid calculation status.

Return Value

String. The code that identifies the calculation status. The valid status codes are as follows:

- OK
- Locked
- OK SC

- CH
- TR
- CN

For descriptions of these statuses, see the *Oracle Hyperion Financial Management, Fusion Edition User's Guide*.

Example

`FormatStatusToText` is used in the example for [GetCell](#).

FormatStoredNumberToText

Converts a number passed as a `Double` to a `String`, formatting the `String` with the decimal and thousands delimiters that you specify. Use this method to format numbers returned by Financial Management methods such as `GetCell` and `GetCells`.

Syntax

```
<HsvData>.FormatStoredNumberToText (dNumber, lStatus, sDecimalChar,
sThousandsChar)
```

Argument	Description
<i>dNumber</i>	<code>Double (ByVal)</code> . The number to be formatted.
<i>lStatus</i>	<code>Long (ByVal)</code> . A valid cell status. The expected usage is that you pass the cell status returned by a method such as <code>GetCell</code> or <code>GetCells</code> .
<i>sDecimalChar</i>	<code>Integer (ByVal)</code> . The double-byte <code>Integer</code> that identifies the decimal to be used in the return value. Use <code>AscW</code> to get the delimiter's <code>Integer</code> equivalent.
<i>sThousandsChar</i>	<code>Integer (ByVal)</code> . The double-byte <code>Integer</code> that identifies the thousands delimiter to be used in the return value. Use <code>AscW</code> to get the delimiter's <code>Integer</code> equivalent.

Return Value

`String`. Returns the formatted `String` equivalent of the `Double` passed to the *dNumber* argument.

Example

The following example formats the number passed to `FormatStoredNumberToText` using a comma as a decimal character and periods as the thousands separator character. For example, if 19398736.23 is passed, 19,398.736,23 is assigned to the `sRet` variable.

```
Dim cData As HsvData, iDec As Integer
Dim iThous As Integer, sRet As String
Set cData = m_cSession.Data
iDec = AscW(",")
iThous = AscW(".")
sRet = cData.FormatStoredNumberToText(m_dNum, m_lStat, iDec, _
iThous)
```

FormatTextToNumber

Takes a number stored in a String and converts it to a Double. If the *IAccount* argument identifies a revenue, expense, asset, or liability account, `FormatTextToNumber` also scales the return value.

Syntax

```
<HsvData>.FormatTextToNumber(lEntity, lParent, lValue, lAccount, bstrText, vbUseDefaultScale, sAlternateScale, vbUseTheUsersFormattingParameters, pbIsValidNumber, pbIsNumberNoData)
```

Argument	Description
<i>lEntity</i>	<p>Long (ByVal). If the <i>lValue</i> argument specifies the <Entity Currency>, <Entity Curr Adjs>, or <Entity Curr Total> Value dimension member, <code>FormatTextToNumber</code> scales the return value by using the Scale attribute of the entity's default currency.</p> <p>Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the <i>lParent</i> or <i>lValue</i> argument.</p>
<i>lParent</i>	<p>Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity. If the <i>lValue</i> argument specifies the <Parent Currency>, <Parent Curr Adjs>, or <Parent Curr Total> Value dimension member, <code>FormatTextToNumber</code> scales the return value by using the Scale attribute of this parent entity's default currency.</p> <p>Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the <i>lEntity</i> or <i>lValue</i> argument.</p>
<i>lValue</i>	<p>Long (ByVal). A valid Value dimension ID. If the <i>IAccount</i> argument identifies a revenue, expense, asset, or liability account, <code>FormatTextToNumber</code> scales the return value. Use this argument to pass the member ID of the Value dimension member for the currency whose Scale attribute is to be applied to the return value.</p> <p>Note: If you pass the member ID of the <Entity Currency>, <Entity Curr Adjs>, <Entity Curr Total>, <Parent Currency>, <Parent Curr Adjs>, or <Parent Curr Total> Value dimension member, the scaling is determined by the <i>lEntity</i> and <i>lParent</i> arguments as described above.</p>
<i>IAccount</i>	<p>Long (ByVal). The member ID of an Account dimension member. If the member is a revenue, expense, asset, or liability account, <code>FormatTextToNumber</code> scales the return value.</p>
<i>bstrText</i>	<p>String (ByVal). The text to be converted to a Double. The text must be numeric; the <i>pbIsValidNumber</i> argument returns FALSE if this argument is non-numeric.</p>
<i>vbUseDefaultScale</i>	<p>Boolean (ByVal). Specify TRUE to apply the Scale attribute of the currency identified by the <i>lValue</i> argument, otherwise FALSE. If you specify FALSE, the <i>sAlternateScale</i> argument sets the scaling to be applied.</p>
<i>sAlternateScale</i>	<p>Integer (ByVal). If you set the <i>vbUseDefaultScale</i> argument to FALSE, this argument specifies the degree of scaling to be applied to the return value.</p> <p>If you set the <i>vbUseDefaultScale</i> argument to TRUE, this argument's value is ignored.</p>
<i>vbUseTheUsersFormattingParameters</i>	<p>Boolean (ByVal). Determines whether <code>FormatTextToNumber</code> uses the user's preferred decimal and thousands separator characters or the system default characters when it reads the <i>bstrText</i> argument. Pass TRUE to use the user's preferred characters, FALSE to use the system defaults.</p>

Argument	Description
<i>pbIsValidNumber</i>	Boolean. Returns TRUE if the <i>bstrText</i> argument is a valid number that can be converted, otherwise FALSE.
<i>pbIsNumberNoData</i>	Boolean. Returns TRUE if the <i>bstrText</i> argument is 0, otherwise FALSE.

Return Value

Double. The number passed as the *bstrText* argument, converted to a String and scaled as specified by the arguments.

FormatTextToStoredNumber

Converts a number passed as a String to a Double. Use this method to convert numbers returned by Financial Management methods such as `GetTextCell` and `GetTextCells`.

Syntax

```
<HsvData>.FormatTextToStoredNumber bstrText, sDecimalChar, sThousandsChar, pbIsValidNumber, pbIsNumberNoData, pdNumber
```

Argument	Description
<i>bstrText</i>	String (ByVal). The String to be converted.
<i>sDecimalChar</i>	Integer (ByVal). The double-byte Integer that identifies the decimal character used in the <i>bstrText</i> argument. Use <code>AscW</code> to get the delimiter's Integer equivalent.
<i>sThousandsChar</i>	Integer (ByVal). The double-byte Integer that identifies the thousands separator character used in the <i>bstrText</i> argument. Use <code>AscW</code> to get the delimiter's Integer equivalent.
<i>pbIsValidNumber</i>	Boolean. Returns TRUE if the <i>bstrText</i> argument contains a number that can be converted to a Double, otherwise FALSE.
<i>pbIsNumberNoData</i>	Boolean. Returns TRUE if the <i>bstrText</i> argument is 0, otherwise FALSE.
<i>pdNumber</i>	Double. Returns the formatted Double equivalent of the String passed to the <i>bstrText</i> argument.

Example

The following example converts the String passed to `FormatTextToStoredNumber`. Note how the decimal and thousands delimiters are specified. To illustrate the example, if 22,445,987.23 is passed, 22445987.23 is assigned to the `dRet` variable.

```
Dim cData As HsvData, iDec As Integer, iThous As Integer
Dim bIsValid As Boolean, bNoData As Boolean, dRet As Double
Set cData = m_cSession.Data
iDec = AscW(".")
iThous = AscW(",")
cData.FormatTextToStoredNumber m_sCellData, iDec, iThous, _
bIsValid, bNoData, dRet
```

GetAllDescriptionsInSubCube

Returns an array containing the cell text descriptions of one or more cells that have the same Scenario, Year, and Value dimension members.

Syntax

```
<HsvData>.GetAllDescriptionsInSubCube lScenario, lYear, varlPeriod,  
varlEntity, varlParent, lValue, varlAccount, pvarlPeriod, pvarlEntity,  
pvarlParent, pvarlValue, pvarlAccount, pvarlICP, pvarlCustom1,  
pvarlCustom2, pvarlCustom3, pvarlCustom4, pvarbstrCellText
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cells' Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cells' Year dimension member.
<i>varlPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varlEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varlParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varlEntity</i> 's entities.
<i>lValue</i>	Long (ByVal). The member ID of the cells' Value dimension member.
<i>varlAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>pvarlPeriod</i>	Variant array. Returns the member IDs of the cells' Period dimension members. The array items are returned as a Long subtype.
<i>pvarlEntity</i>	Variant array. Returns the member IDs of the cells' Entity dimension members. The array items are returned as a Long subtype.
<i>pvarlParent</i>	Variant array. Returns the member IDs of the parent entities of the <i>pvarlEntity</i> argument's entities. The array items are returned as a Long subtype.
<i>pvarlValue</i>	Variant array. Returns the member IDs of the cells' Value dimension members. The array items are returned as a Long subtype.
<i>pvarlAccount</i>	Variant array. Returns the member IDs of the cells' Account dimension members. The array items are returned as a Long subtype.
<i>pvarlICP</i>	Variant array. Returns the member IDs of the cells' Intercompany Partner dimension members. The array items are returned as a Long subtype.
<i>pvarlCustom1</i>	Variant array. Returns the member IDs of the cells' Custom 1 dimension members. The array items are returned as a Long subtype.
<i>pvarlCustom2</i>	Variant array. Returns the member IDs of the cells' Custom 2 dimension members. The array items are returned as a Long subtype.
<i>pvarlCustom3</i>	Variant array. Returns the member IDs of the cells' Custom 3 dimension members. The array items are returned as a Long subtype.
<i>pvarlCustom4</i>	Variant array. Returns the member IDs of the cells' Custom 4 dimension members. The array items are returned as a Long subtype.

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

<i>pvarbstrCellText</i>	Variant array. Returns the cell text descriptions for the cells. The array items are returned as a String subtype.
-------------------------	--

GetAllURLNames

Retrieves the full list of URL names currently in Financial Management.

Syntax

```
<HsvData>.GetAllURLNames pvarabstrURLNames
```

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

<i>pvarabstrURLNames</i>	Variant array. Array of URL names.
--------------------------	------------------------------------

Example

This function is intended to be used prior to calling *LoadDrillableRegions* in “Replace by URL name” mode and *ExtractDrillableRegionsByURLNames*. The returned BSTR array *pvarabstrURLNames* from *GetAllURLNames* can be modified if needed and then used in the load and extract functions as an input. The following example will extract all existing URL names in Financial Management.

```
fsVARIANT *pvarabstrURLNames; GetAllURLNames (pvarabstrURLNames);
```

An example of the returned *BSTR* array *pvarabstrURLNames* would contain “ERPI_USSales,” “ERPI_EUSale,s” and “ERPI_Salary.”

GetAttachedDocumentsToCell

Returns the names and paths of the documents attached to a given cell. The names and paths are returned in arguments that have a one-to-one correspondence.

Syntax

```
<HsvData>.GetAttachedDocumentsToCell lScenario, lYear, lPeriod, lEntity,  
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,  
pvarbstraDocumentPaths, pvarbstraDocumentFiles
```

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

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
------------------	--

<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
--------------	--

<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
----------------	--

<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
----------------	--

Argument	Description
<i>IParent</i>	Long (ByVal). The member ID of the parent of the <i>IEntity</i> argument's entity.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pvarbstraDocumentPaths</i>	Variant. Returns the paths of the files attached to the cell.
<i>pvarbstraDocumentFiles</i>	Variant. Returns the names of the files attached to the cell.

Example

`GetAttachedDocumentsToCell` is used in the example for [DetachDocumentFromCell](#).

GetBaseDataForAccount

For internal use.

GetCalcStatus

Returns the calculation status of a subcube. The subcube is identified by `GetCalcStatus`'s arguments. (For information on subcubes, see [“About Subcubes” on page 43.](#))

Syntax

```
<HsvData>.GetCalcStatus lScenario, lYear, lPeriod, lEntity, lParent, lValue, plCalcStatus
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the subcube's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the subcube's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the subcube's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the subcube's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the parent of the <i>IEntity</i> argument's entity.
<i>IValue</i>	Long (ByVal). The member ID of the subcube's Value dimension member.

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

<i>plCalcStatus</i>	Long. Returns the subcube's calculation status as a bit-field. The valid statuses that this bit-field can contain are represented by constants in the <code>HFMConstants</code> type library; for a list of constants, see "Subcube Period Calculation Status Constants" on page 843 .
---------------------	--

Example

The following example tests a subcube's status to determine whether the subcube contains no data. `GetCalcStatus`'s returned subcube status is assigned to the `lCalcStat` variable. The `If` statement performs a bitwise comparison to see if the subcube's status includes the calculation status code for no data, which is represented by the `CALCSTATUS_NODATA` constant. If `lStat` contains this calculation status, then any code placed within the `If` structure would be executed. The example assumes that the variables in `GetCalcStatus`'s member ID arguments were passed from another procedure.

```
Dim cData As HsvData, lCalcStat As Long
Set cData = m_cHsvSession.Data
cData.GetCalcStatus m_lScen, m_lYear, m_lPer, m_lEnt, m_lPar, _
m_lVal, lCalcStat
If (CALCSTATUS_NODATA And lCalcStat) Then
    ...
End If
```

GetCalcStatusEx

Returns the calculation status of a cell.

Syntax

```
<HsvData>.GetCalcStatusEx lScenario, lYear, lPeriod, lEntity, lParent,
lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, plCalcStatus
```

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

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.

Argument	Description
<i>ICustom3</i>	Long (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>ICustom4</i>	Long (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>pCalcStatus</i>	Long. Returns the cell's calculation status as a bit-field. The valid statuses that this bit-field can contain are represented by constants listed in Table 114, “tagCALCSTATUSHIGHBITS Enumeration,” on page 842.

GetCalcStatusStatistics

Returns arrays of flags that indicate which calculation statuses apply to the specified entities and periods of a subcube. For each specified period, `GetCalcStatusStatistics` returns an array of two-dimensional arrays.

Syntax

```
<HsvData>.GetCalcStatusStatistics(lScenarioID, lYearID, varalPeriodIDs, lValueID, varalEntityIDs, varalParentIDs, vbOBPActiveOnly)
```

Argument	Description
<i>lScenarioID</i>	Long (ByVal). The member ID of the subcube's Scenario dimension member.
<i>lYearID</i>	Long (ByVal). The member ID of the subcube's Year dimension member.
<i>varalPeriodIDs</i>	Long array (ByVal). The member IDs of the Period dimension members for which to return status information.
<i>lValueID</i>	Long (ByVal). The member ID of the subcube's Value dimension member.
<i>varalEntityIDs</i>	Long array (ByVal). The member IDs of the Entity dimension members for which to return status information.
<i>varalParentIDs</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntityIDs</i> argument's entities.
<i>vbOBPActiveOnly</i>	Boolean (ByVal). Specifies whether to return status information for only active entities. This applies only to organization-by-period applications. Pass TRUE to return information for only active entities.

Return Value

Variant. Returns an array of Long two-dimensional arrays that indicate which calculation statuses apply to any cells that intersect the specified dimension members. Each array of two-dimensional arrays has a one-to-one correspondence with the *varalPeriodIDs* argument's array. The two-dimensional arrays contain the following information:

- The first dimension is an index that identifies the type of calculation status returned by the second dimension item. Valid values are represented by the `HFMConstants` enumeration `tagCALCSTATUSSTATISTICS`, which is described in [“Calculation Status Types” on page 846.](#)
- The second dimension indicates whether the calculation status applies to one or more of the specified entities. 1 indicates that the status applies, 0 that it does not.

Example

The following subroutine prints to Visual Basic's Immediate window whether the specified entities contain cells that have calculation statuses of NoData or CN ND.

```
Sub areEntitiesNDorCNND(lScen As Long, lYear As Long, laPers() As Long, _
    lVal As Long, laEnts() As Long, laPars() As Long)
Dim vRet As Variant, cData As HsvData, cTreeInfo As IHsvTreeInfo
Dim sPeriod As String
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
'g_cMetadata is an HsvMetadata object reference
Set cTreeInfo = g_cMetadata.Periods
vRet = cData.GetCalcStatusStatistics(lScen, lYear, laPers, lVal, _
    laEnts, laPars, False)
For i = LBound(vRet) To UBound(vRet)
    cTreeInfo.GetLabel laPers(i), sPeriod
    Debug.Print sPeriod + " period:"
    Debug.Print "    NoData: " + CStr(vRet(i, CALCSTATUS_STATSCOL_NODATA))
    Debug.Print "    CN ND: " + CStr(vRet(i, CALCSTATUS_STATSCOL_CNND))
Next i
End Sub
```

GetCell

Returns the data in a cell, as well as the cell's status.

Tip: `GetCell` returns cell data as a Double data type. To return a cell's data as a formatted string, use `GetTextCell`.

Syntax

```
<HsvData>.GetCell(lScenario, lYear, lPeriod, lView, lEntity, lParent,
    lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, pdData,
    plStatus)
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.

Argument Description

<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pdData</i>	Double. Returns the data in the cell.
<i>plStatus</i>	Long. Returns the cell's status. For details on cell statuses, see “About Cell Statuses” on page 275 .

Return Value

Integer. Indicates the success of the function call; returns 0 for success or an error code for failure.

Example

The following subroutine populates two Label controls with a cell's data and status. The subroutine takes the member IDs of the cell's dimension members. The cell status string is obtained by passing the status Long returned by `GetCell` to `FormatStatusToText`.

```
Sub fillCellLabels(lScen As Long, lYear As Long, lPer As Long, _
    lView As Long, lEnt As Long, lPar As Long, lVal As Long, _
    lAcct As Long, lICP As Long, lCust1 As Long, lCust2 As Long, _
    lCust3 As Long, lCust4 As Long)
Dim cData As HsvData, dData As Double, lStatus As Long
Dim sStatus As String
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
cData.GetCell lScen, lYear, lPer, lView, lEnt, _
    lPar, lVal, lAcct, lICP, lCust1, lCust2, _
    lCust3, lCust4, dData, lStatus
sStatus = cData.FormatStatusToText(lStatus)
'Populate the labels
lblData.Caption = dData
lblStatus.Caption = sStatus
End Sub
```

GetCellDescription

Returns the cell text description of a cell.

Syntax

```
<HsvData>.GetCellDescription lScenario, lYear, lPeriod, lEntity, lParent,
lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
pbstrCellText
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the parent of the <i>IEntity</i> argument's entity.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pbstrCellText</i>	String. Returns the cell text description.

GetCellDescriptions

Returns the cell text descriptions of one or more cells.

`GetCellDescriptions` takes arrays of dimension member IDs as arguments; these arrays must contain identical numbers of elements. The elements of these arrays have a one-to-one correspondence to the elements of the array that is returned; the first elements of the member ID arrays define the first cell to be returned, the second elements of the member ID arrays define the second cell to be returned, and so on.

Syntax

```
<HsvData>.GetCellDescriptions varIScenario, varIYear, varIPeriod,
varIEntity, varIParent, varIValue, varIAccount, varIICP, varICustom1,
varICustom2, varICustom3, varICustom4, pvarbstrCellText
```

Argument	Description
<i>varIScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varIYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varIPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varIEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.

Argument	Description
<i>varlParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varlEntity</i> argument's entities.
<i>varlValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varlAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varlICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varlCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varlCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varlCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varlCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>pvarbstrCellText</i>	Variant array. Returns cell text descriptions of the cells identified by the other arguments. The array items are returned as a String subtype.

Example

This example places the cell text of cells into a ListBox control. The example assumes that the cells' member ID arguments are passed from another procedure. `GetCellDescriptions` puts the array of cell text descriptions for the cells into the `saCellText` variable. The example then loops through this array, putting the array elements into the `lstDesc` ListBox.

```
Dim saCellText, iHighIndex As Integer, iLowIndex As Integer
Set m_cData = m_cSession.Data
m_cData.GetCellDescriptions laScen, laYear, laPer, laEnt, _
laPar, laVal, laAcct, laICP, laCust1, laCust2, laCust3, _
laCust4, saCellText
iHighIndex = UBound(saCellText)
For iLowIndex = LBound(saCellText) To iHighIndex
    lstDesc.AddItem saCellText(iLowIndex)
Next iLowIndex
```

For an example of how to configure the member ID arrays passed as arguments, see the example for [GetCells](#), which takes a similar set of arguments.

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](#). 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

```
<HsvData>.GetCellHistory lScenario, lYear, lPeriod, lEntity, lParent, lValue, lICP, lAccount, lCustom1, lCustom2, lCustom3, lCustom4, pvarabstrServers, pvarabstrUserNames, pvaradTimeModified, pvaralActionCode, pvaradValues, pvaralNoData
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). You must pass the <code>MEMBERNOTUSED</code> constant to this argument. For information on this constant, see “Dimension Member Constants” on page 834 .
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pvarabstrServers</i>	Variant array. Returns the names of the application servers on which the data changes were made. The array items are returned as a String subtype.
<i>pvarabstrUserNames</i>	Variant array. Returns the usernames of the users who made the data changes. The array items are returned as a String subtype.
<i>pvaradTimeModified</i>	Variant array. Returns the timestamps of the data changes. These are returned as Double values that can be cast to the Date format. The array items are returned as a Double subtype.
<i>pvaralActionCode</i>	Variant array. Returns 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 870 . The array items are returned as a Long subtype.
<i>pvaradValues</i>	Variant array. Returns the cell values that the data changes resulted in. The array items are returned as a Double subtype.
<i>pvaralNoData</i>	Variant array. 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.

Argument	Description
	The array items are returned as a Long subtype.

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, lVal, lIcp, lAcct, lCust1, lCust2, _
    lCust3, lCust4) As Variant
Dim cData As HsvData, vaServers, vaUsers, vaTime
Dim vaActivity, vaVal, vaNoData, vaRet(), lItems
'm_cHsvSession is an HsvSession object reference.
Set cData = m_cHsvSession.Data
cData.GetCellHistory lScen, lYear, lPer, lEnt, MEMBERNOTUSED, _
    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)
        vaRet(i, 2) = vaVal(i)
    Next i
    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](#). 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

```
<HsvData>.GetCellHistory2 lScenario, lYear, lPeriod, lEntity, lParent,
lValue, lICP, lAccount, lCustom1, lCustom2, lCustom3, lCustom4,
pvarabstrServers, pvarabstrUserNames, pvaradTimeModified,
pvaralActivityCode, pvarabstrValues, pvaralNoData
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>IParent</i>	Long (ByVal). You must pass the <code>MEMBERNOTUSED</code> constant to this argument. For information on this constant, see “Dimension Member Constants” on page 834 .
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pvarabstrServers</i>	Variant array. Returns the names of the application servers on which the data changes were made. The array items are returned as a String subtype.
<i>pvarabstrUserNames</i>	Variant array. Returns the usernames of the users who made the data changes. The array items are returned as a String subtype.
<i>pvaradTimeModified</i>	Variant array. Returns the timestamps of the data changes. These are returned as Double values that can be cast to the Date format. The array items are returned as a Double subtype.
<i>pvaralActivityCode</i>	Variant array. Returns 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 870 . The array items are returned as a Long subtype.
<i>pvarabstrValues</i>	Variant array. Returns the cell values that the data changes resulted in. The values are formatted according to the user’s preferences for decimal separator and thousands separator characters. Tip: To programmatically get and set these preferences, use the GetNumberFormattingUserParameters and SetNumberFormattingUserParameters methods of the <code>HsvSystemInfo</code> object. The array items are returned as a String subtype.
<i>pvaralNoData</i>	Variant array. 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. The array items are returned as a Long subtype.

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, lVal, lIcp, lAcct, lCust1, lCust2, _
    lCust3, lCust4) As Variant
Dim cData As HsvData, vaServers, vaUsers, vaTime
Dim vaActivity, vaVal, vaNoData, vaRet(), lItems
'm_cHsvSession is an HsvSession object reference.
Set cData = m_cHsvSession.Data
cData.GetCellHistory2 lScen, lYear, lPer, lEnt, MEMBERNOTUSED, _
    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)
        vaRet(i, 2) = vaVal(i)
    Next i
    GetCellDatesUsersValuesStr = vaRet
Else
    GetCellDatesUsersValuesStr = ""
End If
End Function
```

GetCellJournalEntries

Returns arrays containing the data in and IDs of the journal entries for a cell. The data is returned in two arrays; one array returns the data as a Double subtype and the second array returns the data as a String subtype.

The arrays have a one-to-one correspondence. For example, the third element in the journal ID array identifies the journal entry that contains the data in the third elements of the data arrays.

Syntax

```
<HsvData>.GetCellJournalEntries lScenario, lYear, lPeriod, lEntity,
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
pvaradData, pvarabstrData, pvaralJournalIDs
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.

Argument	Description
<i>IParent</i>	Long (ByVal). The member ID of the parent of the <i>IEntity</i> argument's entity.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pvaradData</i>	Variant array. Returns the data in the cell's journal entries as a Double subtype.
<i>pvarabstrData</i>	Variant array. Returns the data in the cell's journal entries as a String subtype.
<i>pvaralJournalIDs</i>	Variant array. Returns the IDs of the cell's journal entries. The array items are returned as a Long subtype.

Example

The following example prints a cell's journal IDs and data to the Immediate window. The example gets the member IDs of the cell's dimension members with calls to the user-defined `GetMemberID` function; for details on `GetMemberID`, see the example for `GetItemID`. These member IDs are then passed to `GetCellJournalEntries`.

```
Dim lScen As Long, lYear As Long, lPer As Long, lView As Long
Dim lEnt As Long, lPar As Long, lVal As Long, lICP As Long
Dim lCust1 As Long, lCust2 As Long, lCust3 As Long, lCust4 As Long
Dim lAcct As Long, vadData, vasData, vaIds
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lView = GetMemberID(DIMENSIONVIEW, "YTD")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lAcct = GetMemberID(DIMENSIONACCOUNT, "Sales")
lVal = GetMemberID(DIMENSIONVALUE, "<Entity Curr Adjs>")
lICP = GetMemberID(DIMENSIONICP, "[ICP None]")
lCust1 = GetMemberID(DIMENSIONCUSTOM1, "GolfBalls")
lCust2 = GetMemberID(DIMENSIONCUSTOM2, "Customer2")
lCust3 = GetMemberID(DIMENSIONCUSTOM3, "[None]")
lCust4 = GetMemberID(DIMENSIONCUSTOM4, "Increases")
m_chsvData.GetCellJournalEntries lScen, lYear, lPer, lEnt, _
lPar, lVal, lAcct, lICP, lCust1, lCust2, lCust3, lCust4, _
vadData, vasData, vaIds
' Make sure an array is returned.
If IsArray(vaIds) = True Then
    For i = LBound(vaIds) To UBound(vaIds)
        Debug.Print CStr(vaIds(i)) & " ID: " & vasData(i)
    Next i
```

End If

GetCellLineItems

Returns arrays of the data and descriptions for the specified cell's line items. Data is returned in a Double array.

Tip: To return data in a String array, use [GetTextCellLineItems](#).

Syntax

```
<HsvData>.GetCellLineItems lScenario, lYear, lPeriod, lEntity, lAccount,  
lICP, lCustom1, lCustom2, lCustom3, lCustom4, pvardData, pvarbstrDetails
```

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

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
------------------	--

<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
--------------	--

<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
----------------	--

<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
----------------	--

<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
-----------------	---

Tip: Use `HsvAccounts.UsesLineItems` to check whether an account supports line items.

<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
-------------	--

<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
-----------------	--

<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
-----------------	--

<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
-----------------	--

<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
-----------------	--

<i>pvardData</i>	VARIANT. Returns an array of Doubles that contain the line item data.
------------------	---

<i>pvarbstrDetails</i>	VARIANT. Returns an array of Strings that contain the line item descriptions.
------------------------	---

Example

This example prints a cell's line item descriptions and data to Visual Basic's Immediate window. The example assumes the variables passed to `GetCellLineItems` have been set to the cell's member IDs, and uses `UsesLineItems` to test whether the cell's Account dimension member supports line items. If line items are supported, the cell's descriptions and data are printed.

```
Dim bLineItems As Boolean, cAccounts As HsvAccounts  
Dim vaData, vaDescs, cData As HsvData  
'g_cSession is an HsvSession object reference  
Set cData = g_cSession.Data  
'g_cMetadata is an HsvMetadata object reference
```

```

Set cAccounts = g_cMetadata.Accounts
'Exit if the account does not support line items
cAccounts.UsesLineItems lAcct, bLineItems
If bLineItems = False Then Exit Sub
cData.GetCellLineItems lScen, lYear, lPer, lEnt, lAcct, _
    lIcp, lCust1, lCust2, lCust3, lCust4, vaData, vaDescs
' Make sure the cell is not empty
If IsArray(vaData) = True Then
    For i = LBound(vaData) To UBound(vaData)
        Debug.Print vaDescs(i) & ": " & vaData(i) & vbCrLf
    Next i
End If

```

GetCells

Returns the data and statuses of cells. `GetCells` returns the cells' data in one array and the statuses in a second array. The array containing the cells' data is returned as a `Double`.

`GetCells` takes arrays of dimension member IDs as arguments; these arrays must contain identical numbers of elements. The elements of these arrays have a one-to-one correspondence to the elements of the arrays that are returned; the first elements of the member ID arrays define the first cell to be returned, the second elements of the member ID arrays define the second cell to be returned, and so on.

Tip: To return cells' data in a Variant array, use `GetTextCells` instead of `GetCells`. To filter out cells that contain no data, zero, derived data, or that consist of invalid intersections, use `GetCellsWithRowSuppression`.

Syntax

```

<HsvData>.GetCells varalScenario, varalYear, varalPeriod, varalView,
varalEntity, varalParent, varalValue, varalAccount, varalICP, varalCustom1,
varalCustom2, varalCustom3, varalCustom4, pvaradData, pvaralStatus

```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varalView</i>	Long array (ByVal). The member IDs of the cells' View dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.

Argument	Description
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>pvaradData</i>	Variant array. Returns the data in the cells. The array items are returned as a Double subtype.
<i>pvaralStatus</i>	Variant array. Returns the cells' statuses. For details on cell statuses, see "About Cell Statuses" on page 275 . The array items are returned as a Long subtype.

Example

The following subroutine populates a ListBox controls with the data and status of the cells specified with the subroutine's arguments. The cell status strings are obtained by passing the status Longs returned by `GetCells` to `FormatStatusToText`.

```
Sub fillCellsLists(laScen() As Long, laYear() As Long, laPer() As Long, _
    laView() As Long, laEnt() As Long, laPar() As Long, laVal() As Long, _
    laAcct() As Long, laIcp() As Long, laCust1() As Long, _
    laCust2() As Long, laCust3() As Long, laCust4() As Long)
Dim cData As HsvData, vaData, vaStatus
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
cData.GetCells laScen, laYear, laPer, laView, laEnt, _
    laPar, laVal, laAcct, laIcp, laCust1, laCust2, laCust3, _
    laCust4, vaData, vaStatus
'Populate the list box
For i = LBound(vaData) To UBound(vaData)
lstDataStatus.AddItem vaData(i) & " - " & _
    cData.FormatStatusToText(vaStatus(i))
Next i
End Sub
```

GetCellsWithRowSuppression

Returns the data and statuses of cells, optionally excluding rows of cells that match specified criteria. You can exclude rows that contain no data, zero, and derived data, as well as rows consisting of invalid intersections. Data is returned in a Variant array of a Double subtype.

`GetCellsWithRowSuppression` takes arrays of dimension member IDs as arguments. These arrays must have a one-to-one correspondence to each other, as the corresponding array elements define the cells to be queried.

Tip: For a similar function that provides formatting and scaling options, see [GetTextCellsWithRowSuppression](#). This function returns data in a Variant array of a String subtype.

Syntax

```
<HsvData>.GetCellsWithRowSuppression varalScenario, varalYear,  
varalPeriod, varalView, varalEntity, varalParent, varalValue, varalAccount,  
varalICP, varalCustom1, varalCustom2, varalCustom3, varalCustom4,  
lSuppressOptions, lNumRows, pvaralNonSuppressedRowIndex, pvaradData,  
pvaralStatus
```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varalView</i>	Long array (ByVal). The member IDs of the cells' View dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>lSuppressOptions</i>	<p>Long (ByVal). The type of suppression to be applied. You can specify any combination of the following hexadecimal values:</p> <ul style="list-style-type: none">● 0x00. No suppression; data is returned for all cells specified by the dimension member arguments.● 0x01. Suppress rows in which all cells contain no data.● 0x02. Suppress rows in which all cells contain derived data.● 0x04. Suppress rows in which all cells contain a value of zero (0).● 0x08. Suppress rows in which all cells consist of invalid intersections. <p>For example, passing 0x01 suppresses rows consisting of cells that contain no data, while passing 0x05 suppresses rows consisting of cells that contain either zero or no data.</p>
<i>lNumRows</i>	<p>Long (ByVal). Specifies the number of rows that the point of view comprises.</p> <p>Caution! You must pass a valid number of rows; otherwise, an error will occur. For example, if you pass 3 and the member ID arrays cannot be resolved to three rows of data, an error occurs.</p>
<i>pvaralNonSuppressedRowIndex</i>	Variant array. Indicates whether a row is suppressed. Returns TRUE if the row is suppressed, FALSE otherwise.

Argument	Description
	<p>A row is suppressed only when all of the row's cells meet the <i>ISuppressOptions</i> argument's criteria.</p> <p>The array items are returned as a Boolean subtype.</p>
<i>pvaradData</i>	<p>Variante array. Returns data for the rows of cells that are not suppressed.</p> <p>Note: If only some of the cells in a row meet the <i>ISuppressOptions</i> argument's suppression criteria, these cells will not be suppressed. Only rows in which all the cells meet the criteria are suppressed.</p> <p>You can correlate this data with the members passed in the member ID arguments by looping through the array returned in the <i>pvarabRowsSuppressed</i> argument.</p> <p>The array items are returned as a Double subtype.</p>
<i>pvaralStatus</i>	<p>Variante array. Returns the statuses of the cells returned in the <i>pvaradData</i> argument. These two arrays have a one-to-one correspondence. For information on the values returned, see "About Cell Statuses" on page 275.</p> <p>The array items are returned as a Long subtype.</p>

Example

The following subroutine gets data for all accounts that have valid intersections with a given set of dimension members. `EnumAllMemberIDs` gets the member IDs of all the Account dimension members. A count of the Account members is passed to the custom `populateGlobalArrays` procedure, which populates the arrays that identify the non-Account dimension members. The example loops through the array returned by `GetCellsWithRowSuppression`'s *pvarabRowIsSuppressed* argument. When an element is FALSE, `GetLabel` gets the corresponding account's name, which is printed to Visual Basic's Immediate window along with the data.

```
Private Sub getValidAcctRows(lScen As Long, lYear As Long, _
    lPer As Long, lView As Long, lEnt As Long, lPar As Long, _
    lVal As Long, lICP As Long, lCust1 As Long, lCust2 As Long, _
    lCust3 As Long, lCust4 As Long)
    Dim lNumRows As Long, vaRowRetd, vaData, vaStatus
    Dim laAcct() As Long, sAcctName As String, lDataCount As Long
    Dim cHsvTreeInfo As IHsvTreeInfo
    Set cHsvTreeInfo = m_cHsvMetadata.Accounts
    cHsvTreeInfo.EnumAllMemberIDs vaAcctIDs
    lNumRows = UBound(vaAcctIDs)
    ReDim laAcct(lNumRows)
    For i = 0 To lNumRows
        laAcct(i) = vaAcctIDs(i)
    Next i
    populateGlobalArrays lNumRows, lScen, lYear, lPer, lView, _
        lEnt, lPar, lVal, lICP, lCust1, lCust2, lCust3, lCust4
    m_cHsvData.GetCellsWithRowSuppression m_laScen, m_laYear, _
        m_laPer, m_laView, m_laEnt, m_laPar, m_laVal, laAcct, _
        m_laICP, m_laCust1, m_laCust2, m_laCust3, m_laCust4, &H8, _
        lNumRows + 1, vaRowRetd, vaData, vaStatus
    lDataCount = 0
    For i = 0 To UBound(vaRowRetd)
        If vaRowRetd(i) = False Then
            cHsvTreeInfo.GetLabel laAcct(i), sAcctName
```

```

        Debug.Print sAcctName & ": " & vaData(lDataCount)
        lDataCount = lDataCount + 1
    End If
Next i
End Sub

```

GetCellsWithRowSuppression2

Returns the data and statuses of cells, optionally excluding rows of cells that match specified criteria. You can exclude rows that contain no data, zero, and derived data, as well as rows consisting of invalid intersections. Data is returned in a Variant array of a Double subtype.

`GetCellsWithRowSuppression2` takes arrays of dimension member IDs as arguments. These arrays must have a one-to-one correspondence to each other, as the corresponding array elements define the cells to be queried.

Tip: For a similar function that provides formatting and scaling options, see [GetTextCellsWithRowSuppression2](#). This function returns data in a Variant array of a String subtype.

Syntax

```

<HsvData>.GetCellsWithRowSuppression2 varalScenario, varalYear,
varalPeriod, varalView, varalEntity, varalParent, varalValue, varalAccount,
varalICP, varalCustom1, varalCustom2, varalCustom3, varalCustom4,
lSuppressOptions, lNumRows, pvaravbRowIsSuppressed, pvaradData,
pvaralStatus

```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varalView</i>	Long array (ByVal). The member IDs of the cells' View dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.

Argument	Description
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>ISuppressOptions</i>	<p>Long (ByVal). The type of suppression to be applied. You can specify any combination of the following hexadecimal values:</p> <ul style="list-style-type: none"> ● 0x00. No suppression—data will be returned for all cells specified by the dimension member arguments. ● 0x01. Suppress rows in which all cells contain no data. ● 0x02. Suppress rows in which all cells contain derived data. ● 0x04. Suppress rows in which all cells contain a value of zero (0). ● 0x08. Suppress rows in which all cells consist of invalid intersections. <p>For example, passing 0x01 suppresses rows consisting of cells that contain no data, while passing 0x05 suppresses rows consisting of cells that contain either zero or no data.</p>
<i>INumRows</i>	<p>Long (ByVal). Specifies the number of rows that the point of view comprises.</p> <p>Caution! You must pass a valid number of rows; otherwise, an error will occur. For example, if you pass 3, and the member ID arrays cannot be resolved to three rows of data, an error occurs.</p>
<i>pvarvbRowsSuppressed</i>	<p>Variant array. Indicates whether a row is to be suppressed. Set to TRUE if the row is to be suppressed, FALSE otherwise.</p> <p>The array items are returned as a Boolean subtype.</p>
<i>pvaradData</i>	<p>Variant array. Returns data for the rows of cells that are not suppressed. If all rows are suppressed, returns empty array.</p> <p>Note: If only some of the cells in a row meet the <i>ISuppressOptions</i> argument's suppression criteria, these cells will not be suppressed. Only rows in which all the cells meet the criteria are suppressed.</p> <p>You can correlate this data with the members passed in the member ID arguments by looping through the array returned in the <i>pvarabRowsSuppressed</i> argument.</p> <p>The array items are returned as a Double subtype.</p>
<i>pvaralStatus</i>	<p>Variant array. Returns the statuses of the cells that are not suppressed. If all rows are suppressed, returns empty array. These two arrays have a one-to-one correspondence. For information on the values returned, see “About Cell Statuses” on page 275.</p> <p>The array items are returned as a Long subtype.</p>

GetCountOfAttachedDocumentsToCell

Returns a count of the documents attached to a cell.

Syntax

```
<HsvData>.GetCountOfAttachedDocumentsToCell(lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4)
```

Argument Description

<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the parent of the <i>IEntity</i> argument's entity.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.

Return Value

Long. Returns the number of documents attached to the specified cell.

GetCurrencyCube

Returns an object reference to the *HsvCurrencyCube* object. The object reference provides access to the subcube identified by the Scenario, Year, Entity, and Value dimension member IDs passed to *GetCurrencyCube*.

Syntax

```
<HsvData>.GetCurrencyCube lScenario, lYear, lEntity, lValue, ppIUnkCube
```

Argument Description

<i>IScenario</i>	Long (ByVal). The member ID of the subcube's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the subcube's Year dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the subcube's Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the subcube's Value dimension member.
<i>ppIUnkCube</i>	<i>HsvCurrencyCube</i> object. Returns an <i>HsvCurrencyCube</i> object reference.

Example

GetCurrencyCube is used in the [Example](#) for *HsvCurrencyCube.GetCell*.

GetDataForAllMetadataCombinations

For internal use.

GetLineItems

Deprecated – superseded by [GetCellLineItems](#).

GetMaxCellTextSize

Returns the maximum number of characters that can be inserted as cell text.

Tip: To test whether a string exceeds the maximum cell text size, use [IsValidCellText](#).

Syntax

```
<HsvData>.GetMaxCellTextSize()
```

Return Value

Long. Returns the maximum number of characters allowed for cell text.

GetMembersThatHaveData

Returns an array of dimension member IDs (as specified by *IDimID*) that contain data for the specified POV.

Syntax

```
<HsvData>.GetMembersThatHaveData lScenario, lYear, lPeriod, lEntity,  
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,  
lDimID, vbConsiderBaseMembersOnly,  
varalMembersAutomaticallyIncludedInOutput, pvaralMembers
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member. Must be a valid member or MEMBERALL, or MEMBERNOTUSED.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.

Argument	Description
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member. Must be a valid member or MEMBERALL, or MEMBERNOTUSED.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member. Must be a valid member or MEMBERALL, or MEMBERNOTUSED.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member. Must be a valid member or MEMBERALL, or MEMBERNOTUSED.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member. Must be a valid member or MEMBERALL, or MEMBERNOTUSED.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member. Must be a valid member or MEMBERALL, or MEMBERNOTUSED.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member. Must be a valid member or MEMBERALL, or MEMBERNOTUSED.
<i>IDimID</i>	Long (ByVal). The dimension for which you want a list of members. You can use the following dimensions: Account, ICP, Custom 1, Custom 2, Custom 3, or Custom 4. For valid values see “Dimension ID Constants” on page 834 .
<i>vbConsiderBaseMembersOnly</i>	Boolean (ByVal). Determines if base members only are tested.
<i>varalMembersAutomaticallyIncludedInOutput</i>	Variant (ByVal). A list of members to be included in output.
<i>pvaralMembers</i>	Variant. Returns the filtered list of member IDs that contain data for any of the specified POVs.

GetNodeCube

Returns an object reference to the HsvNodeCube object. The object reference provides access to the subcube identified by the Scenario, Year, and parent and child Entity dimension member IDs passed to GetNodeCube.

Syntax

```
<HsvData>.GetNodeCube lScenario, lYear, lEntity, lParent, ppIUnkCube
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the subcube's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the subcube's Year dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the subcube's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the parent of the <i>IEntity</i> argument's entity.
<i>ppIUnkCube</i>	HsvNodeCube object. Returns an HsvNodeCube object reference.

Example

GetCurrencyCube is used in the [Example](#) for HsvNodeCube.GetCell.

GetPhaseSubmissionGridForGivenScenarioPeriod

Returns an array representing the submission groups assigned to the specified combinations of Scenario dimension member, Period dimension members, and submission phases. The input Period member IDs and phase IDs, and the returned submission group IDs, are in arrays that have a one-to-one correspondence.

Syntax

```
<HsvData>.GetPhaseSubmissionGridForGivenScenarioPeriod lScenario,  
varalPeriodIDs, varalPhaseIDs, pvarbstrGroupInfo
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>varalPeriodIDs</i>	Long array (ByVal). The member IDs of the Period dimension members.
<i>varalPhaseIDs</i>	Long array (ByVal). The phase IDs of the submission phases.
<i>pvarbstrGroupInfo</i>	Variant array. Returns the IDs of the submission groups. If no submission group is assigned to a period-submission group combination, the corresponding array item returns an empty string Note: If all submission groups are assigned to a period-submission group combination, the asterisk (*) character is returned instead of an ID. The array's subtype is String.

Example

The following function returns the ID of the submission group assigned to a submission phase for a given scenario and period.

```
Function GetGroupForPhase(lScen As Long, lPer As Long, lPhase As Long) _  
    As String  
    Dim cData As HsvData, laPers(0) As Long, laPhaseId(0) As Long, vaGroup  
    Set cData = g_cSession.Data  
    laPers(0) = lPer  
    laPhaseId(0) = lPhase  
    cData.GetPhaseSubmissionGridForGivenScenarioPeriod lScen, laPers, _  
        laPhaseId, vaGroup  
    GetGroupForPhase = vaGroup(0)  
End Function
```

GetStatus

Returns the transaction status, metadata status, and calculation status of a cell. The *bIncludeTransType* argument enables you to exclude the transaction status from the return value.

Note: To return additional cell status information, use [GetStatusEx](#).

Syntax

```
<HsvData>.GetStatus(lScenario, lYear, lPeriod, lView, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, bIncludeTransType, plStatus)
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>bIncludeTransType</i>	Boolean (ByVal). Determines whether the cell's transaction status will be returned in the <i>plStatus</i> argument. Specify TRUE to return the transaction status, otherwise FALSE.
<i>plStatus</i>	Long. Returns the cell's status. For details on cell statuses, see "About Cell Statuses" on page 275 .

Return Value

Integer. Indicates the success of the function call; returns 0 for success or an error code for failure.

Example

The following example tests a cell's status to determine whether the cell requires consolidation. `GetStatus`'s returned cell status is assigned to the `lStat` variable. The `If` statement performs a bitwise comparison to see if the cell's status includes the calculation status for requiring consolidation, which is represented by the `CELLSTATUS_NEEDSCONSOLIDATION` constant. If `lStat` contains this status, then any code placed within the `If` structure would be executed. The example assumes that the variables in `GetStatus`'s member ID arguments were passed from another procedure.


```

Dim cData as HsvData, lStat as Long
Set cData = m_cHsvSession.Data
cData.GetStatus m_lScen, m_lYear, m_lPer, m_lView, m_lEnt, _
m_lPar, m_lVal, m_lAcct, m_lICP, m_lCust1, m_lCust2, _
m_lCust3, m_lCust4, True, lStat
If (CELLSTATUS_NEEDSCONSOLIDATION And lStat) Then
    ...
End If

```

GetStatusEx

Returns the transaction status, metadata status, and calculation status of a cell, as well as additional status information such as whether the cell supports intercompany transactions. The *bIncludeTransType* argument enables you to exclude the transaction status from the return value.

Syntax

```

<HsvData>.GetStatusEx lScenario, lYear, lPeriod, lView, lEntity, lParent,
lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
bIncludeTransType, plStatus, plExtendedStatus

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>bIncludeTransType</i>	Boolean (ByVal). Determines whether the cell's transaction status will be returned in the <i>plStatus</i> argument. Specify TRUE to return the transaction status, otherwise FALSE.
<i>plStatus</i>	Long. Returns the cell's status. For details on cell statuses, see "About Cell Statuses" on page 275 .

Argument	Description
<i>pExtendedStatus</i>	Long. Returns additional status information. Valid values are represented by the <code>HFMConstants</code> type library constants listed in “Additional Status Information Constant” on page 847 .

GetStatusUsingPhaseID

Returns the transaction status, metadata status, and calculation status of either a cell or a submission phase. You must pass one of the following:

- To get the status of a submission phase, pass the phase ID to the *lPhaseID* argument and the `HFMConstant` `MEMBERNOTUSED` to the *Account*, *Intercompany Partner*, and *Custom dimension* arguments.
- To get the status of a cell, pass member IDs to all dimension member arguments and `MEMBERNOTUSED` to the *lPhaseID* argument.

Note: The *bIncludeTransType* argument enables you to exclude the transaction status from the return value.

Syntax

```
<HsvData>.GetStatusUsingPhaseID lScenario, lYear, lPeriod, lView, lEntity,
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
lPhaseID, bIncludeTransType, plStatus, psReturn
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> • To get a cell's status, pass the member ID of the cell's Account dimension member. • To get a submission phase's status, pass the <code>HFMConstant</code> <code>MEMBERNOTUSED</code>.
<i>lICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> • To get a cell's status, pass the member ID of the cell's Intercompany Partner dimension member. • To get a submission phase's status, pass the <code>HFMConstant</code> <code>MEMBERNOTUSED</code>.
<i>lCustom1</i>	Long (ByVal). Pass one of the following:

Argument	Description
	<ul style="list-style-type: none"> ● To get a cell's status, pass the member ID of the cell's Custom1 dimension member. ● To get a submission phase's status, pass the HFMConstant <code>MEMBERNOTUSED</code>.
<i>ICustom2</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● To get a cell's status, pass the member ID of the cell's Custom2 dimension member. ● To get a submission phase's status, pass the HFMConstant <code>MEMBERNOTUSED</code>.
<i>ICustom3</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● To get a cell's status, pass the member ID of the cell's Custom3 dimension member. ● To get a submission phase's status, pass the HFMConstant <code>MEMBERNOTUSED</code>.
<i>ICustom4</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● To get a cell's status, pass the member ID of the cell's Custom4 dimension member. ● To get a submission phase's status, pass the HFMConstant <code>MEMBERNOTUSED</code>.
<i>IPhaseID</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● To get a cell's status, pass the HFMConstant <code>MEMBERNOTUSED</code>. ● To get a submission phase's status, pass the phase ID.
<i>blIncludeTransType</i>	Boolean (ByVal). Determines whether the cell's transaction status will be returned in the <i>plStatus</i> argument. Specify TRUE to return the transaction status, otherwise FALSE.
<i>plStatus</i>	Long. Returns the cell's status. For details on cell statuses, see “About Cell Statuses” on page 275 .
<i>psReturn</i>	Integer. <i>For internal use.</i>

GetTextCell

Returns the data in a cell in a String data type. `GetTextCell` also returns the cell's status, and provides arguments to specify the returned data's scale and decimal precision.

Tip: To return a cell's data as a Double data type, use `GetCell` instead of `GetTextCell`.

Syntax

```
<HsvData>.GetTextCell lScenario, lYear, lPeriod, lView, lEntity, lParent,
lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
sNumDecimals, sScale, pbstrData, plStatus
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.

Argument	Description
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>sNumDecimals</i>	Integer (ByVal). The number of decimal places that the <i>pbstrData</i> argument's return value will contain. To use the application's default number of decimals, pass the HFMConstants type library constant <code>DEFAULT_NUM_DECIMALS</code> , described in "Number Defaults Constants" on page 881 .
<i>sScale</i>	Integer (ByVal). The degree of scaling to be applied to the <i>pbstrData</i> argument's return value. Rules for this argument are described below: <ul style="list-style-type: none"> ● Pass 0 to return the number without scaling it. ● Each integer greater than 0 scales the number by dividing it by 10. Pass 1 to divide the number by 10, 2 to divide the number by 100, and so on. For example, if you pass 3, and the cell contains 1,000,000.00, the <i>pbstrData</i> argument would return 1,000.00. ● Similarly, each integer less than 0 scales the number by multiplying it by 10. Pass -1 to multiply the number by 10, -2 to multiply the number by 100, and so on. For example, if you pass -3, and the cell contains 1,000,000.00, the <i>pbstrData</i> argument would return 1,000,000,000.00. ● To use the system's default scaling, pass the HFMConstants type library constant <code>DEFAULT_NUM_DECIMALS</code>, described in "Number Defaults Constants" on page 881.
<i>pbstrData</i>	String. Returns the data in the cell.
<i>plStatus</i>	Long. Returns the cell's status. For details on cell statuses, see "About Cell Statuses" on page 275 .

Example

The following subroutine populates two Label controls with a cell's data and status. The subroutine takes the member IDs of the cell's dimension members. The cell status string is obtained by passing the status Long returned by `GetTextCell` to `FormatStatusToText`.

```
Sub fillCellLabels(lScen As Long, lYear As Long, lPer As Long, _
    lView As Long, lEnt As Long, lPar As Long, lVal As Long, _
    lAcct As Long, lICP As Long, lCust1 As Long, lCust2 As Long, _
    lCust3 As Long, lCust4 As Long)
Dim cData As HsvData, sData As String, lStatus As Long
Dim sStatus As String
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
```

```

cData.GetTextCell lScen, lYear, lPer, lView, lEnt, lPar, _
    lVal, lAcct, lICP, lCust1, lCust2, lCust3, lCust4, _
    DEFAULT_NUM_DECIMALS, DEFAULT_SCALE, sData, lStatus
sStatus = cData.FormatStatusToText(lStatus)
'Populate the labels
lblData.Caption = sData
lblStatus.Caption = sStatus
End Sub

```

GetTextCellLineItems

Returns arrays of the data and descriptions for the specified cell's line items. Data is returned in a String array.

Tip: To return data in a Double array, use [GetCellLineItems](#).

Syntax

```

<HsvData>.GetTextCellLineItems lScenario, lYear, lPeriod, lEntity,
lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, pvarabstrData,
pvarabstrDetails

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member. Tip: Use <code>HsvAccounts.UsesLineItems</code> to check whether an account supports line items.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pvarabstrData</i>	Variant. Returns an array of Strings that contain the line item data.
<i>pvarabstrDetails</i>	Variant. Returns an array of Strings that contain the line item descriptions.

Example

This example prints a cell's line item descriptions and data to Visual Basic's Immediate window. The example assumes the variables passed to `GetTextCellLineItems` have been set to the

cell's member IDs, and uses [UsesLineItems](#) to test whether the cell's Account dimension member supports line items. If line items are supported, the cell's descriptions and data are printed.

```
Dim bLineItems As Boolean, cAccounts As HsvAccounts
Dim vaData, vaDescs, cData As HsvData
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
'g_cMetadata is an HsvMetadata object reference
Set cAccounts = g_cMetadata.Accounts
'Exit if the account does not support line items
cAccounts.UsesLineItems lAcct, bLineItems
If bLineItems = False Then Exit Sub
cData.GetTextCellLineItems lScen, lYr, lPer, lEnt, lAcct, _
    lICP, lCust1, lCust2, lCust3, lCust4, vaData, vaDescs
' Make sure the cell is not empty
If IsArray(vaData) = True Then
    For i = LBound(vaData) To UBound(vaData)
        Debug.Print vaDescs(i) & ": " & vaData(i) & vbCrLf
    Next i
End If
```

GetTextCells

Returns Variant arrays containing the data and statuses of cells. `GetTextCells` also enables you to specify the scaling and the number of decimals for the returned data.

`GetTextCells` takes arrays of dimension member IDs as arguments; these arrays must contain identical numbers of elements. The elements of these arrays have a one-to-one correspondence to the elements of the arrays that are returned; the first elements of the member ID arrays define the first cell to be returned, the second elements of the member ID arrays define the second cell to be returned, and so on.

Tip: To return cells' data in a Double array, use [GetCells](#) instead of `GetTextCells`. To filter out cells that contain no data, zero, derived data, or that consist of invalid intersections, use [GetTextCellsWithRowSuppression](#).

Syntax

```
<HsvData>.GetTextCells varalScenario, varalYear, varalPeriod, varalView,
varalEntity, varalParent, varalValue, varalAccount, varalICP, varalCustom1,
varalCustom2, varalCustom3, varalCustom4, sNumDecimals As Integer, sScale
As Integer, pvarabstrData, pvaralStatus
```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.

Argument	Description
<i>varalView</i>	Long array (ByVal). The member IDs of the cells' View dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>sNumDecimals</i>	Integer (ByVal). The number of decimal places that the amounts in the <i>pvarabstrData</i> argument's return value will contain. To use the application's default number of decimals, pass the HFMCConstants type library constant <code>DEFAULT_NUM_DECIMALS</code> , described in "Number Defaults Constants" on page 881 .
<i>sScale</i>	Integer (ByVal). Specifies the degree of scaling to be applied to the return value. To use the application's default scaling, pass the HFMCConstants type library constant <code>DEFAULT_SCALE</code> , described in "Number Defaults Constants" on page 881 .
<i>pvarabstrData</i>	Variant array. Returns the data in the cells identified by the dimension arguments. The array items are returned as a String subtype.
<i>pvaralStatus</i>	Variant array. Returns the calculation statuses of the cells. For details on calculation statuses, see "About Cell Statuses" on page 275 . The array items are returned as a Long subtype.

Example

The following example creates a function that takes arrays of cells' member IDs and returns the cells' data scaled by a tenth.

```
Function scaleCellsByTenth(laScen() As Long, laYear() As _
    Long, laPer() As Long, laView() As Long, laEnt() As Long, _
    laPar() As Long, laVal() As Long, laAcct() As Long, laICP() _
    As Long, laCust1() As Long, laCust2() As Long, laCust3() _
    As Long, laCust4() As Long) _
    As Variant
Dim cData As HsvData, vaData, vaStatus
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
cData.GetTextCells laScen, laYear, laPer, laView, laEnt, _
    laPar, laVal, laAcct, laICP, laCust1, laCust2, laCust3, _
    laCust4, DEFAULT_NUM_DECIMALS, 1, vaData, vaStatus
scaleCellsByTenth = vaData
```

End Function

GetTextCellsWithRowSuppression

Returns the data and statuses of cells, optionally applying scaling and formatting, and excluding rows of cells that match specified criteria. You can exclude rows that contain no data, zero, and derived data, as well as rows consisting of invalid intersections. Data is returned in a Variant array of a String subtype.

`GetTextCellsWithRowSuppression` takes arrays of dimension member IDs as arguments. These arrays must have a one-to-one correspondence to each other, as the corresponding array elements define the cells to be queried.

Tip: For a similar function that returns data as a Double subtype and does not provide formatting and scaling options, see [GetCellsWithRowSuppression](#). This function returns data in a Variant array of a Double subtype.

Syntax

```
<HsvData>.GetTextCellsWithRowSuppression varalScenario, varalYear,  
varalPeriod, varalView, varalEntity, varalParent, varalValue, varalAccount,  
varalICP, varalCustom1, varalCustom2, varalCustom3, varalCustom4,  
sNumDecimals, sScale, lSuppressOptions, lNumRows, pvarabRowIsSuppressed,  
pvarabstrData, pvaralStatus
```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varalView</i>	Long array (ByVal). The member IDs of the cells' View dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.

Argument	Description
<i>sNumDecimals</i>	<p>Integer (ByVal). The number of decimal places that the data returned in the <i>pvarabstrData</i> argument will contain.</p> <p>To use the application's default number of decimals, pass the HFMConstants type library constant <code>DEFAULT_NUM_DECIMALS</code>, described in "Number Defaults Constants" on page 881.</p>
<i>sScale</i>	<p>Integer (ByVal). Indicates the degree of scaling to apply to the data returned in the <i>pvarabstrData</i> argument. To leave the data unscaled, pass 0. To scale the data, each whole-number increment over 0 scales by ten. In other words, passing 1 scales by ten, passing 2 scales by a hundred, and so on.</p> <p>To apply the application's default scaling, pass the HFMConstants type library constant <code>DEFAULT_SCALE</code>, described in "Number Defaults Constants" on page 881.</p>
<i>ISuppressOptions</i>	<p>Long (ByVal). The type of suppression to be applied. You can specify any combination of the following hexadecimal values:</p> <ul style="list-style-type: none"> ● 0x00. No suppression—data will be returned for all cells specified by the dimension member arguments. ● 0x01. Suppress rows in which all cells contain no data. ● 0x02. Suppress rows in which all cells contain derived data. ● 0x04. Suppress rows in which all cells contain a value of zero (0). ● 0x08. Suppress rows in which all cells consist of invalid intersections. <p>For example, passing 0x01 suppresses rows consisting of cells that contain no data, while passing 0x05 suppresses rows consisting of cells that contain either zero or no data.</p>
<i>INumRows</i>	<p>Long (ByVal). Specifies the number of rows that the point of view comprises.</p> <p>Caution! You must pass a valid number of rows; otherwise, an error will occur. For example, if you pass 3 and the member ID arrays cannot be resolved to three rows of data, an error occurs.</p>
<i>pvarabRowsSuppressed</i>	<p>Variant array. Indicates whether a row is suppressed. Returns TRUE if the row is suppressed, FALSE otherwise.</p> <p>A row is suppressed only when all of the row's cells meet the <i>ISuppressOptions</i> argument's criteria.</p> <p>The array items are returned as a Boolean subtype.</p>
<i>pvarabstrData</i>	<p>Variant array. Returns data for the rows of cells that are not suppressed.</p> <p>Note: If only some of the cells in a row meet the <i>ISuppressOptions</i> argument's suppression criteria, these cells will not be suppressed. Only rows in which all the cells meet the criteria are suppressed.</p> <p>You can correlate this data with the members passed in the member ID arguments by looping through the array returned in the <i>pvarabRowsSuppressed</i> argument.</p> <p>The array items are returned as a String subtype.</p>
<i>pvaralStatus</i>	<p>Variant array. Returns the statuses of the cells returned in the <i>pvarabstrData</i> argument. These two arrays have a one-to-one correspondence. For information on the values returned, see "About Cell Statuses" on page 275.</p> <p>The array items are returned as a Long subtype.</p>

Example

The following subroutine gets data for all accounts that have valid intersections with a given set of dimension members, and scales the data according to the scale factor passed in.

`EnumAllMemberIDs` gets the member IDs of all the Account dimension members. A count of the Account members is passed to the custom `populateGlobalArrays` procedure, which populates the arrays that identify the non-Account dimension members. The example loops through the array returned by `GetTextCellsWithRowSuppression`'s *pvarabRowIsSuppressed* argument. When an element is FALSE, `GetLabel` gets the corresponding account's name, which is printed to Visual Basic's Immediate window along with the data.

```
Private Sub getValidAcctRowsScale(lScen As Long, lYear As Long, _
lPer As Long, lView As Long, lEnt As Long, lPar As Long, _
lVal As Long, lICP As Long, lCust1 As Long, lCust2 As Long, _
lCust3 As Long, lCust4 As Long, iScale As Integer)
Dim lNumRows As Long, vaRowRetd, vaData, vaStatus
Dim laAcct() As Long, sAcctName As String, lDataCount As Long
Dim cHsvTreeInfo As IHsvTreeInfo
Set cHsvTreeInfo = m_cHsvMetadata.Accounts
cHsvTreeInfo.EnumAllMemberIDs vaAcctIDs
lNumRows = UBound(vaAcctIDs)
ReDim laAcct(lNumRows)
For i = 0 To lNumRows
    laAcct(i) = vaAcctIDs(i)
Next i
populateGlobalArrays lNumRows, lScen, lYear, lPer, lView, _
lEnt, lPar, lVal, lICP, lCust1, lCust2, lCust3, lCust4
m_cHsvData.GetTextCellsWithRowSuppression m_laScen, m_laYear, _
m_laPer, m_laView, m_laEnt, m_laPar, m_laVal, laAcct, _
m_laICP, m_laCust1, m_laCust2, m_laCust3, m_laCust4, _
DEFAULT_NUM_DECIMALS, iScale, &H8, lNumRows + 1, vaRowRetd, _
vaData, vaStatus
lDataCount = 0
For i = 0 To UBound(vaRowRetd)
    If vaRowRetd(i) = False Then
        cHsvTreeInfo.GetLabel laAcct(i), sAcctName
        Debug.Print sAcctName & ": " & vaData(lDataCount)
        lDataCount = lDataCount + 1
    End If
Next i
End Sub
```

GetTextCellsWithRowSuppression2

Returns the data and status for rows that are not suppressed.

Syntax

```
<HsvData>.GetTextCellsWithRowSuppression2 varalScenario, varalYear,
varalPeriod, varalView, varalEntity, varalParent, varalValue, varalAccount,
varalICP, varalCustom1, varalCustom2, varalCustom3, varalCustom4,
sNumDecimals, sScale, lSuppressOptions, lNumRows, pvaravbRowIsSuppressed,
pvarabstrData, pvaralStatus
```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varalView</i>	Long array (ByVal). The member IDs of the cells' View dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>sNumDecimals</i>	<p>Integer (ByVal). The number of decimal places that the data returned in the <i>pvarabstrData</i> argument contains.</p> <p>To use the application's default number of decimals, pass the HFMConstants type library constant <code>DEFAULT_NUM_DECIMALS</code>, described in "Number Defaults Constants" on page 881.</p>
<i>sScale</i>	<p>Integer (ByVal). Indicates the degree of scaling to apply to the data returned in the <i>pvarabstrData</i> argument. To leave the data unscaled, pass 0. To scale the data, each whole-number increment over 0 scales by 10. In other words, passing 1 scales by 10, passing 2 scales by 100, and so on.</p> <p>To apply the application's default scaling, pass the HFMConstants type library constant <code>DEFAULT_SCALE</code>, described in "Number Defaults Constants" on page 881.</p>
<i>ISuppressOptions</i>	<p>Long (ByVal). The type of suppression to be applied. You can specify any combination of the following hexadecimal values:</p> <ul style="list-style-type: none"> ● 0x00. No suppression; data is returned for all cells specified by the dimension member arguments. ● 0x01. Suppress rows in which all cells contain no data. ● 0x02. Suppress rows in which all cells contain derived data. ● 0x04. Suppress rows in which all cells contain a value of zero (0). ● 0x08. Suppress rows in which all cells consist of invalid intersections. <p>For example, passing 0x01 suppresses rows consisting of cells that contain no data, while passing 0x05 suppresses rows consisting of cells that contain either zero or no data.</p>
<i>INumRows</i>	Long (ByVal). Specifies the number of rows that the point of view comprises.

Argument	Description
	<p>Caution! You must pass a valid number of rows; otherwise, an error will occur. For example, if you pass 3, and the member ID arrays cannot be resolved to three rows of data, an error occurs.</p>
<i>pvaravbRowsSuppressed</i>	<p>Variant array. Indicates whether a row is suppressed. Returns TRUE if the row is suppressed, FALSE otherwise.</p> <p>A row is suppressed only when all of the row's cells meet the <i>ISuppressOptions</i> argument's criteria.</p> <p>The array items are returned as a Boolean subtype.</p>
<i>pvarabstrData</i>	<p>Variant array. Returns data for the rows of cells that are not suppressed.</p> <p>Note: If only some of the cells in a row meet the <i>ISuppressOptions</i> argument's suppression criteria, these cells are not suppressed. Only rows in which all the cells meet the criteria are suppressed.</p> <p>You can correlate this data with the members passed in the member ID arguments by looping through the array returned in the <i>pvarabRowsSuppressed</i> argument.</p> <p>The array items are returned as a String subtype.</p>
<i>pvaralStatus</i>	<p>Variant array. Returns the statuses of the cells returned in the <i>pvarabstrData</i> argument. These two arrays have a one-to-one correspondence. For information on the values returned, see “About Cell Statuses” on page 275.</p> <p>The array items are returned as a Long subtype.</p>

GetTextLineItems

Deprecated – superseded by [GetTextCellLineItems](#).

GetTransactionData

Populates an *HsvTransactionData* object with an array of transaction data. Selection criteria for the data is specified with *HsvTransactionData.SetQueryItem*.

Caution! Do not confuse this method with the *HsvTransactionData* object's *GetTransactionData* method.

Syntax

```
<HsvData>.GetTransactionData pUnkTransactionData
```

Argument	Description
<i>pUnkTransactionData</i>	<p><i>HsvTransactionData</i> object (ByVal). Pass an <i>HsvTransactionData</i> object for which you have already set selection criteria. For information on using <i>GetTransactionData</i> with the <i>HsvTransactionData</i> object's methods, see “HsvTransactionData Object Methods” on page 667.</p>

Example

GetTransactionData is used in the [Example](#) for HsvTransactionData.GetTransactionData.

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

```
<HsvData>.GetUnassignedGroups lScenario, lPeriodID, pbstrMetaNoPhase,  
pbstrPhaseNoMeta
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lPeriodID</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>pbstrMetaNoPhase</i>	String. Returns the names of the submission groups assigned to dimension members but not to submission phases.
<i>pbstrPhaseNoMeta</i>	String. Returns the names of the submission groups assigned to submission phases but not to members.

Example

The following function returns the names of submission groups assigned to submission phases but not to members.

```
Function getGroupsNoMembers(lScen As Long, lPer As Long)  
Dim cData As HsvData, sMetaNoPhase As String, sRet As String  
'g_csession is an HsvSession object reference  
Set cData = g_cSession.Data  
cData.GetUnassignedGroups lScen, lPer, sMetaNoPhase, sRet  
getGroupsNoMembers = sRet  
End Function
```

GetURLByName

Returns a single XML block representing the specified URL.

Syntax

```
<HsvData>.GetURLByName bstrURLName, pbstrURL
```

Argument	Description
<i>bstrURLName</i>	String (ByVal).The URL name.

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

<i>pbstrURL</i>	String. A XML block representing the URL link and its multi-language display names.
-----------------	---

Examples

Oracle Hyperion Smart View for Office, Fusion Edition requires an additional function call to obtain the URL because its *HFMOfficeProvider* module cannot cache previous queries. Financial Management server will provide a function to retrieve the URL by specifying a URL name. Other clients should be able to get enough information using function *GetURLsForCell*. The following example will return a single XML block representing the URL with the specified URL name “ERPI_Sales.”

```
GetURLByName (_T("ERPI_Sales"), pbstrURL);
```

An example of the returned BSTR array *pbstrURL* would be:

```
<foldercontents path="/">
  <resource name="Sales Report" description="" type="application/x-hyperion-
applicationbuilder-report">
    <attribute name="name" type="string" xml:lang="fr" value="Rapport de
ventes" />
    <attribute name="name" type="string" xml:lang="es" value="Informe de
ventas" />
    <action name="Display Drill-Back" description="Launch Drill-Back"
shortdesc="drill-back">
      <url>./HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?
&fdmAppName=CommaFDM&fdmTargetAppName=Comma95&sso_token=
$SSO_TOKEN&sso_username=$SSO_USERNAME&sso_password=$SSO_PASSWORD
&$ATTR(id)&RCP_VERSION=$RCP_VERSION$</url>
    </action>
  </resource>
</foldercontents>
```

GetURLsForCell

Retrieves all URLs whose regions cover a specified POV.

Syntax

```
<HsvData>.GetURLsForCell bstrScenario, bstrYear, bstrPeriod, bstrEntity,
bstrAccount, pvarabstrURLNames, pvarabstrURLs
```

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

<i>bstrScenario</i>	String (ByVal). The Scenario dimension.
---------------------	---

<i>bstrYear</i>	String (ByVal). The Year dimension.
-----------------	-------------------------------------

<i>bstrPeriod</i>	String (ByVal). The Period dimension.
-------------------	---------------------------------------

<i>bstrEntity</i>	String (ByVal). The Entity dimension.
-------------------	---------------------------------------

Argument	Description
<i>bstrAccount</i>	String (ByVal). The Account dimension.
<i>pvarabstrURLNames</i>	Variant array. Array of URL names.
<i>pvarabstrURLs</i>	Variant array. Array of XML blocks representing URL links and their multi-language display names.

Example

The POV should contain exactly one element in each dimension. The function will not support *MEMBERALL* or member lists. This function will perform a cache lookup on the region cache for all URL names and URLs matching this POV. The following example will return a list of all URL names and XML blocks for URL for the given POV scenario “Actual,” year “2008,” period “January,” entity “Stamford,” and account “Sales.”

```
GetURLsForCell (_T("Actual"), _T("2008"), _T("January"),
_T("Stamford"),_T("Sales"), pvarabstrURLNames, pvarabstrURLs);
```

An example of the returned BSTR array *pvarabstrURLNames* would contain “ERPI_SalesReport” and “ERPI_EUSalesReport.” An example of the returned BSTR array *pvarabstrURLs* would contain two BSTR. The first BSTR looks like the following:

```
<foldercontents path="/">
  <resource name="Sales Report" description="" type="application/x-hyperion-
applicationbuilder-report">
    <attribute name="name" type="string" xml:lang="fr" value="Rapport de
ventes" />
    <attribute name="name" type="string" xml:lang="es" value="Informe de
ventas" />
    <action name="Display Drill-Back" description="Launch Drill-Back"
shortdesc="drill-back">
      <url>./HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?
&fdmAppName=CommaFDM&fdmTargetAppName=Comma95&sso_token=
$SSO_TOKEN$&sso_username=$SSO_USERNAME$&sso_password=$SSO_PASSWORD
&$ATTR(id) $&RCP_VERSION=$RCP_VERSION$</url>
    </action>
  </resource>
</foldercontents>
```

The second BSTR looks like the following:

```
<foldercontents path="/">
  <resource name="EU Sales Report" description="" type="application/x-
hyperion-applicationbuilder-report">
    <attribute name="name" type="string" xml:lang="fr" value="EU Rapport de
ventes" />
    <attribute name="name" type="string" xml:lang="es" value="EU Informe de
ventas" />
    <action name="Display Drill-Back" description="Launch Drill-Back"
shortdesc="drill-back">
      <url>./HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?
&fdmAppName=CommaFDM&fdmTargetAppName=Comma95&sso_token=
$SSO_TOKEN$&sso_username=$SSO_USERNAME$&sso_password=$SSO_PASSWORD
&$ATTR(id) $&RCP_VERSION=$RCP_VERSION$</url>
```

```
</action>
</resource>
</foldercontents>
```

InsertLineItemDetails

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

Syntax

```
<HsvData>.InsertLineItemDetails lScenario, lYear, lEntity, lAccount, lICP,
lCustom1, lCustom2, lCustom3, lCustom4, varabstrDetail, varalInsertPos
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member for the intersecting cells.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member for the intersecting cells.
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member for the intersecting cells.
<i>lAccount</i>	Long (ByVal). The member ID of the Account dimension member for the intersecting cells. Tip: Use <code>HsvAccounts.UsesLineItems</code> to check whether an account supports line items.
<i>lICP</i>	Long (ByVal). Long (ByVal). The member ID of the Intercompany Partner dimension member for the intersecting cells.
<i>lCustom1</i>	Long (ByVal). The member ID of the Custom 1 dimension member for the intersecting cells.
<i>lCustom2</i>	Long (ByVal). The member ID of the Custom 2 dimension member for the intersecting cells.
<i>lCustom3</i>	Long (ByVal). The member ID of the Custom 3 dimension member for the intersecting cells.
<i>lCustom4</i>	Long (ByVal). The member ID of the Custom 4 dimension member for the intersecting cells.
<i>varabstrDetail</i>	String array (ByVal). An array of strings that contain the line item descriptions to be added.
<i>varalInsertPos</i>	Long array (ByVal). An array of Longs that indicate the positions in the intersecting cells in which the line item details 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.

Example

The following example inserts two line item descriptions in the first two positions of the intersecting cells. The example assumes the variables passed to `InsertLineItemDetails` have been set to the member IDs for the intersecting cells.

```
Dim saDesc(1) As String, laPos(1) As Long
saDesc(0) = "Chairs"
saDesc(1) = "Desks"
laPos(0) = 0
laPos(1) = 1
m_cHsvData.InsertLineItemDetails lScen, lYr, lEnt, lAcct, _
lIcp, lCust1, lCust2, lCust3, lCust4, saDesc, laPos
```


IsValidCellText

Indicates whether a string exceeds the application's maximum cell text size.

Tip: To return the maximum number of characters allowed for cell text, use [GetMaxCellTextSize](#).

Syntax

```
<HsvData>.IsValidCellText (bstrCellText)
```

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

<i>bstrCellText</i>	String (ByVal). The string to test.
---------------------	-------------------------------------

Return Value

Boolean. Returns TRUE if the string does not exceed the maximum cell text size, FALSE otherwise.

Load

Loads data into an application, using a load file on the application server.

Tip: You can load files from client computers with the HsvcDataLoad type library. This library also offers properties and methods that simplify handling of the data load options. For more information, see [“Loading Data” on page 776](#).

Syntax

```
<HsvData>.Load bstrServerFileName, bstrLogFileName, varavOptions
```

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

<i>bstrServerFileName</i>	String (ByVal). The name and path of the data load file. This file must exist on the application server.
---------------------------	--

<i>bstrLogFileName</i>	String (ByVal). The name and path of the log file. The path must exist on the application server.
------------------------	---

<i>varavOptions</i>	Variant array (ByVal). The load options for the data load operation. The array is 1-based. For details on indexes and valid values, see Table 61 .
---------------------	--

Tip: Use [EnumLoadOptions](#) to return information about the valid load options.

The following table describes the data load options. Some of the valid values are represented by constants of the HsvcDataLoad type library. To use these constants, you must reference HsvcDataLoad in your project; for information on this library, see [“HsvcDataLoad Type Library” on page 775](#).

The listed indexes apply to the array passed to `Load` and to the first dimension of the array returned by `EnumLoadOptions`.

Table 61 Data Load Options

Index	Load Option Information
1	<p>Option: Delimiter</p> <p>Usage: Specifies a load file's delimiter.</p> <p>Pass to Load: A valid delimiter character (String subtype).</p>
2	<p>Option: Append to Log File</p> <p>Usage: Specifies whether log data is appended to or overwrites the existing log file.</p> <p>Pass to Load: Boolean – TRUE to append, FALSE to overwrite.</p>
3	<p>Option: Load Calculated</p> <p><i>For internal use.</i></p>
4	<p>Option: Duplicates</p> <p>Usage: Sets the data load mode to either replace or merge.</p> <p>Tip: This option corresponds to the Lode Mode option buttons in Financial Management's Load Data workspace frame. For details on the load modes, see the <i>Oracle Hyperion Financial Management, Fusion Edition User's Guide</i>.</p> <p>Pass to Load: One of the constants listed in "Update Mode Constants" on page 276.</p>
5	<p>Option: Accumulate within file</p> <p>Usage: Specifies whether multiple values for the same cell within the load file should be accumulated or allowed to overwrite each other.</p> <p>If the load file does not contain multiple values for the same cell, then this option has no effect. In addition, this load option applies to line items and values, but not to descriptions – multiple descriptions always overwrite.</p> <p>Note: Data for system accounts never gets accumulated.</p> <p>Pass to Load: TRUE to accumulate, FALSE to overwrite.</p>
6	<p>Option: Does the file contain shares data</p> <p>Usage: Specifies whether the load file contains shares data such as "shares outstanding" or "voting outstanding" or "owned."</p> <p>Pass to Load: TRUE if the file contains shares data, FALSE otherwise.</p>
7	<p>Option: Mode</p> <p>Usage: Specifies whether the data file is loaded or is merely scanned when <code>Load</code> is called.</p> <p>Pass to Load: One of the following <code>HsvcDataLoad</code> type library constants:</p> <ul style="list-style-type: none"> ● <code>HSV_DATALOAD_LOAD</code>. Data will be loaded. ● <code>HSV_DATALOAD_SCAN</code>. Data will be scanned but not loaded. <p>The Mode option defaults to scanning, so you must set the option to <code>HSV_DATALOAD_LOAD</code> in order to load data.</p>
8	<p>Option: Does the file contain submission phase data</p>

Index	Load Option Information
	<p>Usage: Specifies whether the load file contains data for phased submissions.</p> <p>Pass to Load: TRUE if the file contains phased submissions data, FALSE otherwise.</p>
9	<p>Option: FileFormat</p> <p>Usage: Specifies the load file format. In this release, only the native file format is supported.</p> <p>Pass to Load: The following HsvcDataLoad type library constant, which represents the only supported option: HSV_DATALOAD_FILE_FORMAT_NATIVE</p>
10	<p>Option: DecimalChar</p> <p>Usage: Specifies the decimal character used in the load file. By default this option is set to an empty string, which indicates the load operation will use the decimal character set as the user's preference.</p> <p>Tip: To get the user preferences for the decimal and thousands separator characters, use the HsvSystemInfo method GetNumberFormattingUserParameters.</p> <p>Pass to Load: If the load file's decimal character differs from that specified as the user's preference, specify the load file's decimal character. (String subtype).</p>
11	<p>Option: ThousandsChar</p> <p>Usage: Specifies the thousands separator character used in the load file. By default this option is set to an empty string, which indicates the load operation will use the thousands separator character set as the user's preference.</p> <p>Pass to Load: If the load file's decimal character differs from that specified as the user's preference, specify the load file's thousands separator character. (String subtype).</p>

Example

The following example loads data. `EnumLoadOptions` gets the default data load options. The array passed to `Load` is built: the default `Delimiter` option is added, the `Duplicates` option is set to `HSV_DATA_UPDATE_ACCUMULATE`, and the `Mode` option is set to `HSV_DATALOAD_LOAD`. This array is then passed to `Load`.

```
Dim cData As HsvData, vaOpts, vaSettings(1 To 11)
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
cData.EnumLoadOptions vaOpts
vaSettings(1) = vaOpts(1, 2)
vaSettings(4) = HSV_DATA_UPDATE_ACCUMULATE
vaSettings(7) = HSV_DATALOAD_LOAD
cData.Load "c:\acme\myapp.dat", "c:\acme\myapp.log", _
    vaSettings
```

LoadDrillableRegions

Loads the definition file for the drillable regions. `LoadDrillableRegions` enables drill through on data values from Financial Management to the source data in the Oracle General Ledger (OGL) application. The function allows caller to define drillable regions in the Scenario, Year, Period, Entity, and Account dimensions. The drillable region allows the definition of the slice of data that was loaded from a GL Source System using EPMI or Oracle Hyperion Financial Data Quality Management, Fusion Edition.

Syntax

```
<HsvData>.LoadDrillableRegions bstrRegionsFilename, bstrLogFileName,  
lLoadMode, pvbErrors, pvbWarnings
```

Argument	Description
<i>bstrRegionsFilename</i>	String (ByVal). The file path to the definition file on the Financial Management application server.
<i>bstrLogFileName</i>	String (ByVal). The name and path of the log file. The path must exist on the application server.
<i>lLoadMode</i>	Long (ByVal). The load mode. Must be set to one of the following values: REPLACE_ALL = 0, REPLACE_BY_URL_NAME = 1, MERGE = 2, SCAN_ONLY = 3.
<i>pvbErrors</i>	Boolean. Returns TRUE if errors occurred during the load task. See the log file for details.
<i>pvbWarnings</i>	Boolean. Returns TRUE if warnings occurred during the load task.

lLoadMode Parameter

- Replace all — This mode will clear all existing URL definitions before loading the new definition from the load file. It is intended to be used by the system administrator or a user with a special role.
- Replace by URL name — This mode will clear and reload a URL definition in Financial Management if a URL with the same name is in the load file. All other URL definitions will remain unchanged. This mode is intended to be used by products that manage data input, either manually or automatically.
- Merge — This mode will add new region definitions to the existing URL. If a new XML block for URL is supplied for an existing URL, it will replace the prior XML block. If it is not provided, existing XML block will be kept for the specified URL. A new URL will still be created if there are no existing URLs with the same URL name. URLs currently existed in Financial Management but are not included in the load file will not be removed.
- Scan Only — This mode will scan the input file for invalid format in the definition. The new definition will not be stored to the database. The existing definition will remain unchanged.

Note: For the Merge mode, if a URL link is not supplied and the URL definition exists, the new region definitions will be merged in. However, if the URL does not exist in Financial Management, an error will be logged and that URL definition will be rejected. Similarly, in Replace mode the URL link must be provided.

Examples

This function is intended to be used by *HFMAwbAgent*. The following example loads the definition file from `C:\temp\~RD8a13.tmp` in Merge mode. Progress messages and errors are saved in the log file `C:\temp\~RD9bc1.tmp`. If any errors or warnings occur, *vbError* or *vbWarnings* will be set to `VARIANT_TRUE`.

```
LoadDrillableRegions (_T("C:\temp\~RD8a13.tmp"), _T("C:\temp\  
\~RD9bc1.tmp"), 2, &vbError, &vbWarnings);  
The content of the input file C:\temp\~RD8a13.tmp:
```

```

!URLName=ERPI_ew3qnio
!URL=<foldercontents path="/">
<resource name="Sales Report" description="" type="application/x-hyperion-
applicationbuilder-report">
    <attribute name="name" type="string" xml:lang="fr" value="Rapport de
ventes" />
    <attribute name="name" type="string" xml:lang="es" value="Informe de
ventas" />
    <action name="Display Drill-Back" description="Launch Drill-Back"
shortdesc="drill-back">
        <url>./HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?
&f;fdmAppName=CommaFDM&f;fdmTargetAppName=Comma95&sso_token=
$SSO_TOKEN$&sso_username=$SSO_USERNAME$&sso_password=$SSO_PASSWORD
$&$ATTR(id)$&RCP_VERSION=$RCP_VERSION$</url>
    </action>
</resource>
</foldercontents>
!Regions
Actual;2008;January;Stamford;Payroll
Actual;2008;January;California;MEMBERALL
!URLName=FDM_adf120agj
!URL=<foldercontents path="/">
    <resource name="EU Sales Report" description="" type="application/x-
hyperion-applicationbuilder-report">
        <attribute name="name" type="string" xml:lang="fr" value="EU Rapport de
ventes" />
        <attribute name="name" type="string" xml:lang="es" value="EU Informe de
ventas" />
        <action name="Display Drill-Back" description="Launch Drill-Back"
shortdesc="drill-back">
            <url>./HyperionFDM/AuthorizedPages/IntersectionSummaryByLocation.aspx?
&f;fdmAppName=CommaFDM&f;fdmTarge
tAppName=Comma95&sso_token=$SSO_TOKEN$&sso_username=$SSO_USERNAME
$&sso_password=$SSO_PASSWORD$&$ATTR(id)$&RCP_VERSION=
$RCP_VERSION$</url>
        </action>
    </resource>
</foldercontents>
!Regions
Actual;2008;MEMBERALL;Germany; Sales
Actual;2008; MEMBERALL;England;Sales
Actual;2008; MEMBERALL;France;Sales

```

SetCalcStatusLocked

Locks the cells for a period in a subcube. (For information on subcubes, see [“About Subcubes” on page 43.](#))

After calling `SetCalcStatusLocked`, data cannot be entered in the cells until the cell is unlocked. Use `SetCalcStatusUnlocked` to unlock the cells.

Syntax

```
<HsvData>.SetCalcStatusLocked lScenario, lYear, lPeriod, lEntity, lParent,
lValue
```

Argument Description

<i>IScenario</i>	Long (ByVal). The member ID of the subcube's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the subcube's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the Period dimension member for which data is being locked.
<i>IEntity</i>	Long (ByVal). The member ID of the subcube's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the parent of the <i>IEntity</i> argument's entity.
<i>IValue</i>	Long (ByVal). The member ID of the subcube's Value dimension member.

SetCalcStatusLockedForMultipleProcessUnits

Locks the cells for one or more process units.

After calling `SetCalcStatusLockedForMultipleProcessUnits`, data cannot be entered in the cells until the process units are unlocked. You can use [SetCalcStatusUnlockedForMultipleProcessUnits](#) to unlock all the cells, or [SetCalcStatusUnlocked](#) to unlock a specific process unit.

Syntax

```
<HsvData>.SetCalcStatusLockedForMultipleProcessUnits varalScenario,  
varalYear, varalPeriod, varalEntity, varalParent, varalValue
```

Argument Description

<i>varalScenario</i>	Long array (ByVal). The member IDs of the process units' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the process units' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the process units' Period dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the process units' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the process units' Value dimension members.

SetCalcStatusUnlocked

Unlocks the cells for a period in a subcube. (For information on subcubes, see [“About Subcubes” on page 43.](#))

Syntax

```
<HsvData>.SetCalcStatusUnlocked lScenario, lYear, lPeriod, lEntity,  
lParent, lValue
```

Argument Description

<i>IScenario</i>	Long (ByVal). The member ID of the subcube's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the subcube's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the Period dimension member for which data is being locked.
<i>IEntity</i>	Long (ByVal). The member ID of the subcube's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the parent of the <i>IEntity</i> argument's entity.
<i>IValue</i>	Long (ByVal). The member ID of the subcube's Value dimension member.

SetCalcStatusUnlockedForMultipleProcessUnits

Unlocks cells for one or more process units. You can use this to unlock all the process units that have been locked with [SetCalcStatusLockedForMultipleProcessUnits](#).

Syntax

```
<HsvData>.SetCalcStatusUnlockedForMultipleProcessUnits varalScenario,  
varalYear, varalPeriod, varalEntity, varalParent, varalValue
```

Argument Description

<i>varalScenario</i>	Long array (ByVal). The member IDs of the process units' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the process units' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the process units' Period dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the process units' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the process units' Value dimension members.

SetCell

Sets a cell's data; you can either insert data or set the cell to Null.

Tip: `SetCell` passes the data to be inserted as a `Double`. To insert a cell's data by passing a `String`, use [SetTextCell](#).

Syntax

```
<HsvData>.SetCell(lScenario, lYear, lPeriod, lView, lEntity, lParent,  
lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, dData,  
bIsNoData)
```

Argument Description

<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>Year</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>View</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>Entity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the parent of the <i>Entity</i> argument's entity.
<i>Value</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>dData</i>	Double (ByVal). The data to be inserted in the cell.
<i>blsNoData</i>	Double (ByVal). Determines whether the <i>dData</i> argument's data is inserted or the cell is set to Null. Pass FALSE to insert data, TRUE to set the cell to Null. You might find it useful to conditionally pass TRUE in cases where the <i>dData</i> argument's amount is 0 (zero). Caution! Passing TRUE will delete any existing data in the cell.

Return Value

Integer. Indicates the success of the function call; returns 0 for success or an error code for failure.

Example

The following subroutine inserts data into a cell; if 0 is passed as the cell data, the cell is set to Null. The cell's dimension members and data are specified by the subroutine's arguments.

```
Sub setCellDataOrNull(lScen As Long, lYear As Long, lPer As _
    Long, lView As Long, lEnt As Long, lPar As Long, lVal As _
    Long, lAcct As Long, lICP As Long, lCust1 As Long, _
    lCust2 As Long, lCust3 As Long, lCust4 As Long, _
    dData As Double)
Dim cData As HsvData
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
If dData = 0 Then
    cData.SetCell lScen, lYear, lPer, lView, lEnt, lPar, _
        lVal, lAcct, lICP, lCust1, lCust2, lCust3, lCust4, _
        dData, True
Else
```



```

        cData.SetCell lScen, lYear, lPer, lView, lEnt, lPar, _
            lVal, lAcct, lICP, lCust1, lCust2, lCust3, lCust4, _
            dData, False
    End If
End Sub

```

SetCellDescriptions

Inserts cell text into one or more cells.

`SetCellDescriptions` takes arrays of dimension member IDs as arguments, as well as an array of strings containing the cell text to be inserted. These arrays must contain identical numbers of elements. The elements of the member ID arrays have a one-to-one correspondence to the elements of the cell text array; the first elements of the member ID arrays define the first cell in which cell text is to be inserted, the second elements of the member ID arrays define the second cell, and so on.

Syntax

```

<HsvData>.SetCellDescriptions varlScenario, varlYear, varlPeriod,
varlEntity, varlParent, varlValue, varlAccount, varlICP, varlCustom1,
varlCustom2, varlCustom3, varlCustom4, varbstrCellText

```

Argument	Description
<i>varlScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varlYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varlPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varlEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varlParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varlEntity</i> argument's entities.
<i>varlValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varlAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varlICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varlCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varlCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varlCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varlCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>varbstrCellText</i>	String array (ByVal). The cell text to be inserted for the cells. The elements in this array correspond to the elements in the other arguments. For example, the fifth element in this array is the cell text for the cell defined by the member IDs in the fifth elements of the other arguments' arrays.

Example

The following example puts cell text into cells for the Sales and OtherCosts accounts. The example gets the member IDs for these accounts with calls to the user-defined `GetMemberID` function; for details on `GetMemberID`, see the example for `GetItemID`. The example then assigns these IDs to the `laAcct1` variable, which is the array passed to `SetCellDescriptions`' *varlAccount* argument. The cell text is then added to the `saCellText` variable, and `SetCellDescriptions` inserts this array's elements into the cells.

Note: The example assumes that the member ID arguments for the non-Account dimensions are set in another procedure.

```
Dim laAcct1 As Long, laAcct2 As Long
Dim laAcct(1) as Long, saCellText(1) As String
laAcct1 = GetMemberID(DIMENSIONACCOUNT, "Sales")
laAcct2 = GetMemberID(DIMENSIONACCOUNT, "OtherCosts")
laAcct(0) = laAcct1
laAcct(1) = laAcct2
saCellText(0) = "Acme sale"      'For the Sales account
saCellText(1) = "Wining/dining"  'For the OtherCosts account
m_chsvData.SetCellDescriptions laScen, laYear, laPer, laEnt, _
laPar, laVal, laAcct, laICP, laCust1, laCust2, laCust3, _
laCust4, saCellText
```

SetCellLineItems

Appends or updates line items for the specified cell. Line item data is passed in a Double array.

Tip: To pass line item data in a String array, use `SetTextCellLineItems`.

Syntax

```
<HsvData>.SetCellLineItems lScenario, lYear, lPeriod, lEntity, lAccount,
lICP, lCustom1, lCustom2, lCustom3, lCustom4, vardData, varbstrDetails
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member. Tip: Use <code>HsvAccounts.UsesLineItems</code> to check whether an account supports line items
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.

Argument	Description
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>vardata</i>	Double array (ByVal). An array of the line item data to be added, formatted according to the connected user's preferences.
<i>varbstrDetails</i>	String array (ByVal). An array of the descriptions that identify the line items to be updated.

Example

The following example inserts data for line item descriptions named “Desks” and “Chairs”. The example assumes the variables passed to `SetCellLineItems` have been set to the cell's member IDs, and then uses `UsesLineItems` to test whether the cell's Account dimension member supports line items. If line items are supported, the line item data is added.

```
Dim daData(1) As Double, saDescs(1) As String, cData As HsvData
Dim cAccounts As HsvAccounts, bLineItems As Boolean
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
'g_cMetadata is an HsvMetadata object reference
Set cAccounts = g_cMetadata.Accounts
'Exit if the account does not support line items
cAccounts.UsesLineItems lAcct, bLineItems
If bLineItems = False Then Exit Sub
daData(0) = 3497#
saDescs(0) = "Desks"
daData(1) = 10000.01
saDescs(1) = "Chairs"
cData.SetCellLineItems lScen, lYear, lPer, lEnt, lAcct, _
    lICP, lCust1, lCust2, lCust3, lCust4, daData, saDescs
```

SetCells

Sets data for an array of cells. For each cell in the array, you can either insert data or set the cell to Null.

Note: If any of the specified cells are not writable, `SetCells` fails. For a similar method that does not fail when non-writable cells are encountered, use `SetCells2`. `SetCells` also returns the statuses of the cells.

`SetCells` takes arrays of dimension member IDs as arguments, as well as an array of data for the cells to be set. These arrays must contain identical numbers of elements. The elements of the member ID arrays have a one-to-one correspondence to the elements of the data array.

Tip: `SetCells` passes the cells' data in a `Double` array. To insert data into cells by passing a `String` array, use `SetTextCells` instead of `SetCells`. For more information, see [“SetTextCells” on page 371](#).

Syntax

```
<HsvData>.SetCells varalScenario, varalYear, varalPeriod, varalView,  
varalEntity, varalParent, varalValue, varalAccount, varalICP, varalCustom1,  
varalCustom2, varalCustom3, varalCustom4, varadData, varabIsNoData
```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varalView</i>	Long array (ByVal). The member IDs of the cells' View dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>varadData</i>	Double array (ByVal). The data to be inserted in the cells. The elements in this array correspond to the elements in the other arguments. For example, the fifth element in this array is inserted in the cell defined by the member IDs in the fifth elements of the other arguments' arrays.
<i>varabIsNoData</i>	Boolean array (ByVal). Determines whether the corresponding element in the <i>varadData</i> argument's array is inserted or the cell is set to Null. Pass FALSE to insert data, TRUE to set a cell to Null. You might find it useful to conditionally pass TRUE in cases where a <i>varadData</i> element is 0 (zero). Caution! Passing TRUE deletes any existing data in a cell.

Example

The following subroutine inserts data into cells; if 0 is passed for a cell's data, the cell is set to Null. The cells' dimension members and data are passed in the subroutine's arguments. The subroutine constructs the array passed to `SetCells`' *varabIsNoData* argument; if 0 is passed for a cell's data, the corresponding item in the *baNull* array is set to TRUE.

```
Sub setCellsOrNull(laScen() As Long, laYear() As _
```

```

        Long, laPer() As Long, laView() As Long, laEnt() As Long, _
        laPar() As Long, laVal() As Long, laAcct() As Long, laICP() _
        As Long, laCust1() As Long, laCust2() As Long, laCust3() _
        As Long, laCust4() As Long, daData() As Double)
Dim cData As HsvData, baNull() As Boolean
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
'Build the baNull array
ReDim baNull(UBound(daData))
For i = LBound(daData) To UBound(daData)
    If daData(i) = 0 Then
        baNull(i) = True
    Else
        baNull(i) = False
    End If
Next i
cData.SetCells laScen, laYear, laPer, laView, laEnt, _
    laPar, laVal, laAcct, laICP, laCust1, laCust2, laCust3, _
    laCust4, daData, baNull
End Sub

```

SetCells2

Sets data for an array of cells and returns the cells' statuses; if any of the cells are not writable, SetCells2 inserts data in the writable cells. For each cell in the array, you can either insert data or set the cell to Null.

SetCells2 takes arrays of dimension member IDs as arguments, as well as an array of data for the cells to be set. These arrays must contain identical numbers of elements. The elements of the member ID arrays have a one-to-one correspondence to the elements of the data array.

Syntax

```

<HsvData>.SetCells2 varalScenario, varalYear, varalPeriod, varalView,
varalEntity, varalParent, varalValue, varalAccount, varalICP, varalCustom1,
varalCustom2, varalCustom3, varalCustom4, varadData, varabIsNoData,
pvaralStatus

```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varalView</i>	Long array (ByVal). The member IDs of the cells' View dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.

Argument	Description
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>varadData</i>	Double array (ByVal). The data to be inserted in the cells. The elements in this array correspond to the elements in the other arguments. For example, the fifth element in this array is inserted in the cell defined by the member IDs in the fifth elements of the other arguments' arrays.
<i>varabIsNoData</i>	Boolean array (ByVal). Determines whether the corresponding element in the <i>varadData</i> argument's array is inserted or the cell is set to Null. Pass FALSE to insert data, TRUE to set a cell to Null. You might find it useful to conditionally pass TRUE in cases where a <i>varadData</i> element is 0 (zero). Passing TRUE deletes any existing data in a cell.
<i>pvaralStatus</i>	Variant array. Returns the cells' statuses. For details on cell statuses, see “About Cell Statuses” on page 275 . The array items are returned as a Long subtype.

Example

The following function inserts data into cells; if 0 is passed for a cell's data, the cell is set to Null. The cells' dimension members and data are passed in the subroutine's arguments. The subroutine constructs the array passed to the *varabIsNoData* argument of *SetCells2*; if 0 is passed for a cell's data, the corresponding item in the *baNull* array is set to TRUE.

```
Function setCells2OrNull(laScen() As Long, laYear() As _
    Long, laPer() As Long, laView() As Long, laEnt() As Long, _
    laPar() As Long, laVal() As Long, laAcct() As Long, laICP() _
    As Long, laCust1() As Long, laCust2() As Long, laCust3() _
    As Long, laCust4() As Long, daData() As Double) As Variant
Dim cData As HsvData, baNull() As Boolean, vaRet
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
'Build the baNull array
ReDim baNull(UBound(daData))
For i = LBound(daData) To UBound(daData)
    If daData(i) = 0 Then
        baNull(i) = True
    Else
        baNull(i) = False
    End If
Next i
cData.SetCells2 laScen, laYear, laPer, laView, laEnt, _
    laPar, laVal, laAcct, laICP, laCust1, laCust2, laCust3, _
    laCust4, daData, baNull, vaRet
setCells2OrNull = vaRet
End Function
```

SetCellsLineItems

Appends or updates line items for the specified cells. Line item data is passed in a Double array.

Tip: To update cells by passing data in a String array, use [SetTextCellsLineItems](#).

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

Syntax

```
<HsvData>.SetCellsLineItems varalScenario, varalYear, varalPeriod,  
varalEntity, varalAccount, varalICP, varalCustom1, varalCustom2,  
varalCustom3, varalCustom4, varadData, varalDetail
```

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

<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
----------------------	--

<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
------------------	--

<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
--------------------	--

<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
--------------------	--

<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
---------------------	---

Tip: Use `HsvAccounts.UsesLineItems` to check whether an account supports line items.

<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
-----------------	--

<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
---------------------	--

<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
---------------------	--

<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
---------------------	--

<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
---------------------	--

<i>varadData</i>	Double array (ByVal). The data to be added to the cells, formatted according to the user's preferences.
------------------	---

<i>varalDetail</i>	String array (ByVal). The line item descriptions that identify the line item details to be updated.
--------------------	---

Example

The following example adds data for two line items. The example assumes that the member ID arrays passed to `SetCellsLineItems` have been previously set.

```
Dim saDesc(1) As String, daData(1) As Double  
daData(0) = 100000  
daData(1) = 200000  
saDesc(0) = "Chairs"  
saDesc(1) = "Desks"  
m_chsvData.SetCellsLineItems laScen, laYr, laPer, laEnt, _  
laAcct, laIcp, laCust1, laCust2, laCust3, laCust4, _  
daData, saDesc
```

SetFileForLoad

For internal use.

SetLineItems

Deprecated – superseded by [SetCellLineItems](#).

SetMinMaxPeriod

For internal use.

SetPhaseSubmissionGridForGivenScenarioPeriod

Assigns submission groups to the specified combinations of Scenario dimension member, Period dimension members, and submission phases. The Period member IDs, phase IDs, and submission group IDs are passed in arrays that have a one-to-one correspondence.

Syntax

```
<HsvData>.SetPhaseSubmissionGridForGivenScenarioPeriod lScenario,  
varalPeriodIDs, varalPhaseIDs, pvarbstrGroupInfo
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>varalPeriodIDs</i>	Long array (ByVal). The member IDs of the Period dimension members.
<i>varalPhaseIDs</i>	Long array (ByVal). The phase IDs of the submission phases.
<i>pvarbstrGroupInfo</i>	String array (ByVal). The IDs of the submission groups.

Example

For the specified scenario and periods, the following method assigns all submission groups to Phase 1.

```
Sub SetPhaseOneAllGroups(lScen As Long, laPers() As Long)  
Dim laPhaseIds(0) As Long, saGroupInfo() As String  
laPhaseIds(0) = 1  
Dim cData As HsvData  
'g_cSession represents an HsvSession instance  
Set cData = g_cSession.Data  
ReDim saGroupInfo(UBound(laPers))  
For i = 0 To UBound(laPers)  
    saGroupInfo(i) = "*"   
Next i  
cData.SetPhaseSubmissionGridForGivenScenarioPeriod lScen, laPers, _  
    laPhaseIds, saGroupInfo  
End Sub
```


SetTextCell

Inserts data into a cell, passing the data as a `String`. `SetTextCell` also enables you to scale the data that is passed.

Tip: To insert data into a cell by passing a `Double`, use `SetCell` instead of `SetTextCell`. For more information, see [“SetCell” on page 359](#).

Syntax

```
<HsvData>.SetTextCell(lScenario, lYear, lPeriod, lView, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, sScale, bstrData)
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>sScale</i>	Integer (ByVal). Indicates the degree of scaling to apply to the <i>bstrData</i> argument's amount. To leave this amount unscaled, pass 0. To scale the amount, each whole-number increment over 0 scales by ten. In other words, passing 1 scales by ten, passing 2 scales by a hundred, and so on. To apply the application's default scaling, pass the <code>HFMConstants</code> type library constant <code>DEFAULT_SCALE</code> , described in “Number Defaults Constants” on page 881 .
<i>bstrData</i>	String (ByVal). The data to be inserted in the cell.

Return Value

Integer. Indicates the success of the function call; returns 0 for success or an error code for failure.

Example

The following subroutine inserts data into a cell, scaling the data by ten. The cell's dimension members and data are specified by the subroutine's arguments.

```
Sub setCellDataScaledbyTen(lScen As Long, lYear As Long, lPer _
    As Long, lView As Long, lEnt As Long, lPar As Long, lVal As _
    Long, lAcct As Long, lICP As Long, lCust1 As Long, _
    lCust2 As Long, lCust3 As Long, lCust4 As Long, _
    sData As String)
Dim cData As HsvData
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
cData.SetTextCell lScen, lYear, lPer, lView, lEnt, lPar, _
    lVal, lAcct, lICP, lCust1, lCust2, lCust3, lCust4, _
    1, sData
End Sub
```

SetTextCellLineItems

Appends or updates line items for the specified cell. Line item data is passed in a String array.

Tip: To pass line item data in a Double array, use [SetCellLineItems](#).

Syntax

```
<HsvData>.SetTextCellLineItems lScenario, lYear, lPeriod, lEntity,
lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, varabstrData,
varabstrDetails
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member. Tip: Use <code>HsvAccounts.UsesLineItems</code> to check whether an account supports line items.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.

Argument	Description
<i>varabstrData</i>	String array (ByVal). An array of the line item data to be added, formatted according to the connected user's preferences.
<i>varabstrDetails</i>	String array (ByVal). An array of the descriptions that identify the line items to be updated.

Example

The following example inserts data for line item descriptions named “Desks” and “Chairs”. The example assumes the variables passed to `SetTextCellLineItems` have been set to the cell's member IDs, and then uses `UsesLineItems` to test whether the cell's Account dimension member supports line items. If line items are supported, the line item data is added.

```
Dim saData(1) As String, saDescs(1) As String, cData As HsvData
Dim cAccounts As HsvAccounts, bLineItems As Boolean
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
'g_cMetadata is an HsvMetadata object reference
Set cAccounts = g_cMetadata.Accounts
'Exit if the account does not support line items
cAccounts.UsesLineItems lAcct, bLineItems
If bLineItems = False Then Exit Sub
saData(0) = 3497#
saDescs(0) = "Desks"
saData(1) = 10000.01
saDescs(1) = "Chairs"
cData.SetTextCellLineItems lScen, lYr, lPer, lEnt, lAcct, _
    lICP, lCust1, lCust2, lCust3, lCust4, saData, saDescs
```

SetTextCells

Inserts data into cells. The cells' data is passed in a String array. `SetTextCells` also enables you to scale the data that is passed.

`SetTextCells` takes arrays of dimension member IDs as arguments, as well as an array of data for the cells to be set. These arrays must contain identical numbers of elements. The elements of the member ID arrays have a one-to-one correspondence to the elements of the data array; the first elements of the member ID arrays define the first cell to be set, the second elements of the member ID arrays define the second cell to be set, and so on.

Tip: To pass data into cells with a Double array, use `SetCells` instead of `SetTextCells`. For more information, see [“SetCells” on page 363](#).

Syntax

```
<HsvData>.SetTextCells varalScenario, varalYear, varalPeriod, varalView,
varalEntity, varalParent, varalValue, varalAccount, varalICP, varalCustom1,
varalCustom2, varalCustom3, varalCustom4, varabstrData, sScale
```

Argument	Description
<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
<i>varalView</i>	Long array (ByVal). The member IDs of the cells' View dimension members.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the parents of the <i>varalEntity</i> argument's entities.
<i>varalValue</i>	Long array (ByVal). The member IDs of the cells' Value dimension members.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
<i>varabstrData</i>	String array (ByVal). The data to be inserted in the cells. The elements in this array correspond to the elements in the other arguments. For example, the fourth element in this array is inserted in the cell defined by the member IDs in the fourth elements of the other arguments' arrays.
<i>sScale</i>	Integer (ByVal). Indicates the degree of scaling to apply to the <i>varabstrData</i> argument's amounts. To leave the amounts unscaled, pass 0. To scale the amounts, each whole-number increment over 0 scales by ten. In other words, passing 1 scales by ten, passing 2 scales by a hundred, and so on. To apply the application's default scaling, pass the HFMConstants type library constant <code>DEFAULT_SCALE</code> , described in "Number Defaults Constants" on page 881 .

Example

The following subroutine inserts data into cells, scaling the data by ten. The cells' dimension members and data are passed in the subroutine's arguments.

```
Sub setCellsScaledByTen(laScen() As Long, laYear() As _
    Long, laPer() As Long, laView() As Long, laEnt() As Long, _
    laPar() As Long, laVal() As Long, laAcct() As Long, laICP() _
    As Long, laCust1() As Long, laCust2() As Long, laCust3() _
    As Long, laCust4() As Long, saData() As String)
Dim cData As HsvData
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
cData.SetTextCells laScen, laYear, laPer, laView, laEnt, _
    laPar, laVal, laAcct, laICP, laCust1, laCust2, laCust3, _
    laCust4, saData, 1
End Sub
```

SetTextCellsLineItems

Appends or updates line items for the specified cells. Line item data is passed in a String array.

Tip: To update cells by passing data in a Double array, use [SetCellsLineItems](#).

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

Syntax

```
<HsvData>.SetTextCellsLineItems varalScenario, varalYear, varalPeriod,  
varalEntity, varalAccount, varalICP, varalCustom1, varalCustom2,  
varalCustom3, varalCustom4, varabstrData, varabstrDetail
```

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

<i>varalScenario</i>	Long array (ByVal). The member IDs of the cells' Scenario dimension members.
----------------------	--

<i>varalYear</i>	Long array (ByVal). The member IDs of the cells' Year dimension members.
------------------	--

<i>varalPeriod</i>	Long array (ByVal). The member IDs of the cells' Period dimension members.
--------------------	--

<i>varalEntity</i>	Long array (ByVal). The member IDs of the cells' Entity dimension members.
--------------------	--

<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
---------------------	---

Tip: Use `HsvAccounts.UsesLineItems` to check whether an account supports line items.

<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
-----------------	--

<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom 1 dimension members.
---------------------	--

<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom 2 dimension members.
---------------------	--

<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom 3 dimension members.
---------------------	--

<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom 4 dimension members.
---------------------	--

<i>varabstrData</i>	String array (ByVal). The data to be added to the cells, formatted according to the user's preferences.
---------------------	---

<i>varabstrDetail</i>	String array (ByVal). The line item descriptions that identify the line item details to be updated.
-----------------------	---

Example

The following example adds data for two line items. The example assumes that the member ID arrays passed to `SetTextCellsLineItems` have been previously set.

```
Dim saDesc(1) As String, saData(1) As String  
saData(0) = "100000"  
saData(1) = "200000"  
saDesc(0) = "Chairs"  
saDesc(1) = "Desks"  
m_cHsvData.SetTextCellsLineItems laScen, laYr, laPer, laEnt, _  
laAcct, laIcp, laCust1, laCust2, laCust3, laCust4, _  
saData, saDesc
```

SetTextLineItems

Deprecated – superseded by [SetTextCellLineItems](#).

StartLoad

For internal use.

UpdateDataUsingMDDataBuffer

Inserts an HsvMDDataBuffer or HsvMDDataBufferLite object's cells into the corresponding cells of an application.

Syntax

```
<HsvData>.UpdateDataUsingMDDataBuffer pIUnkDataBuffer, lEnumUpdateMode,  
bAccumulateWithinBuffer
```

Argument	Description
<i>pIUnkDataBuffer</i>	HsvMDDataBuffer or HsvMDDataBufferLite object (ByVal). The object reference for the object.
<i>lEnumUpdateMode</i>	Long (ByVal). Determines the update mode. For more information and valid values, see “Update Mode Constants” on page 276 .
<i>bAccumulateWithinBuffer</i>	Boolean (ByVal). Determines whether multiple values for the same cell within the HsvMDDataBuffer or HsvMDDataBufferLite object should be accumulated or allowed to overwrite each other. Pass TRUE to accumulate, FALSE to overwrite. If the data buffer does not contain multiple values for the same cell, then this argument has no effect. In addition, this argument applies to line items and values, but not to descriptions – multiple descriptions always overwrite. Note: Data for system accounts never gets accumulated.

Example

The following example inserts an HsvMDDataBuffer object's cells into an application. The arguments set UpdateDataUsingMDDataBuffer to the accumulate update mode and to accumulate within the HsvMDDataBuffer object's cells.

```
m_cData.UpdateDataUsingMDDataBuffer cMDBuffer, _  
HSV_DATA_UPDATE_ACCUMULATE, True
```

In This Chapter

HsvCalculate Object Methods	376
Allocate	376
Allocate2	376
CalcEPU	377
ChartLogic	377
ChartLogic2	378
Consolidate	379
Consolidate2	380
CustomLogic	381
FindOverlappingConsolidation	381
GetCOMDLLRules	382
GetConsolidationProgress	382
GetDefaultExchangeRate	383
GetEPUInfo	384
GetVBScriptRules	385
IsEntityAnEPUOwner	385
LoadCalcManagerRules	386
LoadCalcManagerRules2	386
SetCOMDLLRules	386
SetVBScriptRules	386
SetVBScriptRules2	388
StopConsolidation	390
Translate	390
Translate2	391

This chapter describes the members of the HsvCalculate type library. The methods of this type library are used to execute consolidations, translations, and calculations.

To use the HsvCalculate type library, you must reference `HsvCalculate.dll` in your project. The HsvCalculate type library contains one object—the HsvCalculate object.

HsvCalculate Object Methods

The HsvCalculate object's methods execute calculations, consolidations, and translations. These methods are summarized in [Table 22 on page 74](#), and are described in detail in the following topics.

Note: Set HsvCalculate object references with the `Calculate` property of the HsvSession object. For an example, see [“HsvCalculate Type Library Overview” on page 74](#).

Allocate

Allocates an entity's data for the specified Scenario, Year, Period, and Value dimension members.

Syntax

```
<HsvCalculate>.Allocate lScenario, lYear, lPeriod, lEntity, lParent, lValue
```

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

<i>IScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
------------------	---

<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
--------------	---

<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
----------------	---

<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member for which the allocation is to be run.
----------------	---

<i>lParent</i>	Long (ByVal). The member ID of the parent entity of the <i>lEntity</i> argument's entity.
----------------	---

<i>lValue</i>	Long (ByVal). The member ID of the Value dimension member.
---------------	--

Allocate2

Allocates an entity's data across a range of periods for the specified Scenario, Year, and Value dimension members.

Syntax

```
<HsvCalculate>.Allocate2 lScenario, lYear, lStartPeriod, lEndPeriod,  
lEntity, lParent, lValue
```

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

<i>IScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
------------------	---

<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
--------------	---

<i>lStartPeriod</i>	Long (ByVal). The member ID of the first Period dimension member in the range of periods to be allocated.
---------------------	---

<i>lEndPeriod</i>	Long (ByVal). The member ID of the last Period dimension member in the range of periods to be allocated.
-------------------	--

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

<i>IEntity</i>	Long (ByVal). The member ID of the Entity dimension member for which the allocation is to be run.
<i>IParent</i>	Long (ByVal). The member ID of the parent entity of the <i>IEntity</i> argument's entity.
<i>IValue</i>	Long (ByVal). The member ID of the Value dimension member.

CalcEPU

Runs the Equity Pickup for the specified Scenario, Year, and Period.

Syntax

```
<HsvCalculate>.CalcEPU lScenario, lYear, lPeriod, varbForceEPU
```

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

<i>IScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>varbForceEPU</i>	Boolean (ByVal). True enables the Force EPU option, False disables the Force EPU option.

ChartLogic

Calculates an entity's data for the specified Scenario, Year, Period, and Value dimension members.

Syntax

```
<HsvCalculate>.ChartLogic lScenario, lYear, lPeriod, lEntity, lParent, lValue, varbForceChartLogic
```

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

<i>IScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the Entity dimension member for which the calculation is to be run.
<i>IParent</i>	Long (ByVal). The member ID of the parent entity of the <i>IEntity</i> argument's entity.
<i>IValue</i>	Long (ByVal). The member ID of the Value dimension member.

Argument	Description
<i>varbForceChartLogic</i>	Boolean (ByVal). Specifies whether the calculation should be forced if the entity does not require calculation. Specify TRUE to force the calculation, otherwise FALSE.

Example

This example runs a calculation for the dimension members passed to the user-defined `GetMemberID` function, which returns their member IDs. (For information on `GetMemberID`, see the example for `GetItemID`.) These member IDs are then passed to `ChartLogic`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cCalculate As HsvCalculate
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cCalculate = m_cHsvSession.Calculate
cCalculate.ChartLogic lScen, lYear, lPer, lEnt, lPar, lVal, True
```

ChartLogic2

Calculates an entity's data across a range of periods for the specified Scenario, Year, and Value dimension members.

Syntax

```
<HsvCalculate>.ChartLogic2 lScenario, lYear, lStartPeriod, lEndPeriod,
lEntity, lParent, lValue, varbForceChartLogic
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lStartPeriod</i>	Long (ByVal). The member ID of the first Period dimension member in the range of periods to be calculated.
<i>lEndPeriod</i>	Long (ByVal). The member ID of the last Period dimension member in the range of periods to be calculated.
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member for which the calculation is to be run.
<i>lParent</i>	Long (ByVal). The member ID of the parent entity of the <i>lEntity</i> argument's entity.
<i>lValue</i>	Long (ByVal). The member ID of the Value dimension member.

Argument	Description
<i>varbForceChartLogic</i>	Boolean (ByVal). Specifies whether the calculation should be forced if the entity does not require calculation. Specify TRUE to force the calculation, otherwise FALSE.

Example

This example runs a calculation for the dimension members passed to the user-defined `GetMemberID` function, which returns their member IDs. (For information on `GetMemberID`, see the example for [GetItemID](#).) These member IDs are then passed to `ChartLogic2`.

```
Dim lScen As Long, lYear As Long, lStartPer As Long
Dim lEndPer As Long, lEnt As Long, lPar As Long, lVal As Long
Dim cCalculate As HsvCalculate
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lStartPer = GetMemberID(DIMENSIONPERIOD, "July")
lEndPer = GetMemberID(DIMENSIONPERIOD, "September")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cCalculate = m_cHsvSession.Calculate
cCalculate.ChartLogic2 lScen, lYear, lStartPer, lEndPer, _
lEnt, lPar, lVal, True
```

Consolidate

Consolidates an entity's data for the specified Scenario, Year, and Period dimension members.

Tip: You can avoid executing redundant consolidations by calling [FindOverlappingConsolidation](#) before launching a consolidation.

Syntax

```
<HsvCalculate>.Consolidate lScenario, lYear, lPeriod, lEntity, lParent,
sConsolidationType
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member for which the consolidation is to be run.
<i>lParent</i>	Long (ByVal). The member ID of the parent entity of the <i>lEntity</i> argument's entity.

Argument	Description
<i>sConsolidationType</i>	Integer (ByVal). Identifies the type of consolidation to be run. Pass one of the HFMCconstants type library constants listed in “Consolidation Type Constants” on page 855 .

Example

This example runs consolidation for the dimension members passed to the user-defined `GetMemberID` function, which returns their member IDs. (For information on `GetMemberID`, see the example for `GetItemID`.) These member IDs are then passed to `Consolidate`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long
Dim cCalculate As HsvCalculate
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lPar = GetMemberID(DIMENSIONENTITY, "Regional")
Set cCalculate = m_cHsvSession.Calculate
cCalculate.Consolidate lScen, lYear, lPer, lEnt, lPar, _
CONSOLIDATE_ALL
```

Consolidate2

Consolidates an entity’s data across a range of periods for the specified Scenario and Year dimension members.

Tip: You can avoid executing redundant consolidations by calling [FindOverlappingConsolidation](#) before launching a consolidation.

Syntax

```
<HsvCalculate>.Consolidate2 lScenario, lYear, lStartPeriod, lEndPeriod,
lEntity, lParent, sConsolidationType
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lStartPeriod</i>	Long (ByVal). The member ID of the first Period dimension member in the range of periods to be consolidated.
<i>lEndPeriod</i>	Long (ByVal). The member ID of the last Period dimension member in the range of periods to be consolidated.
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member for which the consolidation is to be run.

Argument	Description
<i>lParent</i>	Long (ByVal). The member ID of the parent entity of the <i>lEntity</i> argument's entity.
<i>sConsolidationType</i>	Integer (ByVal). Identifies the type of consolidation to be run. Pass one of the HFMConstants type library constants listed in "Consolidation Type Constants" on page 855 .

Example

This example runs consolidation for the dimension members passed to the user-defined `GetMemberID` function, which returns their member IDs. (For information on `GetMemberID`, see the example for `GetItemID`.) These member IDs are then passed to `Consolidate2`.

```
Dim lScen As Long, lYear As Long, lPerStart As Long
Dim lPerEnd As Long, lEnt As Long, lPar As Long
Dim cCalculate As HsvCalculate
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPerStart = GetMemberID(DIMENSIONPERIOD, "July")
lPerEnd = GetMemberID(DIMENSIONPERIOD, "September")
lEnt = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lPar = GetMemberID(DIMENSIONENTITY, "Regional")
Set cCalculate = m_cHsvSession.Calculate
cCalculate.Consolidate2 lScen, lYear, lPerStart, lPerEnd, _
lEnt, lPar, CONSOLIDATE_ALL
```

CustomLogic

For internal use.

FindOverlappingConsolidation

Indicates whether any consolidations are currently running or queued for the specified entity, scenario, year, and range of periods. If `FindOverlappingConsolidation` finds any such consolidations, the consolidations' types and dimension member labels are returned.

Tip: You can avoid executing redundant consolidations by calling `FindOverlappingConsolidation` before launching a consolidation.

The information returned by `FindOverlappingConsolidation` includes several arrays. These arrays have a one-to-one correspondence, and contain one item for each consolidation that matches the specified dimension members.

Syntax

```
<HsvCalculate>.FindOverlappingConsolidation lEntity, lParent, lScenario,
lYear, lStartPeriod, lEndPeriod, lConsolType, pbstrYear, pbstrScenario,
```

`pvarabstrEntity`, `pvarabstrParent`, `pvarabstrStartPeriod`,
`pvarabstrEndPeriod`, `pvaralConsolType`, `plTotalOverlapRecords`

Argument	Description
<code>lEntity</code>	Long (ByVal). The member ID of the Entity dimension member.
<code>lParent</code>	Long (ByVal). The member ID of the <code>lEntity</code> argument's parent.
<code>lScenario</code>	Long (ByVal). The member ID of the Scenario dimension member.
<code>lYear</code>	Long (ByVal). The member ID of the Year dimension member.
<code>lStartPeriod</code>	Long (ByVal). The member ID of the first Period dimension member in the range of periods.
<code>lEndPeriod</code>	Long (ByVal). The member ID of the last Period dimension member in the range of periods.
<code>lConsolType</code>	Long (ByVal). Identifies the type of consolidation to test for. Pass one of the constants in the <code>HFMConstants</code> type library enumeration <code>tagCONSOLIDATIONTYPE</code> , which is described in "Consolidation Type Constants" on page 855 .
<code>pbstrYear</code>	String. Returns the label of the Year dimension member for the consolidations, or a blank string if no matching consolidations are found.
<code>pbstrScenario</code>	String. Returns the label of the Scenario dimension member for the consolidations, or a blank string if no matching consolidations are found.
<code>pvarabstrEntity</code>	Variant. Returns an array containing the labels of the consolidations' Entity dimension members, or an empty variant if no matching consolidations are found.
<code>pvarabstrParent</code>	Variant. Returns an array containing the parents of the entities returned by the <code>pvarabstrEntity</code> argument.
<code>pvarabstrStartPeriod</code>	Variant. Returns an array containing the labels of the first periods in the consolidations' ranges of periods, or an empty variant if no matching consolidations are found.
<code>pvarabstrEndPeriod</code>	Variant. Returns an array containing the labels of the last periods in the consolidations' ranges of periods, or an empty variant if no matching consolidations are found.
<code>pvaralConsolType</code>	Variant. Returns an array containing the IDs of the consolidations' types, or an empty variant if no matching consolidations are found. Consolidation types are represented by the <code>HFMConstants</code> type library enumeration <code>tagCONSOLIDATIONTYPE</code> , which is described in "Consolidation Type Constants" on page 855 .
<code>plTotalOverlapRecords</code>	Long. Returns a count of the consolidations currently queued or running for the specified dimension members.

GetCOMDLLRules

For internal use.

GetConsolidationProgress

For internal use.

GetDefaultExchangeRate

Returns the exchange rate between two currencies for the specified Point of View. You must specify a valid Point of View for entering currency rates; for information on entering currency rates, see the *Oracle Hyperion Financial Management, Fusion Edition User's Guide*.

Syntax

```
<HsvCalculate>.GetDefaultExchangeRate lScenario, lYear, lPeriod, lEntity,  
lAccount, lCustom1, lCustom2, lCustom3, lCustom4, lFromCurrencyId,  
lToCurrencyId, pdRate
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Point of View's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Point of View's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Point of View's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the Point of View's Entity dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the Point of View's Account dimension member. This member must have an account type of Currencyrate.
<i>lCustom1</i>	Long (ByVal). The member ID of the source currency for which to obtain the exchange rate. This must be one of the currencies on the system-generated [Currencies] member list for the Custom 1 dimension.
<i>lCustom2</i>	Long (ByVal). The member ID of the destination currency for which to obtain the exchange rate. This must be one of the currencies on the system-generated [Currencies] member list for the Custom 2 dimension.
<i>lCustom3</i>	Long (ByVal). The member ID of the Custom 3 dimension member [None]. Tip: The [None] member of a dimension can be represented by the HFMCconstants type library constant <code>MEMBERNONE</code> .
<i>lCustom4</i>	Long (ByVal). The member ID of the Custom 4 dimension member [None].
<i>lFromCurrencyId</i>	Long (ByVal). The currency ID of the source currency. Tip: You can get a currency's ID from the HsvCurrencies method GetCurrencyID .
<i>lToCurrencyId</i>	Long (ByVal). The currency ID of the destination currency.
<i>pdRate</i>	Double. Returns the exchange rate of the currencies.

Example

The following function returns the exchange rate of two currencies for the specified Point of View. The function take currency names as strings; the *lCustom1* and *lCustom2* arguments' values are obtained by passing these names to `IHsvTreeInfo.GetItemID`, and the *lFromCurrencyId* and *lToCurrencyId* arguments' values are obtained by passing the names to `HsvCurrencies.GetCurrencyID`. The function takes the member IDs of the other dimension members.

```
Function getExchangeRate(lScen As Long, lYear As Long, lPer As Long, _
```

```

    lEnt As Long, sFromCurr As String, sToCurr As String, lAcct As Long) _
    As Double
Dim cCalculate As HsvCalculate, lFromCurr As Long, lToCurr As Long
Dim dRate As Double, cTreeInfo As IHsvTreeInfo
Dim cCurrencies As HsvCurrencies
Dim lCust1 As Long, lCust2 As Long, lCust3 As Long, lCust4 As Long
' g_cMetadata and g_cSession are previously set HsvMetadata and
' HsvSession instances
Set cCurrencies = g_cMetadata.Currencies
Set cCalculate = g_cSession.Calculate
Set cTreeInfo = g_cMetadata.Custom1
lCust1 = cTreeInfo.GetItemID(sFromCurr)
Set cTreeInfo = g_cMetadata.Custom2
lCust2 = cTreeInfo.GetItemID(sToCurr)
lFromCurr = cCurrencies.GetCurrencyID(sFromCurr)
lToCurr = cCurrencies.GetCurrencyID(sToCurr)
cCalculate.GetDefaultExchangeRate lScen, lYear, lPer, lEnt, lAcct, _
    lCust1, lCust2, MEMBERNONE, MEMBERNONE, lFromCurr, lToCurr, dRate
getExchangeRate = dRate
End Function

```

GetEPUInfo

Retrieves the equity pickup information for the specified Scenario, Year, and Period. The method returns an array that includes: owner, owned, percentage ownership, and status.

Syntax

```

<HsvCalculate>.GetEPUInfo lScenario, lYear, lPeriod, varalOwners,
varalOwned, lStatus, lPctEPUOp, dPctEPU, lCirOwn, pvara2DvEPUInfo

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>varalOwners</i>	Variant array of Long (ByVal). Input filtered list of Owner entities. The return list will only include the input owners.
<i>varalOwned</i>	Variant array of Long (ByVal). Input filtered list of owned entities. The return list will only include the input owned entities.
<i>lStatus</i>	Long (ByVal). The filtered EPU Status. Only ownership pairs with selected status are returned.
<i>lPctEPUOp</i>	Long (ByVal). The operator to be used with the dPctEPU value.
<i>dPctEPU</i>	Double (ByVal). The filtered percentage ownership value. Only ownership pairs with percentage ownership value match the operator condition are returned.
<i>lCirOwn</i>	Long (ByVal). The filtered circular ownership status.

Argument	Description
<i>pvara2DvEPUInfo</i>	Variant array. Returns two-dimensional array. The values are returned as Owner (Long), Owned (Long), Level (Long), Percentage Ownership (Double), Status (Long), Circular Ownership flag (Long).

GetVBScriptRules

Returns the rules that have been loaded into an application. The rules are returned as an array of bytes.

Caution! This method returns the contents of the rules file (.RLE file) that has been *loaded* into an application. If the rules file was changed after the file was loaded, the updated file's contents are *not* returned by GetVBScriptRules.

Syntax

```
<HsvCalculate>.GetVBScriptRules pvarabRules, pbRulesExist
```

Argument	Description
<i>pvarabRules</i>	Variant array. Returns the application's rules. The array is returned as a Byte subtype.
<i>pbRulesExist</i>	Boolean. Returns TRUE if a rules file has been loaded into the application, FALSE if no rules file has been loaded.

Example

The following example outputs an application's rules to a text file. The array of bytes returned by GetVBScriptRules is converted to a String and inserted into a file with various Visual Basic methods.

```
Dim cCalc As HsvCalculate, vaRules As Variant
Dim bRulesExist As Boolean, iFile As Integer
Set cCalc = m_cHsvSession.Calculate
cCalc.GetVBScriptRules vaRules, bRulesExist
If bRulesExist = True Then
    iFile = FreeFile
    Open "c:\temp\appRules.rle" For Output As #iFile
    Print #iFile, StrConv(CStr(vaRules), 64)
    Close #iFile
End If
```

IsEntityAnEPUOwner

Indicates whether the entity is an EPU owner for specified Scenario, Year, and Period.

Syntax

```
<HsvCalculate>.IsEntityAnEPUOwner lScenario, lYear, lPeriod, lEntity,
pvbEntityIsAnEPUOwner
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the Entity dimension member for which the query is to be run.
<i>pvbEntityIsAnEPUOwner</i>	Boolean. Returns True if specified Entity is an EPU owner, False otherwise.

LoadCalcManagerRules

For internal use.

LoadCalcManagerRules2

For internal use.

SetCOMDLLRules

For internal use.

SetVBScriptRules

Loads or scans a rules file.

Note: To validate rules for intercompany transactions, load the rules with [SetVBScriptRules2](#).

`SetVBScriptRules` returns arrays that contain the line numbers, severity levels, descriptions, contents, and details of the lines in a rules file that do not pass validation. These arrays have a one-to-one correspondence, with one item for each line that does not pass validation.

Tip: Variables cannot be tested for validity, so whenever `SetVBScriptRules` encounters a line containing a variable, items with a severity level of Information are included in the arrays. The descriptions for these lines say that “validation was not performed.”

Syntax

```
<HsvCalculate>.SetVBScriptRules varabRules, vbScanOnly,
pvbErrorsWereFound, pvbWarningsWereFound, pvbInfoWasProvided,
```

pvaralErrorLineNumbers, *pvaralErrorSeverity*, *pvarabstrErrorDescriptions*,
pvarabstrErrorVBScript, *pvarabstrErrorDetails*

Argument	Description
<i>varabRules</i>	Variant (ByVal). The rules file, passed as an array of bytes.
<i>vbScanOnly</i>	Boolean (ByVal). A flag that specifies whether to load or scan the rules file. Pass TRUE to scan, FALSE to load. Note: If you pass FALSE, the rules file is scanned before loading.
<i>pvbErrorsWereFound</i>	Boolean. Indicates whether <i>SetVBScriptRules</i> found any validation errors. Returns TRUE if errors were found, FALSE otherwise.
<i>pvbWarningsWereFound</i>	Boolean. Indicates whether <i>SetVBScriptRules</i> found any validation warnings. Returns TRUE if warnings were found, FALSE otherwise.
<i>pvbInfoWasProvided</i>	Boolean. Indicates whether <i>SetVBScriptRules</i> returns any information not classified as errors or warnings. Returns TRUE if non-error and non-warning information is returned, FALSE otherwise.
<i>pvaralErrorLineNumbers</i>	Variant array. Returns the line numbers to which the errors, warnings, and information apply. The array is returned as a Long subtype.
<i>pvaralErrorSeverity</i>	Variant array. Returns numbers that indicate the severity levels of the errors, warnings, and information. Valid return values are described in the following list: <ul style="list-style-type: none">● 1 = Error severity level.● 2 = Warning severity level.● 3 = Information severity level. The array is returned as a Long subtype.
<i>pvarabstrErrorDescriptions</i>	Variant array. Returns descriptions of the errors, warnings, and information. The array is returned as a String subtype.
<i>pvarabstrErrorVBScript</i>	Variant array. Returns the rules file statements to which the errors, warnings, and information apply. The array is returned as a String subtype.
<i>pvarabstrErrorDetails</i>	Variant array. Returns details that describe why the lines caused validation to fail. The array is returned as a String subtype.

Example

The following example creates a log file that lists the line numbers, descriptions, and details for any validation messages of a warning severity level. Various Visual Basic methods convert the rules file to an array of bytes, which is then passed to *SetVBScriptRules*; note that *SetVBScriptRules* is set to scan. If the *pvbWarningsWereFound* argument returns TRUE, the example loops through the *pvaralErrorSeverity* argument array to find items flagged with a warning severity level. For each such item, the corresponding items returned in the *pvaralErrorLineNumbers*, *pvarabstrErrorDescriptions*, and *pvarabstrErrorDetails* arguments are concatenated to the *vLogText* variable. When the loop finishes, *vLogText*'s value is written to the log file, which is created by using various Visual Basic methods.

```
Dim cCalculate As HsvCalculate, vFileName, lFile As Long
```

```

Dim lSize As Long, vaRules, bytaRules() As Byte
Dim bErrs As Boolean, bWarnings As Boolean, bInfo As Boolean
Dim valLines, valSeverity, vasDescs, vasErr, vasDetails
Dim vLogText, iFile As Integer
Set cCalculate = m_cHsvSession.Calculate
vFileName = "C:\Program Files\Acme\AppRules.rle"
lFile = FreeFile
lSize = FileLen(vFileName)
Open vFileName For Binary Access Read As #lFile
ReDim bytaRules(lSize)
Get #lFile, , bytaRules
Close #lFile
vaRules = bytaRules
cCalculate.SetVBScriptRules vaRules, True, bErrs, bWarnings, _
bInfo, valLines, valSeverity, vasDescs, vasErr, vasDetails
If bWarnings = True Then
    For i = LBound(valLines) To UBound(valLines)
        If valSeverity(i) = 2 Then
            vLogText = vLogText & "Line #: " & valLines(i) & _
                vbCrLf & "    Description: " & vasDescs(i) & vbCrLf _
                & "    Details: " & vasDetails(i) & vbCrLf & vbCrLf
        End If
    Next i
iFile = FreeFile
Open "C:\Program Files\Acme\warnings.log" For Output As #iFile
Print #iFile, vLogText
Close #iFile
End If

```

SetVBScriptRules2

Loads or scans a rules file, optionally validating whether the rules violate the referential integrity of any intercompany transactions.

SetVBScriptRules2 returns arrays that contain the line numbers, severity levels, descriptions, contents, and details of the lines in a rules file that do not pass validation. These arrays have a one-to-one correspondence, with one item for each line that does not pass validation.

Tip: Variables cannot be tested for validity, so whenever SetVBScriptRules2 encounters a line containing a variable, items with a severity level of Information are included in the arrays. The descriptions for these lines say that “validation was not performed.”

Syntax

```

<HsvCalculate>.SetVBScriptRules2 varabRules, vbScanOnly,
vbCheckRefIntegrity, pvbErrorsWereFound, pvbWarningsWereFound,
pvbInfoWasProvided, pvaralErrorLineNumbers, pvaralErrorSeverity,
pvarabstrErrorDescriptions, pvarabstrErrorVBScript, pvarabstrErrorDetails,
pbstrRefIntegDiagXML

```

Argument	Description
<i>varabRules</i>	Variant (ByVal). The rules file, passed as an array of bytes.
<i>vbScanOnly</i>	Boolean (ByVal). A flag that specifies whether to load or scan the rules file. Pass TRUE to scan, FALSE to load. If you pass FALSE, the rules file is scanned before loading.
<i>vbCheckRefIntegrity</i>	Boolean (ByVal). A flag that specifies whether to validate the referential integrity of intercompany transactions. Pass TRUE to validate intercompany transactions, FALSE otherwise. If you pass TRUE, the <i>pbstrRefIntegDiagXML</i> argument will return information regarding any intercompany transactions that are rendered invalid.
<i>pvbErrorsWereFound</i>	Boolean. Indicates whether <i>SetVBScriptRules2</i> found any validation errors. Returns TRUE if errors were found, FALSE otherwise.
<i>pvbWarningsWereFound</i>	Boolean. Indicates whether <i>SetVBScriptRules2</i> found any validation warnings. Returns TRUE if warnings were found, FALSE otherwise.
<i>pvbInfoWasProvided</i>	Boolean. Indicates whether <i>SetVBScriptRules2</i> returns any information not classified as errors or warnings. Returns TRUE if non-error and non-warning information is returned, FALSE otherwise.
<i>pvaralErrorLineNumbers</i>	Variant array. Returns the line numbers to which the errors, warnings, and information apply. The array is returned as a Long subtype.
<i>pvaralErrorSeverity</i>	Variant array. Returns numbers that indicate the severity levels of the errors, warnings, and information. Valid return values are described in the following list: <ul style="list-style-type: none"> ● 1 = Error severity level. ● 2 = Warning severity level. ● 3 = Information severity level. The array is returned as a Long subtype.
<i>pvarabstrErrorDescriptions</i>	Variant array. Returns descriptions of the errors, warnings, and information. The array is returned as a String subtype.
<i>pvarabstrErrorVBScript</i>	Variant array. Returns the rules file statements to which the errors, warnings, and information apply. The array is returned as a String subtype.
<i>pvarabstrErrorDetails</i>	Variant array. Returns details that describe why the lines caused validation to fail. The array is returned as a String subtype.
<i>pbstrRefIntegDiagXML</i>	String. Returns an XML document that indicates any referential integrity errors; this argument is used if TRUE is passed to the <i>vbCheckRefIntegrity</i> argument. The XML document is structured as follows, with one <code><cell></code> tag for each intercompany transaction for which the rules file violates the referential integrity: <pre> <refint> <module name="ICM"> <table name="ICT_TRANSACTIONS"> <section name="supportstran"> <cell scenario="label" year="label" period="label" entity="label" account=="label" icp=="label" custom1="label" custom2="label" custom3="label" custom4="label" /> </section> </table> </module> </refint> </pre>

Argument**Description**

```

</table>
</module>
</refint>

```

The attributes of `<cell>` specify the member labels of a transaction's dimension members

StopConsolidation

For internal use.

Translate

Translates an entity's data from one currency to another for the specified Scenario, Year, Period, and Value dimension members.

Tip: `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

```

<HsvCalculate>.Translate lScenario, lYear, lPeriod, lEntity, lParent,
lValue, varbForceTranslate, bTranslateAll

```

Argument**Description***lScenario*

Long (ByVal). The member ID of the Scenario dimension member.

lYear

Long (ByVal). The member ID of the Year dimension member.

lPeriod

Long (ByVal). The member ID of the Period dimension member.

lEntity

Long (ByVal). The member ID of the Entity dimension member for which the consolidation is to be run.

*lParent*Long (ByVal). The member ID of the parent entity of the *lEntity* argument's entity.*lValue*

Long (ByVal). The member ID of the Value dimension member for the currency into which the amount is to be translated.

varbForceTranslate

Boolean (ByVal). Determines whether translations will be forced. Specify TRUE to force translations, otherwise FALSE.

Tip: Specifying TRUE is analogous to Financial Management's Force Translate command.

bTranslateAll

Boolean (ByVal). This argument has no effect in the current release. You must pass a value, but the value is ignored.

Example

This example translates the Italy entity's amount to the USD currency. The dimension members' labels are passed to the user-defined `GetMemberID` function, which returns their member IDs. (For information on `GetMemberID`, see the example for `GetItemID`.) These member IDs are then passed to `Translate`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cCalculate As HsvCalculate
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Italy")
lPar = GetMemberID(DIMENSIONENTITY, "Europe")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cCalculate = m_cHsvSession.Calculate
cCalculate.Translate lScen, lYear, lPer, lEnt, lPar, lVal, _
False, False
```

Translate2

Translates an entity's data from one currency to another across a range of periods for the specified Scenario, Year, and Value dimension members.

Tip: `Translate2` uses the default exchange rates defined for an application. If an application has a translation rule, `Translate2` executes the rule instead of applying the default exchange rates.

Syntax

```
<HsvCalculate>.Translate2 lScenario, lYear, lStartPeriod, lEndPeriod,  
lEntity, lParent, lValue, varbForceTranslate
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lStartPeriod</i>	Long (ByVal). The member ID of the first Period dimension member in the range of periods to be translated.
<i>lEndPeriod</i>	Long (ByVal). The member ID of the last Period dimension member in the range of periods to be translated.
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member for which the consolidation is to be run.
<i>lParent</i>	Long (ByVal). The member ID of the parent entity of the <i>lEntity</i> argument's entity.

Argument	Description
<i>lValue</i>	Long (ByVal). The member ID of the Value dimension member for the currency into which the amount is to be translated.
<i>varbForceTranslate</i>	Boolean (ByVal). Determines whether translations will be forced. Specify TRUE to force translations, otherwise FALSE. Tip: Specifying TRUE is analogous to Financial Management's Force Translate command

Example

This example translates the Italy entity's amount to the USD currency. The dimension members' labels are passed to the user-defined `GetMemberID` function, which returns their member IDs. (For information on `GetMemberID`, see the example for `GetItemID`.) These member IDs are then passed to `Translate2`.

```
Dim lScen As Long, lYear As Long, lStartPer As Long
Dim lEndPer As Long, lEnt As Long, lPar As Long, lVal As Long
Dim cCalculate As HsvCalculate
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lStartPer = GetMemberID(DIMENSIONPERIOD, "July")
lEndPer = GetMemberID(DIMENSIONPERIOD, "September")
lEnt = GetMemberID(DIMENSIONENTITY, "Italy")
lPar = GetMemberID(DIMENSIONENTITY, "Europe")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cCalculate = m_cHsvSession.Calculate
cCalculate.Translate2 lScen, lYear, lStartPer, lEndPer, lEnt, _
lPar, lVal, False
```


11

HsvJournals Type Library

In This Chapter

Journal Column Return Values.....	393
Template Column Return Values.....	394
HsvJournals Object Methods.....	395
IHsvJournalsEx Interface Methods.....	399
IHsvJournalsReport Interface Methods.....	444

This chapter describes the members of the HsvJournals type library. The properties and methods of this type library are used to manage periods, create journals and templates, and process journals.

To use the HsvJournals type library, you must reference `HsvJournals.dll` in your project.

The HsvJournals type library provides the HsvJournals object and the IHsvJournalsEx and IHsvJournalsReport interfaces. For an overview of these objects and interfaces, see “[HsvJournals Type Library Overview](#)” on page 75.

Journal Column Return Values

Some methods return arrays of values that correspond to journals’ display columns; these are the columns listed in the Columns tab of the Filters And Sorting dialog box. The following table describes the array elements returned for these columns.

Tip: For methods that take journal display columns, use the HFMConstants type library constants listed in “[Journal Column Display Constants](#)” on page 848.

Table 62 Journal Column Return Values

Column	Return Value (Subtype and Description)
Label	String. The journal label.
Status	Integer. The journal status. The valid return values are represented by the HFMConstants type library constants listed in “ Journal Status Constants ” on page 851.
Type	Integer. The journal type. The valid return values are represented by the HFMConstants type library constants listed in “ Journal Type Constants ” on page 851.

Column	Return Value (Subtype and Description)
Balance Type	Integer. The balance type. The valid return values are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 848 .
Group	String. The name of the journal group.
Description	String. The journal description.
Short Description	String. The truncated version of the journal description.
Created By	String. The fully qualified username of the journal creator.
Date Created	Double. The time and date on which the journal was created. This is returned as a Double that you can cast to a Date format.
Approved By	String. The fully qualified username of the journal approver. If a journal has not been approved, a blank String is returned.
Approved On	Double. The time and date on which the journal was approved. This is returned as a Double that you can cast to a Date format. If a journal has not been approved, 0 is returned.
Posted By	String. The fully qualified username of the journal poster. If a journal has not been posted, a blank String is returned.
Date Posted	Double. The time and date on which the journal was posted. This is returned as a Double that you can cast to a Date format. If a journal has not been posted, 0 is returned.
Security Class	Long. The ID number of the journal's security class.
Entity	Long. The member ID of the journal's entity.
Parent	Long. The member ID of the journal's parent entity.

Template Column Return Values

Some methods return arrays of values that correspond to templates' display columns; these are the columns listed in the Columns tab of the Filters And Sorting dialog box. The following table describes the array elements returned for these columns.

Tip: For methods that take template display columns, use the HFMConstants type library constants listed in [“Template Column Display Constants” on page 851](#).

Table 63 Template Column Return Values

Column	Return Value (Subtype and Description)
Label	String. The template label.

Column	Return Value (Subtype and Description)
Balance Type	Integer. The balance type. The valid return values are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 848 .
Group	String. The name of the journal group.
Description	String. The template description.
Short Description	String. The truncated version of the template description.
Type	Integer. The template type. The valid return values are represented by the HFMConstants type library constants listed in “Template Type Constants” on page 852 .
Entity	Long. The member ID of the template’s entity.
Parent	Long. The member ID of the template’s parent entity.
Value	Long. The member ID of the template’s Value dimension member.

HsvJournals Object Methods

The HsvJournals object’s methods open and close periods and get IDs of templates and journals. These methods are summarized in [Table 23 on page 76](#), and are described in detail in the following topics.

Assign HsvJournals references with the `Journals` property of the HsvSession object. For an example, see [“Journals” on page 158](#).

ClosePeriod

Closes a period, meaning that journals can no longer be posted for the period. Since periods in Financial Management apply to years and scenarios, you must pass the member IDs of the applicable scenario and year in addition to the member ID of the period.

Tip: For information on getting member IDs, see [“About Member IDs” on page 161](#).

Syntax

```
<HsvJournals>.ClosePeriod lCategory, lYear, lPeriod
```

Argument Description

lCategory Long (ByVal). The member ID of the Scenario dimension member.

lYear Long (ByVal). The member ID of the Year dimension member.

lPeriod Long (ByVal). The member ID of the Period dimension member to be closed.

Example

This example closes the August period for the Actual scenario in the year 2000. The member IDs of the Scenario, Year, and Period dimension members are obtained with calls to the user-defined `GetMemberID` function; for details on `GetMemberID`, see the example for [GetItemID](#).

```
Dim lScen As Long, lYear As Long, lPer As Long
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "August")
Dim cHsvJournals As HsvJournals
Set cHsvJournals = m_cHsvSession.Journals
cHsvJournals.ClosePeriod lScen, lYear, lPer
```

GetItemID

Returns the internal ID of a journal. You pass the journal's label and `GetItemID` returns the journal ID. Use `GetItemID` to get journal IDs for methods that take these IDs as arguments.

Syntax

```
<HsvJournals>.GetItemID(lCategory, lYear, lPeriod, bstrLabel)
```

Argument Description

lCategory Long (ByVal). The member ID of the Scenario dimension member.

lYear Long (ByVal). The member ID of the Year dimension member.

lPeriod Long (ByVal). The member ID of the Period dimension member.

bstrLabel String (ByVal). The label of the journal for which you want to get the ID.

Return Value

Long. The ID number of the journal.

Example

`GetItemID` is used in the [Example](#) for `ApproveJournals`.

GetJournalTemplateItemID

Returns the internal ID of a journal template. You pass the template's label and `GetJournalTemplateItemID` returns the template ID. Use `GetJournalTemplateItemID` to get template IDs for methods that take these IDs as arguments.

Syntax

```
<HsvJournals>.GetJournalTemplateItemID(bstrLabel)
```

Argument Description

bstrLabel String (ByVal). The label of the template.

Return Value

Long. The ID of the template.

Example

`GetJournalTemplateItemID` is used in the [Example](#) for `DeleteTemplates`.

GetPeriodStatusList

Returns a two-dimensional array that indicates whether the periods for a scenario and year are opened, unopened, or closed.

Syntax

```
<HsvJournals>.GetPeriodStatusList lCategory, lYear, pvarlarrIDs
```

Argument Description

lCategory Long (ByVal). The member ID of the Scenario dimension member.

lYear Long (ByVal). The member ID of the Year dimension member.

pvarlarrIDs Variant array. Returns a two-dimensional array, with the first dimension listing the periods' member IDs and the second dimension listing the periods' statuses.

The valid return values for the second dimension are represented by the `HFMConstants` type library constants listed in "[Period Status Constants](#)" on page 849.

The first dimension is returned as a Long subtype, and the second dimension is returned as an Integer subtype.

Example

This example prints the labels of the open periods to the Immediate window by looping through the array returned by `GetPeriodStatusList`. The calls to the user-defined `GetMemberID` function get the example's Scenario and Year member IDs; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`.

```
Dim lScen As Long, lYear As Long, vaIds
Dim cTreeInfo As IHsvTreeInfo, sPer As String
Set cTreeInfo = m_cHsvMetadata.Periods
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
Set m_cHsvJournals = m_cHsvSession.Journals
m_cHsvJournals.GetPeriodStatusList lScen, lYear, vaIds
For i = LBound(vaIds, 2) To UBound(vaIds, 2)
    If vaIds(1, i) = JPS_OPENED Then
        cTreeInfo.GetLabel vaIds(0, i), sPer
        Debug.Print sPer & " is open."
    End If
```

IsPeriodOpen

Indicates whether a period is open. Since periods in Financial Management apply to years and scenarios, you must pass the member IDs of the applicable scenario and year in addition to the member ID of the period.

Tip: You should call `IsPeriodOpen` before calling methods such as `PostJournals` and `UnpostJournals` that require an open period. If a period is not open, open it with `OpenPeriod`. For information on opening periods, see “[OpenPeriod](#)” on page 398.

Syntax

```
<HsvJournals>.IsPeriodOpen lCategory, lYear, lPeriod, pnRc
```

Argument Description

lCategory Long (ByVal). The member ID of the Scenario dimension member.

lYear Long (ByVal). The member ID of the Year dimension member.

lPeriod Long (ByVal). The member ID of the Period dimension member to be checked.

pnRc Integer. Returns -1 if the period is open, or 0 if the period is unopened or closed.

OpenPeriod

Opens a period, allowing users to post journals to the period. Since periods in Financial Management apply to years and scenarios, you must pass the member IDs of the applicable scenario and year in addition to the member ID of the period.

Syntax

```
<HsvJournals>.OpenPeriod lCategory, lYear, lPeriod
```

Argument Description

lCategory Long (ByVal). The member ID of the Scenario dimension member.

lYear Long (ByVal). The member ID of the Year dimension member.

lPeriod Long (ByVal). The member ID of the Period dimension member to be opened.

Example

`OpenPeriod` is used in the example for [SaveJournal](#).

IHsvJournalsEx Interface Methods

The IHsvJournalsEx interface's methods create templates, create journals, and process journals. The following list describes common usages of the IHsvJournalsEx methods:

- `SaveTemplate` creates journal templates.
- `SaveJournal` creates journals. Note that you can create a journal from a template by passing the template information returned by `GetTemplate` to `SaveJournal`.
- `SubmitJournals`, `UnsubmitJournals`, `ApproveJournals`, `RejectJournals`, `PostJournals`, and `UnpostJournals` process journals.

Tip: `SaveTemplate`, `SaveJournal`, and `GetTemplate` pass line item amounts as `Double` arrays. To pass line item amounts as `String` arrays, use the corresponding `SaveTextTemplate`, `SaveTextJournal`, and `GetTextTemplate` methods.

These methods are summarized in [Table 24 on page 77](#), and are described in detail in the following topics.

Note: Assign IHsvJournalsEx object references with the `Journals` property of the `HsvSession` object. For an example, see [“Journals” on page 158](#).

AddJournalGroup

Creates a journal group.

Syntax

```
<IHsvJournalsEx>.AddJournalGroup bstrGroup, bstrDescription
```

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

<code>bstrGroup</code>	String (ByVal). The name of the journal group.
------------------------	--

<code>bstrDescription</code>	String (ByVal). The description of the journal group.
------------------------------	---

ApproveJournals

Approves one or more journals. The `varar1JournalIDs` argument takes the IDs of the journals that are being approved.

Caution! The user must be assigned to the Journals Administrator or Approve Journals role, otherwise error number 40D (hexadecimal) will occur.

Syntax

```
<IHsvJournalsEx>.ApproveJournals lScenario, lYear, varar1JournalIDs,  
pvararnRc
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>vararJnlIDs</i>	Long array (ByVal). The IDs of the journals to be approved. Get these IDs with the HsvJournals object's <code>GetItemID</code> method; for more information, see "GetItemID" on page 396 .
<i>pvararnRc</i>	Variant array. Returns codes that identify the success or failure of the journal approvals. The elements in this array have a one-to-one correspondence with the journal IDs passed in the <i>vararJnlIDs</i> argument. Successfully approved journals return 0. The array is returned as an Integer subtype.

Example

This example approves two journals. The calls to the user-defined `GetMemberID` function get the example's dimension member IDs; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`. The `laJnlIDs` array variable is then populated with the journal IDs by `HsvJournals.GetItemID`. The member IDs for the Scenario and Year dimension are passed along with this array to `ApproveJournals`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim laJnlIDs(1) As Long, vaRetVal
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
laJnlIDs(0) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
    "Jnl230")
laJnlIDs(1) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
    "Jnl231")
m_cIHsvJournalEx.ApproveJournals lScen, lYear, laJnlIDs, _
vaRetVal
```

DeleteJournals

Deletes one or more journals. The *vararJnlIDs* argument takes the IDs of the journals that are being deleted.

Caution! The user must be assigned to the Journals Administrator or Create Journals role, otherwise error number 40D (hexadecimal) will occur.

Syntax

```
<IHsvJournalsEx>.DeleteJournals lScenario, lYear, vararJnlIDs,
pvararnRc
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.

Argument	Description
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>vararlJournalIDs</i>	Long array (ByVal). The IDs of the journals to be deleted. Get these IDs with the HsvJournals object's <code>GetItemID</code> method; for more information, see “GetItemID” on page 396 . Caution! You cannot delete journals that have been posted or approved.
<i>pvararnRc</i>	Variant array. Returns codes that identify the success or failure of the journal deletions. The elements in this array have a one-to-one correspondence with the journal IDs passed in the <i>vararlJournalIDs</i> argument. Successfully deleted journals return 0. The array is returned as an Integer subtype. Tip: If the user does not have ALL access to a journal's security class, error code 40F (hexadecimal) is returned. If a journal has been posted or approved it cannot be deleted, and error code 418 (hexadecimal) is returned.

Example

This example deletes two journals. The calls to the user-defined `GetMemberID` function get the example's dimension member IDs; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`. The `laJnlIDs` array variable is then populated with the journal IDs by `HsvJournals.GetItemID`. The member IDs for the Scenario and Year dimension are passed along with this array to `DeleteJournals`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim laJnlIDs(1) As Long, vaRetVal
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
laJnlIDs(0) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
    "Jnl392")
laJnlIDs(1) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
    "Jnl393")
m_cIHsvJournalEx.DeleteJournals lScen, lYear, laJnlIDs, vaRetVal
```

DeleteTemplates

Deletes one or more journal templates.

Caution! The user must be assigned to the Journals Administrator role, otherwise error number 40D (hexadecimal) will occur.

Syntax

```
<IHsvJournalsEx>.DeleteTemplates vararlTemplateIDs, pvararnRc
```

Argument	Description
<i>vararlTemplateIDs</i>	Long array (ByVal). The IDs of the templates to be deleted. Use <code>GetJournalTemplateItemID</code> to get template IDs; for more information, see “GetJournalTemplateItemID” on page 396 .

Argument	Description
<i>pvararrRc</i>	Variant array. Returns codes that identify the success or failure of the template deletions. The elements in this array have a one-to-one correspondence with the template IDs passed in the <i>vararrTemplateIDs</i> argument. Successfully deleted templates return 0. The array is returned as an Integer subtype.

Example

This example deletes a template named TempUsAdjs. The example gets the template ID with a call to `GetJournalTemplateItemID`, then passes the ID to `DeleteTemplates`.

```
Dim laID(0) As Long, vaRetVal
laID(0) = m_cHsvJournals.GetJournalTemplateItemID("TempUsAdjs")
m_cIHsvJournalEx.DeleteTemplates laIDs, vaRetVal
```

EnumJournalGroups

Returns the names and descriptions of an application's journal groups. The information is returned in arrays that have a one-to-one correspondence.

Syntax

```
<IHsvJournalsEx>.EnumJournalGroups pvarabstrGroups, pvarabstrDescriptions
```

Argument	Description
<i>pvarabstrGroups</i>	Variant array. Returns the names of the journal groups. The array is returned as a String subtype.
<i>pvarabstrDescriptions</i>	Variant array. Returns the descriptions of the journal groups. The array is returned as a String subtype.

EnumJournalGroupsForScenarioYear

Returns the names and descriptions of the journal groups assigned to journals for a given scenario and year. The information is returned in arrays that have a one-to-one correspondence.

Syntax

```
<IHsvJournalsEx>.EnumJournalGroupsForScenarioYear lScenario, lYear,
pvarabstrGroups, pvarabstrDescriptions
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>pvarabstrGroups</i>	Variant array. Returns the names of the journal groups.

Argument	Description
	The array is returned as a String subtype.
<i>pvarabstrDescriptions</i>	Variant array. Returns the descriptions of the journal groups. The array is returned as a String subtype.

EnumJournalIDsForExtractFilter

Returns the IDs of journals and templates that match the specified filtering criteria. The flag passed to the *IJournalTypes* argument determines whether IDs of journals, standard templates, recurring templates, or some combination thereof are returned.

`EnumJournalIDsForExtractFilter` provides arguments for numerous filtering criteria. For filtering criteria you do not want to use, pass an empty variable to the corresponding argument; only arguments containing values are used to filter. This principle applies to all arguments other than the *IScenario* and *IYear* arguments, for which valid member IDs are required.

Syntax

```
<IHsvJournalsEx>.EnumJournalIDsForExtractFilter IScenario, IYear,
varalPeriods, varalEntities, varalParents, varalValues, IJournalTypes,
varabstrLabels, varabstrGroups, varalStatus, varalAutoTypes,
varalBalanceTypes, pvaralJournalIDs, pvaralTemplateIDs
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the Scenario dimension member by which to filter.
<i>IYear</i>	Long (ByVal). The member ID of the Year dimension member by which to filter.
<i>varalPeriods</i>	Long array (ByVal). The member IDs of the Period dimension members by which to filter.
<i>varalEntities</i>	Long array (ByVal). The member IDs of the Entity dimension members by which to filter.
<i>varalParents</i>	Long array (ByVal). The member IDs of the parents of the Entity dimension members passed to the <i>varalEntities</i> argument.
<i>varalValues</i>	Long array (ByVal). The member IDs of the Value dimension members by which to filter.
<i>IJournalTypes</i>	Long (ByVal). A flag that determines whether journals, standard templates, recurring templates, or some combination thereof are returned. Valid values are represented by the <code>HFMConstants</code> type library constants listed in “Journal and Template Type Constants” on page 852 . To filter with more than one of these items, use <code>Or</code> with the applicable constants.
<i>varabstrLabels</i>	String array (ByVal). The journal labels by which to filter. You can use the percent sign (%) as a wildcard.
<i>varabstrGroups</i>	String array (ByVal). The journal groups by which to filter. You can use the percent sign (%) as a wildcard.
<i>varalStatus</i>	Long array (ByVal). The journal statuses by which to filter. Valid values are represented by the <code>HFMConstants</code> type library constants listed in “Journal Status Constants” on page 851 .

Argument	Description
<i>varalAutoTypes</i>	Long array (ByVal). The journal types by which to filter. Valid values are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 851 .
<i>varalBalanceTypes</i>	Long array (ByVal). The balance types by which to filter. Valid values are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 848 .
<i>pvaralJournalIDs</i>	Variante array. Returns the journal IDs that match the filtering criteria, if the <i>IJournalTypes</i> argument specifies that journals are to be returned. The array is returned as a Long subtype.
<i>pvaralTemplateIDs</i>	Variante array. Returns the template IDs that match the filtering criteria, if the <i>IJournalTypes</i> argument specifies that templates are to be returned. The array is returned as a Long subtype.

GenerateRecurring

Generates a journal from a recurring template.

Syntax

```
<IHsvJournalsEx>.GenerateRecurring lScenario, lYear, lPeriod, lTemplateID
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>lTemplateID</i>	Long (ByVal). The ID of the journal template.

Example

The following subroutine creates a recurring journal. The subroutine takes the member IDs of the journal’s Scenario, Year, and Period dimension members and the label of the recurring template. `HsvJournals.GetJournalTemplateItemID` gets the template’s ID, which is then passed to `GenerateRecurring`.

```
Sub createRecurJnl(lScen As Long, lYear As Long, _
    lPer As Long, sTemplate As String)
Dim cJournals As HsvJournals, cJournalsEx As IHsvJournalsEx
Dim lTemplateId As Long
Set cJournals = g_cSession.Journals
Set cJournalsEx = g_cSession.Journals
lTemplateId = cJournals.GetJournalTemplateItemID(sTemplate)
cJournalsEx.GenerateRecurring lScen, lYear, lPer, lTemplateId
End Sub
```

GetEntityJournals

Returns journal entry information for all journals that match the specified Point of View. The information is returned in arrays that have a one-to-one correspondence.

Syntax

```
<IHsvJournalsEx>.GetEntityJournals lScenario, lYear, lPeriod, lValue,  
lEntity, lParent, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,  
pvaralCreditDebitUnit, pvaradAmount, pvarabstrLabel, pvaralICP,  
pvaralCustom1, pvaralCustom2, pvaralCustom3, pvaralCustom4, pvarabstrGroup
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Point of View's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Point of View's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Point of View's Period dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the Point of View's Value dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the Point of View's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's Entity dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the Point of View's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the Point of View's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the Point of View's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the Point of View's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the Point of View's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the Point of View's Custom 4 dimension member.
<i>pvaralCreditDebitUnit</i>	Variant array. Indicates whether the journal entries are debits or credits. Valid return values are represented by the HFConstants type library constants listed in "Debit/Credit Constants" on page 849 . The array is returned as an Long subtype.
<i>pvaradAmount</i>	Variant array. Returns the amounts of the journal entries. The array is returned as a Double subtype.
<i>pvarabstrLabel</i>	Variant array. Returns the labels of the journal entries. The array is returned as a String subtype.
<i>pvaralICP</i>	Variant array. Returns the member IDs of the Intercompany Partner dimension members for the journal entries. The array is returned as a Long subtype.
<i>pvaralCustom1</i>	Variant array. Returns the member IDs of the Custom 1 dimension members for the journal entries. The array is returned as a Long subtype.

Argument	Description
<i>pvaralCustom2</i>	Variant array. Returns the member IDs of the Custom 2 dimension members for the journal entries. The array is returned as a Long subtype.
<i>pvaralCustom3</i>	Variant array. Returns the member IDs of the Custom 3 dimension members for the journal entries. The array is returned as a Long subtype.
<i>pvaralCustom4</i>	Variant array. Returns the member IDs of the Custom 4 dimension members for the journal entries. The array is returned as a Long subtype.
<i>pvarabstrGroup</i>	Variant array. Returns the labels of the journal groups for the journal entries. The array is returned as a Long subtype.

GetJournal

Returns a variety of information for a journal.

Caution! To successfully call `GetJournal`, the user must be assigned to the Journals Administrator or Read Journals role, and must have Read or All access to the journal's security class. For posted journals, the user must also have Read or All access for the entities in the journal's line items.

The amounts for the journal's line items are returned in an array with a Double subtype. To return line item amounts in an array with a String subtype, use `GetTextJournal`. `GetTextJournal` is almost identical to `GetJournal`; the only difference is the subtype of the line item amount array. For more information, see [“GetTextJournal” on page 420](#).

The first three arguments specify the scenario, year, and journal IDs. The next several arguments return journal header information, and the remaining arguments return line item data in arrays. The arrays have a one-to-one correspondence; for example, the second elements in the *pvaradAmount* and *pvararlAccount* arguments' arrays return the amount and account for the journal's second line item.

Syntax

```
<IHsvJournalsEx>.GetJournal lScenario, lYear, lJournalID, plPeriod,
plValue, pnType, pnStatus, pnAttribute, pbstrLabel, pbstrDescription,
pbstrGroup, plSingleEntity, plSingleParent, plSecurityClass,
pvararlEntryID, pvararnDebitCreditUnit, pvaradAmount,
pvararbstrDescription, pvararlEntity, pvararlParent, pvararlAccount,
pvararlICP, pvararlCustom1, pvararlCustom2, pvararlCustom3, pvararlCustom4
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.

Argument	Description
<i>IJournalID</i>	Long (ByVal). The ID of the journal. Get this ID with <code>HsvJournals.GetItemID</code> ; for details, see “GetItemID” on page 396 .
<i>plPeriod</i>	Long. Returns the ID of the journal's period.
<i>plValue</i>	Long. Returns the member ID of the Value dimension member for the journal's currency.
<i>pnType</i>	Integer. Returns the journal's type. The valid return values are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 851 .
<i>pnStatus</i>	Integer. Returns the journal's status. The valid return values are represented by the HFMConstants type library constants listed in “Journal Status Constants” on page 851 .
<i>pnAttribute</i>	Integer. Returns a value that indicates whether the journal must be balanced. The valid return values are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 848 .
<i>pbstrLabel</i>	String. Returns the label of the journal.
<i>pbstrDescription</i>	String. Returns the description of the journal.
<i>pbstrGroup</i>	String. Returns the description of the journal group to which the journal has been assigned.
<i>plSingleEntity</i>	Long. For single entity journals, this argument returns the ID of the journal's base entity. If the journal is a multi-entity journal, this argument returns -1.
<i>plSingleParent</i>	Long. For single entity journals, this argument returns the ID of the journal's parent entity. If the journal is a multi-entity journal, this argument returns -1.
<i>plSecurityClass</i>	Long. Returns the ID of the journal's security class. Tip: To get the label of the security class, pass this ID to <code>HsvSecurityAccess.GetSecurityClassLabel</code> . For more information, see “GetSecurityClassLabel” on page 473 .
<i>pvararEntryID</i>	Variant array. <i>For internal use.</i>
<i>pvararnDebitCreditUnit</i>	Variant array. Indicates whether the line items are debits or credits. Valid return values are represented by the HFMConstants type library constants listed in “Debit/Credit Constants” on page 849 . The array is returned as an Integer subtype.
<i>pvaradAmount</i>	Variant array. Returns the amounts for the journal's line items. The array is returned as a Double subtype. Tip: Use <code>GetVariance</code> to return the total debit and credit amounts and the difference between these amounts. For details, see “GetVariance” on page 425 .
<i>pvararbstrDescription</i>	Variant array. Returns the descriptions for the journal's line items. The array is returned as a String subtype.
<i>pvararEntity</i>	Variant array. Returns the member IDs for the base entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>plSingleEntity</i> argument. The array is returned as a Long subtype.

Argument	Description
<i>pvararParent</i>	Variant array. Returns the member IDs for the parent entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>plSingleParent</i> argument. The array is returned as a Long subtype.
<i>pvararAccount</i>	Variant array. Returns the member IDs of the line items' accounts. The array is returned as a Long subtype.
<i>pvararICP</i>	Variant array. Returns the member IDs of the line items' Intercompany Partner dimension members. The array is returned as a Long subtype.
<i>pvararCustom1</i>	Variant array. Returns the member IDs of the line items' Custom 1 dimension members. The array is returned as a Long subtype.
<i>pvararCustom2</i>	Variant array. Returns the member IDs of the line items' Custom 2 dimension members. The array is returned as a Long subtype.
<i>pvararCustom3</i>	Variant array. Returns the member IDs of the line items' Custom 3 dimension members. The array is returned as a Long subtype.
<i>pvararCustom4</i>	Variant array. Returns the member IDs of the line items' Custom 4 dimension members. The array is returned as a Long subtype.

Example

The following example shows how to call `GetJournal`; after calling `GetJournal`, you can get information about the journal from any of the non-ByVal arguments. The example declares several variables, most of which are used as `GetJournal`'s arguments. The user-defined `GetMemberID` function gets the member IDs for the dimension members passed to `GetJournal`; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`. `HsvJournals.GetItemID` gets the ID of the journal passed to `GetJournal`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lJnlPer As Long, lVal As Long, iType As Integer
Dim iStatus As Integer, iAttr As Integer, sLabel As String
Dim sDescHead As String, sGroup As String, lSingleEnt As Long
Dim lSinglePar As Long, lSecClass As Long, vaEntryIDs
Dim vaDebCredUnit, vaAmt, vaItemDesc, vaEnt, vaPar, vaAcct
Dim vaICP, vaCust1, vaCust2, vaCust3, vaCust4
'Get IDs for the ByVal arguments
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lJnlID = m_cHsvJournals.GetItemID(lScen, lYear, lPer, "Jnl1993")
m_cIHsvJournalEx.GetJournal lScen, lYear, lJnlID, lJnlPer, _
lVal, iType, iStatus, iAttr, sLabel, sDescHead, sGroup, _
lSingleEnt, lSinglePar, lSecClass, vaEntryIDs, _
vaDebCredUnit, vaAmt, vaItemDesc, vaEnt, vaPar, vaAcct, _
vaICP, vaCust1, vaCust2, vaCust3, vaCust4
```


GetJournal2

Returns a variety of information for a journal. This is the same as *GetJournal* except with the addition of the VARIANT_BOOL *vbScaleAmounts* flag which is used to determine whether the textual representation of the amounts are scaled or not.

Caution! To successfully call *GetJournal2*, the user must be assigned to the Journals Administrator or Read Journals role, and must have Read or All access to the journal's security class. For posted journals, the user must also have Read or All access for the entities in the journal's line items.

The amounts for the journal's line items are returned in an array with a Double subtype. To return line item amounts in an array with a String subtype, use *GetTextJournal*. *GetTextJournal* is almost identical to *GetJournal2*; the only difference is the subtype of the line item amount array. For more information, see "[GetTextJournal](#)" on page 420.

The first three arguments specify the scenario, year, and journal IDs.

The fourth argument is the *vbScaleAmounts* flag. The call to *GetJournal* calls *GetJournal2* with a *vbScaleAmounts* = VARIANT_TRUE. This scales the amounts based on the cell's scale factor. Setting the flag to FALSE means that no scaling will occur.

The next several arguments return journal header information, and the remaining arguments return line item data in arrays. The arrays have a one-to-one correspondence; for example, the second elements in the *pvaradAmount* and *pvararlAccount* arguments' arrays return the amount and account for the journal's second line item.

Syntax

```
<HsvJournalsEx>.GetJournal2 lScenario, lYear, lJournalID,  
vbScaleDataValues, plPeriod, plValue, pnType, pnStatus, pnAttribute,  
pbstrLabel, pbstrDescription, pbstrGroup, plSingleEntity, plSingleParent,  
plSecurityClass, pvararlEntryID, pvararnDebitCreditUnit, pvaradAmount,  
pvararbstrDescription, pvararlEntity, pvararlParent, pvararlAccount,  
pvararlICP, pvararlCustom1, pvararlCustom2, pvararlCustom3, pvararlCustom4
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lJournalID</i>	Long (ByVal). The ID of the journal. Get this ID with <i>HsvJournals.GetItemID</i> ; for details, see " GetItemID " on page 396.
<i>vbScaleDataValues</i>	Boolean (ByVal). Determines whether the textual representation of the amounts are scaled or not. Setting to TRUE will scale the amounts based on the cell's scale factor. Setting the flag to FALSE results in no scaling.
<i>plPeriod</i>	Long. Returns the ID of the journal's period.
<i>plValue</i>	Returns the member ID of the Value dimension member for the journal's currency.

Argument	Description
<i>pnType</i>	Integer. Returns the journal's type. The valid return values are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 851 .
<i>pnStatus</i>	Integer. Returns the journal's status. The valid return values are represented by the HFMConstants type library constants listed in “Journal Status Constants” on page 851 .
<i>pnAttribute</i>	Integer. Returns a value that indicates whether the journal must be balanced. The valid return values are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 848 .
<i>pbstrLabel</i>	String. Returns the label of the journal.
<i>pbstrDescription</i>	String. Returns the description of the journal.
<i>pbstrGroup</i>	String. Returns the description of the journal group to which the journal has been assigned.
<i>plSingleEntity</i>	Long. For single entity journals, this argument returns the ID of the journal's base entity. If the journal is a multi-entity journal, this argument returns -1.
<i>plSingleParent</i>	Long. For single entity journals, this argument returns the ID of the journal's parent entity. If the journal is a multi-entity journal, this argument returns -1.
<i>plSecurityClass</i>	Long. Returns the ID of the journal's security class. Tip: To get the label of the security class, pass this ID to <code>HsvSecurityAccess.GetSecurityClassLabel</code> . For more information, see “GetSecurityClassLabel” on page 473 .
<i>pvararEntryID</i>	Variant array. <i>For internal use.</i>
<i>pvararnDebitCreditUnit</i>	Variant array. Indicates whether the line items are debits or credits. Valid return values are represented by the HFMConstants type library constants listed in “Debit/Credit Constants” on page 849 . The array is returned as an Integer subtype.
<i>pvaradAmount</i>	Variant array. Returns the amounts for the journal's line items. The array is returned as a Double subtype. Tip: Use <code>GetVariance</code> to return the total debit and credit amounts and the difference between these amounts. For details, see “GetVariance” on page 425 .
<i>pvararbstDescription</i>	Variant array. Returns the descriptions for the journal's line items. The array is returned as a String subtype.
<i>pvararEntity</i>	Variant array. Returns the member IDs for the base entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>plSingleEntity</i> argument. The array is returned as a Long subtype.
<i>pvararParent</i>	Variant array. Returns the member IDs for the parent entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>plSingleParent</i> argument. The array is returned as a Long subtype.
<i>pvararAccount</i>	Variant array. Returns the member IDs of the line items' accounts. The array is returned as a Long subtype.

Argument	Description
<i>pvararICP</i>	Variant array. Returns the member IDs of the line items' Intercompany Partner dimension members. The array is returned as a Long subtype.
<i>pvararCustom1</i>	Variant array. Returns the member IDs of the line items' Custom 1 dimension members. The array is returned as a Long subtype.
<i>pvararCustom2</i>	Variant array. Returns the member IDs of the line items' Custom 2 dimension members. The array is returned as a Long subtype.
<i>pvararCustom3</i>	Variant array. Returns the member IDs of the line items' Custom 3 dimension members. The array is returned as a Long subtype.
<i>pvararCustom4</i>	Variant array. Returns the member IDs of the line items' Custom 4 dimension members. The array is returned as a Long subtype.

GetJournalDisplayData

Returns various types of information for journals. The types of information returned correspond to the display columns in the Columns tab of the Filters And Sorting dialog box. The journal information is returned as a multidimensional array, with one array dimension for each display column that you specify in the *varalColumns* argument.

Syntax

```
<IHsvJournalsEx>.GetJournalDisplayData lScenario, lYear, varalColumns, varlJournalIDs, pvar2DvarData
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>varalColumns</i>	Long array (ByVal). Specifies the display columns for which journal information will be returned. Use the HFMConstants type library constants listed in “Journal Column Display Constants” on page 848 to create this array.
<i>varlJournalIDs</i>	Long array (ByVal). The ID numbers of the journals for which you want to return information.
<i>pvar2DvarData</i>	Variant array. Returns information for the journals and columns that match the criteria specified in the arguments. This is a multidimensional array, with one dimension for each column specified in the <i>varalColumns</i> argument; in each dimension, there is one element per journal. For details on the values returned in the array elements, see “Journal Column Return Values” on page 393 .

Example

The following example gets the labels, date created, and creators of journals that have labels beginning with “Je,” that have statuses of Working, Submitted, or Approved, and that apply to the following dimension members:

- Scenario = Actual
- Year = 2000
- Period = July
- Value = <Entity Curr Adjs>

The calls to the user-defined `GetMemberID` function get these members’ IDs; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`.

`GetJournalQueryDefinitionIDs` returns the journal IDs of the journals; note that the IDs are sorted in ascending order. These IDs are then passed to `GetJournalDisplayData`, which returns the journal information in the `vaDisplayData` variable. This variable’s information is then printed to Visual Basic’s Immediate window.

```
Dim lScen As Long, lYear As Long, lPer As Long, lVal As Long
Dim laCols(2) As Long, vaJnlIDs, vaDisplayData
Dim laSort(0, 1) As Long, vaFilter(31)
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lVal = GetMemberID(DIMENSIONVALUE, "<Entity Curr Adjs>")
laCols(0) = COLUMN_JOURNALLABEL
laCols(1) = COLUMN_JOURNALCREATEDON
laCols(2) = COLUMN_JOURNALCREATEDBY
laSort(0, 0) = COLUMN_JOURNALLABEL
laSort(0, 1) = 0
vaFilter(0) = "Je%"
vaFilter(1) = Array(JSF_WORKING, JSF_SUBMITTED, JSF_APPROVED)
m_cIHsvJournalEx.GetJournalQueryDefinitionIDs lScen, lYear, _
lPer, lVal, laCols, vaFilter, laSort, vaJnlIDs
m_cIHsvJournalEx.GetJournalDisplayData lScen, lYear, laCols, _
vaJnlIDs, vaDisplayData
For i = LBound(vaJnlIDs) To UBound(vaJnlIDs)
    Debug.Print "Label: " & vaDisplayData(i, 0)
    Debug.Print "Created On: " & CDate(vaDisplayData(i, 1))
    Debug.Print "Created By: " & vaDisplayData(i, 2)
Next i
```

GetJournalLabelsForIDs

Returns the labels of the journals that contain the specified Scenario and Year dimension members and that correspond to a given set of journal IDs.

Syntax

```
<IHsvJournalsEx>.GetJournalLabelsForIDs lScenario, lYear, varalIDs,
pvarabstrLabels
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>varallIDs</i>	Long array (ByVal). The IDs of the journals.
<i>pvarabstrLabels</i>	Variant array. Returns the labels of the journals. This array has a one-to-one correspondence with the array of IDs passed to the <i>varallIDs</i> argument.

GetJournalQueryDefinitionIDs

Returns IDs of one or more journals. You can return IDs of all journals or return only those IDs that meet filtering criteria that you specify.

Syntax

```
<IHsvJournalsEx>.GetJournalQueryDefinitionIDs lScenario, lYear, lPeriod, lValue, varalColumns, vararvFilters, var2DalColumnsSort, pvararlJournalIDs
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the Value dimension member.
<i>varalColumns</i>	Long array (ByVal). This array specifies the display columns. To create this array, use the HFMConstants type library constants listed in “Journal Column Display Constants” on page 848 .
<i>vararvFilters</i>	<p>Variant (ByVal). Enables you to filter out journals:</p> <ul style="list-style-type: none"> ● To return all journal IDs, pass an empty variable. ● To filter out journals, pass this as a Variant array that consists of 32 elements. <p>When filtering journals, each array element corresponds to a type of filter. The available filters and their indexes in this 32-element array are described in Table 64 on page 414. Define only those elements that correspond to the desired filtering criteria; if you do not want to use a filter, leave the corresponding array element empty.</p> <p>Note: In this release only the first six elements and the fourteenth and fifteenth elements of this array are supported; the remaining elements are reserved for future use.</p>
<i>var2DalColumnsSort</i>	<p>Long array (ByVal). Specifies the columns to be sorted by and whether the sort order is ascending or descending. This is a two-dimensional array, where the first dimension specifies the sort columns and the second dimension specifies the sort order. Apply the following rules when defining this array:</p> <ul style="list-style-type: none"> ● The first dimension takes the HFMConstants type library constants listed in “Journal Column Display Constants” on page 848. ● The second dimension takes 0 to sort in ascending order, 1 in descending order.

Argument**Description**

For example, suppose you want to sort by status and then by label, with both sorts in ascending order. This array would be defined as follows:

```
(0, 0) = COLUMN_JOURNALSTATUS
(0, 1) = 0
(1, 0) = COLUMN_JOURNALLABEL
(1, 1) = 0
```

pvararJournalIDs Variant array. Returns the ID numbers of the journals that match the criteria specified in the arguments. The array is returned as a Long subtype.

Journal Filter Array Elements

The following table lists the index numbers, corresponding display columns, and descriptions of the supported elements in the array passed to the *vararvFilters* argument.

Note: The array passed to the *vararvFilters* argument is a 32-element array. Indexes for this array that are not listed in the following table are reserved for future use.

Table 64 Journal Filter Array Elements

Index	Display Column	Description
0	Label	A String containing the desired label. You can use the percentage sign (%) as a wildcard character.
1	Status	An array of Longs containing the journal statuses to be applied. These statuses are represented by the HFMCconstants type library constants listed in “Journal Status Constants” on page 851 .
2	Type	An array of Longs containing the journal types to be applied. These journal types are represented by the HFMCconstants type library constants listed in “Journal Type Constants” on page 851 .
3	Balance Type	An array of Longs containing the balance types to be applied. These balance types are represented by the HFMCconstants type library constants listed in “Balance Type Constants” on page 848 .
4	Group	A String containing the name of the desired journal group. You can use the percentage sign (%) as a wildcard character.

Index	Display Column	Description
5	Description	A String containing the name of the desired description. You can use the percentage sign (%) as a wildcard character.
14	Entity	The member ID (Long) of the entity by which to filter.
15	Parent	The member ID (Long) of the parent entity by which to filter.

Example

GetJournalQueryDefinitionIDs is used in the [Example](#) for GetJournalDisplayData.

GetTemplate

Returns a variety of information for a journal template.

Tip: Use GetTemplate to create a journal from a template; GetTemplate's return values can be passed to SaveJournal.

The first argument specifies the template's ID. The next several arguments return template header information, and the remaining arguments return the template's line item data in arrays. The arrays have a one-to-one correspondence; for example, the second elements in the *pvaradAmount* and *pvararlAccount* arguments' arrays return the amount and account for the template's second line item.

The amounts for the template's line items are returned in an array with a Double subtype. To return line item amounts in an array with a String subtype, use GetTextTemplate. GetTextTemplate is almost identical to GetTemplate; the only difference is the subtype of the line item amount array. For more information, see ["GetTextTemplate" on page 422](#).

Syntax

```
<IHsvJournalsEx>.GetTemplate lTemplateID, pnType, pnAttribute, pbstrLabel,
pbstrDescription, pbstrGroup, plSingleEntity, plSingleParent,
pnTemplateType, plValueID, pvararlEntryID, pvararnDebitCreditUnit,
pvaradAmount, pvararbstrDescription, pvararlEntity, pvararlParent,
pvararlAccount, pvararlICP, pvararlCustom1, pvararlCustom2,
pvararlCustom3, pvararlCustom4
```

Argument

Description

lTemplateID

Long (ByVal). The ID of the template. Pass the template label to GetJournalTemplateItemID to get this ID; for more information, see ["GetJournalTemplateItemID" on page 396](#).

Argument	Description
<i>pnType</i>	Integer. Returns the journal type for the template. The valid return values are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 851 .
<i>pnAttribute</i>	Integer. Returns a value that indicates whether journals created from the template must be balanced. The valid return values are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 848 .
<i>pbstrLabel</i>	String. Returns the label of the template.
<i>pbstrDescription</i>	String. Returns the description of the template.
<i>pbstrGroup</i>	String. Returns the journal group of the template.
<i>plSecurityClass</i>	Long. Returns the ID of the security class for the template. Tip: To get the ID from a security class name, use <code>HsvSecurityAccess.GetSecurityClassLabel</code> .
<i>plSingleEntity</i>	Long. For single entity templates, this argument returns the ID of the template’s base entity. If the template is a multi-entity template, this argument returns -1.
<i>plSingleParent</i>	Long. For single entity templates, this argument returns the ID of the template’s parent entity. If the template is a multi-entity template, this argument returns -1.
<i>pnTemplateType</i>	Integer. Returns a value that indicates the template type. The valid return values are represented by the HFMConstants type library constants listed in “Template Type Constants” on page 852 .
<i>plValueID</i>	Long. For recurring templates, this argument returns the member ID of the template’s Value dimension member. Since standard templates are not assigned a Value dimension member, this argument returns -1 when <code>GetTemplate</code> is called for a standard template.
<i>pvararEntryID</i>	Variant array. <i>For internal use.</i>
<i>pvararnDebitCreditUnit</i>	Variant array. Returns a value that indicates whether the line items are debits or credits. The valid return values are represented by the HFMConstants type library constants listed in “Debit/Credit Constants” on page 849 . The array is returned as an Integer subtype.
<i>pvaradAmount</i>	Variant array. Returns the amounts for the template’s line items. The array is returned as a Double subtype.
<i>pvararbstDescription</i>	Variant array. Returns the descriptions for the template’s line items. The array is returned as a String subtype.
<i>pvararEntity</i>	Variant array. Returns the member IDs for the base entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>plSingleEntity</i> argument. The array is returned as a Long subtype.
<i>pvararParent</i>	Variant array. Returns the member IDs for the parent entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>plSingleParent</i> argument. The array is returned as a Long subtype.

Argument	Description
<i>pvararAccount</i>	Variant array. Returns the member IDs of the line items' accounts. The array is returned as a Long subtype.
<i>pvararICP</i>	Variant array. Returns the member IDs of the line items' Intercompany Partner dimension members. The array is returned as a Long subtype.
<i>pvararCustom1</i>	Variant array. Returns the member IDs of the line items' Custom 1 dimension members. The array is returned as a Long subtype.
<i>pvararCustom2</i>	Variant array. Returns the member IDs of the line items' Custom 2 dimension members. The array is returned as a Long subtype.
<i>pvararCustom3</i>	Variant array. Returns the member IDs of the line items' Custom 3 dimension members. The array is returned as a Long subtype.
<i>pvararCustom4</i>	Variant array. Returns the member IDs of the line items' Custom 4 dimension members. The array is returned as a Long subtype.

Example

GetTemplate is used in the [Example](#) for SaveJournal . This example shows how to pass GetTemplate's return values to SaveJournal .

GetTemplateDisplayData

Returns various types of information for templates. The types of information returned correspond to the display columns in the Columns tab of the Filters And Sorting dialog box. The template information is returned as a multidimensional array, with one array dimension for each display column that you specify in the *varalColumns* argument.

Syntax

```
<IHsvJournalsEx>.GetTemplateDisplayData varalColumns, var1TemplateIDs,
pvar2DvarData
```

Argument	Description
<i>varalColumns</i>	Long array (ByVal). Specifies the display columns for which template information will be returned. To create this array, use the HFMCconstants type library constants listed in "Template Column Display Constants" on page 851 .
<i>var1TemplateIDs</i>	Long array (ByVal). The ID numbers of the templates for which you want to return information.
<i>pvar2DvarData</i>	Variant array. Returns information for the templates and columns that match the criteria specified in the arguments. This is a multidimensional array, with one dimension for each column specified in the <i>varalColumns</i> argument; in each dimension, there is one element per template. For details on the values returned in the array elements, see "Template Column Return Values" on page 394 .

Example

The following example returns the labels, descriptions, and groups of templates that have labels beginning with “jStd” and that have balance types of either Balanced or Balanced By Entity. The IDs of these templates are returned by `GetTemplateQueryDefinitionIDs`; note that the templates are sorted by label in ascending order. The template IDs are then passed to `GetTemplateDisplayData`, and the template information returned is printed to Visual Basic’s Immediate window.

```
Dim laCols(2) As Long, vaDisplayData, laSort(0, 1) As Long
Dim vaFilter(31), vaTemplIDs
laCols(0) = COLUMN_TEMPLATELABEL
laCols(1) = COLUMN_TEMPLATEDESCRIPTION
laCols(2) = COLUMN_TEMPLATEGROUP
laSort(0, 0) = COLUMN_TEMPLATELABEL
laSort(0, 1) = 0
vaFilter(0) = "jStd%"
vaFilter(1) = Array(JBTF_BALANCED, JBTF_BALANCED_BY_ENTITY)
m_cIHsvJournalEx.GetTemplateQueryDefinitionIDs laCols, _
vaFilter, laSort, vaTemplIDs
m_cIHsvJournalEx.GetTemplateDisplayData laCols, vaTemplIDs, _
vaDisplayData
For i = LBound(vaTemplIDs) To UBound(vaTemplIDs)
    Debug.Print "Label: " & vaDisplayData(i, 0)
    Debug.Print "Description: " & vaDisplayData(i, 1)
    Debug.Print "Group: " & vaDisplayData(i, 2)
Next i
```

GetTemplateLabelsForIDs

Returns the labels of the journal templates for a given set of journal template IDs.

Syntax

```
<IHsvJournalsEx>.GetTemplateLabelsForIDs varalIDs, pvarabstrLabels
```

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

<i>varalIDs</i>	Long array (ByVal). The journal template IDs.
-----------------	---

<i>pvarabstrLabels</i>	Variant array. Returns the labels of the templates. This array has a one-to-one correspondence with the array of IDs passed to the <i>varalIDs</i> argument.
------------------------	--

GetTemplateQueryDefinitionIDs

Returns IDs of one or more templates. You can return IDs of all templates or return only those IDs that meet the filtering criteria you specify.

Syntax

```
<IHsvJournalsEx>.GetTemplateQueryDefinitionIDs varalColumns,
vararvFilters, var2DalColumnsSort, pvarar1TemplateIDs
```

Argument	Description
<i>varalColumns</i>	Long array (ByVal). This array specifies the display columns. To create this array, use the HFMConstants type library constants listed in “Template Column Display Constants” on page 851 .
<i>vararvFilters</i>	<p>Variant (ByVal). Enables you to filter out templates:</p> <ul style="list-style-type: none"> ● To return all template IDs, pass an empty variable. ● To filter out templates, pass this as a Variant array that consists of 32 elements. <p>When filtering templates, each array element corresponds to a type of filter. The available filters and their indexes in this 32-element array are described in Table 65 on page 419. Define only those elements that correspond to the desired filtering criteria; if you do not want to use a filter, leave the corresponding array element empty.</p> <p>Note: In this release only the first four elements and the sixth through eighth elements of this array are supported; the remaining elements are reserved for future use.</p>
<i>var2DalColumnsSort</i>	<p>Long array (ByVal). Specifies the columns on which to sort and whether the sort order is ascending or descending. This is a two-dimensional array, where the first dimension specifies the sort columns and the second dimension specifies the sort order. Apply the following rules when defining this array:</p> <ul style="list-style-type: none"> ● The first dimension takes the HFMConstants type library constants listed in “Template Column Display Constants” on page 851. ● The second dimension takes 0 to sort in ascending order, 1 in descending order. <p>For example, suppose you want to sort by label and then by group, with both sorts in ascending order. This array would be defined as follows:</p> <pre>(0, 0) = COLUMN_TEMPLATELABEL (0, 1) = 0 (1, 0) = COLUMN_TEMPLATEGROUP (1, 1) = 0</pre>
<i>pvararvTemplateIDs</i>	Variant array. Returns the ID numbers of the templates that match the criteria specified in the arguments. The array is returned as a Long subtype.

Template Filter Array Elements

The following table lists the index numbers, corresponding display columns, and descriptions of the supported elements in the array passed to the *vararvFilters* argument.

Note: The array passed to the *vararvFilters* argument is a 32-element array. Indexes for this array that are not listed in the following table are reserved for future use.

Table 65 Template Filter Array Elements

Index	Display Column	Description
0	Label	A String containing the desired label. You can use the percentage sign (%) as a wildcard character.

Index	Display Column	Description
1	Balance Type	An array of Longs containing the balance types to be applied. These balance types are represented by the HFMCconstants type library constants listed in “Balance Type Constants” on page 848 .
2	Group	A String containing the name of the desired journal group. You can use the percentage sign (%) as a wildcard character.
3	Description	A String containing the name of the desired description. You can use the percentage sign (%) as a wildcard character.
5	Type	An array of Longs containing the template types to be applied. These journal types are represented by the HFMCconstants type library constants listed in “Template Type Constants” on page 852 .
6	Entity	The member ID (Long) of the entity by which to filter.
7	Parent	The member ID (Long) of the parent entity by which to filter.

Example

GetTemplateQueryDefinitionIDs is used in the [Example](#) for GetTemplateDisplayData.

GetTextJournal

Returns a variety of information for a journal.

Caution! To successfully call `GetTextJournal`, the user must be assigned to the Journals Administrator or Read Journals role, and must have Read or All access to the journal’s security class. For posted journals, the user must also have Read or All access for the entities in the journal’s line items.

The first three arguments specify the scenario, year, and journal IDs. The next several arguments return journal header information, and the remaining arguments return line item data in arrays. The arrays have a one-to-one correspondence; for example, the second elements in the *pvarabstrAmount* and *pvararAccount* arguments’ arrays return the amount and account for the journal’s second line item.

The amounts for the journal's line items are returned in an array with a String subtype. To return line item amounts in an array with a Double subtype, use `GetJournal`. `GetJournal` is almost identical to `GetTextJournal`; the only difference is the subtype of the line item amount array. For more information, see [“GetJournal” on page 406](#).

Syntax

```
<IHsvJournalsEx>.GetTextJournal lScenario, lYear, lJournalID, plPeriod,
plValue, pnType, pnStatus, pnAttribute, pbstrLabel, pbstrDescription,
pbstrGroup, plSingleEntity, plSingleParent, plSecurityClass,
pvarar1EntryID, pvararnDebitCreditUnit, pvarabstrAmount,
pvararbstrDescription, pvarar1Entity, pvarar1Parent, pvarar1Account,
pvarar1ICP, pvarar1Custom1, pvarar1Custom2, pvarar1Custom3, pvarar1Custom4
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lJournalID</i>	Long (ByVal). The ID number of the journal. Get this ID with <code>HsvJournals.GetItemID</code> ; for details, see “GetItemID” on page 396 .
<i>plPeriod</i>	Long. Returns the member ID of the journal's period.
<i>plValue</i>	Long. Returns the member ID of the Value dimension member for the journal's currency.
<i>pnType</i>	Integer. Returns a numeric constant that identifies the journal's type. The valid return values are represented by the <code>HFMConstants</code> type library constants listed in “Journal Type Constants” on page 851 .
<i>pnStatus</i>	Integer. Returns the journal's status. The valid return values are represented by the <code>HFMConstants</code> type library constants listed in “Journal Status Constants” on page 851 .
<i>pnAttribute</i>	Integer. Returns a value that indicates whether the journal must be balanced. The valid return values are represented by the <code>HFMConstants</code> type library constants listed in “Balance Type Constants” on page 848 .
<i>pbstrLabel</i>	String. Returns the label of the journal.
<i>pbstrDescription</i>	String. Returns the description of the journal.
<i>pbstrGroup</i>	String. Returns the description of the journal group to which the journal has been assigned.
<i>plSingleEntity</i>	Long. For single entity journals, this argument returns the member ID of the journal's base entity. If the journal is a multi-entity journal, this argument returns -1.
<i>plSingleParent</i>	Long. For single entity journals, this argument returns the member ID of the journal's parent entity. If the journal is a multi-entity journal, this argument returns -1.
<i>plSecurityClass</i>	Long. Returns the ID of the journal's security class. Tip: To get the label of the security class, pass this ID to <code>HsvSecurityAccess.GetSecurityClassLabel</code> . For more information, see “GetSecurityClassLabel” on page 473 .
<i>pvarar1EntryID</i>	Variant array. <i>For internal use.</i>

Argument	Description
<i>pvararrDebitCreditUnit</i>	Variant array. Indicates whether the line items are debits or credits. Valid return values are represented by the <code>HFMConstants</code> type library constants listed in “Debit/Credit Constants” on page 849 . The array is returned as an Integer subtype.
<i>pvarabstrAmount</i>	Variant array. Returns the amounts for the journal’s line items. The array is returned as a String subtype. Tip: Use <code>GetVariance</code> to return the total debit and credit amounts and the difference between these amounts. For details, see “GetVariance” on page 425 .
<i>pvarabstrDescription</i>	Variant array. Returns the descriptions for the journal’s line items. The array is returned as a String subtype.
<i>pvararrEntity</i>	Variant array. Returns the member IDs for the base entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>pISingleEntity</i> argument. The array is returned as a Long subtype.
<i>pvararrParent</i>	Variant array. Returns the member IDs for the parent entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>pISingleParent</i> argument. The array is returned as a Long subtype.
<i>pvararrAccount</i>	Variant array. Returns the member IDs of the line items’ accounts. The array is returned as a Long subtype.
<i>pvararrICP</i>	Variant array. Returns the member IDs of the line items’ Intercompany Partner dimension members. The array is returned as a Long subtype.
<i>pvararrCustom1</i>	Variant array. Returns the member IDs of the line items’ Custom 1 dimension members. The array is returned as a Long subtype.
<i>pvararrCustom2</i>	Variant array. Returns the member IDs of the line items’ Custom 2 dimension members. The array is returned as a Long subtype.
<i>pvararrCustom3</i>	Variant array. Returns the member IDs of the line items’ Custom 3 dimension members. The array is returned as a Long subtype.
<i>pvararrCustom4</i>	Variant array. Returns the member IDs of the line items’ Custom 4 dimension members. The array is returned as a Long subtype.

Example

See the [Example](#) for `GetJournal`. The only difference between the two methods is the data subtype of the array that returns the line item amounts. All other arguments are identical.

GetTextTemplate

Returns a variety of information for a journal template.

Tip: Use `GetTextTemplate` to create a journal from a template; `GetTextTemplate`'s return values can be passed to `SaveTextJournal`.

The first argument specifies the template's ID. The next several arguments return template header information, and the remaining arguments return the template's line item data in arrays. The arrays have a one-to-one correspondence; for example, the second elements in the `pvarabstrAmount` and `pvararlAccount` arguments' arrays return the amount and account for the template's second line item.

The amounts for the template's line items are returned in an array with a `String` subtype. To return line item amounts in an array with a `Double` subtype, use `GetTemplate`. `GetTemplate` is almost identical to `GetTextTemplate`; the only difference is the subtype of the line item amount array. For more information, see [“GetTemplate” on page 415](#).

Syntax

```
<IHsvJournalsEx>.GetTextTemplate lTemplateID, pnType, pnAttribute,
pbstrLabel, pbstrDescription, pbstrGroup, plSingleEntity, plSingleParent,
pnTemplateType, plValueID, pvararlEntryID, pvararnDebitCreditUnit,
pvarabstrAmount, pvararbstrDescription, pvararlEntity, pvararlParent,
pvararlAccount, pvararlICP, pvararlCustom1, pvararlCustom2,
pvararlCustom3, pvararlCustom4
```

Argument	Description
<code>lTemplateID</code>	Long (ByVal). The ID of the template. Pass the template label to <code>GetJournalTemplateItemID</code> to get this ID; for more information, see “GetJournalTemplateItemID” on page 396 .
<code>pnType</code>	Integer. Returns the journal type for the template. The valid return values are represented by the <code>HFMConstants</code> type library constants listed in “Journal Type Constants” on page 851 .
<code>pnAttribute</code>	Integer. Returns a value that indicates whether journals created from the template must be balanced. The valid return values are represented by the <code>HFMConstants</code> type library constants listed in “Balance Type Constants” on page 848 .
<code>pbstrLabel</code>	String. Returns the label of the template.
<code>pbstrDescription</code>	String. Returns the description of the template.
<code>pbstrGroup</code>	String. Returns the journal group of the template.
<code>plSecurityClass</code>	Long. Returns the ID of the security class for the template. Tip: To get the ID from a security class name, use <code>HsvSecurityAccess.GetSecurityClassLabel</code> .
<code>plSingleEntity</code>	Long. For single entity templates, this argument returns the member ID of the template's base entity. If the journal is a multi-entity template, this argument returns -1.
<code>plSingleParent</code>	Long. For single entity templates, this argument returns the member ID of the template's parent entity. If the journal is a multi-entity template, this argument returns -1.
<code>pnTemplateType</code>	Integer. Returns the template type. The template types are represented by the <code>HFMConstants</code> type library constants listed in “Template Type Constants” on page 852 .

Argument	Description
<i>pValueID</i>	<p>Long. For recurring templates, this argument returns the member ID of the template's Value dimension member.</p> <p>Since standard templates are not assigned a Value dimension member, this argument returns -1 when <code>GetTextTemplate</code> is called for a standard template.</p>
<i>pvararEntryID</i>	Variant array. <i>For internal use.</i>
<i>pvararDebitCreditUnit</i>	<p>Variant array. Indicates whether the line items are debits or credits. Valid return values are represented by the <code>HFMConstants</code> type library constants listed in "Debit/Credit Constants" on page 849.</p> <p>The array is returned as an Integer subtype.</p>
<i>pvarabstrAmount</i>	<p>Variant array. Returns the amounts for the template's line items.</p> <p>The array is returned as a Double subtype.</p>
<i>pvarabstrDescription</i>	<p>Variant array. Returns the descriptions for the template's line items.</p> <p>The array is returned as a String subtype.</p>
<i>pvararEntity</i>	<p>Variant array. Returns the member IDs for the base entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>pISingleEntity</i> argument.</p> <p>The array is returned as a Long subtype.</p>
<i>pvararParent</i>	<p>Variant array. Returns the member IDs for the parent entities of the line items. For single-entity items, these IDs will be the same as the member ID returned in the <i>pISingleParent</i> argument.</p> <p>The array is returned as a Long subtype.</p>
<i>pvararAccount</i>	<p>Variant array. Returns the member IDs of the line items' accounts.</p> <p>The array is returned as a Long subtype.</p>
<i>pvararICP</i>	<p>Variant array. Returns the member IDs of the line items' Intercompany Partner dimension members.</p> <p>The array is returned as a Long subtype.</p>
<i>pvararCustom1</i>	<p>Variant array. Returns the member IDs of the line items' Custom 1 dimension members.</p> <p>The array is returned as a Long subtype.</p>
<i>pvararCustom2</i>	<p>Variant array. Returns the member IDs of the line items' Custom 2 dimension members.</p> <p>The array is returned as a Long subtype.</p>
<i>pvararCustom3</i>	<p>Variant array. Returns the member IDs of the line items' Custom 3 dimension members.</p> <p>The array is returned as a Long subtype.</p>
<i>pvararCustom4</i>	<p>Variant array. Returns the member IDs of the line items' Custom 4 dimension members.</p> <p>The array is returned as a Long subtype.</p>

Example

`GetTextTemplate` is used in the [Example](#) for `SaveTextJournal`. This example shows how to pass `GetTextTemplate`'s return values to `SaveTextJournal`.

GetVariance

Returns a journal's total debit and credit amounts and the difference between these amounts.

You do not pass a journal ID to `GetVariance`. Instead, you pass journal information returned by `GetJournal` or `GetTextJournal` as shown in the example.

Syntax

```
<IHsvJournalsEx>.GetVariance lSingleParent, lSingleEntity, varalParent,
varalEntity, varalAccount, lValue, varanDebitCreditUnit, varabstrAmount,
pbstrTotalDebits, pbstrTotalCredits, pbstrAmount
```

Argument	Description
<i>lSingleParent</i>	Long (ByVal). Pass the value returned by the <i>plSingleParent</i> argument of <code>GetJournal</code> or <code>GetTextJournal</code> .
<i>lSingleEntity</i>	Long (ByVal). Pass the value returned by the <i>plSingleEntity</i> argument of <code>GetJournal</code> or <code>GetTextJournal</code> .
<i>varalParent</i>	Variant (ByVal). Pass the array of parent entity member IDs returned by the <i>pvaralParent</i> argument of <code>GetJournal</code> or <code>GetTextJournal</code> .
<i>varalEntity</i>	Variant (ByVal). Pass the array of entity member IDs returned by the <i>pvaralEntity</i> argument of <code>GetJournal</code> or <code>GetTextJournal</code> .
<i>varalAccount</i>	Variant (ByVal). Pass the array of account member IDs returned by the <i>pvaralAccount</i> argument of <code>GetJournal</code> or <code>GetTextJournal</code> .
<i>lValue</i>	Long (ByVal). Pass the Value dimension member ID returned by the <i>plValue</i> argument of <code>GetJournal</code> or <code>GetTextJournal</code> .
<i>varanDebitCreditUnit</i>	Variant (ByVal). Pass the array flagging line items as debits or credits that is returned by the <i>pvaranDebitCreditUnit</i> argument of <code>GetJournal</code> or <code>GetTextJournal</code> .
<i>varabstrAmount</i>	Variant (ByVal). Pass the array of line item amounts that is returned by the <i>pvarabstrAmount</i> argument of <code>GetTextJournal</code> or the <i>pvaradAmount</i> argument of <code>GetJournal</code> . Caution! The <code>GetJournal</code> array would need to be converted to a String subtype.
<i>pbstrTotalDebits</i>	String. Returns the total debit amount.
<i>pbstrTotalCredits</i>	String. Returns the total credit amount.
<i>pbstrAmount</i>	String. Returns the difference between the <i>pbstrTotalDebits</i> and <i>pbstrTotalCredits</i> arguments' amounts.

Example

The following example uses `GetVariance` to place the difference between a journal's debits and credits in a text box control named `txtVariance`. Note how `GetVariance`'s `ByVal` arguments are passed with information that is returned by `GetTextJournal`.

```
m_cIHsvJournalEx.GetTextJournal lScen, lYear, lJnlID, lPer, _
lVal, iType, iStatus, iAttr, sLabel, sDesc, sGroup, _
lSingEnt, lSingPar, lSec, vaIds, vaDebCred, vaAmounts, _
```

```

vaDescs, vaEnts, vaPars, vaAccts, vaICPs, vaCust1, _
vaCust2, vaCust3, vaCust4
m_cIHsvJournalEx.GetVariance lSingEnt, lSingPar, vaPars, _
vaEnts, vaAccts, lVal, vaDebCred, vaAmounts, sTotDebs, _
sTotCreds, sVariance
txtVariance.Text = sVariance

```

PostJournals

Posts one or more journals. The *vararJournalIDs* argument takes the IDs of the journals that are being posted.

Caution! The user must be assigned to the Journals Administrator or Post Journals role, otherwise error number 40D (hexadecimal) will occur.

Syntax

```
<IHsvJournalsEx>.PostJournals lScenario, lYear, vararJournalIDs, pvararnRc
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>vararJournalIDs</i>	Long array (ByVal). The IDs of the journals to be posted. Get these IDs with the HsvJournals object's <code>GetItemID</code> method; for more information, see “GetItemID” on page 396 . Caution! To post a journal, the connected user must have All access to the journal's security class and to the entities in the journal's line items.
<i>pvararnRc</i>	Variant array. Returns codes that identify the success or failure of the journal postings. The elements in this array have a one-to-one correspondence with the journal IDs passed in the <i>vararJournalIDs</i> argument. Successfully posted journals return 0. The following list identifies some of the likely error codes in hexadecimal form: <ul style="list-style-type: none"> ● 40F ● 3F9 ● 418 ● 40D ● 411 ● 408 ● 409 <p>The array is returned as an Integer subtype.</p>

Example

This example posts two journals. The calls to the user-defined `GetMemberID` function get the example's dimension member IDs; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`. The `laJnlIDs` array variable is then populated with the journal

IDs by `HsvJournals.GetItemID`. The member IDs for the Scenario and Year dimension are passed along with this array to `PostJournals`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim laJnlIDs(1) As Long, vaRetVal
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
laJnlIDs(0) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
    "Jnl1990")
laJnlIDs(1) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
    "Jnl1994")
m_cIHsvJournalEx.PostJournals lScen, lYear, laJnlIDs, vaRetVal
```

RejectJournals

Rejects one or more journals. The *vararJnlIDs* argument takes the IDs of the journals that are being rejected.

Caution! The user must be assigned to the Journals Administrator or Approve Journals role, otherwise error number 40D (hexadecimal) will occur.

Syntax

```
<IHsvJournalsEx>.RejectJournals lScenario, lYear, vararJnlIDs,
pvararnRc
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>vararJnlIDs</i>	Long array (ByVal). The IDs of the journals to be rejected. Get these IDs with the <code>HsvJournals</code> object's <code>GetItemID</code> method; for more information, see “GetItemID” on page 396 .
<i>pvararnRc</i>	Variant array. Returns codes that identify the success or failure of the journal rejections. The elements in this array have a one-to-one correspondence with the journal IDs passed in the <i>vararJnlIDs</i> argument. Successfully rejected journals return 0. The array is returned as an Integer subtype.

Example

This example rejects two journals. The calls to the user-defined `GetMemberID` function get the example's dimension member IDs; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`. The `laJnlIDs` array variable is then populated with the journal IDs by `HsvJournals.GetItemID`. The member IDs for the Scenario and Year dimension are passed along with this array to `RejectJournals`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim laJnlIDs(1) As Long, vaRetVal
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
```

```
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
laJnlIDs(0) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl810")
laJnlIDs(1) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl811")
m_cIHsvJournalEx.RejectJournals lScen, lYear, laJnlIDs, vaRetVal
```

RemoveAllJournalGroups

Removes all journal groups from an application.

Syntax

```
<IHsvJournalsEx>.RemoveAllJournalGroups
```

RemoveJournalGroup

Removes the specified journal group.

Syntax

```
<IHsvJournalsEx>.RemoveJournalGroup bstrGroup
```

Argument Description

bstrGroup String (ByVal). The name of the journal group to remove.

SaveJournal

Creates a new journal, or saves changes to an existing journal that has a Working or Submitted status.

Caution! To successfully call `SaveJournal`, the connected user must be assigned to the Journals Administrator or Create Journals role, and must have All access to the journal's security class.

The arguments for journal line items consist of arrays. The arrays have a one-to-one correspondence; for example, the second elements in the *varadAmount* and *vararAccount* arguments' arrays specify the amount and account for a journal's second line item.

`SaveJournal` is almost identical to `SaveTextJournal`; the only difference is that `SaveJournal` passes line item amounts in a Double array while `SaveTextJournal` passes line item amounts in a String array. For more information, see [“SaveTextJournal” on page 434](#).

Tip: To create a journal from a template, use `GetTemplate` to get the template's information, then pass the applicable `GetTemplate` return values to `SaveJournal`. For information on `GetTemplate`, see [“GetTemplate” on page 415](#).

Syntax

```
<IHsvJournalsEx>.SaveJournal lScenario, lYear, lPeriod, lValue, nType,
nStatus, nAttribute, bstrLabel, bstrDescription, bstrGroup, lSingleEntity,
lSingleParent, lSecurityClass, vararnDebitCreditUnit, varadAmount,
vararbstrDescription, vararlEntity, vararlParent, vararlAccount,
vararlICP, vararlCustom1, vararlCustom2, vararlCustom3, vararlCustom4,
plJournalID
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the Value dimension member for the journal's currency.
<i>nType</i>	Integer (ByVal). Specifies the journal's type. Pass one of the HFMConstants type library constants listed in “Journal Type Constants” on page 851 . Note: You cannot use this method to save journals of type <code>JTF_AUTOREVERSAL</code> , as these are system-generated journals. Attempting to save a journal as this type will throw error number 8004041A (hexadecimal).
<i>nStatus</i>	Integer (ByVal). Specifies the journal's status. Pass one of the HFMConstants type library constants listed in “Journal Status Constants” on page 851 .
<i>nAttribute</i>	Integer (ByVal). Specifies whether the journal must be balanced. Pass one of the HFMConstants type library constants listed in “Balance Type Constants” on page 848 .
<i>bstrLabel</i>	String (ByVal). The label of the journal.
<i>bstrDescription</i>	String (ByVal). The description of the journal. You can pass a blank string if there is no description.
<i>bstrGroup</i>	String (ByVal). The journal group to which the journal is assigned. You can pass a blank string if the journal is not being assigned to a journal group.
<i>lSingleEntity</i>	Long (ByVal). For single entity journals, specify the member ID of the journal's base entity. For multi-entity journals, pass a value of -1.
<i>lSingleParent</i>	Long (ByVal). For single entity journals, specify the member ID of the journal's parent entity. For multi-entity journals, pass a value of -1.
<i>lSecurityClass</i>	Long (ByVal). The ID of the journal's security class. To get this ID, pass the security class name to <code>HsvSecurityAccess.GetSecurityClassID</code> . For more information, see “GetSecurityClassID” on page 472 .
<i>vararnDebitCreditUnit</i>	Integer array (ByVal). Specifies whether the line items are debits or credits. Pass one of the HFMConstants type library constants listed in “Debit/Credit Constants” on page 849 .
<i>varadAmount</i>	Double array (ByVal). The amounts of the line items.

Argument	Description
<i>vararbstrDescription</i>	String array (ByVal). The descriptions of the line items.
<i>vararEntity</i>	Long array (ByVal). The member IDs for the base entities of the line items. For single-entity items, set the IDs in this array to the same member ID that is in the <i>ISingleEntity</i> argument.
<i>vararParent</i>	Long array (ByVal). The member IDs for the parent entities of the line items. For single-entity items, set the IDs in this array to the same member ID that is in the <i>ISingleParent</i> argument.
<i>vararAccount</i>	Long array (ByVal). The member IDs of the accounts for the line items.
<i>vararICP</i>	Long array (ByVal). The member IDs of the line items' Intercompany Partner dimension members.
<i>vararCustom1</i>	Long array (ByVal). The member IDs of the line items' Custom 1 dimension members.
<i>vararCustom2</i>	Long array (ByVal). The member IDs of the line items' Custom 2 dimension members.
<i>vararCustom3</i>	Long array (ByVal). The member IDs of the line items' Custom 3 dimension members.
<i>vararCustom4</i>	Long array (ByVal). The member IDs of the line items' Custom 4 dimension members.
<i>pJournalID</i>	<p>Long. The purpose of this argument depends upon whether you are creating a new journal or updating an existing journal:</p> <ul style="list-style-type: none"> ● For a newly created journal, this argument returns the automatically-generated ID of the journal. ● To update an existing journal, pass the journal's ID with this argument. You can get journal IDs with <code>GetItemID</code>; for details, see "GetItemID" on page 396.

Example

This example uses `SaveJournal` in a custom function that creates journals from a template. The custom function is named `CreateJournal`, and takes the following items as arguments:

- The member IDs of the applicable Scenario, Year, Value, and Period dimension members.
- The name of the template to be used.
- The name to be assigned to the journal.
- An array of the line item amounts, and a corresponding array that flags the line items as debits or credits.

The example gets the template's information with `GetTemplate`; note how most of `GetTemplate`'s return values are passed to `SaveJournal`.

```
Function CreateJournal(lScenID As Long, lYearID _
    As Long, lPerID As Long, lValID As Long, sTemplate _
    As String, sJnlName As String, daAmount() As Double, _
    iaDebCredUnit() As Integer) As Long
Dim cHsvJournals As HsvJournals
Dim cIHsvJournalEx As IHsvJournalsEx
Dim lTempID As Long, iType As Integer, iAttr As Integer
Dim sLabel As String, sDescHead As String, sGroup As String
Dim lSecId As Long, lSingleEnt As Long, lSinglePar As Long
Dim iTempType As Integer, lTempVal As Long, vaEntryIDs
Dim vaDebCredUnit, vaAmt, vaItemDesc, vaEnt, vaPar, vaAcct
Dim vaICP, vaCust1, vaCust2, vaCust3, vaCust4, lJnlID As Long
```

```

'm_cSession is an HsvSession object reference
Set cHsvJournals = m_cSession.Journals
Set cIHsvJournalEx = m_cSession.Journals
'Get the template's ID
lTempID = cHsvJournals.GetJournalTemplateItemID(sTemplate)
'Get the template's data
cIHsvJournalEx.GetTemplate lTempID, iType, iAttr, sLabel, _
    sDescHead, sGroup, lSecId, lSingleEnt, lSinglePar, _
    iTempType, lTempVal, vaEntryIDs, vaDebCredUnit, vaAmt, _
    vaItemDesc, vaEnt, vaPar, vaAcct, vaICP, vaCust1, _
    vaCust2, vaCust3, vaCust4
'Create the journal.
cIHsvJournalEx.SaveJournal lScenID, lYearID, lPerID, lValID, _
    iType, 1, iAttr, sJnlName, "", "", lSingleEnt, lSinglePar, _
    lSecId, iaDebCredUnit, daAmount, vaItemDesc, vaEnt, _
    vaPar, vaAcct, vaICP, vaCust1, vaCust2, vaCust3, _
    vaCust4, lJnlID
'Assign the return value
CreateJournal = lJnlID
End Function

```

SaveTemplate

Creates a new journal template, or updates an existing journal template.

Caution! To successfully call `SaveTemplate`, the connected user must be assigned to the Journals Administrator role.

The arguments for template line items consist of arrays. The arrays have a one-to-one correspondence; for example, the second elements in the *varadAmount* and *vararlAccount* arguments' arrays specify the amount and account for a template's second line item.

`SaveTemplate` takes a Double array for the template's line item's amounts. To create a template by passing a String array of amounts, use `SaveTextTemplate` instead of `SaveTemplate`. The methods are almost identical; the only difference is the subtype of the line item amount array. For more information, see "[SaveTextTemplate](#)" on page 437.

Syntax

```

<IHsvJournalsEx>.SaveTemplate nType, nAttribute, bstrLabel,
bstrDescription, bstrGroup, lSingleEntity, lSingleParent, nTemplateType,
lValueID, vararnDebitCreditUnit, varadAmount, vararbstrDescription,
vararlEntity, vararlParent, vararlAccount, vararlICP, vararlCustom1,
vararlCustom2, vararlCustom3, vararlCustom4, plTemplateID

```

Argument	Description
<i>nType</i>	Integer (ByVal). Specifies the journal type of the journals that will be created from the template. Pass one of the HFMConstants type library constants listed in " Journal Type Constants " on page 851.

Argument	Description
<i>nAttribute</i>	Integer (ByVal). Specifies whether journals created from the template must be balanced. Pass one of the “Balance Type Constants” on page 848 .
<i>bstrLabel</i>	String (ByVal). The label for the template.
<i>bstrDescription</i>	String (ByVal). The description for the template. You can pass a blank string if there is no description.
<i>bstrGroup</i>	String (ByVal). The journal group to which the journals created from the template will be assigned. You can pass a blank string if the journals will not be assigned to a journal group
<i>ISecurityClass</i>	Long. The ID of the security class for the template. Tip: To get the ID from a security class name, use <code>HsvSecurityAccess.GetSecurityClassLabel</code> .
<i>ISingleEntity</i>	Long (ByVal). For single entity journal templates, specify the member ID of the journal's base entity. For multi-entity journal templates, pass a value of -1.
<i>ISingleParent</i>	Long (ByVal). For single entity journal templates, specify the member ID of the journal's parent entity. For multi-entity journal templates, pass a value of -1.
<i>nTemplateType</i>	Integer (ByVal). Determines whether the template is standard or recurring. Pass one of the HFMConstants type library constants listed in “Template Type Constants” on page 852 .
<i>IValueID</i>	Long (ByVal). For recurring templates, specify the member ID of the template's Value dimension member. Since standard templates are not assigned a Value dimension member, specify -1 when <code>SetTemplate</code> is called for a standard template.
<i>varamDebitCreditUnit</i>	Integer array (ByVal). Specifies whether the line items are debits or credits. Pass one of the HFMConstants type library constants listed in “Debit/Credit Constants” on page 849 .
<i>varadAmount</i>	Double array (ByVal). The amounts of the line items.
<i>varabstrDescription</i>	String array (ByVal). The descriptions of the line items.
<i>vararEntity</i>	Long array (ByVal). The member IDs for the base entities of the line items. For single-entity items, set the IDs in this array to the same member ID that is in the <i>ISingleEntity</i> argument.
<i>vararParent</i>	Long array (ByVal). The member IDs for the parent entities of the line items. For single-entity items, set the IDs in this array to the same member ID that is in the <i>ISingleParent</i> argument.
<i>vararAccount</i>	Long array (ByVal). The member IDs of the accounts for the line items.
<i>vararICP</i>	Long array (ByVal). The member IDs of the line items' Intercompany Partner dimension members.
<i>vararCustom1</i>	Long array (ByVal). The member IDs of the line items' Custom 1 dimension members.
<i>vararCustom2</i>	Long array (ByVal). The member IDs of the line items' Custom 2 dimension members.
<i>vararCustom3</i>	Long array (ByVal). The member IDs of the line items' Custom 3 dimension members.
<i>vararCustom4</i>	Long array (ByVal). The member IDs of the line items' Custom 4 dimension members.
<i>pITemplateID</i>	Long. The purpose of this argument depends upon whether you are creating a new template or updating an existing template:

Argument	Description
	<ul style="list-style-type: none"> For a newly created template, this argument returns the automatically-generated ID of the template. To update an existing template, pass the template's ID with this argument. You can get template IDs with <code>GetJournalTemplateItemID</code>; for details, see “GetJournalTemplateItemID” on page 396.

Example

This example defines a custom function named `NewTemplate` that creates new templates from an existing template, changing the entities while carrying over the other properties of the existing template. The existing template's name and the new template's name, description, and Entity dimension member IDs are passed as the function's arguments. The example calls `GetTemplate` to get the existing template's information.

```
Function NewTemplate(sTemplate As String, sNewLabel As String, _
    sDesc As String, lNewEntID As Long, lNewParID As Long) As Long
Dim cHsvJournals As HsvJournals, iType As Integer
Dim cIHsvJournalEx As IHsvJournalsEx, lTempID As Long
Dim sLabel As String, sDescHead As String, sGroup As String
Dim lSecID As Long, lSingleEnt As Long, lSinglePar As Long
Dim iTempType As Integer, lVal As Long, vaEntryIDs, vaDebCredUnit
Dim vaAmt, vaItemDesc, vaEnt, vaPar, vaAcct, vaICP, vaCust1
Dim vaCust2, vaCust3, vaCust4, lNewTempID As Long
Dim iAttr As Integer, laNewEnt() As Long, laNewPar() As Long
Dim lUpBounds As Long
'm_cSession is an HsvSession object reference
Set cHsvJournals = m_cSession.Journals
Set cIHsvJournalEx = m_cSession.Journals
lTempID = cHsvJournals.GetJournalTemplateItemID(sTemplate)
cIHsvJournalEx.GetTemplate lTempID, iType, iAttr, sLabel, _
    sDescHead, sGroup, lSecID, lSingleEnt, lSinglePar, _
    iTempType, lVal, vaEntryIDs, vaDebCredUnit, vaAmt, _
    vaItemDesc, vaEnt, vaPar, vaAcct, vaICP, vaCust1, _
    vaCust2, vaCust3, vaCust4
lUpBounds = UBound(vaEnt)
'Create the entity ID arrays that will be passed to SaveTemplate
ReDim laNewEnt(lUpBounds)
ReDim laNewPar(lUpBounds)
For i = LBound(vaEnt) To lUpBounds
    laNewEnt(i) = lNewEntID
    laNewPar(i) = lNewParID
Next i
cIHsvJournalEx.SaveTemplate iType, iAttr, sNewLabel, _
    sDesc, sGroup, lSecID, lNewEntID, lNewParID, _
    iTempType, lVal, vaDebCredUnit, vaAmt, vaItemDesc, _
    laNewEnt, laNewPar, vaAcct, vaICP, vaCust1, vaCust2, _
    vaCust3, vaCust4, lNewTempID
NewTemplate = lNewTempID
End Function
```

SaveTextJournal

Creates a new journal, or saves changes to an existing journal that has a Working or Submitted status.

Caution! To successfully call `SaveTextJournal`, the connected user must be assigned to the Journals Administrator or Create Journals role, and must have All access to the journal's security class.

The arguments for journal line items consist of arrays. The arrays have a one-to-one correspondence; for example, the second elements in the `varabstrAmount` and `vararAccount` arguments' arrays specify the amount and account for a journal's second line item.

`SaveTextJournal` is almost identical to `SaveJournal`; the only difference is that `SaveTextJournal` passes line item amounts in a String array while `SaveJournal` passes line item amounts in a Double array. For more information, see [“SaveJournal” on page 428](#).

Tip: To create a journal from a template, use `GetTextTemplate` to get the template's information, then pass the applicable `GetTextTemplate` return values to `SaveTextJournal`. For information on `GetTextTemplate`, see [“GetTextTemplate” on page 422](#).

Syntax

```
<IHsvJournalsEx>.SaveTextJournal lScenario, lYear, lPeriod, lValue, nType, nStatus, nAttribute, bstrLabel, bstrDescription, bstrGroup, lSingleEntity, lSingleParent, lSecurityClass, vararnDebitCreditUnit, varabstrAmount, vararbstrDescription, vararEntity, vararParent, vararAccount, vararICP, vararCustom1, vararCustom2, vararCustom3, vararCustom4, plJournalID
```

Argument	Description
<code>lScenario</code>	Long (ByVal). The member ID of the Scenario dimension member.
<code>lYear</code>	Long (ByVal). The member ID of the Year dimension member.
<code>lPeriod</code>	Long (ByVal). The member ID of the Period dimension member.
<code>lValue</code>	Long (ByVal). The member ID of the Value dimension member for the journal's currency.
<code>nType</code>	Integer (ByVal). Specifies the journal's type. Pass one of the <code>HFMConstants</code> type library constants listed in “Journal Type Constants” on page 851 . Note: You cannot use this method to save journals of type <code>JTF_AUTOREVERSAL</code> , as these are system-generated journals. Attempting to save a journal as this type will throw error number 8004041A (hexadecimal).
<code>nStatus</code>	Integer (ByVal). Specifies the journal's status. Pass one of the <code>HFMConstants</code> type library constants listed in “Journal Status Constants” on page 851 .

Argument	Description
<i>nAttribute</i>	Integer (ByVal). Specifies whether the journal must be balanced. Pass one of the “Balance Type Constants” on page 848 .
<i>bstrLabel</i>	String (ByVal). The label of the journal.
<i>bstrDescription</i>	String (ByVal). The description of the journal. You can pass a blank string if there is no description.
<i>bstrGroup</i>	String (ByVal). The journal group to which the journal is assigned. You can pass a blank string if the journal is not being assigned to a journal group.
<i>ISingleEntity</i>	Long (ByVal). For single entity journals, specify the member ID of the journal’s base entity. For multi-entity journals, pass a value of -1.
<i>ISingleParent</i>	Long (ByVal). For single entity journals, specify the member ID of the journal’s parent entity. For multi-entity journals, pass a value of -1.
<i>ISecurityClass</i>	Long (ByVal). The ID of the journal’s security class. To get this ID, pass the security class name to <code>HsvSecurityAccess.GetSecurityClassID</code> . For more information, see “GetSecurityClassID” on page 472 .
<i>vararrDebitCreditUnit</i>	Integer array (ByVal). Specifies whether the line items are debits or credits. Pass one of the <code>HFMConstants</code> type library constants listed in “Debit/Credit Constants” on page 849 .
<i>varabstrAmount</i>	String array (ByVal). The amounts of the line items.
<i>varabstrDescription</i>	String array (ByVal). The descriptions of the line items.
<i>vararrEntity</i>	Long array (ByVal). The member IDs for the base entities of the line items. For single-entity items, set the IDs in this array to the same member ID that is in the <i>ISingleEntity</i> argument.
<i>vararrParent</i>	Long array (ByVal). The member IDs for the parent entities of the line items. For single-entity items, set the IDs in this array to the same member ID that is in the <i>ISingleParent</i> argument.
<i>vararrAccount</i>	Long array (ByVal). The member IDs of the accounts for the line items.
<i>vararrICP</i>	Long array (ByVal). The member IDs of the line items’ Intercompany Partner dimension members.
<i>vararrCustom1</i>	Long array (ByVal). The member IDs of the line items’ Custom 1 dimension members.
<i>vararrCustom2</i>	Long array (ByVal). The member IDs of the line items’ Custom 2 dimension members.
<i>vararrCustom3</i>	Long array (ByVal). The member IDs of the line items’ Custom 3 dimension members.
<i>vararrCustom4</i>	Long array (ByVal). The member IDs of the line items’ Custom 4 dimension members.
<i>pJournalID</i>	<p>Long. The purpose of this argument depends upon whether you are creating a new journal or updating an existing journal:</p> <ul style="list-style-type: none"> ● For a newly created journal, this argument returns the automatically-generated ID of the journal. ● To update an existing journal, pass the journal’s ID with this argument. You can get journal IDs with <code>GetItemID</code>; for details, see “GetItemID” on page 396.

Example

This example uses `SaveTextJournal` in a custom function that creates journals from a template. The custom function is named `CreateTextJournal`, and takes the following items as arguments:

- The member IDs of the applicable Scenario, Year, Period, and Value dimension members.
- The name of the template to be used.
- The name to be assigned to the journal.
- An array of the line item amounts, and a corresponding array that flags the line items as debits or credits.

The example gets the template's information with `GetTextTemplate`; note how most of `GetTextTemplate`'s return values are passed to `SaveTextJournal`. (More details on the example are provided in the comments.)

```
Function CreateTextJournal(lScenID As Long, lYearID _
    As Long, lPerID As Long, lValID As Long, _
    sTemplate As String, sJnlName As String, _
    saAmount() As String, iaDebCredUnit() As Integer) As Long
Dim cHsvJournals As HsvJournals
Dim cIHsvJournalEx As IHsvJournalsEx
'Variables for GetTextTemplate - many are also passed to
'SaveTextJournal.
Dim lTempID As Long, iType As Integer, iAttr As Integer
Dim sLabel As String, sDescHead As String, sGroup As String
Dim lSecID As Long, lSingleEnt As Long, lSinglePar As Long
Dim iTempType As Integer, lTempVal As Long, vaEntryIDs
Dim vaDebCredUnit, vaAmt, vaItemDesc, vaEnt, vaPar, vaAcct
Dim vaICP, vaCust1, vaCust2, vaCust3, vaCust4
'Variables for SaveTextJournal
Dim lSecClass As Long, lJnlID As Long
'm_cSession is an HsvSession object reference
Set cHsvJournals = m_cSession.Journals
'Set IHsvJournalsEx object reference
Set cIHsvJournalEx = m_cSession.Journals
'Get the template's ID
lTempID = cHsvJournals.GetJournalTemplateItemID(sTemplate)
'Get the template's data
cIHsvJournalEx.GetTextTemplate lTempID, iType, iAttr, sLabel, _
    sDescHead, sGroup, lSecID, lSingleEnt, lSinglePar, _
    iTempType, lTempVal, vaEntryIDs, vaDebCredUnit, vaAmt, _
    vaItemDesc, vaEnt, vaPar, vaAcct, vaICP, vaCust1, vaCust2, _
    vaCust3, vaCust4
'Create the journal.
cIHsvJournalEx.SaveTextJournal lScenID, lYearID, lPerID, _
    lValID, iType, 1, iAttr, sJnlName, "", "", lSingleEnt, _
    lSinglePar, lSecID, iaDebCredUnit, saAmount, vaItemDesc, _
    vaEnt, vaPar, vaAcct, vaICP, vaCust1, vaCust2, vaCust3, _
    vaCust4, lJnlID
'Assign the return value
CreateTextJournal = lJnlID
End Function
```

SaveTextTemplate

Creates a new journal template, or updates an existing journal template.

Caution! To successfully call `SaveTextTemplate`, the connected user must be assigned to the Journals Administrator role.

The arguments for template line items consist of arrays. The arrays have a one-to-one correspondence; for example, the second elements in the `varabstrAmount` and `vararlAccount` arguments' arrays specify the amount and account for a template's second line item.

`SaveTextTemplate` takes a `String` array for the template's line item's amounts. To create a template by passing a `Double` array of amounts, use `SaveTemplate` instead of `SaveTextTemplate`. The methods are almost identical; the only difference is the subtype of the line item amount array. For more information, see [“SaveTemplate” on page 431](#).

Syntax

```
<IHsvJournalsEx>.SaveTextTemplate nType, nAttribute, bstrLabel,  
bstrDescription, bstrGroup, lSingleEntity, lSingleParent, nTemplateType,  
lValueID, vararnDebitCreditUnit, varabstrAmount, varabstrDescription,  
vararlEntity, vararlParent, vararlAccount, vararlICP, vararlCustom1,  
vararlCustom2, vararlCustom3, vararlCustom4, plTemplateID
```

Argument	Description
<i>nType</i>	Integer (ByVal). Specifies the journal type of the journals that will be created from the template. Pass one of the <code>HFMConstants</code> type library constants listed in “Journal Type Constants” on page 851 .
<i>nAttribute</i>	Integer (ByVal). Specifies whether journals created from the template must be balanced. Pass one of the <code>HFMConstants</code> type library constants listed in “Balance Type Constants” on page 848 .
<i>bstrLabel</i>	String (ByVal). The label of the template.
<i>bstrDescription</i>	String (ByVal). The description for the template. You can pass a blank string if there is no description.
<i>bstrGroup</i>	String (ByVal). The journal group to which the journals created from the template will be assigned. You can pass a blank string if the journals will not be assigned to a journal group.
<i>lSecurityClass</i>	Long. The ID of the security class for the template. Tip: To get the ID from a security class name, use <code>HsvSecurityAccess.GetSecurityClassLabel</code> .
<i>lSingleEntity</i>	Long (ByVal). For single entity journal templates, specify the member ID of the journal's base entity. For multi-entity journal templates, pass a value of -1.
<i>lSingleParent</i>	Long (ByVal). For single entity journal templates, specify the member ID of the journal's parent entity. For multi-entity journal templates, pass a value of -1.
<i>nTemplateType</i>	Integer (ByVal). Determines whether the template is standard or recurring. Pass one of the <code>HFMConstants</code> type library constants listed in “Template Type Constants” on page 852 .

Argument	Description
<i>lValueID</i>	Long (ByVal). For recurring templates, specify the member ID of the template's Value dimension member. Since standard templates are not assigned a Value dimension member, this argument returns -1 when <code>GetTemplate</code> is called for a standard template.
<i>vararrDebitCreditUnit</i>	Integer array (ByVal). Specifies whether the line items are debits or credits. Pass one of the HFMConstants type library constants listed in “Debit/Credit Constants” on page 849 .
<i>varabstrAmount</i>	String array (ByVal). The amounts of the line items.
<i>varabstrDescription</i>	String array (ByVal). The descriptions of the line items.
<i>vararrEntity</i>	Long array (ByVal). The member IDs for the base entities of the line items. For single-entity items, set the IDs in this array to the same member ID that is in the <i>lSingleEntity</i> argument.
<i>vararrParent</i>	Long array (ByVal). The member IDs for the parent entities of the line items. For single-entity items, set the IDs in this array to the same member ID that is in the <i>lSingleParent</i> argument.
<i>vararrAccount</i>	Long array (ByVal). The member IDs of the accounts for the line items.
<i>vararrICP</i>	Long array (ByVal). The member IDs of the line items' Intercompany Partner dimension members.
<i>vararrCustom1</i>	Long array (ByVal). The member IDs of the line items' Custom 1 dimension members.
<i>vararrCustom2</i>	Long array (ByVal). The member IDs of the line items' Custom 2 dimension members.
<i>vararrCustom3</i>	Long array (ByVal). The member IDs of the line items' Custom 3 dimension members.
<i>vararrCustom4</i>	Long array (ByVal). The member IDs of the line items' Custom 4 dimension members.
<i>plTemplateID</i>	Long. The purpose of this argument depends upon whether you are creating a new template or updating an existing template: <ul style="list-style-type: none"> ● For a newly created template, this argument returns the automatically-generated ID of the template. ● To update an existing template, pass the template's ID with this argument. You can get template IDs with <code>GetJournalTemplateItemID</code>; for details, see “GetJournalTemplateItemID” on page 396.

Example

This example defines a custom function named `NewTextTemplate` that creates new templates from an existing template, changing the entities while carrying over the other properties of the existing template. The existing template's name and the new template's name, description, and Entity dimension member IDs are passed as the custom function's arguments. The example calls `GetTextTemplate` to get the existing template's information.

```
Function NewTextTemplate(sTemplate As String, sNewLabel _
    As String, sDesc As String, lNewEntID As Long, _
    lNewParID As Long) As Long
Dim cHsvJournals As HsvJournals, iType As Integer
Dim cIHsvJournalEx As IHsvJournalsEx, lTempID As Long
Dim sLabel As String, sDescHead As String, sGroup As String
Dim lSecID As Long, lSingleEnt As Long, lSinglePar As Long
Dim iTempType As Integer, lVal As Long, vaEntryIDs
```

```

Dim vaDebCredUnit, vaAmt, vaItemDesc, vaEnt, vaPar, vaAcct
Dim vaICP, vaCust1, vaCust2, vaCust3, vaCust4
Dim lNewTempID As Long, iAttr As Integer, laNewEnt() As Long
Dim laNewPar() As Long, lUpBounds As Long
'm_cSession is an HsvSession object reference
Set cHsvJournals = m_cSession.Journals
Set cIHsvJournalEx = m_cSession.Journals
lTempID = cHsvJournals.GetJournalTemplateItemID(sTemplate)
cIHsvJournalEx.GetTextTemplate lTempID, iType, iAttr, sLabel, _
    sDescHead, sGroup, lSecID, lSingleEnt, lSinglePar, _
    iTempType, lVal, vaEntryIDs, vaDebCredUnit, vaAmt, _
    vaItemDesc, vaEnt, vaPar, vaAcct, vaICP, vaCust1, vaCust2, _
    vaCust3, vaCust4
    lUpBounds = UBound(vaEnt)
'Create the entity ID arrays that will be passed to SaveTemplate
ReDim laNewEnt(lUpBounds)
ReDim laNewPar(lUpBounds)
For i = LBound(vaEnt) To lUpBounds
    laNewEnt(i) = lNewEntID
    laNewPar(i) = lNewParID
Next i
cIHsvJournalEx.SaveTextTemplate iType, iAttr, sNewLabel, _
    sDesc, sGroup, lSecID, lNewEntID, lNewParID, iTempType, _
    lVal, vaDebCredUnit, vaAmt, vaItemDesc, laNewEnt, _
    laNewPar, vaAcct, vaICP, vaCust1, vaCust2, vaCust3, _
    vaCust4, lNewTempID
NewTextTemplate = lNewTempID
End Function

```

SubmitJournals

Submits one or more journals. The *varar!JournalIDs* argument takes the IDs of the journals that are being rejected.

Caution! The user must be assigned to the Journals Administrator or Create Journals role, otherwise error number 40D (hexadecimal) will occur.

Syntax

```
<IHsvJournalsEx>.SubmitJournals lScenario, lYear, varar!JournalIDs,
pvararnRc
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>varar!JournalIDs</i>	Long array (ByVal). The IDs of the journals to be submitted. Get these IDs with the HsvJournals object's <code>GetItemID</code> method; for more information, see "GetItemID" on page 396 .
<i>pvararnRc</i>	Variant array. Returns codes that identify the success or failure of the journal submissions. The elements in this array have a one-to-one correspondence with the journal IDs passed in the <i>varar!JournalIDs</i> argument. Successfully submitted journals return 0.

Argument	Description
	The array is returned as an Integer subtype.

Example

This example submits two journals. The calls to the user-defined `GetMemberID` function get the example's dimension member IDs; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`. The `laJnlIDs` array variable is then populated with the journal IDs by `HsvJournals.GetItemID`. The member IDs for the Scenario and Year dimension are passed along with this array to `SubmitJournals`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim laJnlIDs(1) As Long, vaRetVal
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
laJnlIDs(0) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl810")
laJnlIDs(1) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl811")
m_cIHsvJournalEx.SubmitJournals lScen, lYear, laJnlIDs, vaRetVal
```

UnpostJournals

Unposts one or more journals. The *vararrlJournalIDs* argument takes the IDs of the journals that are being rejected.

Caution! The user must be assigned to the Journals Administrator or Post Journals role, otherwise error number 40D (hexadecimal) will occur.

Syntax

```
<IHsvJournalsEx>.UnpostJournals lScenario, lYear, vararrlJournalIDs,
pvararrnRc
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>vararrlJournalIDs</i>	Long array (ByVal). The IDs of the journals to be unposted. Get these IDs with the <code>HsvJournals</code> object's <code>GetItemID</code> method; for more information, see "GetItemID" on page 396 .
<i>pvararrnRc</i>	Variant array. Returns codes that identify the success or failure of the journal unpostings. The elements in this array have a one-to-one correspondence with the journal IDs passed in the <i>vararrlJournalIDs</i> argument. Successfully unposted journals return 0. The following list identifies some of the likely error codes in hexadecimal form: <ul style="list-style-type: none"> ● 40F ● 3F9

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

- 418
- 40D
- 411
- 40B

The array is returned as an Integer subtype.

Example

This example unposts two journals. The calls to the user-defined `GetMemberID` function get the example's dimension member IDs; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`. The `laJnlIDs` array variable is then populated with the journal IDs by `HsvJournals.GetItemID`. The member IDs for the Scenario and Year dimension are passed along with this array to `UnpostJournals`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim laJnlIDs(1) As Long, vaRetVal
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
laJnlIDs(0) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl810")
laJnlIDs(1) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl811")
m_cIHsvJournalEx.UnpostJournals lScen, lYear, laJnlIDs, _
vaRetVal
```

UnsubmitJournals

Unsubmits one or more journals. The *vararJJournalIDs* argument takes the IDs of the journals that are being unsubmitted.

Caution! The user must be assigned to the Journals Administrator or Create Journals role, otherwise error number 40D (hexadecimal) will occur.

Syntax

```
<IHsvJournalsEx>.UnsubmitJournals lScenario, lYear, vararJJournalIDs,
pvararnRc
```

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

<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
------------------	---

<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
--------------	---

<i>vararJJournalIDs</i>	Long array (ByVal). The IDs of the journals to be unsubmitted. Get these IDs with the <code>HsvJournals</code> object's <code>GetItemID</code> method; for more information, see "GetItemID" on page 396 .
-------------------------	--

Argument	Description
<i>pvararrRc</i>	Variant array. Returns codes that identify the success or failure of the journal unsubscriptions. The elements in this array have a one-to-one correspondence with the journal IDs passed in the <i>vararrJournalIDs</i> argument. Successfully unsubmitted journals return 0. The array is returned as an Integer subtype.

Example

This example unsubmitted two journals. The calls to the user-defined `GetMemberID` function get the example's dimension member IDs; for details on `GetMemberID`, see the [Examples](#) for `IHsvTreeInfo.GetItemID`. The `laJnlIDs` array variable is then populated with the journal IDs by `HsvJournals.GetItemID`. The member IDs for the Scenario and Year dimension are passed along with this array to `UnsubmitJournals`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim laJnlIDs(1) As Long, vaRetVal
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
laJnlIDs(0) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl810")
laJnlIDs(1) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl811")
m_cIHsvJournalEx.UnsubmitJournals lScen, lYear, laJnlIDs, _
vaRetVal
```

ValidateLineItems

Validates whether the values passed would define a valid journal.

The journal line item data is passed as arrays; these arrays must have a one-to-one correspondence. `ValidateLineItems` returns an array of error numbers that corresponds to these arrays of line item data.

Syntax

```
<IHsvJournalsEx>.ValidateLineItems lScenarioID, lYear, lPeriod, lValueID,
lJournalID, sJournalType, sBalanceAttribute, lSingleEntity, lSingleParent,
vararrLineItemID, vararrEntityID, vararrParent, vararrAccount, vararrICP,
vararrCustom1, vararrCustom2, vararrCustom3, vararrCustom4,
vararrDebitCreditUnit, vararrAmount, pvarHRLLineItems
```

Argument	Description
<i>lScenarioID</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>lValueID</i>	Long (ByVal). The member ID of the Value dimension member.

Argument	Description
<i>lJournalID</i>	Long (ByVal). Pass 0 (zero) to this argument.
<i>sJournalType</i>	Integer (ByVal). Specifies the journal's type. Pass one of the HFMConstants type library constants listed in "Journal Type Constants" on page 851 .
<i>sBalanceAttribute</i>	Integer (ByVal). Specifies whether the journal must be balanced. Pass one of the HFMConstants type library constants listed in "Balance Type Constants" on page 848 .
<i>lSingleEntity</i>	Long (ByVal). For single entity journals, specify the member ID of the journal's base entity. For multi-entity journals, pass a value of -1.
<i>lSingleParent</i>	Long (ByVal). For single entity journals, specify the member ID of the journal's parent entity. For multi-entity journals, pass a value of -1.
<i>vararLineItemID</i>	Long array (ByVal). The line numbers of the journal line items.
<i>vararEntityID</i>	Long array (ByVal). The member IDs of the line items' child Entity dimension members.
<i>vararParent</i>	Long array (ByVal). The member IDs of the line items' parent Entity dimension members.
<i>vararAccount</i>	Long array (ByVal). The member IDs of the line items' Account dimension members.
<i>vararICP</i>	Long array (ByVal). The member IDs of the line items' Intercompany Partner dimension members.
<i>vararCustom1</i>	Long array (ByVal). The member IDs of the line items' Custom 1 dimension members.
<i>vararCustom2</i>	Long array (ByVal). The member IDs of the line items' Custom 2 dimension members.
<i>vararCustom3</i>	Long array (ByVal). The member IDs of the line items' Custom 3 dimension members.
<i>vararCustom4</i>	Long array (ByVal). The member IDs of the line items' Custom 4 dimension members.
<i>vararnDebitCreditUnit</i>	Integer array (ByVal). A flag that determines whether the line items are debits or credits. Pass one of the HFMConstants type library constants listed in "Debit/Credit Constants" on page 849 .
<i>varabstrAmount</i>	String array (ByVal). The amounts of the line items.
<i>pvarHRLineItems</i>	Variant array. Returns the Financial Management error numbers that correspond to the line items. 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. Tip: To look up a description of a Financial Management error number, use the utility described in "Error Message Lookup Utility" on page 822 .

ValidateValue

Indicates whether a String resolves to a Double that Financial Management would then allow to be inserted as a journal amount. To evaluate as valid, the String must not represent a negative Double, as Financial Management does not allow the entry of negative amounts in journals.

Syntax

```
<IHsvJournalsEx>.ValidateValue(lEntity, lParent, lAccount, lValue,
bstAmount)
```

Argument Description

<i>IEntity</i>	Long (ByVal). You must pass a valid Long. The value passed is ignored, this argument is intended for future use.
<i>IParent</i>	Long (ByVal). You must pass a valid Long. The value passed is ignored, this argument is intended for future use.
<i>IAccount</i>	Long (ByVal). You must pass a valid Long. The value passed is ignored, this argument is intended for future use.
<i>IValue</i>	Long (ByVal). You must pass a valid Long. The value passed is ignored, this argument is intended for future use.
<i>bstAmount</i>	String (ByVal). The value to be validated.

Return Value

Boolean. Returns TRUE if the *bstAmount* argument's value is considered valid, FALSE if it is considered invalid.

Example

This example tests the validity of the value in a text box named `txtData`. If the value is invalid, any code placed within the `If` structure would be executed.

```
Dim bRet As Boolean
bRet = m_cIHsvJournalEx.ValidateValue(-1, -1, -1, -1, _
Trim(txtData.Text))
If bRet = False Then
...
End If
```

IHsvJournalsReport Interface Methods

This interface provides methods for returning journal report data.

Assign `IHsvJournalsReport` object references with the `Journals` property of the `HsvSession` object. For an example, see [“Journals” on page 158](#).

The following topics describe the interface methods and the structures of arrays used by these methods.

Report Data Array Structures

The following tables describe structures of arrays used by [GetReportData](#) and [GetReportData2](#). These are the arrays for journal report filters and display columns.

Journal Report Filter Array Elements

The following table lists the index numbers, corresponding display columns, and descriptions of the supported elements in the array that specifies a report's journal filters.

Note: The array passed to the *vararvFilters* argument is a 32-element array. Indexes for this array that are not listed in the following table are reserved for future use.

Table 66 Journal Report Filter Array Elements

Index	Display Column	Description
0	Label	A String containing the desired label. You can use the percentage sign (%) as a wildcard character.
1	Status	An array of Longs containing the journal statuses to be applied. These statuses are represented by the HFMCconstants type library constants listed in “Journal Status Constants” on page 851.
2	Type	An array of Longs containing the journal types to be applied. These balance types are represented by the HFMCconstants type library constants listed in “Journal Type Constants” on page 851.
3	Balance Type	An array of Longs containing the balance types to be applied. These balance types are represented by the HFMCconstants type library constants listed in “Balance Type Constants” on page 848.
4	Group	A String containing the name of the desired journal group. You can use the percentage sign (%) as a wildcard character.
5	Description	A String containing the name of the desired description. You can use the percentage sign (%) as a wildcard character.
14	Entity	The member ID (Long) of the entity by which to filter.
15	Parent	The member ID (Long) of the parent entity by which to filter.
16	Account	The member ID (Long) of the account by which to filter.

Journal Report Display Columns

The journal report methods take an array that specifies the report data to return. The following table lists the constants that identify and the type of data returned for the columns.

Table 67 Report Column Display Constants

Constant	Column	Return Value (Subtype and Description)
0	Label	String. The journal label.
1	Status	Integer. The journal status. The valid return values are represented by the HFMCconstants type library constants listed in “Journal Status Constants” on page 851
2	Type	Integer. The journal type. The valid return values are represented by the HFMCconstants type library constants listed in “Journal Type Constants” on page 851
3	Balance Type	Integer. The balance type. The valid return values are represented by the HFMCconstants type library constants listed in “Balance Type Constants” on page 848 .
4	Group	String. The name of the journal group.
5	Description	String. The journal description.
6	Short Description	String. The truncated version of the journal description.
7	Created By	String. The fully qualified username of the journal creator.
8	Date Created	Double. The time and date on which the journal was created. This is returned as a Double that you can cast to a Date format.
9	Approved By	String. The fully qualified username of the journal approver. If a journal has not been approved, a blank String is returned.
10	Approved On	Double. The time and date on which the journal was approved. This is returned as a Double that you can cast to a Date format. If a journal has not been approved, 0 is returned.
11	Posted By	String. The fully qualified username of the journal poster. If a journal has not been posted, a blank String is returned.

Constant	Column	Return Value (Subtype and Description)
12	Date Posted	Double. The time and date on which the journal was posted. This is returned as a Double that you can cast to a Date format. If a journal has not been posted, 0 is returned.
13	Security Class	Long. The ID number of the journal's security class.
14	Entity	Long. The member ID of the journal's entity.
15	Parent	Long. The member ID of the journal's parent entity.
16	Account	Long. The member ID of the journal's account.
17	ICP	Long. The member ID of the journal's Intercompany Partner dimension member.
18	Custom1	Long. The member ID of the journal's Custom 1 dimension member.
19	Custom2	Long. The member ID of the journal's Custom 2 dimension member.
20	Custom3	Long. The member ID of the journal's Custom 3 dimension member.
21	Custom4	Long. The member ID of the journal's Custom 4 dimension member.

GetReportData

Returns data for journals; only actual journal amounts are returned.

Note: To return both scaled and actual amounts, use [GetReportData2](#).

Syntax

```
<IHsvJournalsReport>.GetReportData lScenario, lYear, lPeriod, lValue,
vararvFilters, varalColumns, varanSortFlags, pvar2DvarData, pvardValues,
pvarlStatus
```

Argument Description

lScenario Long (ByVal). The member ID of the Scenario dimension member for the report data.

Argument	Description
<i>Year</i>	Long (ByVal). The member ID of the Year dimension member for the report data.
<i>IPeriod</i>	Long (ByVal). The member ID of the Period dimension member for the report data.
<i>IValue</i>	Long (ByVal). The member ID of the Value dimension member for the report data.
<i>varvFilters</i>	<p>Variant (ByVal). Filters out journals:</p> <ul style="list-style-type: none"> ● To return all journals in the report, pass an empty variable. ● To filter out journals from the report, pass this as a Variant array that consists of 32 elements <p>When filtering journals, each array element corresponds to a type of filter. The available filters and their indexes in this 32-element array are described in Table 66 on page 445. Specify values for only those elements that correspond to the desired filtering criteria; if you do not want to use a filter, leave the corresponding array element empty.</p> <p>Note: In this release only the first six elements and the fourteenth through sixteenth elements of this array are supported; the remaining elements are reserved for future use.</p>
<i>varalColumns</i>	<p>Long array (ByVal). Specifies the display columns that the report data will contain. To create this array, use the constants listed in Table 67 on page 446.</p> <p>Tip: Specify the sortable columns as the top elements of this array, which has a one-to-one correspondence with the <i>varanSortFlags</i> argument's array.</p>
<i>varanSortFlags</i>	<p>Integer array (ByVal). Specifies the columns on which the report data will be sorted, as well as the sort order. This array has a one-to-one correspondence with the <i>varalColumns</i> argument's array. To specify these columns, use one of the HFMConstants type library constants listed in "Journal Report Sort Option Constants" on page 850.</p>
<i>pvar2DvarData</i>	<p>Variant array. Returns information for the specified columns. This is a multidimensional array, with one dimension for each column specified in the <i>varalColumns</i> argument; in each dimension, there is one element per journal.</p> <p>For details on the values returned in the array elements, see Table 67 on page 446.</p>
<i>pvardValues</i>	<p>Variant array. Returns the debit and credit amounts for the journals in the report. This array has a one-to-one correspondence with the <i>pvar2DvarData</i> argument's array, and is returned as a Double subtype.</p>
<i>pvarIStatus</i>	<p>Variant array. Returns an array that indicates whether the connected user has access to the line items. Each array element returns one of the following values:</p> <ul style="list-style-type: none"> ● 0 if the user has access to the corresponding line item. ● <code>CELLSTATUS_NOREADACCESS</code>, which is an HFMConstants type library constant indicating that the user does not have access to a cell. <p>This array has a one-to-one correspondence with the <i>pvar2DvarData</i> argument's array, and is returned as a Long subtype.</p>

Example

The following subroutine gets data of Approved and Posted journals that match the specified journal label and are for the specified dimension members. The journal labels and the labels of the journals' Entity and Account dimension members are printed to Visual Basic's Immediate window. The report information is sorted by the Label column in ascending order.

```
Sub printPostApprJnlRpt(lScen As Long, lYear As Long, lPer As Long, _
    lVal As Long, sJnlLbl As String)
```



```

Dim cJournalsReport As IHsvJournalsReport
Dim laCols(3) As Long, vaDisplayData, vaFilter(31)
Dim iaSortFlags(0) As Integer, vaVals, vaStatus
Dim cTreeInfo As IHsvTreeInfo, sMemberLabel As String
'g_cSession is an HsvSession object reference
Set cJournalsReport = g_cSession.Journals
'Specify the display columns.
laCols(0) = 0
laCols(1) = 14
laCols(2) = 15
laCols(3) = 16
'Specify the filtering criteria
vaFilter(0) = sJnlLbl
vaFilter(1) = Array(JSF_APPROVED, JSF_POSTED)
'Sort ascending on the Label column
iaSortFlags(0) = JOURNALREPORT_SORT_ASCENDING
'Print the report data for the specified journals.
cJournalsReport.GetReportData lScen, lYear, lPer, _
    lVal, vaFilter, laCols, iaSortFlags, vaDisplayData, _
    vaVals, vaStatus
For i = LBound(vaDisplayData, 1) To UBound(vaDisplayData, 1)
    Debug.Print "Label: " & vaDisplayData(i, 0)
    'g_cMetadata is an HsvMetadata object reference
    Set cTreeInfo = g_cMetadata.Entities
    cTreeInfo.GetLabel vaDisplayData(i, 1), sMemberLabel
    Debug.Print "Entity: " & sMemberLabel
    cTreeInfo.GetLabel vaDisplayData(i, 2), sMemberLabel
    Debug.Print "Parent: " & sMemberLabel
    Set cTreeInfo = g_cMetadata.Accounts
    cTreeInfo.GetLabel vaDisplayData(i, 3), sMemberLabel
    Debug.Print "Account: " & sMemberLabel & vbCrLf
Next i
End Sub

```

GetReportData2

Returns data for journals; scaled and actual journal amounts are returned.

Syntax

```

<IHsvJournalsReport>.GetReportData2 lScenario, lYear, lPeriod, lValue,
varavFilters, varalColumns, varanSortFlags, pvar2DvarData,
pvardScaledValues, pvardValues, pvarlStatus

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member for the report data.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member for the report data.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member for the report data.
<i>lValue</i>	Long (ByVal). The member ID of the Value dimension member for the report data.
<i>varavFilters</i>	Variant (ByVal). Filters out journals:

Argument	Description
	<ul style="list-style-type: none"> ● To return all journals in the report, pass an empty variable. ● To filter out journals from the report, pass this as a Variant array that consists of 32 elements <p>When filtering journals, each array element corresponds to a type of filter. The available filters and their indexes in this 32-element array are described in Table 66 on page 445. Specify values for only those elements that correspond to the desired filtering criteria; if you do not want to use a filter, leave the corresponding array element empty.</p> <p>In this release only the first six elements and the fourteenth through sixteenth elements of this array are supported; the remaining elements are reserved for future use.</p>
<i>varalColumns</i>	<p>Long array (ByVal). Specifies the display columns that the report data will contain. To create this array, use the constants listed in Table 67 on page 446.</p> <p>Tip: Specify the sortable columns as the top elements of this array, which has a one-to-one correspondence with the <i>varanSortFlags</i> argument's array.</p>
<i>varanSortFlags</i>	<p>Integer array (ByVal). Specifies the columns on which the report data will be sorted, as well as the sort order. This array has a one-to-one correspondence with the <i>varalColumns</i> argument's array. To specify these columns, use one of the HFMConstants type library constants listed in "Journal Report Sort Option Constants" on page 850.</p>
<i>pvar2DvarData</i>	<p>Variant array. Returns information for the specified columns. This is a multidimensional array, with one dimension for each column specified in the <i>varalColumns</i> argument; in each dimension, there is one element per journal.</p> <p>For details on the values returned in the array elements, see Table 67 on page 446.</p>
<i>pvardScaledValues</i>	<p>Variant array. Returns the scaled debit and credit amounts for the journals in the report. This array has a one-to-one correspondence with the <i>pvar2DvarData</i> argument's array, and is returned as a Double subtype.</p>
<i>pvardValues</i>	<p>Variant array. Returns the actual debit and credit amounts for the journals in the report. This array has a one-to-one correspondence with the <i>pvar2DvarData</i> argument's array, and is returned as a Double subtype.</p>
<i>pvarlStatus</i>	<p>Variant array. Returns an array that indicates whether the connected user has access to the line items. Each array element returns one of the following values:</p> <ul style="list-style-type: none"> ● 0 if the user has access to the corresponding line item. ● CELLSTATUS_NOREADACCESS, which is an HFMConstants type library constant indicating that the user does not have access to a cell. <p>This array has a one-to-one correspondence with the <i>pvar2DvarData</i> argument's array, and is returned as a Long subtype.</p>

In This Chapter

Access Rights Constants	451
Role ID Constants	452
Task Constants	452
E-mail Alerting Constants	452
User Groups - User Type Flag Constants	452
Identity Type Constants	453
Search Filters	453
Security Identifiers	454
The Connected User	454
HsvSecurityAccess Object Methods	454
IHsvDataSecurity Interface Methods	485

This chapter describes the HsvSecurityAccess type library. Use the methods of this type library to get and set an application's users, roles, and security class access rights, and to get information on user rights to process units.

To use the HsvSecurityAccess type library, you must reference `HsvSecurityAccess.dll` in your project. The HsvSecurityAccess type library contains the HsvSecurityAccess object and the IHsvDataSecurity interface.

Access Rights Constants

Several HsvSecurityAccess methods take or return values that identify users' access rights. These access rights are represented by the constants listed in the following table.

Table 68 Access Rights Constants

Rights level	Enumerated
None	HFM_ACCESS_RIGHTS_NONE
Promote	HFM_ACCESS_RIGHTS_READANDPROMOTE
Read	HFM_ACCESS_RIGHTS_READONLY
All	HFM_ACCESS_RIGHTS_ALL

Rights level	Enumerated
Metadata	HFM_ACCESS_RIGHTS_VIEW

Caution! A constant named `HFM_ACCESS_RIGHTS_UNSPECIFIED` is also visible in Visual Basic's Object Browser. However, this constant is only for internal use; if you attempt to use it, an error will occur.

Role ID Constants

Several `HsvSecurityAccess` methods pass or return internal IDs of security roles. The constants that represent the roles correspond to those in the `HFMConstants` enumeration `tagHFM_ROLE_ENUM`, which is described in “[Role Constants](#)” on page 856.

Tip: You can also return a role's ID by passing the role name to the `HsvSecurityAccess` method [GetRoleID](#).

Task Constants

Some `HsvSecurityAccess` methods use internal IDs of security tasks. The constants that represent the tasks correspond to those in the `HFMConstants` enumeration `tagHFM_TASK_ENUM`, which is described in “[Task Constants](#)” on page 858.

E-mail Alerting Constants

Some `HsvSecurityAccess` methods use bitmasks that represent E-mail alerting rights. The following table lists constants that represent the bits.

Table 69 E-mail Alerting Constants

Notification Right	Constant
E-mail alerting disabled	<code>HFM_EMAIL_ALERTING_DISABLED</code>
E-mail alerting enabled	<code>HFM_EMAIL_ALERTING_ENABLED</code>
	Note: The bits represented by <code>HFM_EMAIL_ALERTING_ENABLED</code> and <code>HFM_EMAIL_ALERTING_DISABLED</code> are mutually exclusive.

User Groups - User Type Flag Constants

Some `HsvSecurityAccess` methods use bitmasks that represent flags for types of users in user groups. The following table lists constants that represent the bits.

Table 70 User Type Flag Constants

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 HsvSecurityAccess methods use bitmasks that represent *identity types*, which are types of users. The following table lists constants that represent the bits.

Table 71 Identity Type Constants

Identity 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 72 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

Security Identifiers

Many `HsvSecurityAccess` methods take or return *security identifiers* (SIDs) that represent users. These security identifiers are automatically generated by Financial Management. To work with security identifiers, the following `HsvSecurityAccess` object methods are helpful:

- `GetUserSID` takes a username and returns the corresponding security identifier.
- `GetUserName2` takes a security identifier and returns the corresponding username.
- `EnumUsers2` returns the security identifiers and usernames of an application's users.

The Connected User

Many `HsvSecurityAccess` methods return information about the *connected user*. The connected user is the user who is logged on when `HsxClient.OpenApplication` or `HsxClientUI.OpenApplication` sets the `HsvSession` object reference from which the `HsvSecurityAccess` or `IHsvDataSecurity` object reference is obtained.

Tip: To get the security identifier of the connected user, call `GetConnectedUser2`.

HsvSecurityAccess Object Methods

Many of the `HsvSecurityAccess` object's methods return information about an application's users, roles, and security classes. The other methods enable you to perform the following security-related actions:

- Add or remove users from an application. In addition, you can assign users to or remove users from roles and security classes.
- Add or remove users from roles.
- Add or remove security classes from an application. In addition, you can specify users' access rights to security classes.

Note: Security classes are assigned to Entity dimension members, and you can get an entity's security class with `HsvEntities.GetSecurityClassID`. For more information, see "[GetSecurityClassID](#)" on page 250.

The `HsvSecurityAccess` object's methods are summarized in [Table 26 on page 79](#), and are described in detail in the following topics.

Note: Assign `HsvSecurityAccess` object references with the `Security` property of the `HsvSession` object.

Caution! For methods that change roles, users, or security class access rights, the connected user must be assigned to the Application Administrator role. Use `IsApplicationAdministrator` to test whether the connected user is assigned to

the Application Administrator role; for more information, see [“IsApplicationAdministrator” on page 478](#).

AddApplicationAdministrator

Deprecated - use [AddApplicationAdministrator2](#).

AddApplicationAdministrator2

Assigns the Application Administrator role to a user.

Syntax

```
<HsvSecurityAccess>.AddApplicationAdministrator2 bstrUserSID
```

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

<i>bstrUserSID</i>	String (ByVal). The user's security identifier.
--------------------	---

Tip: You can get a user's security identifier with [GetUserSID](#).

Example

The following subroutine adds the specified username as an Application Administrator.

```
Sub addAppAdminUsername(sName As String)
Dim cSecurity As HsvSecurityAccess, sSID As String
'g_cSession is an HsvSession object reference
Set cSecurity = g_cSession.Security
cSecurity.GetUserSID sName, sSID
cSecurity.AddApplicationAdministrator2 sSID
End Sub
```

AddOrRemoveApplicationAdministrators

Deprecated - use [AddOrRemoveApplicationAdministrators2](#).

AddOrRemoveApplicationAdministrators2

Adds or removes one or more users from the Application Administrator role.

Syntax

```
<HsvSecurityAccess>.AddOrRemoveApplicationAdministrators2
varabstrUserSIDs, varabAdd
```

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

<i>varabstrUserSIDs</i>	String array (ByVal). The security identifiers of the users being added to or removed from the Application Administrator role.
-------------------------	--

Argument	Description
	This array has a one-to-one correspondence with the <i>varabAdd</i> argument's array. For example, the third element of the <i>varabAdd</i> array indicates whether the user identified by the third element of this array is added or removed.
<i>varabAdd</i>	Boolean array (ByVal). Indicates whether users should be added or removed from the Application Administrator role. Set an array element to TRUE to add a user, or to FALSE to remove a user.

Example

The following subroutine adds the specified users to the Application Administrator role. The subroutine takes an array of security identifiers.

```
Sub addAdmins(saSIDs() As String)
Dim cSecurity As HsvSecurityAccess, bAdd() As Boolean
'g_cSession is an HsvSession object reference
Set cSecurity = g_cSession.Security
ReDim bAdd(UBound(saSIDs))
For i = LBound(saSIDs) To UBound(saSIDs)
    bAdd(i) = True
Next i
cSecurity.AddOrRemoveApplicationAdministrators2 saSIDs, bAdd
End Sub
```

AddOrRemoveRolesFromUser

Deprecated - use [AddOrRemoveRolesFromUser2](#).

AddOrRemoveRolesFromUser2

Adds or removes a user from one or more roles.

Syntax

```
<HsvSecurityAccess>.AddOrRemoveRolesFromUser2 bstrUserSID, varalRoles,
varabAdd
```

Argument	Description
<i>bstrUserSID</i>	String (ByVal). The security identifier of the user being assigned to or removed from the roles.
<i>varalRoles</i>	Long array (ByVal). The IDs of the roles. This array has a one-to-one correspondence with the <i>varabAdd</i> argument's array. For example, the third element of the <i>varabAdd</i> array indicates whether the user is added to or removed from the role identified by the third element of this array. Tip: To get a role's ID from its name, use GetRoleID .
<i>varabAdd</i>	Boolean array (ByVal). Indicates whether the user should be assigned to or removed from the roles. For each array element, specify TRUE to assign, FALSE to remove.

Example

The following subroutine adds the specified roles to a given user. The subroutine takes an array of role IDs and the user's security identifier.

```
Sub assignRolesToUser(laRoles() As Long, sSID As String)
Dim cSecurity As HsvSecurityAccess, bAdd() As Boolean
'g_cSession is an HsvSession object reference
Set cSecurity = g_cSession.Security
ReDim bAdd(UBound(laRoles))
For i = LBound(laRoles) To UBound(laRoles)
    bAdd(i) = True
Next i
cSecurity.AddOrRemoveRolesFromUser2 sSID, laRoles, bAdd
End Sub
```

AddOrRemoveUsersFromRole

Deprecated - use [AddOrRemoveUsersFromRole2](#).

AddOrRemoveUsersFromRole2

Assigns users to or removes them from a given role.

Syntax

```
<HsvSecurityAccess>.AddOrRemoveUsersFromRole2 lRoleID, varabstrUserSIDs,
varabAdd
```

Argument	Description
<i>lRoleID</i>	Long (ByVal). The ID of the role. For a list of valid role IDs, see “Role ID Constants” on page 452 . Tip: To get a role's ID from its name, use GetRoleID .
<i>varabstrUserSIDs</i>	String array (ByVal). The security identifiers of the users being assigned to or removed from the role. This array has a one-to-one correspondence with the <i>varabAdd</i> argument's array. For example, the third element of the <i>varabAdd</i> array indicates whether the user identified by the third element of this array is assigned or removed.
<i>varabAdd</i>	Boolean array (ByVal). Indicates whether users should be assigned to or removed from the role. For each array element, specify TRUE to assign, FALSE to remove.

Example

The following subroutine adds the specified users to a given role. The subroutine takes an array of security identifiers and the name of the role.

```
Sub AssignRole(sId As String, sRole As String)
Dim cSecurity As HsvSecurityAccess, lRoleId As Long
'g_cSession is an HsvSession object reference
Set cSecurity = g_cSession.Security
cSecurity.GetRoleID sRole, lRoleId
cSecurity.AddUserToRole2 lRoleId, sId
```

End Sub

AddSecurityClass

Adds a security class to a Classic application.

Caution! The method will fail if executed against an application created with Performance Management Architect.

Syntax

```
<HsvSecurityAccess>.AddSecurityClass bstrSecurityClass, plSecurityClassID
```

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

<i>bstrSecurityClass</i>	String (ByVal). The name of the security class to add.
--------------------------	--

<i>plSecurityClassID</i>	Long. Returns the ID that Financial Management assigns to the security class.
--------------------------	---

AddSecurityClassWithAccessCode

For internal use.

AddUser

Deprecated - use [AddUser2](#).

AddUser2

Adds a user or user group to an application.

Syntax

```
<HsvSecurityAccess>.AddUser2 bstrName, pbstrFullName, pbstrUserSID
```

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

<i>bstrName</i>	String (ByVal). The username of the user group name.
-----------------	--

This must be a valid Windows username or user group for the application server, otherwise an error is thrown.

<i>pbstrFullName</i>	String. Returns the name of the user or group.
----------------------	--

<i>pbstrUserSID</i>	String. Returns the user's security identifier.
---------------------	---

AddUserEx

This method is not supported as of Release 4.1.

AddUserToRole

Deprecated - use [AddUserToRole2](#).

AddUserToRole2

Assigns a user to a role.

Syntax

```
<HsvSecurityAccess>.AddUserToRole2 lRoleID, bstrUserSID
```

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

<i>lRoleID</i>	Long (ByVal). The ID of the role to which the user is being assigned. For a list of valid role IDs, see “ Role ID Constants ” on page 452.
----------------	--

Tip: To get a role’s ID from its name, use [GetRoleID](#).

<i>bstrUserSID</i>	String (ByVal). The user’s security identifier.
--------------------	---

Example

The following subroutine assigns a user to a given role. The role name is passed to the subroutine, which uses `GetRoleID` to obtain the role ID.

```
Sub AssignRole(sId As String, sRole As String)
Dim cSecurity As HsvSecurityAccess, lRoleID As Long
'g_cSession is an HsvSession object reference
Set cSecurity = g_cSession.Security
cSecurity.GetRoleID sRole, lRoleID
cSecurity.AddUserToRole2 lRoleID, sId
End Sub
```

AllowRulesLoadForEPMAApp

For internal use.

DeleteSecurityClass

Deletes a security class from a Classic application.

Caution! The method will fail if executed against an application created with Performance Management Architect.

Syntax

```
<HsvSecurityAccess>.DeleteSecurityClass lSecurityClassID
```

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

<i>lSecurityClassID</i>	Long (ByVal). The ID of the security class to be deleted. You can get this ID by passing the security class name to GetSecurityClassID .
-------------------------	--

Example

The following method deletes the specified security class.

```
Sub deleteSecClass(sName As String)
Dim lSecClassID As Long, cSecurity As HsvSecurityAccess
'g_cSession is an HsvSession object reference
Set cSecurity = g_cSession.Security
cSecurity.GetSecurityClassID sName, lSecClassID
cSecurity.DeleteSecurityClass lSecClassID
End Sub
```

DeleteSecurityClassWithAccessCode

For internal use.

EnumApplicationAdministrators

Deprecated - use [EnumApplicationAdministrators2](#).

EnumApplicationAdministrators2

Returns arrays containing the security identifiers and usernames of the users assigned to the Application Administrator role. The arrays have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.EnumApplicationAdministrators2 pvarabstrUserSIDs,
pvarabstrNames
```

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

<i>pvarabstrUserSIDs</i>	Variant. Returns an array containing the security identifiers of the users. The array is returned as a String subtype.
--------------------------	--

<i>pvarabstrNames</i>	Variant. Returns an array containing the usernames of the users. The array is returned as a String subtype.
-----------------------	---

EnumRoles

Returns an array containing the names of an application's roles.

Syntax

```
<HsvSecurityAccess>.EnumRoles pvarabstrRoles
```

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

<i>pvarabstrRoles</i>	Variant array. Returns the names of the application's roles. The array is returned as a String subtype.
-----------------------	---

EnumRolesForPrincipal

Returns the IDs of a user's or group's roles.

Syntax

```
<HsvSecurityAccess>.EnumRolesForPrincipal bstrUserSID, pvaralRoleIds
```

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

<i>bstrUserSID</i>	String (ByVal). The user's or group's security identifier.
--------------------	--

<i>pvaralRoleIds</i>	Variant array. Returns the IDs of the assigned roles. Valid values are listed in "Role ID Constants" on page 452 .
----------------------	--

The array's subtype is Long.

Example

The following function returns the roles for a given username.

```
Function GetPrincRolesFromName(sName As String) As Variant
Dim cSecurity As HsvSecurityAccess, vaRoles As Variant
Dim sSID As String
'g_cSession represents an HsvSession instance
Set cSecurity = g_cSession.Security
cSecurity.GetUserSID sName, sSID
cSecurity.EnumRolesForPrincipal sSID, vaRoles
GetPrincRolesFromName = vaRoles
End Function
```

EnumRolesForUser

Returns the localized names of a user's roles.

Syntax

```
<HsvSecurityAccess>.EnumRolesForUser bstrUser, bstrLangId,
pvarabstrRoleNames
```

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

<i>bstrUser</i>	String (ByVal). The user's security identifier.
-----------------	---

Argument	Description
<i>bstrLangId</i>	String (ByVal). The ID of the language in which to return the role names. Valid values are represented by the constants listed in Table 102 on page 834 .
<i>pvarabstrRoleNames</i>	Variant array. Returns the localized role names. The array's subtype is String.

Example

The following function returns the English-language names of the connected user's roles.

```
Function GetUsersRolesEnglish() As Variant
Dim sSid As String, sName As String, cSecurity As HsvSecurityAccess
Dim vaRoles As Variant
'g_cSession represents an HsvSession instance
Set cSecurity = g_cSession.Security
cSecurity.GetConnectedUser2 sSid, sName
cSecurity.EnumRolesForUser sSid, CStr(HFM_LANGUAGE_ENGLISH), vaRoles
GetUsersRolesEnglish = vaRoles
End Function
```

EnumSecurityClasses

Returns arrays containing an application's security class IDs and names. The arrays have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.EnumSecurityClasses pvaralSecurityClassIDs,  
pvarabstrSecurityClasses
```

Argument	Description
<i>pvaralSecurityClassIDs</i>	Variant array. Returns the IDs of the application's security classes. The array is returned as a String subtype.
<i>pvarabstrSecurityClasses</i>	Variant array. Returns the names of the application's security classes. The array is returned as a String subtype.

Example

This example puts the names of an application's security classes into a ComboBox.

```
Dim vaSecIDs, vaSecNames, lHiBound As Long
'g_cSecurity is an HsvSecurityAccess object reference
g_cSecurity.EnumSecurityClasses vaSecIDs, vaSecNames
lHiBound = UBound(vaSecNames)
For i = LBound(vaSecNames) To lHiBound
    'cmbClasses represents the ComboBox
    cmbClasses.AddItem vaSecNames(i)
Next i
```

EnumSecurityClassRightsForPrincipal

Returns arrays representing a user's access and E-mail alerting rights to security classes. The arrays have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.EnumSecurityClassRightsForPrincipal bstrUserSID,  
pvaralSecurityClassIDs, pvarabstrSecurityClasses, pvaralRights,  
pvaralEmailAlerting
```

Argument	Description
<i>bstrUserSID</i>	String (ByVal). The user's security identifier.
<i>pvaralSecurityClassIDs</i>	Variant array. Returns the IDs of the security classes to which the user has been assigned access rights. The array's subtype is Long.
<i>pvarabstrSecurityClasses</i>	Variant array. Returns the names of the security classes. The array's subtype is String.
<i>pvaralRights</i>	Variant array. Returns the user's rights to the security classes. Valid values are represented by the constants in Table 68 on page 451 . The array's subtype is Long.
<i>pvaralEmailAlerting</i>	Variant array. Returns an array of bitmasks that represent user's E-mail alerting rights to the security classes. Valid values for the bits are represented by the enumeration described in "E-mail Alerting Constants" on page 452 . The array's subtype is Long.

Example

The following function returns the names of the security classes to which a user has All access rights.

```
Function GetUserSecClassesAllAccess(sName As String) As Variant  
Dim sSID As String, cSecurity As HsvSecurityAccess, lCounter As Long  
Dim vaSecIds, vaSecNames, vaRights, vaAlerts, vaRet()  
'g_cSession represents an HsvSession instance  
Set cSecurity = g_cSession.Security  
cSecurity.GetUserSID sName, sSID  
cSecurity.EnumSecurityClassRightsForPrincipal sSID, vaSecIds, _  
    vaSecNames, vaRights, vaAlerts  
lCounter = -1  
If IsArray(vaSecNames) = True Then  
    For i = LBound(vaSecNames) To UBound(vaSecNames)  
        If vaRights(i) = HFM_ACCESS_RIGHTS_ALL Then  
            lCounter = lCounter + 1  
            ReDim Preserve vaRet(lCounter)  
            vaRet(lCounter) = vaSecNames(i)  
        End If  
    Next i  
End If  
GetUserSecClassesAllAccess = vaRet  
End Function
```

EnumUserClassAccess

Returns the access and E-mail alerting rights that one or more users have for the specified security classes.

Syntax

```
<HsvSecurityAccess>.EnumUserClassAccess varabstrUserSIDs,  
varabstrSecClasses, pvar2DalRights, pvar2DalAlertable
```

Argument	Description
<i>varabstrUserSIDs</i>	String array (ByVal). The users' security identifiers.
<i>varabstrSecClasses</i>	String array (ByVal). The names of the security classes.
<i>pvar2DalRights</i>	Variant. Returns a two-dimensional array containing the IDs of the users' access rights for the security classes. The first dimension has a one-to-one correspondence with the <i>varabstrUserSIDs</i> argument's array, and the second dimension has a one-to-one correspondence with the <i>varabstrSecClasses</i> argument's array. IDs of access rights are represented by the enumeration described in " Access Rights Constants " on page 451.
<i>pvar2DalAlertable</i>	Variant. Returns a two-dimensional array containing bitmasks that represent the users' E-mail alerting rights for the security classes. The first dimension has a one-to-one correspondence with the <i>varabstrUserSIDs</i> argument's array, and the second dimension has a one-to-one correspondence with the <i>varabstrSecClasses</i> argument's array. Valid values for the bits are represented by the enumeration described in " E-mail Alerting Constants " on page 452.

EnumUsers

Deprecated - use [EnumUsers2](#).

EnumUsers2

Deprecated - use [EnumUsers3](#).

EnumUsers3

Returns arrays containing the security identifiers and usernames of an application's users. If delegated user management is enabled in Shared Services, you can filter users who have access rights not granted to the connected user.

The arrays returned have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.EnumUsers3 AccessType, pvarabstrUserSIDs,  
pvarabstrNames
```


Argument	Description
<i>AccessType</i>	Long (ByVal). If delegated user management is enabled, this argument lets you filter users who have access rights not granted to the connected user. Pass 1 to filter by access rights, 0 otherwise.
<i>pvarabstrUserSIDs</i>	Variant. Returns an array containing the security identifiers. The array is returned as a String subtype.
<i>pvarabstrNames</i>	Variant. Returns an array containing the usernames. The array is returned as a String subtype.

EnumUsersInGroup

Returns the security identifiers, usernames, and identity types of the users in a user group.

Syntax

```
<HsvSecurityAccess>.EnumUsersInGroup bstrGroupSID, lBitmaskFlags,
pvarabstrUserSIDs, pvarabstrUserFullNames, pvaralIdentityTypes
```

Argument	Description
<i>bstrGroupSID</i>	String (ByVal). The group's security identifier.
<i>lBitmaskFlags</i>	Long (ByVal). A bitmask that specifies the identity types to return in the <i>pvaralIdentityTypes</i> argument's array. Valid values for the bits are represented by the enumeration described in "User Groups - User Type Flag Constants" on page 452. For example, to exclude bits for application administrators from the array, pass <code>HFM_USER_GROUP_ENUM_EXCLUDEADMINS</code> .
<i>pvarabstrUserSIDs</i>	Variant. Returns an array that contains the users' security identifiers.
<i>pvarabstrUserFullNames</i>	Variant. Returns an array that contains the users' usernames.
<i>pvaralIdentityTypes</i>	Variant. Returns an array of bitmasks that indicate the users' identity types. Valid values for the bits are represented by the enumeration described in "Identity Type Constants" on page 453.

EnumUsersInRole

Deprecated - use [EnumUsersInRole2](#).

EnumUsersInRole2

Returns the security identifiers and usernames of the users and groups assigned to a given role.

Tip: To return security identifiers and usernames of users who belong to groups assigned to a role, use [EnumUsersInRole3](#).

Syntax

```
<HsvSecurityAccess>.EnumUsersInRole2 lRoleID, pvarabstrUserSIDs,  
pvarabstrNames
```

Argument	Description
<i>lRoleID</i>	Long (ByVal). The ID of the role. For a list of valid role IDs, see “Role ID Constants” on page 452 . Tip: To get a role's ID from its name, use GetRoleID .
<i>pvarabstrUserSIDs</i>	Variant. Returns an array containing the security identifiers of the users and groups assigned to the role. This array has a one-to-one correspondence with the array returned by the <i>pvarabstrNames</i> argument.
<i>pvarabstrNames</i>	Variant. Returns an array containing the usernames of the users and groups assigned to the role.

EnumUsersInRole3

Returns the security identifiers and usernames of the users and groups assigned to a given role, and optionally allows you to return this information for users of any groups assigned to the role.

Syntax

```
<HsvSecurityAccess>.EnumUsersInRole3 lRoleID, vbExpandGroup,  
pvarabstrUserSIDs, pvarabstrNames
```

Argument	Description
<i>lRoleID</i>	Long (ByVal). The ID of the role. For a list of valid role IDs, see “Role ID Constants” on page 452 . Tip: To get a role's ID from its name, use GetRoleID .
<i>vbExpandGroup</i>	Boolean (ByVal). A flag that specifies whether to include security identifiers and usernames of groups' users. Pass TRUE to return details of groups' users, FALSE otherwise.
<i>pvarabstrUserSIDs</i>	Variant. Returns an array containing the security identifiers of the users and groups assigned to the role. This array has a one-to-one correspondence with the array returned by the <i>pvarabstrNames</i> argument.
<i>pvarabstrNames</i>	Variant. Returns an array containing the usernames of the users and groups assigned to the role.

EnumUsersInSecurityClass

For internal use.

EnumUsersInSecurityClass2

For internal use.

EnumUsersInSecurityClass3

For internal use.

EnumUsersOrGroups

Returns the security identifiers, usernames, and identity types of the users in a user group.

Syntax

```
<HsvSecurityAccess>.EnumUsersOrGroups lBitmaskFlags,  
pvarabstrUserAndGroupSIDs, pvarabstrUserAndGroupFullNames,  
pvaralIdentityTypes
```

Argument	Description
<i>lBitmaskFlags</i>	Long (ByVal). A bitmask that specifies the identity types to return in the <i>pvaralIdentityTypes</i> argument's array. Valid values for the bits are represented by the enumeration described in "User Groups - User Type Flag Constants" on page 452. For example, to exclude bits for application administrators from the array, pass HFM_USER_GROUP_ENUM_EXCLUDEADMINS.
<i>pvarabstrUserAndGroupSIDs</i>	Variant. Returns an array that contains the security identifiers of the users or groups.
<i>pvarabstrUserAndGroupFullNames</i>	Variant. Returns an array that contains the usernames of the users or groups.
<i>pvaralIdentityTypes</i>	Variant. Returns an array of bitmasks that indicate the identity types of the users or groups. Valid values for the bits are represented by the enumeration described in "Identity Type Constants" on page 453.

EnumUsersWithFilter

Deprecated. Use [EnumUsersWithFilter2](#).

EnumUsersWithFilter2

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. If delegated user management is enabled in Shared Services, you can filter users who have access rights not granted to the connected user.

Syntax

```
<HsvSecurityAccess>.EnumUsersWithFilter2 bstrFilter,  
lBitFieldPrincipalsToReturn, AccessType, pvarabstrUserSIDs, pvarabstrNames
```

Argument	Description
<i>bstrFilter</i>	String (ByVal). The string by which to search user and group names. For wildcard searches, use asterisks (*).
<i>IBitFieldPrincipalsToReturn</i>	Long (ByVal). A bitmask that represents the user categories by which to filter. Valid values for the bits are listed in Table 72 on page 453 .
<i>AccessType</i>	Long (ByVal). If delegated user management is enabled, this argument lets you to filter users who have access rights not granted to the connected user. Pass 1 to filter by access rights, 0 otherwise.
<i>pvarabstrUserSIDs</i>	VARIANT array. Returns the security identifiers for the users and groups that match the search criteria. The array's subtype is String.
<i>pvarabstrNames</i>	VARIANT array. Returns the names of the users and groups that match the search criteria. The array's subtype is String.

Example

The following function returns an array of the user and group names that begin with the string passed.

```
Function GetUsersByPrefix(sPrefix As String) As Variant
Dim cSecurity As HsvSecurityAccess, vaIds, vaNames
'g_cSession represents an HsvSession instance
Set cSecurity = g_cSession.Security
cSecurity.EnumUsersWithFilter2 sPrefix & "*", HFM_USER_GROUP_ENUM_GROUPS _
    & HFM_USER_GROUP_ENUM_USERS, 0, vaIds, vaNames
GetUsersByPrefix = vaNames
End Function
```

GenerateSecurityReportForBiPub

For internal use.

GetAllSecurityClassRightsForConnectedUser

Returns arrays indicating the connected user's rights to all security classes. The arrays have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.GetAllSecurityClassRightsForConnectedUser
pvaralSecClassIds, pvarabstrSecClassNames, pvaralRights
```

Argument	Description
<i>pvaralSecClassIds</i>	VARIANT array. Returns the IDs of the application's security classes. The array's subtype is Long.
<i>pvarabstrSecClassNames</i>	VARIANT array. Returns the names of the application's security classes. The array's subtype is String.

Argument	Description
<i>pvaralRights</i>	Variant array. Returns the users' access rights to the security class. Valid values are represented by the enumeration described in "Access Rights Constants" on page 451 . The array's subtype is Long.

GetApplicationAdministratorAccessForAllUsers

Deprecated - use [GetApplicationAdministratorAccessForAllUsers2](#).

GetApplicationAdministratorAccessForAllUsers2

Indicates which of an application's users are assigned to the Application Administrator role. This method returns arrays containing the usernames and security identifiers of the application's users and an array of flags indicating whether the users are assigned to the Application Administrator role. The arrays have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.GetApplicationAdministratorAccessForAllUsers2  
pvarabstrUserSIDs, pvarabstrNames, pvarabHasAccess
```

Argument	Description
<i>pvarabstrUserSIDs</i>	Variant. Returns an array containing the security identifiers of the application's users. The array is returned as a String subtype.
<i>pvarabstrNames</i>	Variant. Returns an array containing the usernames of the application's users. The array is returned as a String subtype.
<i>pvarabHasAccess</i>	Variant. Returns an array of Boolean flags indicating whether the users are assigned to the Application Administrator role. Array elements return TRUE for users assigned to the Application Administrator role, FALSE otherwise.

GetConnectedUser

Deprecated - use [GetConnectedUser2](#).

GetConnectedUser2

Returns the security identifier and username of the connected user.

Syntax

```
<HsvSecurityAccess>.GetConnectedUser2 pbstrUserSID, pbstrName
```

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

<i>pbstrUserSID</i>	String. Returns the user's security identifier.
---------------------	---

<i>pbstrName</i>	String. Returns the username.
------------------	-------------------------------

Example

`GetConnectedUser2` is used in the example for [EnumRolesForUser](#).

GetIdentityTypes

Returns the identity types of the specified users.

Syntax

```
<HsvSecurityAccess>.GetIdentityTypes (varabstrSIDs)
```

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

<i>varabstrSIDs</i>	String array (ByVal). The security identifiers of the users.
---------------------	--

Return Value

Variant. Returns an array of bitmasks that represent the users' identity types. Valid values for the bits are listed in [“Identity Type Constants” on page 453](#).

This array has a one-to-one correspondence with the array passed to the *varabstrSIDs* argument.

GetNumRoles

Returns a count of the number of roles in an application.

Syntax

```
<HsvSecurityAccess>.GetNumRoles p1NumRoles
```

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

<i>p1NumRoles</i>	Long. Returns the number of roles.
-------------------	------------------------------------

GetOwner

This method is not supported as of Release 4.1.

GetRoleAccessForAllUsers

Deprecated - use [GetRoleAccessForAllUsers2](#).

GetRoleAccessForAllUsers2

Indicates whether the application's users are assigned to a given role.

`GetRoleAccessForAllUsers2` returns arrays that identify the users and whether they are assigned to a role; the arrays have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.GetRoleAccessForAllUsers2 lRoleID, pvarabstrUserSIDs,  
pvarabstrNames, pvarabHasAccess
```

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

<i>lRoleID</i>	Long (ByVal). The ID of the role. For a list of valid role IDs, see “Role ID Constants” on page 452 .
----------------	---

Tip: To get a role's ID from its name, use [GetRoleID](#).

<i>pvarabstrUserSIDs</i>	Variant. Returns an array containing the security identifiers of an application's users.
--------------------------	--

<i>pvarabstrNames</i>	Variant. Returns an array containing the usernames of an application's users.
-----------------------	---

<i>pvarabHasAccess</i>	Variant. Returns an array of flags that indicate whether the users are assigned to the role. An element returns TRUE if the user is assigned to the role, FALSE otherwise.
------------------------	--

GetRoleID

Returns the ID of a role, given the role's name.

Syntax

```
<HsvSecurityAccess>.GetRoleID bstrRole, plRoleID
```

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

<i>bstrRole</i>	String (ByVal). The name of the role.
-----------------	---------------------------------------

<i>plRoleID</i>	Long. Returns the role ID. For a list of constants that represent role IDs, see “Role ID Constants” on page 452 .
-----------------	---

Example

`GetRoleID` is used in the example for [AddUserToRole2](#).

GetRoleLabel

Returns the name of a role, given a role ID.

Syntax

```
<HsvSecurityAccess>.GetRoleLabel lRoleID, pbstrRole
```

Argument Description

<i>IRoleID</i>	Long (ByVal). The role ID. For a list of constants that represent role IDs, see “Role ID Constants” on page 452 .
<i>pbstrRole</i>	String. Returns the role name.

GetRulesMode

For internal use.

GetSecurityClassAccessForAllUsers

Deprecated - use [GetSecurityClassAccessForAllUsers2](#).

GetSecurityClassAccessForAllUsers2

Returns the access rights to a security class for all of an application’s users. The method returns access rights and user information in arrays that have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.GetSecurityClassAccessForAllUsers2 lSecurityClassID,  
pvarabstrUserSIDs, pvarabstrNames, pvaralRights
```

Argument Description

<i>lSecurityClassID</i>	Long (ByVal). The ID of the security class. You can get a security class’s ID from its name with GetSecurityClassID .
<i>pvarabstrUserSIDs</i>	Variant. Returns an array containing the users’ security identifiers.
<i>pvarabstrNames</i>	Variant. Returns an array containing the users’ usernames.
<i>pvaralRights</i>	Variant. Returns an array containing the users’ access rights to the security class. Valid values are represented by the enumeration described in “Access Rights Constants” on page 451 .

GetSecurityClassID

Returns the ID of a security class, given the security class’s name.

Syntax

```
<HsvSecurityAccess>.GetSecurityClassID bstrSecurityClass, plSecurityClassID
```

Argument Description

<i>bstrSecurityClass</i>	String (ByVal). The name of the security class for which the ID is to be returned.
--------------------------	--

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

Note: If the security class does not exist, error number 8004F003 (hexadecimal) is thrown.

plSecurityClassID Long. Returns the ID of the security class.

Example

GetSecurityClassID is used in the example for [SetSecurityClassLabel](#).

GetSecurityClassLabel

Returns the name of a security class, given a security class ID.

Syntax

```
<HsvSecurityAccess>.GetSecurityClassLabel lSecurityClassID,  
pbstrSecurityClass
```

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

lSecurityClassID Long (ByVal). The ID of the security class.

pbstrSecurityClass String. Returns the name of the security class.

GetSecurityClassRightsForConnectedUser

Indicates the access rights that the connected user has to a security class.

Syntax

```
<HsvSecurityAccess>.GetSecurityClassRightsForConnectedUser  
lSecurityClassID, plAccessRights
```

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

lSecurityClassID Long (ByVal). The ID of the security class. You can get this ID with [GetSecurityClassID](#).

plAccessRights Long. Returns a value that indicates the connected user's access rights to the security class. For a list of constants that represent the valid return values, see ["Access Rights Constants" on page 451](#).

Example

This example tests whether the connected user has All access rights for the Asia security class. GetSecurityClassID assigns the security class ID to the `lSecID` variable, which is passed to GetSecurityClassRightsForConnectedUser. GetSecurityClassRightsForConnectedUser's return value is then tested by the `If` statement: if the user has All access rights, any code placed within the `If` structure would be executed.

```
Dim lSecID As Long, lRights As Long
```

```

m_cSecurity.GetSecurityClassID "Asia", lSecID
m_cSecurity.GetSecurityClassRightsForConnectedUser lSecID, _
lRights
If lRights = HFM_ACCESS_RIGHTS_ALL Then
...
End If

```

GetTaskAccessForConnectedUserFromList

Indicates whether the connected user is allowed to perform one or more tasks. The IDs of the tasks for which you want to return the user's access rights are passed as an array.

Tip: To determine whether the connected user is allowed to perform one specific task, you can use [IsConnectedUserAllowedToPerformTask](#).

Syntax

```

<HsvSecurityAccess>.GetTaskAccessForConnectedUserFromList varalTaskIDs,
pvarabHasAccess

```

Argument	Description
<i>varalTaskIDs</i>	Long array (ByVal). The array of task IDs. Use the IDs listed in "Task Constants" on page 452 .
<i>pvarabHasAccess</i>	Variant array. Indicates whether the connected user has rights to the tasks passed in the <i>varalTaskIDs</i> argument; the two arrays have a one-to-one correspondence. The array is returned as a Boolean subtype: an element is TRUE if the user has rights to the task, FALSE otherwise.

Example

The following example tests whether the connected user has rights to the Create Application, Define Application Profile, and Delete Application tasks. For each of these tasks, if the user has rights, any code placed within the `If` structure would be executed.

```

Dim laTaskIDs(2) As Long, vaRights
laTaskIDs(0) = HFM_TASK_APPLICATION_CREATE_APPLICATION
laTaskIDs(1) = HFM_TASK_APPLICATION_DEFINE_APPLICATION_PROFILE
laTaskIDs(2) = HFM_TASK_APPLICATION_DELETE_APPLICATION
'g_cSecurity is an HsvSecurityAccess object reference
g_cSecurity.GetTaskAccessForConnectedUserFromList laTaskIDs, _
vaRights
For i = 0 To UBound(laTaskIDs)
If vaRights(i) = True Then
...
End If
Next i

```

GetUserAccessForAllRoles

Deprecated - use [GetUserAccessForAllRoles2](#).

GetUserAccessForAllRoles2

Indicates whether a user is assigned to the application's roles.

`GetUserAccessForAllRoles2` returns arrays that enumerate the roles and whether the user is assigned to them; the arrays have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.GetUserAccessForAllRoles2 bstrUserSID, pvarabstrRoles, pvarabHasAccess
```

Argument	Description
<i>bstrUserSID</i>	String (ByVal). The user's security identifier.
<i>pvarabstrRoles</i>	Variant. Returns an array containing the names of the application's roles.
<i>pvarabHasAccess</i>	Variant. Returns an array of flags that indicate whether the user is assigned to the roles. An element returns TRUE if the user is assigned to the role, FALSE otherwise.

GetUserAccessForAllSecurityClasses

Deprecated - use [GetUserAccessForAllSecurityClasses2](#).

GetUserAccessForAllSecurityClasses2

Returns the level of access rights that a user has for each security class in an application. The method returns access rights and security class information in arrays that have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.GetUserAccessForAllSecurityClasses2 bstrUserID, pvaralSecurityClassIDs, pvarabstrSecurityClasses, pvaralRights
```

Argument	Description
<i>bstrUserID</i>	String (ByVal). The user's security identifier.
<i>pvaralSecurityClassIDs</i>	Variant. Returns the IDs of the security classes.
<i>pvarabstrSecurityClasses</i>	Variant. Returns the names of the security classes.
<i>pvaralRights</i>	Variant. Returns an array containing the users' access rights to the security classes. Valid values are represented by the enumeration described in "Access Rights Constants" on page 451.

GetUserID

Deprecated - use [GetUserSID](#).

GetUserIDFromSID

This method is not supported as of Release 4.1.

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

```
<HsvSecurityAccess>.GetUserInfoFromUniqueID2 bstrSSOToken,  
bstrUniqueUserID, pvarbstrUserName, pbastrFullUserName, pvarbstrFirstName,  
pvarbstrLastName, pvarbstrDesc, pvarbstrEMail
```

Argument	Description
<i>bstrSSOToken</i>	String (ByVal). The user's external authentication token. Tip: You can get the token with the HsxClient method GetLogonInfoSSO .
<i>bstrUniqueUserID</i>	String (ByVal). The user's security identifier. Tip: You can get a security identifier with GetUserSID .
<i>pvarbstrUserName</i>	Variant. Returns the username.
<i>pbastrFullUserName</i>	Variant. Returns the user's fully qualified domain name.
<i>pvarbstrFirstName</i>	Variant. Returns the user's first name.
<i>pvarbstrLastName</i>	Variant. Returns the user's last name.
<i>pvarbstrDesc</i>	Variant. Returns the user's description.
<i>pvarbstrEMail</i>	Variant. Returns the user's E-mail address.

Example

The following function returns the fully qualified domain name of the connected user.

```
Function getFQDN() As String  
Dim cSecurity As HsvSecurityAccess, sToken As String, sSID As String  
Dim sDomain As String, sUsername As String, vName As Variant  
Dim vFQDN As Variant, vFirst As Variant, vLast As Variant  
Dim vDesc As Variant, vEmail As Variant  
'g_cSession is an HsvSession object reference  
Set cSecurity = g_cSession.Security  
'g_cClient is an HsxClient object reference  
sToken = g_cClient.GetLogonInfoSSO(sDomain, sUsername)
```

```
cSecurity.GetUserSID sUsername, sSid
cSecurity.GetUserInfoFromUniqueID2 sToken, sSid, vName, vFQDN, _
    vFirst, vLast, vDesc, vEmail
getFQDN = vFQDN
End Function
```

GetUserName

Deprecated - use [GetUserName2](#).

GetUserName2

Returns a username, given a security identifier.

Syntax

```
<HsvSecurityAccess>.GetUserName2 bstrUserSID, pbstrName
```

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

<i>bstrUserSID</i>	String (ByVal). The security identifier.
--------------------	--

<i>pbstrName</i>	String. Returns the username.
------------------	-------------------------------

GetUserNameFromSID

Returns the username of a user, given the user's security identifier (SID).

Syntax

```
<HsvSecurityAccess>.GetUserNameFromSID bstrSID, pbstrName
```

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

<i>bstrSID</i>	String (ByVal). The security identifier.
----------------	--

<i>pbstrName</i>	String. Returns the username
------------------	------------------------------

GetUserSID

Returns the security identifier for a given user.

Syntax

```
<HsvSecurityAccess>.GetUserSID bstrName, pbstrUserSID
```

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

<i>bstrName</i>	String (ByVal). The username of the user.
-----------------	---

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

<code>pbstrUserSID</code>	String. Returns the security identifier.
---------------------------	--

Example

`GetUserSID` is used in the example for [AddApplicationAdministrator2](#).

InsertDefaultSecurityClass

For internal use.

InsertDefaultSecurityClassWithAccessCode

For internal use.

IsApplicationAdministrator

Indicates whether the connected user is assigned to the Application Administrator role.

Syntax

```
<HsvSecurityAccess>.IsApplicationAdministrator pbIsAdmin
```

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

<code>pbIsAdmin</code>	Boolean. Returns TRUE if the connected user is assigned to the Application Administrator role, FALSE otherwise.
------------------------	---

IsClassicHFMAApplication

Indicates whether an application is a Classic application.

Syntax

```
<HsvSecurityAccess>.IsClassicHFMAApplication()
```

Return Value

Boolean. Returns TRUE for a Classic application, FALSE for a Performance Management Architect application.

IsConnectedUserAllowedToPerformTask

Indicates whether the connected user has rights to a task.

Tip: To determine whether the connected user is allowed to perform multiple tasks, use `GetTaskAccessForConnectedUserFromList`. For details, see [“GetTaskAccessForConnectedUserFromList” on page 474](#).

Syntax

```
<HsvSecurityAccess>.IsConnectedUserAllowedToPerformTask lTaskID,  
pbHasAccess
```

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

<code>lTaskID</code>	Long (ByVal). The ID of the task. For a list of valid task IDs, see “Task Constants” on page 452 .
----------------------	--

<code>pbHasAccess</code>	Boolean. Returns TRUE if the connected user has rights to the task, FALSE otherwise.
--------------------------	--

Example

`IsConnectedUserAllowedToPerformTask` is used in the [Example](#) for `HsvProcessFlow.Promote`.

IsConnectedUserInRole

Indicates whether the connected user is assigned to a role.

Syntax

```
<HsvSecurityAccess>.IsConnectedUserInRole lRoleID, pbUserIsInRole
```

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

<code>lRoleID</code>	Long (ByVal). The ID of the role to be tested. For a list of valid role IDs, see “Role ID Constants” on page 452 .
----------------------	--

<code>pbUserIsInRole</code>	Boolean. Returns TRUE if the connected user is assigned to the role, FALSE otherwise.
-----------------------------	---

Example

This example tests whether the connected user is assigned to the Load System role. If `IsConnectedUserInRole` returns TRUE, any code placed within the `If` structure would be executed.

```
Dim lRoleID As Long, bInRole As Boolean  
'g_cSecurity is an HsvSecurityAccess object reference  
g_cSecurity.GetRoleID "Load System", lRoleID  
g_cSecurity.IsConnectedUserInRole lRoleID, bInRole  
If bInRole = True Then  
    ...  
End If
```

IsValidWindowsUser

This method is not supported as of Release 4.1.

RemoveApplicationAdministrator

Deprecated - use [RemoveApplicationAdministrator2](#).

RemoveApplicationAdministrator2

Removes a user from the Application Administrator role.

Syntax

```
<HsvSecurityAccess>.RemoveApplicationAdministrator2 bstrUserSID
```

Argument Description

bstrUserSID String (ByVal). The security identifier of the user to be removed.

Example

The following snippet removes the Jsmith username from the Application Administrator role. EnumApplicationAdministrators2 gets the security identifiers of all the application Administrators. The loop tests whether Jsmith's security identifier is in the array of Administrators' security identifiers; if so, Jsmith is removed from the Application Administrator role.

```
Dim cSecurity As HsvSecurityAccess, sSID As String, vaUserIDs, vaNames
'g_cSession is an HsvSession object reference
Set cSecurity = g_cSession.Security
cSecurity.GetUserSID "myserver\Jsmith", sSID
cSecurity.EnumApplicationAdministrators2 vaUserIDs, vaNames
For i = LBound(vaUserIDs) To UBound(vaUserIDs)
    If vaUserIDs(i) = sSID Then
        cSecurity.RemoveApplicationAdministrator2 sSID
    End If
Next i
```

RemoveUser

Deprecated - use [RemoveUser2](#).

RemoveUser2

Removes a user from an application.

Syntax

```
<HsvSecurityAccess>.RemoveUser2 bstrUserSID
```


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

<i>bstrUserSID</i>	String (ByVal). The security identifier of the user to remove.
--------------------	--

RemoveUserFromRole

Deprecated - use [RemoveUserFromRole2](#).

RemoveUserFromRole2

Removes a user from a role.

Syntax

```
<HsvSecurityAccess>.RemoveUserFromRole2 lRoleID, bstrUserSID
```

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

<i>lRoleID</i>	Long (ByVal). The ID of the role from which to remove the user. For a list of valid role IDs, see "Role ID Constants" on page 452 .
----------------	---

Tip: To get a role's ID from its name, use [GetRoleID](#).

<i>bstrUserSID</i>	String (ByVal). The user's security identifier.
--------------------	---

Example

The following subroutine removes a user from a given role. The role name is passed to the subroutine, which uses `GetRoleID` to obtain the role ID.

```
Sub RemoveFromRole(sId As String, sRole As String)
Dim cSecurity As HsvSecurityAccess, lRoleID As Long
'g_cSession is an HsvSession object reference
Set cSecurity = g_cSession.Security
cSecurity.GetRoleID sRole, lRoleID
cSecurity.RemoveUserFromRole2 lRoleID, sId
End Sub
```

RenameSecurityClass

For internal use.

SetManySecurityClassRightsForUser

Deprecated - use [SetManySecurityClassRightsForUser2](#).

SetManySecurityClassRightsForUser2

Sets a user's access and E-mail alerting rights for one or more security classes. Security and access right IDs are passed in arrays that have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.SetManySecurityClassRightsForUser2 bstrUserSID,  
varalSecurityClassIDs, varalRights, varalEmailAlerting
```

Argument	Description
<i>bstrUserSID</i>	String (ByVal). The user's security identifier.
<i>varalSecurityClassIDs</i>	Long array (ByVal). The IDs of the security classes. You can get a security class's ID by passing its name to GetSecurityClassID .
<i>varalRights</i>	Long array (ByVal). The IDs of the access rights to be assigned. Valid values are represented by the enumeration described in " Access Rights Constants " on page 451.
<i>varalEmailAlerting</i>	Long array (ByVal). An array of bitmasks that represent the E-mail alerting rights to assign. Valid values for the bits are represented by the enumeration described in " E-mail Alerting Constants " on page 452.

SetRolesForUser

Specifies one or more roles for a given user. This method overwrites any previously assigned roles for the user.

Syntax

```
<HsvSecurityAccess>.SetRolesForUser bstrUserSid, varalRoleIds
```

Argument	Description
<i>bstrUserSid</i>	String (ByVal). The user's security identifier.
<i>varalRoleIds</i>	Long array (ByVal). The IDs of the roles. For a list of valid role IDs, see " Role ID Constants " on page 452.

Example

The following example assigns the specified roles to a given username. The user's security identifier is obtained with [GetUserSID](#).

```
Sub assignRolesUsername(sName As String, laRoles() As Long)  
Dim cSecurity As HsvSecurityAccess, sID As String  
'g_cSession is an HsvSystemInfo object reference  
Set cSecurity = g_cSession.Security  
cSecurity.GetUserSID sName, sID  
cSecurity.SetRolesForUser sID, laRoles  
End Sub
```

SetRulesMode

For internal use.

SetSecurityClassLabel

Changes the name of a security class in a Classic application.

Caution! The method will fail if executed against an application created with Performance Management Architect.

Syntax

```
<HsvSecurityAccess>.SetSecurityClassLabel lSecurityClassID,  
bstrSecurityClass
```

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

<i>lSecurityClassID</i>	Long (ByVal). The ID of the security class. You can get this ID by passing the security class name to GetSecurityClassID .
-------------------------	--

<i>bstrSecurityClass</i>	String (ByVal). The new name for the security class.
--------------------------	--

Example

The following method renames a given security class.

```
Sub renameSecClass(sName As String, sNewName As String)  
Dim lSecClassID As Long, cSecurity As HsvSecurityAccess  
'g_cSession is an HsvSession object reference  
Set cSecurity = g_cSession.Security  
cSecurity.GetSecurityClassID sName, lSecClassID  
cSecurity.SetSecurityClassLabel lSecClassID, sNewName  
End Sub
```

SetSecurityClassLabelWithAccessCode

For internal use.

SetSecurityClassRightsForManyUsers

Deprecated - use [SetSecurityClassRightsForManyUsers2](#).

SetSecurityClassRightsForManyUsers2

Sets one or more users' access and E-mail alerting rights to a given security class. Users' security identifiers, access right IDs, and E-mail alerting flags are passed in arrays that have a one-to-one correspondence.

Syntax

```
<HsvSecurityAccess>.SetSecurityClassRightsForManyUsers2 lSecurityClassID,  
varabstrUserSIDs, varalRights, varalEmailAlerting
```

Argument	Description
<i>lSecurityClassID</i>	Long (ByVal). Long (ByVal). The ID of the security class. You can get this ID by passing the security class name to GetSecurityClassID .
<i>varabstrUserSIDs</i>	String array (ByVal). The users' security identifiers.
<i>varalRights</i>	Long array (ByVal). The IDs of the access rights to be assigned. Valid values are represented by the enumeration described in " Access Rights Constants " on page 451.
<i>varalEmailAlerting</i>	Long array (ByVal). An array of bitmasks that represent the E-mail alerting rights to assign. Valid values for the bits are represented by the enumeration described in " E-mail Alerting Constants " on page 452.

SetSecurityClassRightsForUser

Deprecated - use [SetSecurityClassRightsForUser2](#).

SetSecurityClassRightsForUser2

Sets a user's access and E-mail alerting rights for a security class.

Syntax

```
<HsvSecurityAccess>.SetSecurityClassRightsForUser2 lSecurityClassID,  
bstrUserSID, lRights, lEmailAlerting
```

Argument	Description
<i>lSecurityClassID</i>	Long (ByVal). The ID of the security class. You can get this ID by passing the security class name to GetSecurityClassID .
<i>bstrUserSID</i>	String (ByVal). The user's security identifier.
<i>lRights</i>	Long (ByVal). The ID of the access rights to be assigned. Valid values are represented by the enumeration described in " Access Rights Constants " on page 451.
<i>lEmailAlerting</i>	Long (ByVal). An array of bitmasks that represent the E-mail alerting rights to assign. Valid values for the bits are represented by the enumeration described in " E-mail Alerting Constants " on page 452.

SetUserClassAccess

Sets security class access and E-mail alerting rights for one or more users.

Syntax

```
<HsvSecurityAccess>.SetUserClassAccess varabstrUserSIDS,  
varabstrSecClasses, var2DalRights, var2DalAlertable
```

Argument	Description
<code>varabstrUserSIDS</code>	String array (ByVal). The users' security identifiers.
<code>varabstrSecClasses</code>	String array (ByVal). The names of the security classes for which rights are being set.
<code>var2DalRights</code>	<p>Long array (ByVal). A two-dimensional array containing the IDs of the users' access rights to the security classes. The first dimension has a one-to-one correspondence with the <code>varabstrUserSIDS</code> argument's array, and the second dimension has a one-to-one correspondence with the <code>varabstrSecClasses</code> argument's array.</p> <p>IDs of access rights are represented by the enumeration described in "Access Rights Constants" on page 451.</p>
<code>var2DalAlertable</code>	<p>Long array (ByVal). A two-dimensional array containing bitmasks that represent the users' E-mail alerting rights for the security classes. The first dimension has a one-to-one correspondence with the <code>varabstrUserSIDS</code> argument's array, and the second dimension has a one-to-one correspondence with the <code>varabstrSecClasses</code> argument's array.</p> <p>Valid values for the bits are represented by the enumeration described in "E-mail Alerting Constants" on page 452.</p>

TakeOwnership

This method is not supported as of Release 4.1.

IHsvDataSecurity Interface Methods

The IHsvDataSecurity interface's methods return information about the connected user's rights to process units, and also provides a method that refreshes access rights on application servers. These methods are summarized in [Table 27 on page 84](#), and are described in detail in the following topics.

To assign IHsvDataSecurity interface object references, use the [Security](#) property of the HsvSession object.

GetCellLevelAccessRights

Returns the access rights that the connected user has to a cell.

Syntax

```
<IHsvDataSecurity>.GetCellLevelAccessRights lScenario, lYear, lPeriod,  
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,  
lCustom4, plAccessRights
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the cell's child Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the cell's parent Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pAccessRights</i>	Long. Returns a value that identifies the connected user's access rights for the cell. Constants that represent the valid return values are listed in "Access Rights Constants" on page 451 .

GetProcessUnitAccessRights

Returns the access rights that the connected user has to a process unit.

Syntax

```
<IHsvDataSecurity>.GetProcessUnitAccessRights lScenario, lYear, lPeriod, lEntity, lParent, lValue, pAccessRights
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>pAccessRights</i>	Long. Returns the number that identifies the connected user's access rights for the process unit. Constants that represent the valid return values are listed in "Access Rights Constants" on page 451 .

Example

`GetProcessUnitAccessRights` is used in the [Example](#) for `HsvProcessFlow.Promote`.

GetProcessUnitAccessRightsAndState

Returns the following information for a process unit:

- The access rights that the connected user has for the process unit.
- The process unit's current level.

Syntax

```
<IHsvDataSecurity>.GetProcessUnitAccessRightsAndState lScenario, lYear, lPeriod, lEntity, lParent, lValue, plAccessRights, psProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>plAccessRights</i>	Long. Returns a value that identifies the connected user's access rights for the process unit. The valid return values are listed in " Access Rights Constants " on page 451.
<i>psProcessState</i>	Integer. Returns a value that identifies the process unit's level. The valid return values are listed in " Process Management Review Level Constants " on page 853.

Example

`GetProcessUnitAccessRightsAndState` is used in the example for `HsvProcessFlow.Start`.

GetProcessUnitAccessRightsAndStateEx

Returns the following information for a submission phase, given either the member IDs of a cell in the phase or the phase ID:

- The connected user's access rights to the submission phase.
- The review level of the submission phase.

You must pass one of the following:

- If you use a phase ID, pass the `HFMConstants.MEMBERNOTUSED` to the Account, Intercompany Partner, and Custom dimension parameters.

- If you use member IDs for the above-mentioned parameters, pass `MEMBERNOTUSED` to the phase ID parameter.

Syntax

```
<IHsvDataSecurity>.GetProcessUnitAccessRightsAndStateEx lScenario, lYear,
lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2,
lCustom3, lCustom4, lPhaseID, plAccessRights, psProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Account dimension member. ● <code>MEMBERNOTUSED</code>
<i>lICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Intercompany Partner dimension member. ● <code>MEMBERNOTUSED</code>
<i>lCustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom1 dimension member. ● <code>MEMBERNOTUSED</code>
<i>lCustom2</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom2 dimension member. ● <code>MEMBERNOTUSED</code>
<i>lCustom3</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom3 dimension member. ● <code>MEMBERNOTUSED</code>
<i>lCustom4</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom4 dimension member. ● <code>MEMBERNOTUSED</code>
<i>lPhaseID</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The phase ID. <p>Note: You can obtain a phase ID with GetGroupPhaseFromCell.</p> ● <code>MEMBERNOTUSED</code>

Argument	Description
<i>plAccessRights</i>	Long. Returns the number that identifies the connected user's access rights for the submission phase. The valid return values are listed in Table 68, "Access Rights Constants," on page 451.
<i>psProcessState</i>	Integer. Returns a value that identifies the submission phase's review level. The valid return values are listed in Table 136, "CEnumProcessFlowStates Enumeration," on page 853.

Example

`GetProcessUnitAccessRightsAndStateEx` is used in the example for [PhasedSubmissionStart](#).

GetProcessUnitAccessRightsEx

Returns the access rights that the connected user has to a submission phase, given either the member IDs 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 HFMConstants `MEMBERNOTUSED` to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass `MEMBERNOTUSED` to the phase ID parameter.

Syntax

```
<IHsvDataSecurity>.GetProcessUnitAccessRightsEx lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, lPhaseID, plAccessRights
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Account dimension member. ● <code>MEMBERNOTUSED</code>
<i>lICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Intercompany Partner dimension member. ● <code>MEMBERNOTUSED</code>
<i>lCustom1</i>	Long (ByVal). Pass one of the following:

Argument	Description
	<ul style="list-style-type: none"> ● The member ID of the cell's Custom1 dimension member. ● MEMBERNOTUSED
<i>ICustom2</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The member ID of the cell's Custom2 dimension member. ● MEMBERNOTUSED
<i>ICustom3</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The member ID of the cell's Custom3 dimension member. ● MEMBERNOTUSED
<i>ICustom4</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The member ID of the cell's Custom4 dimension member. ● MEMBERNOTUSED
<i>IPhaseID</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The phase ID. <p>Note: You can obtain a phase ID with GetGroupPhaseFromCell.</p> <ul style="list-style-type: none"> ● MEMBERNOTUSED
<i>pAccessRights</i>	<p>Long. Returns the number that identifies the connected user's access rights for the submission phase. The valid return values are listed in Table 68, "Access Rights Constants," on page 451.</p>

RefreshAccessRightsCache

Refreshes the access rights on the application server. Access rights are cached into application servers. In systems with multiple application servers, use `RefreshAccessRightsCache` to ensure that the application server to which the client is connected has the most recently-defined access rights.

Syntax

```
<IHsvDataSecurity>.RefreshAccessRightsCache
```

This chapter describes the members of the HsvSystemInfo type library. The methods of this type library are used to get and set various types of system information such as application directories, server names, and task audit histories.

To use the HsvSystemInfo type library, you must reference `HsvSystemInfo.dll` in your project. The HsvSystemInfo type library contains one object—the HsvSystemInfo object.

HsvSystemInfo Object Methods

The HsvSystemInfo object's methods are summarized in [Table 28 on page 85](#), and are described in detail in the following topics.

Note: Assign HsvSystemInfo object references with the `SystemInfo` property of the HsvSession object. For an example, see [“HsvSystemInfo Type Library Overview” on page 84](#).

AddRefToHsxServer

For internal use.

Syntax

```
<HsvSystemInfo>.AddRefToHsxServer plRefCount
```

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

<code>plRefCount</code>	Long.
-------------------------	-------

AddTaskToAudit

Adds a given task for the user to the audit log. To avoid redundant audit records, use this method only for operations that do not call methods in the object model, as the system updates the audit log when the object model methods are called.

Syntax

```
<HsvSystemInfo>.AddTaskToAudit lActivityCode, dStartTime, dEndTime,  
bstrDescription
```

Argument	Description
<i>lActivityCode</i>	Long (ByVal). The ID of the activity being added. Valid values are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 870 . However, Hyperion recommends that you pass only the USERACTIVITYCODE_EXTERNAL constant, which represents custom activities.
<i>dStartTime</i>	Double (ByVal). The timestamp of the activity’s start time. Pass a Double that can be cast to the Date format.
<i>dEndTime</i>	Double (ByVal). The timestamp of the activity’s end time. Pass a Double that can be cast to the Date format.
<i>bstrDescription</i>	String (ByVal). A description of the activity.

AddTaskToRunningTasks

For internal use.

AddTaskToRunningTasksAndUpdatePOV

For internal use.

CheckAccess

Indicates whether the current user has access to the application. For example, this method returns FALSE if the administrator has disabled connections for the user.

Syntax

```
<HsvSystemInfo>.CheckAccess pvbHasAccess
```

Argument	Description
<i>pvbHasAccess</i>	Boolean. Indicates whether the user has access to the application represented by the HsvSession object from which the HsvSystemInfo object reference was obtained. Returns TRUE if the user has access, FALSE otherwise.

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

```
<HsvSystemInfo>.ClearAuditTasks dStartTime, dEndTime, vbAllServers,  
bstrServer, vbAllUsers, lActivityUserID
```

Argument	Description
<i>dStartTime</i>	Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.
<i>dEndTime</i>	Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.
<i>vbAllServers</i>	Boolean (ByVal). Specifies 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.
<i>bstrServer</i>	String (ByVal). 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.
<i>vbAllUsers</i>	Boolean (ByVal). Specifies 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.
<i>lActivityUserID</i>	Long (ByVal). 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. To get a user's activity user ID, use GetActivityUserID .

Example

The following subroutine deletes the audit history for a given user. The subroutine takes a username; the activity user ID required by `ClearAuditTasks` is obtained by passing the username to [GetActivityUserID](#).

```
Sub ClearUserTaskAudit(sUserName As String)  
Dim cSysInfo As HsvSystemInfo, lUserActID As Long  
'm_cHsvSession is an HsvSession object reference  
Set cSysInfo = m_cHsvSession.SystemInfo  
cSysInfo.GetActivityUserID sUserName, lUserActID  
cSysInfo.ClearAuditTasks 0, Cdbl(Now()), True, "", _  
    False, lUserActID  
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](#).

Syntax

```
<HsvSystemInfo>.ClearAuditTasks2 dStartTime, dEndTime, vbAllServers,  
bstrServer, vbAllUsers, lActivityUserID, vbAllTasks, lActivityTaskID
```

Argument	Description
<i>dStartTime</i>	Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.
<i>dEndTime</i>	Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.
<i>vbAllServers</i>	Boolean (ByVal). Specifies 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.
<i>bstrServer</i>	String (ByVal). 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.
<i>vbAllUsers</i>	Boolean (ByVal). Specifies 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.
<i>lActivityUserID</i>	Long (ByVal). 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. To get a user's activity user ID, use GetActivityUserID .
<i>vbAllTasks</i>	Boolean (ByVal). 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.
<i>lActivityTaskID</i>	Long (ByVal). 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 <code>HFMConstants</code> type library constants listed in "User Activity Constants" on page 870 .

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 As Long)  
Dim cSysInfo As HsvSystemInfo  
'm_cHsvSession is an HsvSession object reference  
Set cSysInfo = m_cHsvSession.SystemInfo  
cSysInfo.ClearAuditTasks2 0, CDb1(Now()), True, "", _  
    True, 0, False, lTaskId  
End Sub
```

ClearRunningTask

For internal use.

DeleteUserParameter

Deletes a user parameter that has been created with `SetUserParameter`.

Syntax

```
<HsvSystemInfo>.DeleteUserParameter bstrKey
```

Argument Description

bstrKey String (ByVal). The key that identifies the parameter to be deleted.

Example

This example deletes a user parameter named “Projects.”

```
Dim cHsvSystemInfo As HsvSystemInfo
'm_cHsvSession is an HsvSession object reference
Set cHsvSystemInfo = m_cHsvSession.SystemInfo
cHsvSystemInfo.DeleteUserParameter "Projects"
```

DisableNewConnections

Disables new Financial Management connections for the specified user and application server criteria.

Tip: To disable new connections for a given cluster or application, use `HsxClient.DisableNewConnections`.

To enable new connections, use `EnableNewConnections`.

Syntax

```
<HsvSystemInfo>.DisableNewConnections vbAllServers, bstrServer,
vbAllUsers, lActivityUserID
```

Argument Description

vbAllServers Boolean (ByVal). Specifies whether to disable connections for all application servers. Pass TRUE for all application servers, FALSE to specify an application server with the *bstrServer* argument.

bstrServer String (ByVal). The name of the application server for which to disable connections. This argument is used only if the *vbAllServers* argument is set to FALSE.

vbAllUsers Boolean (ByVal). Specifies whether to disable connections for all users. Pass TRUE for all users, FALSE to specify a user with the *lActivityUserID* argument.

lActivityUserID Long (ByVal). The activity user ID of the user for whom to disable connections. This argument is used only if the *vbAllUsers* argument is set to FALSE.

To get a user's activity user ID, use `GetActivityUserID`.

EnableNewConnections

Enables new Financial Management connections for the specified user and application server criteria.

Tip: To enable new connections for a given cluster or application, use `HsxClient.EnableNewConnections`.

To disable new connections, use [DisableNewConnections](#).

Syntax

```
<HsvSystemInfo>.EnableNewConnections vbAllServers, bstrServer, vbAllUsers, lActivityUserID
```

Argument	Description
<i>vbAllServers</i>	Boolean (ByVal). Specifies whether to enable connections for all application servers. Pass TRUE for all application servers, FALSE to specify an application server with the <i>bstrServer</i> argument.
<i>bstrServer</i>	String (ByVal). The name of the application server for which to enable connections. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether to enable connections for all users. Pass TRUE for all users, FALSE to specify a user with the <i>lActivityUserID</i> argument.
<i>lActivityUserID</i>	Long (ByVal). 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. To get a user's activity user ID, use GetActivityUserID .

EnumActivityServers

Returns the names of the application servers for which there are task audit and data audit records.

Syntax

```
<HsvSystemInfo>.EnumActivityServers pvarabstrActivityServerNames
```

Argument	Description
<i>pvarabstrActivityServerNames</i>	Variant. Returns an array containing the names of the servers. The array is returned as a String subtype.

EnumActivityUsers

Returns the usernames of all users who have performed at least one activity in the application.

Syntax

```
<HsvSystemInfo>.EnumActivityUsers pvarabstrActivityUserNames
```

Argument

Description

pvarabstrActivityUserNames Variant array. Returns the usernames.

The array is returned as a String subtype.

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 *plTotalNumRecords* 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 *plTotalNumRecords* 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

```
<HsvSystemInfo>.EnumAuditTasks dStartTime, dEndTime, vbAllServers,  
bstrServer, vbAllUsers, lActivityUserID, lStartRecord, lEndRecord,  
pvaralActivityUserID, pvarabstrActivityUserName, pvaralActivitySessionID,  
pvarbstrServerName, pvaralActivityCode, pvaradStartTime, pvaradEndTime,  
pvarbstrDescription, pvarbstrModuleNames, plTotalNumRecords
```

Argument

Description

dStartTime Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.

dEndTime Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.

vbAllServers Boolean (ByVal). Specifies whether audit information for all application servers is returned. Pass TRUE for all application servers, FALSE to specify an application server with the *bstrServer* argument.

bstrServer String (ByVal). The name of the application server for which to return audit information. This argument is used only if the *vbAllServers* argument is set to FALSE.

Argument	Description
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether audit information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <i>lActivityUserID</i> argument.
<i>lActivityUserID</i>	Long (ByVal). The activity user ID of the user for whom to return audit information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. To get a user's activity user ID, use GetActivityUserID .
<i>lStartRecord</i>	Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.
<i>lEndRecord</i>	Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.
<i>pvaralActivityUserID</i>	Variant array. Returns the activity user IDs for the users who performed the tasks. The array is returned as a Long subtype.
<i>pvarabstrActivityUserName</i>	Variant array. Returns the usernames of the users who performed the tasks. The array is returned as a String subtype.
<i>pvaralActivitySessionID</i>	Variant array. Returns the internal IDs of the sessions in which the tasks occurred. The array is returned as a Long subtype.
<i>pvarbstrServerName</i>	Variant array. Returns the names of the application servers on which the tasks occurred. The array is returned as a String subtype.
<i>pvaralActivityCode</i>	Variant array. Returns 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 870 . The array is returned as a Long subtype.
<i>pvaradStartTime</i>	Variant array. Returns the starting times of the tasks. Array items are formatted as Doubles that can be cast to the Date format. The array is returned as a Double subtype.
<i>pvaradEndTime</i>	Variant array. Returns the ending times of the tasks. Array items are formatted as Doubles that can be cast to the Date format. The array is returned as a Double subtype.
<i>pvarbstrDescription</i>	Variant array. Returns the descriptions of the tasks. The array is returned as a String subtype.
<i>pvarbstrModuleNames</i>	Variant array. Returns the names of the modules for the tasks. The array is returned as a String subtype.
<i>pITotalNumRecords</i>	Long. Returns the total number of audit records in the database that meet the filtering criteria.

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 *plTotalNumRecords* 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 *plTotalNumRecords* 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

```
<HsvSystemInfo>.EnumAuditTasks2 dStartTime, dEndTime, vbAllServers,  
bstrServer, vbAllUsers, lActivityUserID, vbAllTasks, lActivityTaskID,  
lStartRecord, lEndRecord, pvaralActivityUserID, pvarabstrActivityUserName,  
pvaralActivitySessionID, pvarbstrServerName, pvaralActivityCode,  
pvaradStartTime, pvaradEndTime, pvarbstrDescription, pvarbstrModuleNames,  
plTotalNumRecords
```

Argument	Description
<i>dStartTime</i>	Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.
<i>dEndTime</i>	Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.
<i>vbAllServers</i>	Boolean (ByVal). Specifies 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.
<i>bstrServer</i>	String (ByVal). 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.
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether audit information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <i>lActivityUserID</i> argument.
<i>lActivityUserID</i>	Long (ByVal). The activity user ID of the user for whom to return audit information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. To get a user's activity user ID, use GetActivityUserID .
<i>vbAllTasks</i>	Boolean (ByVal). 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.

Argument	Description
<i>IActivityTaskID</i>	<p>Long (ByVal). 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 870.</p>
<i>IStartRecord</i>	<p>Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.</p>
<i>IEndRecord</i>	<p>Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.</p>
<i>pvaralActivityUserID</i>	<p>Variant array. Returns the activity user IDs for the users who performed the tasks.</p> <p>The array is returned as a Long subtype.</p>
<i>pvarabstrActivityUserName</i>	<p>Variant array. Returns the usernames of the users who performed the tasks.</p> <p>The array is returned as a String subtype.</p>
<i>pvaralActivitySessionID</i>	<p>Variant array. Returns the internal IDs of the sessions in which the tasks occurred.</p> <p>The array is returned as a Long subtype.</p>
<i>pvarbstrServerName</i>	<p>Variant array. Returns the names of the application servers on which the tasks occurred.</p> <p>The array is returned as a String subtype.</p>
<i>pvaralActivityCode</i>	<p>Variant array. Returns 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 870.</p> <p>The array is returned as a Long subtype.</p>
<i>pvaradStartTime</i>	<p>Variant array. Returns the starting times of the tasks. Array items are formatted as Doubles that can be cast to the Date format.</p> <p>The array is returned as a Double subtype.</p>
<i>pvaradEndTime</i>	<p>Variant array. Returns the ending times of the tasks. Array items are formatted as Doubles that can be cast to the Date format.</p> <p>The array is returned as a Double subtype.</p>
<i>pvarbstrDescription</i>	<p>Variant array. Returns the descriptions of the tasks.</p> <p>The array is returned as a String subtype.</p>
<i>pvarbstrModuleNames</i>	<p>Variant array. Returns the names of the modules for the tasks.</p> <p>The array is returned as a String subtype.</p>
<i>pITotalNumRecords</i>	<p>Long. Returns the total number of audit records in the database that meet the filtering criteria.</p>

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: You can return information on connections disabled for a given cluster with `HsxClient.EnumProhibitConnections`.

Syntax

```
<HsvSystemInfo>.EnumProhibitConnections pvaravbAllApps, pvarabstrAppNames,  
pvaravbAllServers, pvarabstrServerNames, pvaravbAllUsers,  
pvaralActivityUserIDs, pvaralActivityUserNames
```

Argument	Description
<i>pvaravbAllApps</i>	<p>Variant array. 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>pvarabstrAppNames</i> argument's array item are disabled. <p>The array is returned as a Long subtype.</p>
<i>pvarabstrAppNames</i>	<p>Variant array. Returns the application names for which connections are disabled. Application names are returned only when the corresponding <i>pvaravbAllApps</i> argument's array item contains 0.</p> <p>Note: If the corresponding <i>pvaravbAllApps</i> argument's array item contains -1, this array item contains the string "AllApps".</p> <p>The array is returned as a String subtype.</p>
<i>pvaravbAllServers</i>	<p>Variant array. 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>pvarabstrServerNames</i> argument's array item are disabled. <p>The array is returned as a Long subtype.</p>
<i>pvarabstrServerNames</i>	<p>Variant array. Returns the application server names for which connections are disabled. Application server names are returned only when the corresponding <i>pvaravbAllServers</i> argument's array item contains 0.</p> <p>Note: If the corresponding <i>pvaravbAllServers</i> argument's array item contains -1, this array item contains the string "AllServers".</p> <p>The array is returned as a String subtype.</p>
<i>pvaravbAllUsers</i>	<p>Variant array. 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.

Argument	Description
	<ul style="list-style-type: none"> 0 indicates that only connections for the user represented by the corresponding <i>pvaralActivityUserIDs</i> and <i>pvaralActivityUserNames</i> arguments' array items are disabled. <p>The array is returned as a Long subtype.</p>
<i>pvaralActivityUserIDs</i>	<p>Variant array. Returns the activity user IDs of the users for whom connections are disabled. Valid IDs are returned only when the corresponding <i>pvaravbAllUsers</i> argument's array item contains 0.</p> <p>Note: If the corresponding <i>pvaravbAllUsers</i> argument's array item contains -1, this array item contains -1.</p> <p>The array is returned as a Long subtype.</p>
<i>pvaralActivityUserNames</i>	<p>Variant array. Returns the usernames of the users for whom connections are disabled. Usernames are returned only when the corresponding <i>pvaravbAllUsers</i> argument's array item contains 0.</p> <p>Note: If the corresponding <i>pvaravbAllUsers</i> argument's array item contains -1, this array item contains an empty string.</p> <p>The array is returned as a String subtype.</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.

Returns information about the running tasks that meet the selection criteria; the information returned includes an array of flags that indicate whether the tasks are running. 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 this method. For example, a new task could be added.

Syntax

```
<HsvSystemInfo>.EnumRunningTasks vbAllTaskTypes, lTaskType, vbAllUsers, lUserID, vbAllServers, bstrServerName, vbAllSessions, vbAllStatus, lStatus, lStartRecord, lEndRecord, pvaralTaskIDs, pvaralTaskTypes, pvaralTaskProgress, pvaralTaskStatus, pvarabstrUserNames, pvarabstrServerNames, pvaradStartTimes, pvaradStartRunningTimes, pvaradLastUpdateTimes, pvarabstrDescriptions, pvarabstrLogFiles, plTotalRecords
```

Argument	Description
<i>vbAllTaskTypes</i>	Boolean (ByVal). 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.
<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
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether to filter by user. Pass TRUE for all users, FALSE to filter by the user specified with the <i>lUserID</i> argument.
<i>lUserID</i>	<p>Long (ByVal). The activity user ID of the user for whom to return task information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE.</p> <p>To get a user's activity user ID, use GetActivityUserID.</p>
<i>vbAllServers</i>	Boolean (ByVal). 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.
<i>bstrServerName</i>	String (ByVal). 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.
<i>vbAllSessions</i>	Boolean (ByVal). Specifies whether to return information for all user sessions or only the session for the connected user. Pass TRUE for all sessions, FALSE otherwise.

Argument	Description
<i>vbAllStatus</i>	Boolean (ByVal). 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.
<i>lStatus</i>	Long (ByVal). 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 872 .
<i>lStartRecord</i>	Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.
<i>lEndRecord</i>	Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.
<i>pvaralTaskIDs</i>	Variant. Returns an array containing the task IDs that identify the running tasks. The array is returned as a Long subtype.
<i>pvaralTaskTypes</i>	Variant. Returns an array containing the tasks’ types. Valid values are represented by the constants listed as valid values for the <i>lTaskType</i> argument. The array is returned as a Long subtype.
<i>pvaralTaskProgress</i>	Variant. Returns an array containing the progress complete percentages of the running tasks. The array is returned as a Long subtype.
<i>pvaralTaskStatus</i>	Variant. Returns an array containing the tasks’ statuses. Valid values are represented by the HFMConstants type library constants listed in “Task Status Constants” on page 872 . The array is returned as a Long subtype.
<i>pvarabstrUserNames</i>	Variant. Returns an array containing the usernames of the running tasks’ users. The array is returned as a String subtype.
<i>pvarabstrServerNames</i>	Variant. Returns an array containing the names of the application servers for the running tasks. The array is returned as a String subtype.
<i>pvaradStartTimes</i>	Variant. 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. The array is returned as a Double subtype.
<i>pvaradStartRunningTimes</i>	Variant. 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. The array is returned as a Double subtype.
<i>pvaradLastUpdateTimes</i>	Variant. 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. The array is returned as a Double subtype.
<i>pvarabstrDescriptions</i>	Variant. Returns an array containing the descriptions for the tasks. The array is returned as a String subtype.
<i>pvarabstrLogFiles</i>	Variant. Returns an array containing the file names and paths of the tasks’ log files. The array is returned as a String subtype.
<i>plTotalRecords</i>	Long. Returns a count of the number of tasks returned by <code>EnumRunningTasks</code> .

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 this method. For example, a new task could be added.

Syntax

```
<HsvSystemInfo>.EnumRunningTasksEx vbAllTaskTypes, lTaskType, vbAllUsers,
lUserID, vbAllServers, bstrServerName, vbAllSessions, vbAllStatus, lStatus,
lStartRecord, lEndRecord, pvaralTaskIDs, pvaralTaskTypes,
pvaralTaskProgress, pvaralTaskStatus, pvarabstrUserNames,
pvarabstrServerNames, pvaradStartTimes, pvaradStartRunningTimes,
pvaradLastUpdateTimes, pvarabstrDescriptions, pvarbStopTaskFlags,
pvarabstrLogFiles, plTotalRecords
```

Argument	Description
<i>vbAllTaskTypes</i>	Boolean (ByVal). 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.
<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 <code>tagUSERACTIVITYCODE</code>:</p> <ul style="list-style-type: none">● <code>USERACTIVITYCODE_CONSOLIDATION</code>: consolidations● <code>USERACTIVITYCODE_DATA_LOAD</code>: data loads● <code>USERACTIVITYCODE_EA_EXPORT</code>: Extended Analytics exports● <code>USERACTIVITYCODE_IC_POSTALL</code>: post all intercompany transactions● <code>USERACTIVITYCODE_IC_UNPOSTALL</code>: unpost all intercompany transactions● <code>USERACTIVITYCODE_IC_DELETEALL</code>: delete all intercompany transactions● <code>USERACTIVITYCODE_IC_UNMATCHALL</code>: unmatched all intercompany transactions● <code>USERACTIVITYCODE_IC_AUTOMATCHBYID</code>: automatch intercompany transactions by ID● <code>USERACTIVITYCODE_IC_AUTOMATCHBYACCT</code>: automatch intercompany transactions by account● <code>USERACTIVITYCODE_IC_MATCHINGRPTBYID</code>: matching intercompany transactions by ID report

Argument	Description
	<ul style="list-style-type: none"> ● USERACTIVITYCODE_IC_MATCHINGRPTBYACCT: matching intercompany transactions by account report ● USERACTIVITYCODE_IC_TRANSACTIONRPT: intercompany transactions report
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether to filter by user. Pass TRUE for all users, FALSE to filter by the user specified with the <i>lUserID</i> argument.
<i>lUserID</i>	Long (ByVal). The activity user ID of the user for whom to return task information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. To get a user's activity user ID, use GetActivityUserID .
<i>vbAllServers</i>	Boolean (ByVal). 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.
<i>bstrServerName</i>	String (ByVal). 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.
<i>vbAllSessions</i>	Boolean (ByVal). Specifies whether to return information for all user sessions or only the session for the connected user. Pass TRUE for all sessions, FALSE otherwise.
<i>vbAllStatus</i>	Boolean (ByVal). 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.
<i>lStatus</i>	Long (ByVal). 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 872 .
<i>lStartRecord</i>	Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.
<i>lEndRecord</i>	Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.
<i>pvaralTaskIDs</i>	Variant. Returns an array containing the task IDs that identify the running tasks. The array is returned as a Long subtype.
<i>pvaralTaskTypes</i>	Variant. Returns an array containing the tasks' types. Valid values are represented by the constants listed as valid values for the <i>lTaskType</i> argument. The array is returned as a Long subtype.
<i>pvaralTaskProgress</i>	Variant. Returns an array containing the progress complete percentages of the running tasks. The array is returned as a Long subtype.
<i>pvaralTaskStatus</i>	Variant. Returns an array containing the tasks' statuses. Valid values are represented by the HFMConstants type library constants listed in "Task Status Constants" on page 872 . The array is returned as a Long subtype.
<i>pvarabstrUserNames</i>	Variant. Returns an array containing the usernames of the running tasks' users. The array is returned as a String subtype.
<i>pvarabstrServerNames</i>	Variant. Returns an array containing the names of the application servers for the running tasks. The array is returned as a String subtype.
<i>pvaradStartTimes</i>	Variant. 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.

Argument	Description
	The array is returned as a Double subtype.
<i>pvaradStartRunningTimes</i>	Variant. 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. The array is returned as a Double subtype.
<i>pvaradLastUpdateTimes</i>	Variant. 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. The array is returned as a Double subtype.
<i>pvarabstrDescriptions</i>	Variant. Returns an array containing the descriptions for the tasks. The array is returned as a String subtype.
<i>pvarbStopTaskFlags</i>	Variant. 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.
<i>pvarabstrLogFiles</i>	Variant. Returns an array containing the file names and paths of the tasks' log files. The array is returned as a String subtype.
<i>plTotalRecords</i>	Long. Returns a count of the number of tasks returned by <code>EnumRunningTasksEx</code> .

EnumRunningTasksPOV

Returns the dimension members, consolidation types, and other information for the running consolidations that meet the selection criteria. Criteria include user, application server, user session, and task status. Consolidation information is returned in arrays that have a one-to-one correspondence.

Syntax

```
<HsvSystemInfo>.EnumRunningTasksPOV lTaskType, vbAllUsers, lUserID,
vbAllServers, bstrServerName, vbAllSessions, pvaralTaskIds,
pvaralTaskProgress, pvaralTaskStatus, pvarabstrUserNames,
pvarabstrServerNames, pvaralEntity, pvaralParent, pvaralScenario,
pvaralYear, pvaralStartPeriod, pvaralEndPeriod, pvaralConsolType,
plTotalRecords
```

Argument	Description
<i>lTaskType</i>	Long (ByVal). Pass the HFMConstants type library constant <code>USERACTIVITYCODE_CONSOLIDATION</code> .
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether to filter by user. Pass TRUE for all users, FALSE to filter by the user specified with the <i>lUserID</i> argument.
<i>lUserID</i>	Long (ByVal). The activity user ID of the user for whom to return task information. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. To get a user's activity user ID, use GetActivityUserID .

Argument	Description
<i>vbAllServers</i>	Boolean (ByVal). 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.
<i>bstrServerName</i>	String (ByVal). 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.
<i>vbAllSessions</i>	Boolean (ByVal). Specifies whether to return information for all user sessions or only the session for the connected user. Pass TRUE for all sessions, FALSE otherwise.
<i>pvaralTaskIds</i>	Variant. Returns an array containing the task IDs that identify the running tasks. The array is returned as a Long subtype.
<i>pvaralTaskProgress</i>	Variant. Returns an array containing the progress complete percentages of the consolidations. The array is returned as a Long subtype.
<i>pvaralTaskStatus</i>	Variant. Returns an array containing the consolidations' task statuses. Valid values are represented by the HFMConstants type library constants listed in "Task Status Constants" on page 872 . The array is returned as a Long subtype.
<i>pvarabstrUserNames</i>	Variant. Returns an array containing the usernames of the consolidations' users. The array is returned as a String subtype.
<i>pvarabstrServerNames</i>	Variant. Returns an array containing the names of the application servers for the consolidations. The array is returned as a String subtype.
<i>pvaralEntity</i>	Variant. Returns an array containing the member IDs of the Entity dimension members for the running tasks. The array is returned as a Long subtype.
<i>pvaralParent</i>	Variant. Returns an array containing the member IDs of the parents of the entities returned by the <i>pvaralEntity</i> argument. The array is returned as a Long subtype.
<i>pvaralScenario</i>	Variant. Returns an array containing the member IDs of the Scenario dimension members for the consolidations. The array is returned as a Long subtype.
<i>pvaralYear</i>	Variant. Returns an array containing the member IDs of the Year dimension members for the consolidations. The array is returned as a Long subtype.
<i>pvaralStartPeriod</i>	Variant. Returns an array containing the member IDs of the first Period dimension member in the range of periods for the consolidations. The array is returned as a Long subtype.
<i>pvaralEndPeriod</i>	Variant. Returns an array containing the member IDs of the last Period dimension member in the range of periods for the consolidations. The array is returned as a Long subtype.
<i>pvaralConsolType</i>	Variant. Returns an array of flags that indicate the consolidation types of the consolidations. Valid values are represented by the HFMConstants type library constants described in "Consolidation Type Constants" on page 855 .
<i>pITotalRecords</i>	Long. Returns a count of the number of consolidations returned by <code>EnumRunningTasksPOV</code> .

ExtractTaskAudit

Extracts to a file the task audit records that meet the specified criteria. Criteria include date range, application server, user, and task.

Syntax

```
<HsvSystemInfo>.ExtractTaskAudit bstrClientFileName,  
bstrClientLogFileName, bstrDelimitChar, dStartTime, dEndTime,  
vbAllServers, bstrServer, vbAllUsers, lActivityUserID, vbAllTasks,  
lActivityTaskID
```

Argument	Description
<i>bstrClientFileName</i>	String (ByVal). The name and path of the extract file.
<i>bstrClientLogFileName</i>	String (ByVal). The name and path of the log file for the extract operation.
<i>bstrDelimitChar</i>	String (ByVal). The delimiter for records in the extract file.
<i>dStartTime</i>	Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.
<i>dEndTime</i>	Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.
<i>vbAllServers</i>	Boolean (ByVal). Specifies whether to extract task audit records for all application servers. Pass TRUE for all application servers, FALSE to specify an application server with the <i>bstrServer</i> argument.
<i>bstrServer</i>	String (ByVal). The name of the application server for which to extract task audit records. This argument is used only if the <i>vbAllServers</i> argument is set to FALSE.
<i>vbAllUsers</i>	Boolean (ByVal). Specifies whether to extract task audit records for all users. Pass TRUE for all users, FALSE to specify a user with the <i>lActivityUserID</i> argument.
<i>lActivityUserID</i>	Long (ByVal). The activity user ID of the user for whom to extract task audit records. This argument is used only if the <i>vbAllUsers</i> argument is set to FALSE. To get a user's activity user ID, use GetActivityUserID .
<i>vbAllTasks</i>	Boolean (ByVal). Specifies whether to extract task audit records for all tasks or a given task. Pass TRUE for all tasks, FALSE to specify a task with the <i>lActivityTaskID</i> argument.
<i>lActivityTaskID</i>	Long (ByVal). The ID of the task for which to extract audit records. 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 870 .

Example

The following example extracts task audit records for all logons.

```
Dim cSysInfo As HsvSystemInfo  
'g_cSession is an HsvSession object reference  
Set cSysInfo = g_cSession.SystemInfo  
cSysInfo.ExtractTaskAudit "c:\temp\Audit.txt", "c:\temp\Audit.log", _
```

",", 0, CDb1(Now()), True, "", True, -1, False, USERACTIVITYCODE_LOGON

GetActivityCodeDesc

Returns the description of a given type of activity.

Syntax

```
<HsvSystemInfo>.GetActivityCodeDesc lActivityCode, bstrDesc
```

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

<i>lActivityCode</i>	Long (ByVal). The ID of the activity. Valid values are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 870 .
----------------------	---

<i>bstrDesc</i>	String. The description of the activity.
-----------------	--

GetActivityUserID

Returns the activity user ID for a given username.

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.

Note: Activity user IDs differ from the security identifiers that represent usernames. Security identifiers are discussed in [“Security Identifiers” on page 454](#).

Syntax

```
<HsvSystemInfo>.GetActivityUserID bstrUserName, plActivityUserID
```

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

<i>bstrUserName</i>	String (ByVal). The username of the user.
---------------------	---

Note: The username should include the domain.

<i>plActivityUserID</i>	Long. Returns the activity user ID for the user.
-------------------------	--

Example

GetActivityUserID is used in the example for [ClearAuditTasks](#).

GetApplicationDirectory

Returns the name of the application folder for an application.

An application folder is located on the application server. An application folder’s name consists of the application name appended to the system folder. For example, if the system folder is C :

\Financial Management, and the application name is Acme, then `GetApplicationDirectory` would return the following path:

```
C:\Financial Management\Acme\
```

System folders are defined with Financial Management Server Administrator. For details, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

Tip: Financial Management automatically creates an application folder when it generates items such as reports. If no such items have been generated for an application, the path returned by `GetApplicationDirectory` might not yet exist on the application server. In this case, the returned path indicates the directory that Financial Management will create when it generates items such as reports.

Syntax

```
<HsvSystemInfo>.GetApplicationDirectory()
```

Return Value

String. Returns the path of the application directory.

Example

This example prints the application directory to the Immediate window.

```
Dim cHsvSystemInfo As HsvSystemInfo
Set cHsvSystemInfo = m_cHsvSession.SystemInfo
Debug.Print cHsvSystemInfo.GetApplicationDirectory
```

GetApplicationName

Returns the name of the application to which the client is connected.

Syntax

```
<HsvSystemInfo>.GetApplicationName()
```

Return Value

String. Returns the application name.

Example

The following example assigns the application name to the `sApp` variable.

```
Dim cHsvSystemInfo As HsvSystemInfo, sApp As String
'm_cHsvSession is an HsvSession object reference
Set cHsvSystemInfo = m_cHsvSession.SystemInfo
sApp = cHsvSystemInfo.GetApplicationName
```

GetCalcRulesType

For internal use.

GetCOMDLLCalcRules

For internal use.

GetCurrentActivity

Returns information about the user's current activity.

Syntax

```
<HsvSystemInfo>.GetCurrentActivity pbstrAppName, pbstrServerName,  
plActivityUserID, plCurrentActivity, pbstrModuleName, pdTimeStarted,  
pbstrDescription
```

Argument	Description
<i>pbstrAppName</i>	String. Returns the name of the application to which the user is logged on.
<i>pbstrServerName</i>	String. Returns the name of the application server to which the user is logged on.
<i>plActivityUserID</i>	Long. Returns the activity user ID for the user.
<i>plCurrentActivity</i>	Long. Returns the ID of the user's current activity. Valid values are represented by the HFMConstants type library constants listed in "User Activity Constants" on page 870 .
<i>pbstrModuleName</i>	String. Returns the name of the Financial Management module in which the user is working.
<i>pdTimeStarted</i>	Double. Returns the timestamp of when the user started the activity. The timestamp is returned as a Double that can be cast into a Date format.
<i>pbstrDescription</i>	String. Returns the description of the user's current activity.

GetExtractFileEncoding

Returns the file encoding type for extracted files.

Syntax

```
<HsvSystemInfo>.GetExtractFileEncoding plExtractFileEncoding
```

Argument	Description
<i>plExtractFileEncoding</i>	Long. Returns the type of file encoding for extracted files. For valid values, see "Extracted File Encoding Constants" on page 869 .

GetFormattedDateTime

Returns a string representation of the given double-byte date/time value using the language ID for the connected user.

Syntax

```
<HsvSystemInfo>.GetFormattedDateTime dDateTime, bDateValueOnly,  
bTimeValueOnly, pbstrDateTime
```

Argument	Description
<i>dDateTime</i>	Double (ByVal). The timestamp.
<i>bDateValueOnly</i>	Long (ByVal). TRUE to return only the date value.
<i>bTimeValueOnly</i>	Long (ByVal). TRUE to return only the time value.
<i>pbstrDateTime</i>	String. Returned string.

GetFormattedResourceString

For internal use.

GetKillUsersStatus

For internal use.

GetKillUserStatus

Indicates whether an administrator has logged off the current user.

Syntax

```
<HsvSystemInfo>.GetKillUserStatus pvbKill, pvbWarn, pdTimestamp
```

Argument	Description
<i>pvbKill</i>	Boolean. Indicates whether the current user has been logged off. Returns TRUE if the user has been logged off, FALSE otherwise.
<i>pvbWarn</i>	Boolean. <i>For internal use.</i>
<i>pdTimestamp</i>	Double. 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.

GetLanguageUserParameters

Gets the language in which member descriptions are displayed for the connected user.

Syntax

```
<HsvSystemInfo>.GetLanguageUserParameters pLanguageID
```

Argument

Description

GetLanguageUserParameters Long. Returns the language's ID.

Tip: You can get the label for a language ID by using the HsvMetadata object's [EnumLanguages](#) method.

Example

`GetLanguageUserParameters` is used in the [Example](#) for `HsvMetadata.GetConsolidationMethodInfo`.

GetLastModifiedDateForArtifact

For internal use.

GetModuleName

Returns the name of the module represented by a given module ID.

Syntax

```
<HsvSystemInfo>.GetModuleName lModuleID, bstrModName
```

Argument

Description

lModuleID Long (ByVal). The module ID. Valid values are represented by the HFMCConstants enumeration tagMODULEIDS, which is described in ["Module ID Constants" on page 883](#).

bstrModName String. Returns the module name.

GetNumberFormattingUserParameters

Returns the double-byte integers that identify the user's decimal and thousands separator characters.

Syntax

```
<HsvSystemInfo>.GetNumberFormattingUserParameters pwcDecimalChar,  
pwcThousandsChar
```

Argument	Description
<i>pwdDecimalChar</i>	Integer. Returns the double-byte integer equivalent of the user's decimal character.
<i>pwdThousandsChar</i>	Integer. Returns the double-byte integer equivalent of the user's thousands separator character.

Example

This example prints the user's decimal and thousands separator characters to Visual Basic's Immediate window.

```
Dim cSystemInfo As HsvSystemInfo, iDec As Integer
Dim iThous As Integer
'g_cSession is an HsvSession object reference
Set cSystemInfo = g_cSession.SystemInfo
cSystemInfo.GetNumberFormattingUserParameters iDec, iThous
Debug.Print "Decimal: " & ChrW(iDec)
Debug.Print "Thousands: " & ChrW(iThous)
```

GetResourceLanguageUserParameters

Returns the ID of the user's default language for resource strings such as error messages and other strings that are generated on the server.

Syntax

```
<HsvSystemInfo>.GetResourceLanguageUserParameters plResourceLanguageID
```

Argument	Description
<i>plResourceLanguageID</i>	Long. Returns the language's ID. The valid IDs are represented by the HFMConstants type library constants listed in "Supported Language Constants" on page 833 .

GetResourceString

For internal use.

GetRunningTaskLogFilePathName

Returns the name and path of the log file for a given running task.

Syntax

```
<HsvSystemInfo>.GetRunningTaskLogFilePathName lTaskID, pbstrLogFilePathName
```

Argument	Description
<i>lTaskID</i>	Long (ByVal). The ID of the task.

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

Tip: You can obtain the IDs of all running tasks with [EnumRunningTasks](#).

pbstrLogFilePathName String. Returns the name and path of the log file.

GetRunningTaskProgress

Returns a given running task's percentage complete, status, last update time, and description.

Syntax

```
<HsvSystemInfo>.GetRunningTaskProgress lTaskID, plProgress, plStatus,  
pdLastUpdateTime, pbstrDesc
```

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

lTaskID Long (ByVal). The ID of the task.

Tip: You can obtain the IDs of all running tasks with [EnumRunningTasks](#).

plProgress Long. Returns the progress complete percentage of the task.

plStatus Long. Returns the status of the task. Valid values are represented by the HFMConstants type library constants listed in ["Task Status Constants" on page 872](#).

pdLastUpdateTime Double. 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.

pbstrDesc String. Returns the task's description.

GetRunningTasksCount

Returns the number of currently running tasks for the application.

Syntax

```
<HsvSystemInfo>.GetRunningTasksCount plRunningTasksCount
```

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

plRunningTasksCount Long. Returns the number of tasks that are running.

GetRunningTaskStatus

Returns the status of a given running task.

Syntax

```
<HsvSystemInfo>.GetRunningTaskStatus lTaskID, plStatus
```

Argument Description

ITaskID Long (ByVal). The ID of the task.

Tip: You can obtain the IDs of all running tasks with [EnumRunningTasks](#).

pIStatus Long. Returns the status of the task. Valid values are represented by the HFMConstants type library constants listed in [“Task Status Constants” on page 872](#).

GetServerName

Returns the name of the server to which the client is connected.

Syntax

```
<HsvSystemInfo>.GetServerName()
```

Return Value

String. Returns the server name.

Example

The following example prints the server name to the Immediate window.

```
Dim cHsvSystemInfo As HsvSystemInfo, sServ As String
Dim vaRules, bRulesExist As Boolean
'm_cHsvSession is an HsvSession object reference
Set cHsvSystemInfo = m_cHsvSession.SystemInfo
sServ = cHsvSystemInfo.GetServerName
Debug.Print sServ
```

GetUserName

Returns the username of the connected user.

Syntax

```
<HsvSystemInfo>.GetUserName pbstrUserName
```

Argument Description

pbstrUserName String. Returns the username of the connected user.

Example

The following example prints the username of the connected user to Visual Basic's Immediate window.

```
Dim cSysInfo As HsvSystemInfo, sUserName As String
'm_cHsvSession is an HsvSession object reference
Set cSysInfo = m_cHsvSession.SystemInfo
```

```
cSysInfo.GetUserName sUserName  
Debug.Print sUserName
```

GetUserParameter

Returns the value of a user parameter that has been set with `SetUserParameter`. (For details on setting user parameters, see [“SetUserParameter” on page 524.](#))

Syntax

```
<HsvSystemInfo>.GetUserParameter bstrKey, pvarabyData
```

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

<i>bstrKey</i>	String (ByVal). The key that identifies the parameter.
----------------	--

<i>pvarabyData</i>	Variant. Returns the value that has been set for the parameter. This value is returned as a Byte subtype.
--------------------	---

Example

This example tests whether a parameter named “Projects” has been defined for the user. If no parameter has been defined, any code placed within the If structure would be executed.

```
Dim cHsvSystemInfo As HsvSystemInfo, vaGetParam  
'm_cHsvSession is an HsvSession object reference  
Set cHsvSystemInfo = m_cHsvSession.SystemInfo  
cHsvSystemInfo.GetUserParameter "Projects", vaGetParam  
If IsEmpty(vaGetParam) = True Then  
...  
End If
```

GetVBScriptCalcRules

Returns the rules that have been loaded into an application. The rules are returned as an array of bytes.

Note: This method returns the contents of the rules file (.RLE file) that has been loaded into an application. If the rules file was changed after the file was loaded, the updated file’s contents are not returned by `GetVBScriptCalcRules`.

Syntax

```
<HsvSystemInfo>.GetVBScriptCalcRules pvarabRules, pbRulesExist
```

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

<i>pvarabRules</i>	Variant array. Returns the application’s rules. The array is returned as a Byte subtype.
--------------------	--

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

<i>pbRulesExist</i>	Boolean. Returns TRUE if a rules file has been loaded into the application, FALSE if no rules file has been loaded.
---------------------	---

Example

The following example outputs an application's rules to a text file. The array of bytes returned by `GetVBScriptCalcRules` is converted to a String and inserted into a file with various Visual Basic methods.

```
Dim cSysInfo As HsvSystemInfo, vaRules As Variant
Dim bRulesExist As Boolean, iFile As Integer
'm_cHsvSession is an HsvSession object reference
Set cSysInfo = m_cHsvSession.SystemInfo
cSysInfo.GetVBScriptCalcRules vaRules, bRulesExist
If bRulesExist = True Then
    iFile = FreeFile
    Open "C:\Program Files\Acme\appRules.rle" For Output As #iFile
    Print #iFile, StrConv(CStr(vaRules), 64)
    Close #iFile
End If
```

GetVBScriptMemberListRules

Returns an application's member lists as an array of bytes in the LST format.

Syntax

```
<HsvSystemInfo>.GetVBScriptMemberListRules pvarabRules, pbRulesExist
```

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

<i>pvarabRules</i>	Variant array. Returns the contents of the most recently-loaded member lists file. The array is returned as a Byte subtype.
--------------------	---

<i>pbRulesExist</i>	Boolean. Returns TRUE if a member lists file has been loaded into the application, FALSE if no member lists file has been loaded.
---------------------	---

Example

The following example outputs a member lists file's contents to a text file. The array of bytes returned by `GetVBScriptMemberListRules` is then inserted into a file with various Visual Basic methods.

```
Dim cHsvSystemInfo As HsvSystemInfo
Dim vaRules as Variant, bRulesExist As Boolean
'm_cHsvSession is an HsvSession object reference
Set cHsvSystemInfo = m_cHsvSession.SystemInfo
cHsvSystemInfo.GetVBScriptMemberListRules vaRules, bRulesExist
iFile = FreeFile
Open "C:\Program Files\Acme\appList.lst" For Output As #iFile
Print #iFile, StrConv(CStr(vaRules), 64)
```

Close #iFile

GetWorkingDirectory

For internal use.

IsScheduledTaskReadyToRun

For internal use.

KeepRunningTaskStillAlive

For internal use.

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 cluster or application, use `HsxClient.KillUsers`.

Syntax

```
<HsvSystemInfo>.KillUsers vbAllServers, bstrServer, vbAllUsers,  
lActivityUserID, vbAllSessions, lActivitySessionID
```

Argument	Description
<code>vbAllServers</code>	Boolean (ByVal). 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 <code>bstrServer</code> argument.
<code>bstrServer</code>	String (ByVal). The name of the application server for which to log users off. This argument is used only if the <code>vbAllServers</code> argument is set to FALSE.
<code>vbAllUsers</code>	Boolean (ByVal). Determines whether to log off all users. Pass TRUE to log off all users, FALSE to log off the user specified in the <code>lActivityUserID</code> argument.
<code>lActivityUserID</code>	Long (ByVal). The activity user ID of the user to log off. This argument is used only if the <code>vbAllUsers</code> argument is set to FALSE. To get a user's activity user ID, use GetActivityUserID .
<code>vbAllSessions</code>	Boolean (ByVal). Determines whether to log off all user sessions. Pass TRUE to log off all sessions, FALSE to log off the session specified in the <code>lActivitySessionID</code> argument.
<code>lActivitySessionID</code>	Long (ByVal). The ID of the session to log off. This argument is used only if the <code>vbAllSessions</code> argument is set to FALSE.

OutputSystemInfo

For internal use.

ReleaseHsxServer

For internal use.

Syntax

```
<HsvSystemInfo>.ReleaseHsxServer plRefCount
```

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

<i>plRefCount</i>	Long.
-------------------	-------

SetApplicationDirectory

For internal use.

SetCOMDLLCalcRules

For internal use.

SetCurrentActivity

Sets the activity for the current user.

Syntax

```
<HsvSystemInfo>.SetCurrentActivity lActivityCode, bstrDescription
```

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

<i>lActivityCode</i>	Long (ByVal). The ID of the activity to be set. Valid values are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 870 .
----------------------	---

Tip: If you are adding an activity for a custom module, use USERACTIVITYCODE_EXTERNAL, which represents custom activities.

<i>bstrDescription</i>	String (ByVal). The description of the activity.
------------------------	--

SetCurrentModule

Sets the current module using a module name.

Tip: To set the current module with a module ID, use [SetCurrentModuleEx](#).

Syntax

```
<HsvSystemInfo>.SetCurrentModule bstrModuleName
```

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

<i>bstrModuleName</i>	String (ByVal). The name of the module.
-----------------------	---

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 883 .
----------------------------	---

SetExtractFileEncoding

Sets the file encoding type for extracted files.

Syntax

```
<HsvSystemInfo>.SetExtractFileEncoding lExtractFileEncoding
```

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

<i>lExtractFileEncoding</i>	Long (ByVal). Identifies the type of file encoding for extracted files. For valid values see “Extracted File Encoding Constants” on page 869 .
-----------------------------	--

SetLanguageUserParameters

Sets the language in which member descriptions are displayed for the connected user.

Syntax

```
<HsvSystemInfo>.SetLanguageUserParameters lLanguageID
```

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

<i>ILanguageID</i>	Long (ByVal). The ID of the language to be set.
--------------------	---

Tip: You can return the internal IDs and the labels of an application's languages with the HsvMetadata object's [EnumLanguages](#) method.

SetNumberFormattingUserParameters

Sets the user's decimal and thousands separator characters.

Syntax

```
<HsvSystemInfo>.SetNumberFormattingUserParameters wcDecimalChar,  
wcThousandsChar
```

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

<i>wcDecimalChar</i>	Integer (ByVal). The double-byte Integer that identifies the decimal character to be set.
----------------------	---

<i>wcThousandsChar</i>	Integer (ByVal). The double-byte Integer that identifies the thousands separator character to be set.
------------------------	---

Example

The following example sets a comma as the decimal character and a period as the thousands separator character.

```
Dim cSystemInfo As HsvSystemInfo  
'g_cSession is an HsvSession object reference  
Set cSystemInfo = g_cSession.SystemInfo  
cSystemInfo.SetNumberFormattingUserParameters AscW(", "), _  
AscW(".")
```

SetResourceLanguageForCurrentSession

Sets the language for a user's resource strings in the current session. The specified language does not persist beyond the current session; in other words, this language does not become the user's default language for resource strings.

Note: Resource strings include error messages and other strings that are generated on the server.

Syntax

```
<HsvSystemInfo>.SetResourceLanguageForCurrentSession lResourceLanguageID
```

Argument	Description
<i>lResourceLanguageID</i>	Long (ByVal). The ID of the language to be set for the session. The valid IDs are represented by the HFMConstants type library constants listed in “Supported Language Constants” on page 833 .

SetResourceLanguageUserParameters

Sets the user’s default language for resource strings such as error messages and other strings that are generated on the server.

Syntax

```
<HsvSystemInfo>.SetResourceLanguageUserParameters lResourceLanguageID
```

Argument	Description
<i>lResourceLanguageID</i>	Long (ByVal). The ID of the language to be set. The valid IDs are represented by the HFMConstants type library constants listed in “Supported Language Constants” on page 833 .

SetUserParameter

Sets a parameter for the user. Use `SetUserParameter` to create and edit custom parameters for items such as personal preferences. These parameters are stored in the database and thus persist beyond the current session.

Tip: `GetUserParameter` returns a user parameter and `DeleteUserParameter` deletes a user parameter. For details, see [“GetUserParameter” on page 518](#) and [“DeleteUserParameter” on page 495](#).

Syntax

```
<HsvSystemInfo>.SetUserParameter bstrKey, varabyData
```

Argument Description

bstrKey String (ByVal). The key that identifies the parameter.

varabyData Byte array (ByVal). The value to be set for the user parameter.

Example

The following example sets a user parameter named “Projects” to the value specified in the `comboParams` combo box.

```
Dim cHsvSystemInfo As HsvSystemInfo
'm_cHsvSession is an HsvSession object reference
Set cHsvSystemInfo = m_cHsvSession.SystemInfo
```

```
'Example assumes the combo box contains 1-character values.
Dim bytSetParam(0) As Byte
bytSetParam(0) = CByte(comboParams.Text)
cHsvSystemInfo.SetUserParameter "Projects", bytSetParam
```

SetVBScriptCalcRules

Validates and loads a rules file; a flag determines whether `SetVBScriptCalcRules` loads after validation or validates without loading.

`SetVBScriptCalcRules` returns arrays that contain the line numbers, severity levels, descriptions, contents, and details of the lines in a rules file that do not pass validation. These arrays have a one-to-one correspondence, with one item for each line that does not pass validation.

Tip: Variables cannot be tested for validity, so whenever `SetVBScriptCalcRules` encounters a line containing a variable, items with a severity level of Information are included in the arrays. The descriptions for these lines say that “validation was not performed.”

Syntax

```
<HsvSystemInfo>.SetVBScriptCalcRules varabRules, vbScanOnly,
pvbErrorsWereFound, pvbWarningsWereFound, pvbInfoWasProvided,
pvaralErrorLineNumbers, pvaralErrorSeverity, pvarabstrErrorDescriptions,
pvarabstrErrorVBScript, pvarabstrErrorDetails
```

Argument	Description
<i>varabRules</i>	Variant array (ByVal). The rules file, passed as a Variant array of bytes.
<i>vbScanOnly</i>	Boolean (ByVal). Determines whether <code>SetVBScriptCalcRules</code> loads after validating. Pass TRUE to load after validation, FALSE to validate without loading.
<i>pvbErrorsWereFound</i>	Boolean. Indicates whether <code>SetVBScriptCalcRules</code> found any validation errors. Returns TRUE if errors are found, FALSE otherwise.
<i>pvbWarningsWereFound</i>	Boolean. Indicates whether <code>SetVBScriptCalcRules</code> found any validation warnings. Returns TRUE if warnings are found, FALSE otherwise.
<i>pvbInfoWasProvided</i>	Boolean. Indicates whether <code>SetVBScriptCalcRules</code> returns any information not classified as errors or warnings. Returns TRUE if non-error and non-warning information is returned, FALSE otherwise.
<i>pvaralErrorLineNumbers</i>	Variant array. Returns the line numbers to which the errors, warnings, and information apply. The array is returned as a Long subtype.
<i>pvaralErrorSeverity</i>	Variant array. Returns numbers that indicate the severity levels of the errors, warnings, and information. Valid return values are described in the following list: <ul style="list-style-type: none"> ● 1 = Error severity level. ● 2 = Warning severity level. ● 3 = Information severity level.

Argument	Description
	The array is returned as a Long subtype.
<i>pvarabstrErrorDescriptions</i>	Variant array. Returns descriptions of the errors, warnings, and information. The array is returned as a String subtype.
<i>pvarabstrErrorVBScript</i>	Variant array. Returns the rules file statements to which the errors, warnings, and information apply. The array is returned as a String subtype.
<i>pvarabstrErrorDetails</i>	Variant array. Returns details that describe why the lines caused validation to fail. The array is returned as a String subtype.

Example

The following example creates a log file that lists the line numbers, descriptions, and details for any validation messages of a warning severity level. Various Visual Basic methods convert the rules file to a Variant array of bytes, which is then passed to `SetVBScriptCalcRules`; note that `SetVBScriptCalcRules` is set to validate without loading. If the `pvbWarningsWereFound` argument returns `TRUE`, the example loops through the `pvaralErrorSeverity` argument's array to find items flagged with a warning severity level. For each such item, the corresponding items returned in the `pvaralErrorLineNumbers`, `pvarabstrErrorDescriptions`, and `pvarabstrErrorDetails` arguments are concatenated to the `vLogText` variable. `vLogText`'s value then is written to the log file, which is created with various Visual Basic methods.

```
Dim cSysInfo As HsvSystemInfo, vFileName, lFile As Long
Dim lSize As Long, vaRules, bytaRules() As Byte
Dim bErrs As Boolean, bWarnings As Boolean, bInfo As Boolean
Dim valLines, valSeverity, vasDescs, vasErr, vasDetails
Dim vLogText, iFile As Integer
'm_chsvSession is an HsvSession object reference
Set cSysInfo = m_chsvSession.SystemInfo
vFileName = "C:\Program Files\Acme\AppRules.rle"
lFile = FreeFile
lSize = FileLen(vFileName)
Open vFileName For Binary Access Read As #lFile
ReDim bytaRules(lSize)
Get #lFile, , bytaRules
Close #lFile
vaRules = bytaRules
cSysInfo.SetVBScriptCalcRules vaRules, True, bErrs, bWarnings, _
bInfo, valLines, valSeverity, vasDescs, vasErr, vasDetails
If bWarnings = True Then
    For i = LBound(valLines) To UBound(valLines)
        If valSeverity(i) = 2 Then
            vLogText = vLogText & "Line #: " & valLines(i) & _
                vbCrLf & "    Description: " & vasDescs(i) & vbCrLf _
                & "    Details: " & vasDetails(i) & vbCrLf & vbCrLf
        End If
    Next i
    iFile = FreeFile
    Open "C:\Program Files\Acme\warnings.log" For Output As #iFile
    Print #iFile, vLogText
    Close #iFile
End If
```

End If

SetVBScriptMemberListRules

Validates and loads a member lists file that is in the LST format. A flag determines whether `SetVBScriptMemberListRules` loads after validation or validates without loading.

`SetVBScriptMemberListRules` returns arrays that contain the line numbers, severity levels, descriptions, contents, and details of the lines in a member lists file that do not pass validation. These arrays have a one-to-one correspondence, with one item for each line that does not pass validation.

Syntax

```
<HsvSystemInfo>.SetVBScriptMemberListRules varabRules, vbScanOnly,  
pvbErrorsWereFound, pvbWarningsWereFound, pvbInfoWasProvided,  
pvaralErrorLineNumbers, pvaralErrorSeverity, pvarabstrErrorDescriptions,  
pvarabstrErrorVBScript, pvarabstrErrorDetails
```

Argument	Description
<i>varabRules</i>	Variant array (ByVal). The member lists file, passed as a Variant array of bytes.
<i>vbScanOnly</i>	Boolean (ByVal). Determines whether <code>SetVBScriptMemberListRules</code> loads after validating. Pass TRUE to load after validation, FALSE to validate without loading.
<i>pvbErrorsWereFound</i>	Boolean. Indicates whether <code>SetVBScriptMemberListRules</code> found any validation errors. Returns TRUE if errors are found, FALSE otherwise.
<i>pvbWarningsWereFound</i>	Boolean. Indicates whether <code>SetVBScriptMemberListRules</code> found any validation warnings. Returns TRUE if warnings are found, FALSE otherwise.
<i>pvbInfoWasProvided</i>	Boolean. Indicates whether <code>SetVBScriptCalcRules</code> returns any information not classified as errors or warnings. Returns TRUE if non-error and non-warning information is returned, FALSE otherwise.
<i>pvaralErrorLineNumbers</i>	Variant array. Returns the line numbers to which the errors, warnings, and information apply. The array is returned as a Long subtype.
<i>pvaralErrorSeverity</i>	Variant array. Returns numbers that indicate the severity levels of the errors, warnings, and information. Valid return values are described in the following list: <ul style="list-style-type: none">● 1 = Error severity level.● 2 = Warning severity level.● 3 = Information severity level. The array is returned as a Long subtype.
<i>pvarabstrErrorDescriptions</i>	Variant array. Returns descriptions of the errors, warnings, and information. The array is returned as a String subtype.
<i>pvarabstrErrorVBScript</i>	Variant array. Returns the member lists file statements to which the errors, warnings, and information apply. The array is returned as a String subtype.

Argument	Description
<i>pvarabstrErrorDetails</i>	Variant array. Returns details that describe why the lines caused validation to fail. The array is returned as a String subtype.

Example

The following example creates a log file that lists the line numbers, descriptions, and details for any validation messages of an error level. Various Visual Basic methods convert the rules file to a Variant array of bytes, which is then passed to `SetVBScriptMemberListRules`; note that `SetVBScriptMemberListRules` is set to validate without loading. If the `pvbErrorsWereFound` argument returns TRUE, the example loops through the `pvaralErrorSeverity` argument's array to find items flagged with an error severity level. For each such item, the corresponding items returned in the `pvaralErrorLineNumbers`, `pvarabstrErrorDescriptions`, and `pvarabstrErrorDetails` arguments are concatenated to the `vLogText` variable. `vLogText`'s value is then written to the log file, which is created with various Visual Basic methods.

```
Dim cSysInfo As HsvSystemInfo, vFileName, lFile As Long
Dim lSize As Long, vaRules, bytaRules() As Byte
Dim bErrs As Boolean, bWarnings As Boolean, bInfo As Boolean
Dim valLines, valSeverity, vasDescs, vasErr, vasDetails
Dim vLogText, iFile As Integer
'm_cHsvSession is an HsvSession object reference
Set cSysInfo = m_cHsvSession.SystemInfo
vFileName = "C:\Program Files\Acme\AppMemLists.lst"
lFile = FreeFile
lSize = FileLen(vFileName)
Open vFileName For Binary Access Read As #lFile
ReDim bytaRules(lSize)
Get #lFile, , bytaRules
Close #lFile
vaRules = bytaRules
cSysInfo.SetVBScriptMemberListRules vaRules, True, bErrs, _
bWarnings, bInfo, valLines, valSeverity, vasDescs, _
vasErr, vasDetails
If bErrs = True Then
    For i = LBound(valLines) To UBound(valLines)
        If valSeverity(i) = 1 Then
            vLogText = vLogText & "Line #: " & valLines(i) & _
vbCrLf & "    Description: " & vasDescs(i) & vbCrLf _
& "    Details: " & vasDetails(i) & vbCrLf & vbCrLf
        End If
    Next i
    iFile = FreeFile
    Open "C:\Program Files\Acme\listErrors.log" For Output _
As #iFile
    Print #iFile, vLogText
    Close #iFile
End If
```


StopRunningTask

For internal use.

UpdateRunningTaskLogFilePathName

For internal use.

UpdateRunningTaskPOV

For internal use.

UpdateRunningTaskProgress

For internal use.

UpdateRunningTaskProgressDetails

For internal use.

UpdateRunningTaskStatus

For internal use.

WarnUsersForShutDown

For internal use.

This chapter describes the members of the HsvProcessFlow type library. This type library exposes Financial Management's Process Management features. Use the methods of this type library to take actions for and get the histories of process units and submission phases.

To use the HsvProcessFlow type library, you must reference `HsvProcessFlow.dll` in your project.

The HsvProcessFlow type library contains one object—the HsvProcessFlow object. Use the HsvProcessFlow object to take actions for and getting histories of process units. The object's methods are summarized in [Table 29](#), and are described in detail in the following topics.

Note: Assign HsvProcessFlow object references with the `ProcessFlow` property of the HsvSession object. For an example, see [“HsvProcessFlow Type Library Overview” on page 88](#).

When executing actions such as promotions, the connected user must be assigned to the required security role for the action. For information on roles, see the *Oracle Hyperion Financial Management, Fusion Edition User's Guide*.

Process Units versus Submission Phases

An application setting determines whether an application supports process management by process unit or by submission phase. You can check this application setting with the HsvMetadata method `GetUseSubmissionPhaseFlag`. This application setting determines which HsvProcessFlow methods you can use. If the application supports phased submissions, use only methods for phased submissions; otherwise, use only methods for process units.

To distinguish these methods, the names of methods for phased submissions typically begin with the prefix “PhasedSubmission.” For example, if phased submissions are disabled, start a process unit with `Approve`, `Approve2`, or `ApproveEx`. If phased submissions are enabled, start a submission phase with `PhasedSubmissionApprove`, `PhasedSubmissionApprove2`, or `PhasedSubmissionApproveEx`.

If phased submissions are enabled, the following methods can be useful:

- The HsvMetadata type library provides methods for obtaining applicable dimension and member settings:

- To return whether the application uses the Account, Intercompany Partner, and Custom dimensions for process management, use the following HsvMetadata object methods:
 - ❑ [GetSupportSubmissionPhaseForAccountFlag](#)
 - ❑ [GetSupportSubmissionPhaseForICPFlag](#)
 - ❑ [GetSupportSubmissionPhaseForCustom1Flag](#)
 - ❑ [GetSupportSubmissionPhaseForCustom2Flag](#)
 - ❑ [GetSupportSubmissionPhaseForCustom3Flag](#)
 - ❑ [GetSupportSubmissionPhaseForCustom4Flag](#)
- To determine the submission group to which an Account, Intercompany Partner, or Custom dimension member is assigned, use the `GetSubmissionGroup` method of the HsvAccounts, HsvICPs, or HsvCustom object.
- The following HsvData methods assign and obtain submission groups for submission phases:
 - [GetPhaseSubmissionGridForGivenScenarioPeriod](#)
 - [SetPhaseSubmissionGridForGivenScenarioPeriod](#)

Approve

Approves a process unit.

To attach documents when approving, use [Approve2](#). To approve multiple process units, use [ApproveEx](#) or [Approve2](#).

Syntax

```
<HsvProcessFlow>.Approve lScenario, lYear, lPeriod, lEntity, lParent, lValue, vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are approved. Pass TRUE to approve related process units, FALSE to approve only the process unit for the <i>lValue</i> member.
<i>bstrAnnotation</i>	String (ByVal). The comment for the approval.

Argument	Description
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_APPROVED, which is the level constant for the Approved level.

Example

This example approves a process unit. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Approve`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long, iState As Integer
Dim cHsvProcessFlow As HsvProcessFlow, lRights As Long
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2001")
lPer = GetMemberID(DIMENSIONPERIOD, "September")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cHsvProcessFlow = m_cHsvSession.ProcessFlow
cHsvProcessFlow.Approve lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, "Good numbers", iState
```

Approve2

Approves a process unit, and optionally attaches documents and approves all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.

Syntax

```
<HsvProcessFlow>.Approve2 lScenario, lYear, lPeriod, lEntity, lParent,
lValue, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation,
varabstrPaths, varabstrFiles, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether process units for Value dimension members related to the <i>lValue</i> member are approved. Pass TRUE to approve related process units, FALSE to approve only the process unit for the <i>lValue</i> member.

Argument	Description
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). Specifies whether to approve the process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members. Pass TRUE to approve these process units, FALSE otherwise. If you specify TRUE and an error occurs for a period, Financial Management will attempt to approve the other periods' process units.
<i>bstrAnnotation</i>	String (ByVal). The comment for the approval.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_APPROVED, which is the level constant for the Approved level.

Example

This example approves a process unit and attaches documents. The calls to the user-defined GetMemberID function get the process unit's member IDs; for information on GetMemberID, see the [Examples](#) for GetItemID. These member IDs are then passed to Approve2.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim chsvProcessFlow As HsvProcessFlow, iState As Integer
Dim saPaths(1) As String, saNames(1) As String
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2005")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "NewYork")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'g_cSession is an HsvSession object reference
Set chsvProcessFlow = g_cSession.ProcessFlow
chsvProcessFlow.Approve2 lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, False, "see attachments", saPaths, saNames, iState
```

ApproveEx

Approves a process unit, and optionally approves all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.

To attach documents when approving, use [Approve2](#).

Syntax

```
<HsvProcessFlow>.ApproveEx lScenario, lYear, lPeriod, lEntity, lParent, lValue, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether process units for Value dimension members related to the <i>lValue</i> member are approved. Pass TRUE to approve related process units, FALSE to approve only the process unit for the <i>lValue</i> member.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). Specifies whether to approve the process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members. Pass TRUE to approve these process units, FALSE otherwise. If you specify TRUE and an error occurs for a period, Financial Management will attempt to approve the other periods. Note: Specifying FALSE is the same as calling <code>Approve</code> .
<i>bstrAnnotation</i>	String (ByVal). The comment for the approval.
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_APPROVED, which is the level constant for the Approved level. This value applies to the Period and Value dimension members passed in the <i>lPeriod</i> and <i>lValue</i> arguments.

Example

This example approves the specified process unit and the process units that consist of the base Period dimension members and the specified members of the other dimensions. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `ApproveEx`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long, iState As Integer
Dim cHsvProcessFlow As HsvProcessFlow, lRights As Long
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2001")
lPer = GetMemberID(DIMENSIONPERIOD, "September")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
```

```
'g_cSession is an HsvSession object reference
Set cHsvProcessFlow = g_cSession.ProcessFlow
cHsvProcessFlow.ApproveEx lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, True, "Approving base periods", iState
```

GetGroupPhaseFromCell

Returns the IDs of the submission group and submission phase to which a given cell is assigned.

Syntax

```
<HsvProcessFlow>.GetGroupPhaseFromCell lScenario, lYear, lPeriod, lEntity,
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
pbstrGroup, pbstrPhase
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>pbstrGroup</i>	String. Returns the ID of the cell's submission group.
<i>pbstrPhase</i>	String. Returns the phase ID of the cell's submission phase.

Example

The following function obtains the phase ID of the cell's submission phase, then starts the phase with PhasedSubmissionStart2.

```
Function StartPhasedWorkflow(lScen As Long, lYear As Long, lPer As Long, _
    lEnt As Long, lPar As Long, lVal As Long, lAcct As Long, lIcp As _
    Long, lC1 As Long, lC2 As Long, lC3 As Long, lC4 As Long, sNote As _
    String) As Integer
Dim cProcessFlow As HsvProcessFlow, sGroup As String, sPhase As String
Dim vaPaths As Variant, vaNames As Variant, iState As Integer
```



```

'g_cSession represents an HsvSession instance
Set cProcessFlow = g_cSession.ProcessFlow
cProcessFlow.GetGroupPhaseFromCell lScen, lYear, lPer, lEnt, lPar, lVal, _
    lAcct, lIcp, lC1, lC2, lC3, lC4, sGroup, sPhase
cProcessFlow.PhasedSubmissionStart2 lScen, lYear, lPer, lEnt, lPar, _
    lVal, MEMBERNOTUSED, MEMBERNOTUSED, MEMBERNOTUSED, MEMBERNOTUSED, _
    MEMBERNOTUSED, MEMBERNOTUSED, CLng(sPhase), True, False, sNote, _
    vaPaths, vaNames, iState
StartPhasedWorkflow = iState
End Function

```

GetHistory

Returns arrays containing the history of a process unit; the process unit's dates and times, users, actions, levels, and comments are returned. The arrays have a one-to-one correspondence.

To get a process unit history that includes the names and paths of document attachments, use [GetHistory2](#).

Syntax

```

<HsvProcessFlow>.GetHistory lScenario, lYear, lPeriod, lEntity, lParent,
lValue, pvaradTime, pvarabstrUser, pvarasAction, pvarasNewState,
pvarabstrAnnotation

```

Argument	Description
<i>lScenario</i>	<p>Long (ByVal). The member ID of the process unit's Scenario dimension member.</p> <p>If the scenario does not support process management, the 0x4D2 error occurs. This error has a Success severity level, meaning that it is not returned in Visual Basic's <code>Err</code> object; however, the error is returned in Visual C++.</p> <p>Tip: <code>HsvScenarios.SupportsProcessFlow</code> tests whether a scenario supports process management; for details, see “SupportsProcessFlow” on page 267.</p>
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>pvaradTime</i>	<p>Variant. Returns an array of the process unit's times and dates. The array is returned as a Double subtype.</p> <p>The numbers returned in this array are in a standard date format; for example, in Visual Basic you can convert a returned array element by passing it to <code>CDate</code>.</p>
<i>pvarabstrUser</i>	<p>Variant. Returns an array of the usernames that have performed actions for the process unit. The full usernames – the domains and the usernames – are returned.</p> <p>The array is returned as a String subtype.</p>

Argument	Description
<i>pvarasAction</i>	Variant. Returns an array of the process unit's actions. The valid return values are listed in "Process Management Action Constants" on page 853 . The array is returned as an Integer subtype.
<i>pvarasNewState</i>	Variant. Returns an array of the process unit's levels. The valid return values are listed in Table 136 on page 853 . The array is returned as an Integer subtype.
<i>pvarabstrAnnotation</i>	Variant. Returns an array of the process unit's comments. The array is returned as a String subtype.

Example

The following subroutine tests whether a process unit has been rejected. If it has been rejected, the subroutine then displays the username of the rejector and the comment entered for the rejection. The subroutine takes the member IDs of the process unit's dimension members. These IDs are passed to `GetHistory`. The `For...Next` loop tests the array returned by the *pvarasAction* argument; if an array element equals `PROCESS_FLOW_ACTION_REJECT`, then a rejection has occurred, and the corresponding elements of the arrays returned in the *pvarabstrUser* and *pvarabstrAnnotation* arguments are displayed.

```
Sub IsUnitRejected(lScen as Long, lYear as Long, lPer as Long, _
lEnt as Long, lPar as Long, lVal as Long)
Dim cHsvProcessFlow As HsvProcessFlow, vaTime, vaUser
Dim vaAction, vaState, vaNote, iSize As Integer
Set cHsvProcessFlow = m_cHsvSession.ProcessFlow
cHsvProcessFlow.GetHistory lScen, lYear, lPer, lEnt, lPar, _
lVal, vaTime, vaUser, vaAction, vaState, vaNote
iSize = UBound(vaAction)
For i = 0 To iSize
    If vaAction(i) = PROCESS_FLOW_ACTION_REJECT Then
        MsgBox "Rejected by: " & vaUser(i) & vbCrLf & _
            "Reason: " & vaNote(i)
    End If
Next i
End Sub
```

GetHistory2

Returns arrays containing the history of a process unit; the process unit's dates and times, users, actions, levels, comments, and names and paths of attached document are returned. The arrays have a one-to-one correspondence.

Syntax

```
<HsvProcessFlow>.GetHistory2 lScenario, lYear, lPeriod, lEntity, lParent,
lValue, pvaradTime, pvarabstrUser, pvarasAction, pvarasNewState,
pvarabstrAnnotation, pvaravarabstrPaths, pvaravarabstrFiles
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>pvaradTime</i>	Variant. Returns an array of the process unit's times and dates. The array is returned as a Double subtype. The numbers returned in this array are in a standard date format; for example, in Visual Basic you can convert a returned array element by passing it to <code>CDate</code> .
<i>pvarabstrUser</i>	Variant. Returns an array of the usernames that have performed actions for the process unit. The full usernames – the domains and the usernames – are returned. The array is returned as a String subtype.
<i>pvarasAction</i>	Variant. Returns an array of the process unit's actions. The valid return values are listed in “Process Management Action Constants” on page 853 . The array is returned as an Integer subtype.
<i>pvarasNewState</i>	Variant. Returns an array of the process unit's levels. The valid return values are listed in Table 136 on page 853 . The array is returned as an Integer subtype.
<i>pvarabstrAnnotation</i>	Variant. Returns an array of the process unit's comments. The array is returned as a String subtype.
<i>pvaravarabstrPaths</i>	Variant. Returns an array of arrays that contain the paths of attached documents for each action. The array is returned as a String subtype.
<i>pvaravarabstrFiles</i>	Variant. Returns an array of arrays that contain the names of attached documents for each action. The array is returned as a String subtype.

Example

The following example prints the times of a process unit's approvals, as well as the names and paths of documents attached to approvals. The example assumes that the process unit's dimension member IDs have been specified.

```
Dim cProcessFlow As HsvProcessFlow, vaTimes, vaUsers
Dim vaActions, vaStates, vaNotes, vaPaths, vaNames
'g_cSession is an HsvSession object reference
Set cProcessFlow = g_cSession.ProcessFlow
'the example assumes that member IDs have been specified
cProcessFlow.GetHistory2 lScen, lYear, lPer, lEnt, lPar, lVal, vaTimes, _
    vaUsers, vaActions, vaStates, vaNotes, vaPaths, vaNames
For i = LBound(vaActions) To UBound(vaActions)
    If vaActions(i) = PROCESS_FLOW_ACTION_APPROVE Then
        Debug.Print CDate(vaTimes(i))
        If IsArray(vaPaths(i)) Then
```

```

    For j = LBound(vaPaths(i), 1) To UBound(vaPaths(i), 1)
        Debug.Print vaPaths(i)(j) & "\" & vaNames(i)(j)
    Next j
End If
End If
Next i

```

GetPhasedSubmissionHistory

Returns arrays containing the history of a submission phase, given the member IDs of a cell in the phase. The arrays have a one-to-one correspondence.

Syntax

```

<HsvProcessFlow>.GetPhasedSubmissionHistory lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, pvaradTime, pvarabstrUser, pvarasAction, pvarasNewState,
pvarabstrAnnotation

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>pvaradTime</i>	Variant array. Returns an array of the submission phase's times and dates. The array is returned as a Double sub-type. The numbers returned in this array are in a standard date format; for example, in Visual Basic you can convert an array element by passing it to <code>CDate</code> .
<i>pvarabstrUser</i>	Variant array. Returns an array of the usernames that have performed actions for the submission phase. The usernames are fully qualified. The array is returned as a String sub-type.

Argument	Description
<i>pvarasAction</i>	Variant array. Returns an array of the submission phase's actions. The valid return values are listed in Table 135 on page 853 . The array is returned as an Integer sub-type.
<i>pvarasNewState</i>	Variant array. Returns an array of the submission phase's review levels. The valid return values are listed in Table 136 on page 853 . The array is returned as an Integer sub-type.
<i>pvarabstrAnnotation</i>	Variant array. Returns an array of the submission phase's comments. The array is returned as a String sub-type.

Example

The following function returns the review levels that resulted from a given user's actions upon a submission phase.

```
Function GetUserPhaseStates(lScen As Long, lYear As Long, lPer As Long, _
    lEnt As Long, lPar As Long, lVal As Long, lAcct As Long, lIcp As _
    Long, lC1 As Long, lC2 As Long, lC3 As Long, lC4 As Long, sUser _
    As String) As Variant
Dim cProcessFlow As HsvProcessFlow, vaTimes, vaUsers, vaActions, vaStates
Dim vaNotes, vaRet(), lCounter As Long
lCounter = -1
'g_cSession represents an HsvSession instance
Set cProcessFlow = g_cSession.ProcessFlow
cProcessFlow.GetPhasedSubmissionHistory lScen, lYear, lPer, lEnt, lPar, _
    lVal, lAcct, lIcp, lC1, lC2, lC3, lC4, vaTimes, vaUsers, vaActions, _
    vaStates, vaNotes
For i = LBound(vaUsers) To UBound(vaUsers)
    If vaUsers(i) = sUser Then
        lCounter = lCounter + 1
        ReDim Preserve vaRet(lCounter)
        vaRet(lCounter) = vaStates(i)
    End If
Next i
GetUserPhaseStates = vaRet
End Function
```

GetPhasedSubmissionState

Returns the current review level of a given cell's submission phase.

Syntax

```
<HsvProcessFlow>.GetPhasedSubmissionState lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, psProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.

Argument	Description
<i>Year</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>psProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase's review level. Valid values are listed in Table 136 on page 853 .

Example

`GetPhasedSubmissionState` is used in the example for [PhasedSubmissionPromote](#).

GetPhasedSubmissionStateUsingPhaseID

Returns the review level of a submission phase, given either the member IDs 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 HFMConstants `MEMBERNOTUSED` to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass `MEMBERNOTUSED` to the phase ID parameter.

Syntax

```
<HsvProcessFlow>.GetPhasedSubmissionStateUsingPhaseID lScenario, lYear,
lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2,
lCustom3, lCustom4, lPhaseID, vbEnsureUptodate, psProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.

Argument	Description
<i>IEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>IAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Account dimension member. ● MEMBERNOTUSED
<i>IICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Intercompany Partner dimension member. ● MEMBERNOTUSED
<i>ICustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom1 dimension member. ● MEMBERNOTUSED
<i>ICustom2</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom2 dimension member. ● MEMBERNOTUSED
<i>ICustom3</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom3 dimension member. ● MEMBERNOTUSED
<i>ICustom4</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom4 dimension member. ● MEMBERNOTUSED
<i>IPhaseID</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The phase ID, which you can obtain with GetGroupPhaseFromCell. ● MEMBERNOTUSED
<i>vbEnsureUptodate</i>	Boolean (ByVal). Forces a read from the database before returning the review level. Pass TRUE to force a read, FALSE to have the system read from the database only if necessary. <p>Note: The force option applies only to multi-server systems.</p>
<i>psProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase's review level. See Table 136 on page 853 .

GetState

Returns the current level of a process unit.

Syntax

```
<HsvProcessFlow>.GetState lScenario, lYear, lPeriod, lEntity, lParent, lValue, psProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>psProcessState</i>	Integer. Returns the level constant that identifies the process unit's current level. The valid return values are listed in Table 136 on page 853 .

Example

This example tests whether a process unit's current level is Not Started. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `GetState`. If `GetState` returns 1, then the current level is Not Started, and any code placed within the `If` structure would be executed.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cHsvProcessFlow As HsvProcessFlow, iState As Integer
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2001")
lPer = GetMemberID(DIMENSIONPERIOD, "August")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cHsvProcessFlow = m_cHsvSession.ProcessFlow
cHsvProcessFlow.GetState lScen, lYear, lPer, lEnt, lPar, lVal, _
iState
If iState = 1 Then
    ...
End If
```

PhasedSubmissionApprove

Approves a submission phase, given the member IDs of a cell in the phase.

Note: To approve by passing a phase ID, use [PhasedSubmissionApprove2](#).

Syntax

```
<HsvProcessFlow>.PhasedSubmissionApprove lScenario, lYear, lPeriod,  
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,  
lCustom4, vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

PhasedSubmissionApprove2

Approves a submission phase and optionally attaches documents, given either the member IDs 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 HFMConstants MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionApprove2 lScenario, lYear, lPeriod,  
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
```

`lCustom4, lPhaseID, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, varabstrPaths, varabstrFiles, psNewProcessState`

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Account dimension member.● MEMBERNOTUSED
<i>lICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Intercompany Partner dimension member.● MEMBERNOTUSED
<i>lCustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom1 dimension member.● MEMBERNOTUSED
<i>lCustom2</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom2 dimension member.● MEMBERNOTUSED
<i>lCustom3</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom3 dimension member.● MEMBERNOTUSED
<i>lCustom4</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom4 dimension member.● MEMBERNOTUSED
<i>lPhaseID</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The phase ID, which you can obtain with GetGroupPhaseFromCell.● MEMBERNOTUSED
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). 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.

Argument	Description
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

PhasedSubmissionApproveEx

Syntax

Approves a submission phase, and optionally applies the action to submission phases for all base periods that intersect with the specified non-Period dimension members, given the member IDs of a cell in the phase.

```
<HsvProcessFlow>.PhasedSubmissionApproveEx lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation,
psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.

Argument	Description
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). 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.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

PhasedSubmissionGetHistory2

Returns arrays containing the history of a submission phase, given the member IDs of a cell in the phase. The arrays have a one-to-one correspondence.

Note: The difference between this method and [GetPhasedSubmissionHistory](#) is that `PhasedSubmissionGetHistory2` returns the filenames and paths of attachments.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionGetHistory2 lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, pvaradTime, pvarabstrUser, pvarasAction, pvarasNewState, pvarabstrAnnotation, pvaravarabstrPaths, pvaravarabstrFiles
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.

Argument	Description
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>pvaradTime</i>	Variant array. Returns an array of the submission phase's times and dates. The array is returned as a Double sub-type. The numbers returned in this array are in a standard date format; for example, in Visual Basic you can convert an array element by passing it to <code>CDate</code> .
<i>pvarabstrUser</i>	Variant array. Returns an array of the usernames that have performed actions for the submission phase. The usernames are fully qualified. The array is returned as a String sub-type.
<i>pvarasAction</i>	Variant array. Returns an array of the submission phase's actions. The valid return values are listed in Table 135 on page 853 . The array is returned as an Integer sub-type.
<i>pvarasNewState</i>	Variant array. Returns an array of the submission phase's review levels. The valid return values are listed in Table 136 on page 853 . The array is returned as an Integer sub-type.
<i>pvarabstrAnnotation</i>	Variant array. Returns an array of the submission phase's comments. The array is returned as a String sub-type.
<i>pvaravarabstrPaths</i>	Variant array. Returns an array of arrays that contain the paths of attached documents for each action. The array is returned as a String sub-type.
<i>pvaravarabstrFiles</i>	Variant array. Returns an array of arrays that contain the names of attached documents for each action. The array is returned as a String sub-type.

PhasedSubmissionGetHistory2UsingPhaseID

Returns arrays containing the history of a submission phase, given either the member IDs 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 `HFMConstants.MEMBERNOTUSED` to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass `MEMBERNOTUSED` to the phase ID parameter.

The arrays have a one-to-one correspondence.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionGetHistory2UsingPhaseID lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, lPhaseID, pvaradTime, pvarabstrUser, pvarasAction, pvarasNewState, pvarabstrAnnotation, pvaravarabstrPaths, pvaravarabstrFiles
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.

Argument	Description
<i>IYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>IAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Account dimension member. ● MEMBERNOTUSED
<i>IICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Intercompany Partner dimension member. ● MEMBERNOTUSED
<i>ICustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom1 dimension member. ● MEMBERNOTUSED
<i>ICustom2</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom2 dimension member. ● MEMBERNOTUSED
<i>ICustom3</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom3 dimension member. ● MEMBERNOTUSED
<i>ICustom4</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom4 dimension member. ● MEMBERNOTUSED
<i>IPhaseID</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The phase ID, which you can obtain with GetGroupPhaseFromCell. ● MEMBERNOTUSED
<i>pvaradTime</i>	Variant array. Returns an array of the submission phase's times and dates. The array is returned as a Double sub-type. The numbers returned in this array are in a standard date format; for example, in Visual Basic you can convert an array element by passing it to <code>CDate</code> .
<i>pvarabstrUser</i>	Variant array. Returns an array of the usernames that have performed actions for the submission phase. The usernames are fully qualified. The array is returned as a String sub-type.
<i>pvarasAction</i>	Variant array. Returns an array of the submission phase's actions. The valid return values are listed in Table 135 on page 853 .

Argument	Description
	The array is returned as an Integer sub-type.
<i>pvarasNewState</i>	Variant array. Returns an array of the submission phase's review levels. The valid return values are listed in Table 136 on page 853 . The array is returned as an Integer sub-type.
<i>pvarabstrAnnotation</i>	Variant array. Returns an array of the submission phase's comments. The array is returned as a String sub-type.
<i>pvaravarabstrPaths</i>	Variant array. Returns an array of arrays that contain the paths of attached documents for each action. The array is returned as a String sub-type.
<i>pvaravarabstrFiles</i>	Variant array. Returns an array of arrays that contain the names of attached documents for each action. The array is returned as a String sub-type.

PhasedSubmissionProcessManagementChangeStateForMultipleEntities2

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

```
<HsvProcessFlow>.PhasedSubmissionProcessManagementChangeStateForMultipleEntities2 lScenarioIn, lYearIn, lPeriodIn, varalEntity, varalParent, lValueIn, varalAccount, varalICP, varalCustom1, varalCustom2, varalCustom3, varalCustom4, bstrAnnotation, lAction, vbUseAllValueMembers, vbApplyToAllPeriods, sProcessStateToPromoteTo, varabstrPaths, varabstrFiles, psNewProcessState
```

Argument	Description
<i>lScenarioIn</i>	Long (ByVal). The member ID of the process units' Scenario dimension member
<i>lYearIn</i>	Long (ByVal). The member ID of the process units' Year dimension member.
<i>lPeriodIn</i>	Long (ByVal). The member ID of the process units' Period dimension member.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the process units' child Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the process units' parent Entity dimension members. This array has a one-to-one correspondence with the <i>varalEntity</i> argument's array.
<i>lValueIn</i>	Long (ByVal). The member ID of the process units' Value dimension member.
<i>varalAccount</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICP</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.

Argument	Description
<i>varalCustom1</i>	Long array (ByVal). The member IDs of the cells' Custom1 dimension members.
<i>varalCustom2</i>	Long array (ByVal). The member IDs of the cells' Custom2 dimension members.
<i>varalCustom3</i>	Long array (ByVal). The member IDs of the cells' Custom3 dimension members.
<i>varalCustom4</i>	Long array (ByVal). The member IDs of the cells' Custom4 dimension members.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>lAction</i>	Long (ByVal). 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 853 .
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValueIn</i> member. Pass TRUE to apply the action to related submission phases.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). 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.
<i>sProcessStateToPromoteTo</i>	Integer (ByVal). 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 853 .
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns the review level applied to the process units. Valid values are represented by the HFMConstants type library constants listed in “Process Management Review Level Constants” on page 853 .

PhasedSubmissionPromote

Promotes a submission phase to a specified review level, given the member IDs of a cell in the phase.

To promote by passing a phase ID, use [PhasedSubmissionPromote2](#).

Syntax

```
<HsvProcessFlow>.PhasedSubmissionPromote lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, vbUseAllValueMembers, bstrAnnotation, sProcessStateToPromoteTo,
psNewProcessState
```


Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>sProcessStateToPromoteTo</i>	Integer (ByVal). The constant that identifies the level to which to promote the submission phase. Use one of the CEnumProcessFlowStates enumeration constants listed in Table 136 on page 853 .
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action.

Example

The following function promotes a given cell's submission phase to a given review level. The function uses [GetPhasedSubmissionState](#) to test whether the submission phase's current review level is less than the specified review level, and returns an array that indicates the submission phase's review level and whether it was promoted.

```
Function PromotePhase(lScen As Long, lYear As Long, lPer As Long, lEnt As _
    Long, lPar As Long, lVal As Long, lAcct As Long, lIcp As Long, _
    lC1 As Long, lC2 As Long, lC3 As Long, lC4 As Long, iNewState As _
    Integer, sNote As String) As Variant
Dim cProcessFlow As HsvProcessFlow, iState As Integer
Dim iRetState As Integer, vaRet(1) As Variant
vaRet(0) = False
'g_cSession represents an HsvSession instance
Set cProcessFlow = g_cSession.ProcessFlow
cProcessFlow.GetPhasedSubmissionState lScen, lYear, lPer, lEnt, _
    lPar, lVal, lAcct, lIcp, lC1, lC2, lC3, lC4, iState
If iState < iNewState Then
```

```

    cProcessFlow.PhasedSubmissionPromote lScen, lYear, lPer, lEnt, _
    lPar, lVal, lAcct, lIcp, lC1, lC2, lC3, lC4, False, sNote, _
    iNewState, iRetState
    vaRet(1) = iRetState
    vaRet(0) = True
Else
    vaRet(1) = iState
End If
PromotePhase = vaRet
End Function

```

PhasedSubmissionPromote2

Promotes a submission phase and optionally attaches documents, given either the member IDs 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 HFMConstants `MEMBERNOTUSED` to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass `MEMBERNOTUSED` to the phase ID parameter.

Syntax

```

<HsvProcessFlow>.PhasedSubmissionPromote2 lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, lPhasedID, vbUseAllValueMembers, bstrAnnotation,
sProcessStateToPromoteTo, varabstrPaths, varabstrFiles, psNewProcessState

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Account dimension member. ● <code>MEMBERNOTUSED</code>
<i>lICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Intercompany Partner dimension member. ● <code>MEMBERNOTUSED</code>
<i>lCustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom1 dimension member.

Argument	Description
	<ul style="list-style-type: none"> ● MEMBERNOTUSED
<i>ICustom2</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The member ID of the cell's Custom2 dimension member. ● MEMBERNOTUSED
<i>ICustom3</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The member ID of the cell's Custom3 dimension member. ● MEMBERNOTUSED
<i>ICustom4</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The member ID of the cell's Custom4 dimension member. ● MEMBERNOTUSED
<i>IPhaseID</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The phase ID, which you can obtain with GetGroupPhaseFromCell. ● MEMBERNOTUSED
<i>vbUseAllValueMembers</i>	<p>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>IValue</i> member. Pass TRUE to apply the action to related submission phases.</p>
<i>bstrAnnotation</i>	<p>String (ByVal). The comment for the process management action.</p>
<i>sProcessStateToPromoteTo</i>	<p>Integer (ByVal). 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 853.</p>
<i>varabstrPaths</i>	<p>String array (ByVal). 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>
<i>varabstrFiles</i>	<p>String array (ByVal). 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>
<i>psNewProcessState</i>	<p>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853.</p>

PhasedSubmissionPublish

Publishes a submission phase, given the member IDs of a cell in the phase.

To publish by passing a phase ID, use [PhasedSubmissionPublish2](#).

Syntax

```
<HsvProcessFlow>.PhasedSubmissionPublish lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

PhasedSubmissionPublish2

Publishes a submission phase and optionally attaches documents, given either the member IDs 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 HFMConstants `MEMBERNOTUSED` to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass `MEMBERNOTUSED` to the phase ID parameter.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionPublish2 lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
```

`lCustom4`, `lPhaseID`, `vbUseAllValueMembers`, `vbApplyToAllPeriods`,
`bstrAnnotation`, `varabstrPaths`, `varabstrFiles`, `psNewProcessState`

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>IAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Account dimension member.● MEMBERNOTUSED
<i>IICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Intercompany Partner dimension member.● MEMBERNOTUSED
<i>ICustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom1 dimension member.● MEMBERNOTUSED
<i>ICustom2</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom2 dimension member.● MEMBERNOTUSED
<i>ICustom3</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom3 dimension member.● MEMBERNOTUSED
<i>ICustom4</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom4 dimension member.● MEMBERNOTUSED
<i>IPhaseID</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The phase ID, which you can obtain with GetGroupPhaseFromCell.● MEMBERNOTUSED
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>IValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). 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.

Argument	Description
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

PhasedSubmissionPublishEx

Publishes a submission phase, and optionally applies the action to submission phases for all base periods that intersect with the specified non-Period dimension members, given the member IDs of a cell in the phase.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionPublishEx lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation,
psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.

Argument	Description
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). 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.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

PhasedSubmissionReject

Rejects a submission phase, given the member IDs of a cell in the phase.

To reject by passing a phase ID, use [PhasedSubmissionReject2](#).

Syntax

```
<HsvProcessFlow>.PhasedSubmissionReject lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.

Argument	Description
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

PhasedSubmissionReject2

Rejects a submission phase and optionally attaches documents, given either the member IDs 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 HFMConstants MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionReject2 lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, lPhaseID, vbUseAllValueMembers, bstrAnnotation, varabstrPaths,
varabstrFiles, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Account dimension member. ● MEMBERNOTUSED
<i>lICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Intercompany Partner dimension member. ● MEMBERNOTUSED
<i>lCustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom1 dimension member. ● MEMBERNOTUSED
<i>lCustom2</i>	Long (ByVal). Pass one of the following:

Argument	Description
	<ul style="list-style-type: none"> ● The member ID of the cell's Custom2 dimension member. ● MEMBERNOTUSED
<i>ICustom3</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The member ID of the cell's Custom3 dimension member. ● MEMBERNOTUSED
<i>ICustom4</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The member ID of the cell's Custom4 dimension member. ● MEMBERNOTUSED
<i>IPhaseID</i>	<p>Long (ByVal). Pass one of the following:</p> <ul style="list-style-type: none"> ● The phase ID, which you can obtain with GetGroupPhaseFromCell. ● MEMBERNOTUSED
<i>vbUseAllValueMembers</i>	<p>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.</p>
<i>bstrAnnotation</i>	<p>String (ByVal). The comment for the process management action.</p>
<i>varabstrPaths</i>	<p>String array (ByVal). 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>
<i>varabstrFiles</i>	<p>String array (ByVal). 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>
<i>psNewProcessState</i>	<p>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853.</p>

PhasedSubmissionSignOff

Signs off on a submission phase, given the member IDs of a cell in the phase.

To sign off by passing a phase ID, use [PhasedSubmissionSignOff2](#).

Syntax

```
<HsvProcessFlow>.PhasedSubmissionSignOff lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, vbUseAllValueMembers, bstrAnnotation
```

Argument	Description
<i>lScenario</i>	<p>Long (ByVal). The member ID of the process unit's Scenario dimension member.</p>
<i>lYear</i>	<p>Long (ByVal). The member ID of the process unit's Year dimension member.</p>

Argument	Description
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.

PhasedSubmissionSignOff2

Signs off on a submission phase and optionally attaches documents, given either the member IDs 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 HFMConstants `MEMBERNOTUSED` to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass `MEMBERNOTUSED` to the phase ID parameter.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionSignOff2 lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, lPhaseID, vbUseAllValueMembers, bstrAnnotation, varabstrPaths,
varabstrFiles
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.

Argument	Description
<i>IParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>IAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Account dimension member. ● MEMBERNOTUSED
<i>IICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Intercompany Partner dimension member. ● MEMBERNOTUSED
<i>ICustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom1 dimension member. ● MEMBERNOTUSED
<i>ICustom2</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom2 dimension member. ● MEMBERNOTUSED
<i>ICustom3</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom3 dimension member. ● MEMBERNOTUSED
<i>ICustom4</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom4 dimension member. ● MEMBERNOTUSED
<i>IPhaseID</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The phase ID, which you can obtain with GetGroupPhaseFromCell. ● MEMBERNOTUSED
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>IValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.

PhasedSubmissionStart

Starts a submission phase, given the member IDs of a cell in the phase.

To start by passing a phase ID, use [PhasedSubmissionStart2](#).

Syntax

```
<HsvProcessFlow>.PhasedSubmissionStart lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

Example

The following function starts a given cell's submission phase if the following conditions are met:

- `HsvScenarios.SupportsProcessFlow` indicates that the scenario is enabled for process management.
- `IHsvDataSecurity.GetProcessUnitAccessRightsAndStateEx` indicates that the connected user has All access rights and that the submission phase's review level is Not Started.

The function returns an array that includes the SupportsProcessFlow value, the user's access rights, and the submission phase's review level.

```
Function StartPhase(lScen As Long, lYear As Long, lPer As Long, lEnt As _  
  
    Long, lPar As Long, lVal As Long, lAcct As Long, lIcp As Long, _  
    lC1 As Long, lC2 As Long, lC3 As Long, lC4 As Long, sNote As String) _  
    As Variant  
Dim cProcessFlow As HsvProcessFlow, lRights As Long, iState As Integer  
Dim iNewState As Integer, bEnabled As Boolean, cDataSec As IHsvDataSecurity  
Dim cScenario As HsvScenarios, vaRet(2) As Variant  
'g_cSession represents an HsvSession instance  
Set cDataSec = g_cSession.Security  
Set cProcessFlow = g_cSession.ProcessFlow  
'g_cMetadata represents an HsvMetadata instance  
Set cScenario = g_cMetadata.Scenarios  
cScenario.SupportsProcessFlow lScen, bEnabled  
vaRet(0) = bEnabled  
If bEnabled = True Then  
    cDataSec.GetProcessUnitAccessRightsAndStateEx lScen, lYear, lPer, _  
    lEnt, lPar, lVal, lAcct, lIcp, lC1, lC2, lC3, lC4, MEMBERNOTUSED, _  
    lRights, iState  
    vaRet(1) = lRights  
    If lRights = HFM_ACCESS_RIGHTS_ALL Then  
        If iState = PROCESS_FLOW_STATE_NOT_STARTED Then  
            cProcessFlow.PhasedSubmissionStart lScen, lYear, lPer, lEnt, _  
            lPar, lVal, lAcct, lIcp, lC1, lC2, lC3, lC4, False, sNote, _  
            iNewState  
            vaRet(2) = iNewState  
        Else  
            vaRet(2) = iState  
        End If  
    End If  
End If  
StartPhase = vaRet  
End Function
```

PhasedSubmissionStart2

Starts a submission phase and optionally attaches documents, given either the member IDs 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 HFMConstants MEMBERNOTUSED to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass MEMBERNOTUSED to the phase ID parameter.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionStart2 lScenario, lYear, lPeriod,  
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,  
lCustom4, lPhaseID, vbUseAllValueMembers, vbApplyToAllPeriods,  
bstrAnnotation, varabstrPaths, varabstrFiles, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Account dimension member.● MEMBERNOTUSED
<i>lICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Intercompany Partner dimension member.● MEMBERNOTUSED
<i>lCustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom1 dimension member.● MEMBERNOTUSED
<i>lCustom2</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom2 dimension member.● MEMBERNOTUSED
<i>lCustom3</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom3 dimension member.● MEMBERNOTUSED
<i>lCustom4</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The member ID of the cell's Custom4 dimension member.● MEMBERNOTUSED
<i>lPhaseID</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none">● The phase ID, which you can obtain with GetGroupPhaseFromCell.● MEMBERNOTUSED
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.

Argument	Description
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). 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.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

Example

See the example for [GetGroupPhaseFromCell](#).

PhasedSubmissionStartEx

Starts a submission phase, and optionally applies the action to submission phases for all base periods that intersect with the specified non-Period dimension members, given the member IDs of a cell in the phase.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionStartEx lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation,
psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.

Argument	Description
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). 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.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

Example

The following function starts a given cell's submission phase, as well as submission phases with base periods that intersect the specified dimension members, if the following conditions are met:

- `HsvScenarios.SupportsProcessFlow` indicates that the scenario is enabled for process management.
- `IHsvDataSecurity.GetProcessUnitAccessRightsAndStateEx` indicates that the connected user has All access rights and that the submission phase's review level is Not Started.

The function returns an array that includes the `SupportsProcessFlow` value, the user's access rights, and the submission phase's review level.

```
Function StartPhaseBasePers(lScen As Long, lYear As Long, lPer As Long,
lEnt As _
    Long, lPar As Long, lVal As Long, lAcct As Long, lIcp As Long, _
    lC1 As Long, lC2 As Long, lC3 As Long, lC4 As Long, sNote As String) _
    As Variant
Dim cProcessFlow As HsvProcessFlow, lRights As Long, iState As Integer
Dim iNewState As Integer, bEnabled As Boolean, cDataSec As IHsvDataSecurity
Dim cScenario As HsvScenarios, vaRet(2) As Variant
'g_cSession represents an HsvSession instance
Set cDataSec = g_cSession.Security
Set cProcessFlow = g_cSession.ProcessFlow
'g_cMetadata represents an HsvMetadata instance
Set cScenario = g_cMetadata.Scenarios
cScenario.SupportsProcessFlow lScen, bEnabled
vaRet(0) = bEnabled
If bEnabled = True Then
    cDataSec.GetProcessUnitAccessRightsAndStateEx lScen, lYear, lPer, _
```



```

lEnt, lPar, lVal, lAcct, lIcp, lC1, lC2, lC3, lC4, MEMBERNOTUSED, _
lRights, iState
vaRet(1) = lRights
If lRights = HFM_ACCESS_RIGHTS_ALL Then
    If iState = PROCESS_FLOW_STATE_NOT_STARTED Then
        cProcessFlow.PhasedSubmissionStartEx lScen, lYear, lPer, _
        lEnt, lPar, lVal, lAcct, lIcp, lC1, lC2, lC3, lC4, False, _
        True, sNote, iNewState
        vaRet(2) = iNewState
    Else
        vaRet(2) = iState
    End If
End If
End If
StartPhaseBasePers = vaRet
End Function

```

PhasedSubmissionSubmit

Submits a submission phase, given the member IDs of a cell in the phase.

To submit by passing a phase ID, use [PhasedSubmissionSubmit2](#).

Syntax

```

<HsvProcessFlow>.PhasedSubmissionSubmit lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, vbUseAllValueMembers, bstrAnnotation, psNewProcessState

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom4 dimension member.

Argument	Description
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

PhasedSubmissionSubmit2

Submits a submission phase and optionally attaches documents, given either the member IDs 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 HFMConstants `MEMBERNOTUSED` to the Account, Intercompany Partner, and Custom dimension parameters.
- If you use member IDs for the above-mentioned parameters, pass `MEMBERNOTUSED` to the phase ID parameter.

Syntax

```
<HsvProcessFlow>.PhasedSubmissionSubmit2 lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, lPhaseID, vbUseAllValueMembers, bstrAnnotation, varabstrPaths,
varabstrFiles, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>lAccount</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Account dimension member. ● <code>MEMBERNOTUSED</code>
<i>lICP</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Intercompany Partner dimension member. ● <code>MEMBERNOTUSED</code>
<i>lCustom1</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> ● The member ID of the cell's Custom1 dimension member.

Argument	Description
	<ul style="list-style-type: none"> MEMBERNOTUSED
<i>ICustom2</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> The member ID of the cell's Custom2 dimension member. MEMBERNOTUSED
<i>ICustom3</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> The member ID of the cell's Custom3 dimension member. MEMBERNOTUSED
<i>ICustom4</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> The member ID of the cell's Custom4 dimension member. MEMBERNOTUSED
<i>IPhaseID</i>	Long (ByVal). Pass one of the following: <ul style="list-style-type: none"> The phase ID, which you can obtain with GetGroupPhaseFromCell. MEMBERNOTUSED
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related submission phases.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 136 on page 853 .

ProcessManagementChangeStateForMultipleEntities

Applies a given process management action to process units for one or more Entity dimension members.

To attach documents when applying a process management action to process units, use [ProcessManagementChangeStateForMultipleEntities2](#).

Syntax

```
<HsvProcessFlow>.ProcessManagementChangeStateForMultipleEntities
lScenarioIn, lYearIn, lPeriodIn, varalEntity, varalParent, lValueIn,
bstrAnnotation, lAction, vbUseAllValueMembers, vbApplyToAllPeriods,
sProcessStateToPromoteTo, psNewProcessState
```

Argument	Description
<i>lScenarioIn</i>	Long (ByVal). The member ID of the process units' Scenario dimension member.
<i>lYearIn</i>	Long (ByVal). The member ID of the process units' Year dimension member.
<i>lPeriodIn</i>	Long (ByVal). The member ID of the process units' Period dimension member.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the process units' child Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the process units' parent Entity dimension members. This array has a one-to-one correspondence with the <i>varalEntity</i> argument's array.
<i>lValueIn</i>	Long (ByVal). The member ID of the process units' Value dimension member.
<i>bstrAnnotation</i>	String (ByVal). A comment to apply to the action.
<i>lAction</i>	Long (ByVal). The action to apply to the process units. Valid values are represented by the HFMConstants type library constants listed in "Process Management Action Constants" on page 853 .
<i>vbUseAllValueMembers</i>	Boolean (ByVal). A flag that determines whether the action applies to process units for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related process units, FALSE to apply the action to only the process unit for the <i>lValue</i> member.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). A flag that determines whether the action applies to process units for all base periods. Pass TRUE to apply the action to all base periods, FALSE otherwise.
<i>sProcessStateToPromoteTo</i>	Integer (ByVal). 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 853 .
<i>psNewProcessState</i>	Integer. Returns the review level applied to the process units. Valid values are represented by the HFMConstants type library constants listed in "Process Management Review Level Constants" on page 853 .

ProcessManagementChangeStateForMultipleEntities2

Applies a given process management action and attaches documents to process units for one or more Entity dimension members. The specified documents are attached to the process units for all of the specified Entity members.

Syntax

```
<HsvProcessFlow>.ProcessManagementChangeStateForMultipleEntities2
lScenarioIn, lYearIn, lPeriodIn, varalEntity, varalParent, lValueIn,
bstrAnnotation, lAction, vbUseAllValueMembers, vbApplyToAllPeriods,
sProcessStateToPromoteTo, varabstrPaths, varabstrFiles, psNewProcessState
```

Argument	Description
<i>lScenarioIn</i>	Long (ByVal). The member ID of the process units' Scenario dimension member.
<i>lYearIn</i>	Long (ByVal). The member ID of the process units' Year dimension member.

Argument	Description
<i>lPeriodIn</i>	Long (ByVal). The member ID of the process units' Period dimension member.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the process units' child Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the process units' parent Entity dimension members. This array has a one-to-one correspondence with the <i>varalEntity</i> argument's array.
<i>lValueIn</i>	Long (ByVal). The member ID of the process units' Value dimension member.
<i>bstrAnnotation</i>	String (ByVal). A comment to apply to the action.
<i>lAction</i>	Long (ByVal). The action to apply to the process units. Valid values are represented by the HFMConstants type library constants listed in "Process Management Action Constants" on page 853 .
<i>vbUseAllValueMembers</i>	Boolean (ByVal). A flag that determines whether the action applies to process units for Value dimension members related to the <i>lValue</i> member. Pass TRUE to apply the action to related process units, FALSE to apply the action to only the process units for the <i>lValue</i> member.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). A flag that determines whether the action applies to process units for all base periods. Pass TRUE to apply the action to all base periods, FALSE otherwise.
<i>sProcessStateToPromoteTo</i>	Integer (ByVal). 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 853 .
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns the review level applied to the process units. Valid values are represented by the HFMConstants type library constants listed in "Process Management Review Level Constants" on page 853 .

Example

The following example attached two documents in a folder named "docs" to the specified process units. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`.

```
Dim lScen As Long, lYear As Long, lPer As Long, lVal As Long
Dim cProcessFlow As HsvProcessFlow, laEnts(1) As Long
Dim laPars(1) As Long, saPaths(1) As String
Dim saNames(1) As String, iState As Integer
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2004")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
laEnts(0) = GetMemberID(DIMENSIONENTITY, "Virginia")
```

```

laEnts(1) = GetMemberID(DIMENSIONENTITY, "NewYork")
laPars(0) = GetMemberID(DIMENSIONENTITY, "UnitedStates")
laPars(1) = GetMemberID(DIMENSIONENTITY, "UnitedStates")
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'g_cSession is an HsvSession object reference
Set cProcessFlow = g_cSession.ProcessFlow
cProcessFlow.ProcessManagementChangeStateForMultipleEntities2 lScen,
lYear, lPer, _
    laEnts, laPars, lVal, "see attached", PROCESS_FLOW_ACTION_START, False,
False, _
    PROCESS_FLOW_STATE_FIRST_PASS, saPaths, saNames, iState

```

ProcessManagementChangeStateForMultipleEntitiesEx

Applies a given process management action 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

```

<HsvProcessFlow>.ProcessManagementChangeStateForMultipleEntitiesEx
lScenarioIn, lYearIn, lPeriodIn, varalEntity, varalParent, lValueIn,
varalAccounts, varalICPs, varalCustom1s, varalCustom2s, varalCustom3s,
varalCustom4s, bstrAnnotation, lAction, vbUseAllValueMembers,
vbApplyToAllPeriods, sProcessStateToPromoteTo, psNewProcessState

```

Argument	Description
<i>lScenarioIn</i>	Long (ByVal). The member ID of the process units' Scenario dimension member
<i>lYearIn</i>	Long (ByVal). The member ID of the process units' Year dimension member.
<i>lPeriodIn</i>	Long (ByVal). The member ID of the process units' Period dimension member.
<i>varalEntity</i>	Long array (ByVal). The member IDs of the process units' child Entity dimension members.
<i>varalParent</i>	Long array (ByVal). The member IDs of the process units' parent Entity dimension members. This array has a one-to-one correspondence with the <i>varalEntity</i> argument's array.
<i>lValueIn</i>	Long (ByVal). The member ID of the process units' Value dimension member.
<i>varalAccounts</i>	Long array (ByVal). The member IDs of the cells' Account dimension members.
<i>varalICPs</i>	Long array (ByVal). The member IDs of the cells' Intercompany Partner dimension members.
<i>varalCustom1s</i>	Long array (ByVal). The member IDs of the cells' Custom1 dimension members.
<i>varalCustom2s</i>	Long array (ByVal). The member IDs of the cells' Custom2 dimension members.
<i>varalCustom3s</i>	Long array (ByVal). The member IDs of the cells' Custom3 dimension members.

Argument	Description
<i>varalCustom4s</i>	Long array (ByVal). The member IDs of the cells' Custom4 dimension members.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process management action.
<i>IAction</i>	Long (ByVal). 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 853 .
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <i>lValueIn</i> member. Pass TRUE to apply the action to related submission phases.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). 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.
<i>sProcessStateToPromoteTo</i>	Integer (ByVal). 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 853 .
<i>psNewProcessState</i>	Integer. Returns the review level applied to the process units. Valid values are represented by the HFMConstants type library constants listed in “Process Management Review Level Constants” on page 853 .

Promote

Promotes a process unit to a specified review level.

To attach documents when promoting, use [Promote2](#).

Syntax

```
<HsvProcessFlow>.Promote lScenario, lYear, lPeriod, lEntity, lParent, lValue, vbUseAllValueMembers, bstrAnnotation, sProcessStateToPromoteTo, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are promoted. Pass TRUE to promote related process units, FALSE to promote only the process unit for the <i>lValue</i> member.

Argument	Description
<i>bstrAnnotation</i>	String (ByVal). The comment for the promotion.
<i>sProcessStateToPromoteTo</i>	Integer (ByVal). The constant that identifies the level to which the process unit is being promoted. Use one of the level constants listed in Table 136 on page 853 .
<i>psNewProcessState</i>	Integer. Returns the constant that identifies the level to which the process unit has been promoted. If the call to <code>Promote</code> succeeds, this should be equal to the constant passed in the <i>sProcessStateToPromoteTo</i> argument.

Example

The following subroutine promotes process units. The subroutine takes the following arguments:

- The process unit's dimension member IDs.
- The ID of the role that corresponds to the review level immediately beneath the level to which the process unit is being promoted.
- The constant that identifies the level to which the process unit is being promoted.

`IHsvDataSecurity.GetProcessUnitAccessRights` gets the connected user's access rights to the process unit. If the user has All access, the `HsvSecurityAccess` method `IsConnectedUserAllowedToPerformTask` tests whether the connected user has been assigned to the role passed to `ProcPromote`; if this returns `TRUE`, `Promote` is called.

```
Sub ProcPromote(lScen As Long, lYear As Long, lPer As Long, _
    lEnt As Long, lPar As Long, lVal As Long, lRoleID As Long, _
    iState As Integer)
    Dim cProcessFlow As HsvProcessFlow, iRetState As Integer
    Dim bInRole As Boolean, lUnitRights As Long
    Dim cSecurity As HsvSecurityAccess, cDataSec As IHsvDataSecurity
    'g_cSession is an HsvSession object reference
    Set cProcessFlow = g_cSession.ProcessFlow
    Set cSecurity = g_cSession.Security
    Set cDataSec = g_cSession.Security
    cDataSec.GetProcessUnitAccessRights lScen, lYear, lPer, lEnt, lPar, _
        lVal, lUnitRights
    If lUnitRights = HFM_ACCESS_RIGHTS_ALL Then
        cSecurity.IsConnectedUserAllowedToPerformTask lRoleID, bInRole
        If bInRole = True Then
            cProcessFlow.Promote lScen, lYear, lPer, lEnt, lPar, lVal, _
                False, "", iState, iRetState
        End If
    End If
End Sub
```

Promote2

Promotes a process unit to a specified review level, and provides the option of attaching one or more documents.

Syntax

```
<HsvProcessFlow>.Promote2 lScenario, lYear, lPeriod, lEntity, lParent,  
lValue, vbUseAllValueMembers, bstrAnnotation, sProcessStateToPromoteTo,  
varabstrPaths, varabstrFiles, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are promoted. Pass TRUE to promote related process units, FALSE to promote only the process unit for the <i>lValue</i> member.
<i>bstrAnnotation</i>	String (ByVal). The comment for the promotion.
<i>sProcessStateToPromoteTo</i>	Integer (ByVal). The constant that identifies the level to which the process unit is being promoted. Use one of the level constants listed in Table 136 on page 853 .
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns the constant that identifies the level to which the process unit has been promoted. If the call to <code>Promote</code> succeeds, this should be equal to the constant passed in the <i>sProcessStateToPromoteTo</i> argument.

Example

The following subroutine promotes process units and attaches documents. The subroutine takes the following arguments:

- The process unit's dimension member IDs.
- The ID of the role that corresponds to the review level immediately beneath the level to which the process unit is being promoted.
- The constant that identifies the level to which the process unit is being promoted.
- The names and paths of the documents to attach.

`IHsvDataSecurity.GetProcessUnitAccessRights` gets the connected user's access rights to the process unit. If the user has All access,

`HsvSecurityAccess.IsConnectedUserAllowedToPerformTask` tests whether the connected user has been assigned to the role passed to the subroutine; if this returns TRUE, `Promote2` is called.

```
Sub ProcPromoteAttach(lScen As Long, lYear As Long, lPer As Long, _
    lEnt As Long, lPar As Long, lVal As Long, lRoleID As Long, _
    saPaths() As String, saNames() As String, iState As Integer)
Dim cProcessFlow As HsvProcessFlow, iRetState As Integer
Dim bInRole As Boolean, lUnitRights As Long
Dim cSecurity As HsvSecurityAccess, cDataSec As IHsvDataSecurity
'g_cSession is an HsvSession object reference
Set cProcessFlow = g_cSession.ProcessFlow
Set cSecurity = g_cSession.Security
Set cDataSec = g_cSession.Security
cDataSec.GetProcessUnitAccessRights lScen, lYear, lPer, lEnt, _
    lPar, lVal, lUnitRights
If lUnitRights = HFM_ACCESS_RIGHTS_ALL Then
    cSecurity.IsConnectedUserAllowedToPerformTask lRoleID, bInRole
    If bInRole = True Then
        cProcessFlow.Promote2 lScen, lYear, lPer, lEnt, lPar, lVal, _
            False, "", iState, saPaths, saNames, iRetState
    End If
End If
End Sub
```

Publish

Publishes a process unit.

To attach documents when publishing, use [Publish2](#). To publish multiple process units, use [PublishEx](#) or [Publish2](#).

Syntax

```
<HsvProcessFlow>.Publish lScenario, lYear, lPeriod, lEntity, lParent,
lValue, vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	The member ID of the process unit's Period dimension member.
<i>lEntity</i>	The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are published. Pass TRUE to publish related process units, FALSE to publish only the process unit for the <i>lValue</i> member.

Argument	Description
<i>bstrAnnotation</i>	The comment for the publishing of the process unit.
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_PUBLISHED, which is the level constant for the Published level.

This example publishes a process unit. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Publish`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cHsvProcessFlow As HsvProcessFlow, iState As Integer
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cHsvProcessFlow = m_cHsvSession.ProcessFlow
cHsvProcessFlow.Publish lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, "", iState
```

Publish2

Publishes a process unit, and optionally attaches documents and publishes all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.

Syntax

```
<HsvProcessFlow>.Publish2 lScenario, lYear, lPeriod, lEntity, lParent,
lValue, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation,
varabstrPaths, varabstrFiles, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether process units for Value dimension members related to the <i>lValue</i> member are published. Pass TRUE to publish related process units, FALSE to publish only the process unit for the <i>lValue</i> member.

Argument	Description
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). Specifies whether to publish the process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members. Pass TRUE to publish these process units, FALSE otherwise. If you specify TRUE and an error occurs for a period, Financial Management will attempt to publish the other periods.
<i>bstrAnnotation</i>	String (ByVal). The comment for the publishing of the process unit.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_PUBLISHED, which is the level constant for the Published level. This value applies to the Period and Value dimension members passed in the <i>lPeriod</i> and <i>lValue</i> arguments.

Example

This example publishes a process unit and attaches documents. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Publish2`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim chsvProcessFlow As HsvProcessFlow, iState As Integer
Dim saPaths(1) As String, saNames(1) As String
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2005")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Virginia")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'g_cSession is an HsvSession object reference
Set chsvProcessFlow = g_cSession.ProcessFlow
chsvProcessFlow.Publish2 lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, False, "see attached", saPaths, saNames, iState
```

PublishEx

Publishes a process unit, and optionally publishes all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.

To attach documents when publishing, use [Publish2](#).

Syntax

```
<HsvProcessFlow>.PublishEx lScenario, lYear, lPeriod, lEntity, lParent, lValue, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	The member ID of the process unit's Period dimension member.
<i>lEntity</i>	The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether process units for Value dimension members related to the <i>lValue</i> member are published. Pass TRUE to publish related process units, FALSE to publish only the process unit for the <i>lValue</i> member.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). Specifies whether to publish the process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members. Pass TRUE to publish these process units, FALSE otherwise. If you specify TRUE and an error occurs for a period, Financial Management will attempt to publish the other periods. Note: Specifying FALSE is the same as calling Publish .
<i>bstrAnnotation</i>	The comment for the publishing of the process unit.
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_PUBLISHED, which is the level constant for the Published level. This value applies to the Period and Value dimension members passed in the <i>lPeriod</i> and <i>lValue</i> arguments.

Example

This example publishes the specified process unit and the process units that consist of the base Period dimension members and the specified members of the other dimensions. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `PublishEx`.

```
Dim lScen As Long, lYear As Long, lPer As Long  
Dim lEnt As Long, lPar As Long, lVal As Long
```

```

Dim cHsvProcessFlow As HsvProcessFlow, iState As Integer
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cHsvProcessFlow = m_cHsvSession.ProcessFlow
cHsvProcessFlow.PublishEx lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, True, "", iState

```

Reject

Demotes a process unit from its current level to its previous level.

To attach documents when rejecting, use [Reject2](#).

Syntax

```

<HsvProcessFlow>.Reject lScenario, lYear, lPeriod, lEntity, lParent,
lValue, vbUseAllValueMembers, bstrAnnotation, psNewProcessState

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are demoted. Pass TRUE to demote related process units, FALSE to demote only the process unit for the <i>lValue</i> member.
<i>bstrAnnotation</i>	String (ByVal). The comment for the rejection.
<i>psNewProcessState</i>	Integer. Returns the constant that identifies the level to which the process unit has been demoted. For a listing of valid return values, see Table 136 on page 853 .

Example

This example demotes a process unit. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Reject`.

```

Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cHsvProcessFlow As HsvProcessFlow, iState As Integer
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")

```

```

lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cHsvProcessFlow = m_cHsvSession.ProcessFlow
cHsvProcessFlow.Reject lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, "", iState

```

Reject2

Demotes a process unit from its current level to its previous level and optionally attaches documents.

Syntax

```

<HsvProcessFlow>.Reject2 lScenario, lYear, lPeriod, lEntity, lParent,
lValue, vbUseAllValueMembers, bstrAnnotation, varabstrPaths,
varabstrFiles, psNewProcessState

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are demoted. Pass TRUE to demote related process units, FALSE to demote only the process unit for the <i>lValue</i> member.
<i>bstrAnnotation</i>	String (ByVal). The comment for the rejection.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). The file names of the documents to attach. 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.
<i>psNewProcessState</i>	Integer. Returns the constant that identifies the level to which the process unit has been demoted. For a listing of valid return values, see Table 136 on page 853 .

Example

This example demotes a process unit and attaches documents. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Reject2`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cProcessFlow As HsvProcessFlow, iState As Integer
Dim saPaths(1) As String, saNames(1) As String
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2005")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Virginia")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'g_cSession is an HsvSession object reference
Set cProcessFlow = g_cSession.ProcessFlow
cProcessFlow.Reject2 lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, "see attached", saPaths, saNames, iState
```

SignOff

Signs off on a process unit.

To attach documents when signing off, use [SignOff2](#).

Syntax

```
<HsvProcessFlow>.SignOff lScenario, lYear, lPeriod, lEntity, lParent,
lValue, vbUseAllValueMembers, bstrAnnotation
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are signed off on. Pass TRUE to sign off on related process units, FALSE to sign off on only the process unit for the <i>lValue</i> member.

Argument	Description
<i>bstrAnnotation</i>	String (ByVal). The comment for the signoff.

Example

The following example prompts the user for a comment, then signs off on a process unit. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Signoff`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cHsvProcessFlow As HsvProcessFlow
Dim sNote As String, iState As Integer
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cHsvProcessFlow = m_cHsvSession.ProcessFlow
sNote = InputBox("Enter comment.", "Comment")
cHsvProcessFlow.SignOff lScen, lYear, lPer, lEnt, lPar, lVal, _
False, sNote
```

SignOff2

Signs off on a process unit and attaches documents.

Syntax

```
<HsvProcessFlow>.SignOff2 lScenario, lYear, lPeriod, lEntity, lParent,
lValue, vbUseAllValueMembers, bstrAnnotation, varabstrPaths, varabstrFiles
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are signed off on. Pass TRUE to sign off on related process units, FALSE to sign off on only the process unit for the <i>lValue</i> member.
<i>bstrAnnotation</i>	String (ByVal). The comment for the signoff.

Argument	Description
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.

Example

The following example signs off on a process unit and attaches documents. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Signoff2`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cProcessFlow As HsvProcessFlow, iState As Integer
Dim saPaths(1) As String, saNames(1) As String
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2005")
lPer = GetMemberID(DIMENSIONPERIOD, "October")
lEnt = GetMemberID(DIMENSIONENTITY, "Virginia")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'g_cSession is an HsvSession object reference
Set cProcessFlow = g_cSession.ProcessFlow
cProcessFlow.SignOff2 lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, "see attached", saPaths, saNames
```

Start

Starts a process unit

To attach documents when starting, use [Start2](#). To start multiple process units, use [StartEx](#) or [Start2](#).

Syntax

```
<HsvProcessFlow>.Start lScenario, lYear, lPeriod, lEntity, lParent, lValue,
vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.

Argument	Description
<i>Year</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>IValue</i> member are to be started. Pass TRUE to start related process units, FALSE to start only the process unit for the <i>IValue</i> member.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process unit.
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_FIRST_PASS, which is the level constant for the First Pass level.

Example

This example creates a function that takes a process unit's member IDs and comment string, and calls *Start* if certain conditions are met. The function tests for the following conditions:

- *HsvScenarios.SupportsProcessFlow* is used to test whether the scenario is enabled for process management.
- *IHsvDataSecurity.GetProcessUnitAccessRightsAndState* is used to test whether the connected user has All access rights and whether the process unit's state is Not Started.

If all of these conditions are met, *Start* is called and the function — named *StartWorkflow* — returns a blank String. If a condition is not met, *StartWorkflow* returns a String describing the unmet condition.

```
Function StartWorkflow(lScen As Long, lYear As Long, _
lPer As Long, lEnt As Long, lPar As Long, lVal As Long, _
sNote As String) As String
Dim cProcessFlow As HsvProcessFlow, lRights As Long
Dim iState As Integer, iNewState As Integer, bEnabled As Boolean
Dim cDataSec As IHsvDataSecurity, cScenario As HsvScenarios
Set cDataSec = m_cSession.Security
Set cProcessFlow = m_cSession.ProcessFlow
Set cScenario = m_cMetadata.Scenarios
cScenario.SupportsProcessFlow lScen, bEnabled
If bEnabled = True Then
cDataSec.GetProcessUnitAccessRightsAndState lScen, _
lYear, lPer, lEnt, lPar, lVal, lRights, iState
If lRights = HFM_ACCESS_RIGHTS_ALL Then
If iState = 1 Then
cProcessFlow.Start lScen, lYear, lPer, lEnt, lPar, _
lVal, False, sNote, iNewState
StartWorkflow = ""
Else
StartWorkflow = "Invalid state; state must be Not Started"
End If
End If
```

```

Else
    StartWorkflow = "User does not have All access rights"
End If
Else
    StartWorkflow = "Scenario not enabled for process management"
End If
End Function

```

Start2

Starts a process unit, and optionally attaches documents and starts all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.

Syntax

```

<HsvProcessFlow>.Start2 lScenario, lYear, lPeriod, lEntity, lParent,
lValue, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation,
varabstrPaths, varabstrFiles, psNewProcessState

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are to be started. Pass TRUE to start related process units, FALSE to start only the process unit for the <i>lValue</i> member.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). Specifies whether to start the process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members. Pass TRUE to start these process units, FALSE otherwise. If you specify TRUE and an error occurs for a period, Financial Management will attempt to start the other periods.
<i>bstrAnnotation</i>	String (ByVal). The comment for the process unit.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.

Argument	Description
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_FIRST_PASS, which is the level constant for the First Pass level. This value applies to the Period and Value dimension members passed in the <i>lPeriod</i> and <i>lValue</i> arguments

Example

This example starts a process unit and attaches two files. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Start2`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cHsvProcessFlow As HsvProcessFlow, iState As Integer
Dim saPaths(1) As String, saNames(1) As String
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2005")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "Virginia")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'g_cSession is an HsvSession object reference
Set cHsvProcessFlow = g_cSession.ProcessFlow
cHsvProcessFlow.Start2 lScen, lYear, lPer, lEnt, lPar, lVal, _
    False, False, "see attachments", saPaths, saNames, iState
```

StartEx

Starts a process unit, and optionally starts all other process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members.

To attach documents when starting, use [Start2](#).

Syntax

```
<HsvProcessFlow>.StartEx lScenario, lYear, lPeriod, lEntity, lParent,
lValue, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation,
psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.

Argument	Description
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Specifies whether process units for Value dimension members related to the <i>lValue</i> member are to be started. Pass TRUE to start related process units, FALSE to start only the process unit for the <i>lValue</i> member.
<i>vbApplyToAllPeriods</i>	Boolean (ByVal). Specifies whether to start the process units that consist of the specified Scenario, Year, Entity, and Value dimension members and the other base Period dimension members. Pass TRUE to start these process units, FALSE otherwise. If you specify TRUE and an error occurs for a period, Financial Management will attempt to start the other periods. Note: Specifying FALSE is the same as calling <i>Start</i> .
<i>bstrAnnotation</i>	String (ByVal). The comment for the process unit.
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_FIRST_PASS, which is the level constant for the First Pass level. This value applies to the Period and Value dimension members passed in the <i>lPeriod</i> and <i>lValue</i> arguments.

Example

This example starts the specified process unit and the process units that consist of the base Period dimension members and the specified members of the other dimensions. The calls to the user-defined *GetMemberID* function get the process unit's member IDs; for information on *GetMemberID*, see the [Examples](#) for *GetItemID*. These member IDs are then passed to *StartEx*.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cHsvProcessFlow As HsvProcessFlow, iState As Integer
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2000")
lPer = GetMemberID(DIMENSIONPERIOD, "October")
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
Set cHsvProcessFlow = m_cHsvSession.ProcessFlow
cHsvProcessFlow.StartEx lScen, lYear, lPer, lEnt, lPar, lVal, _
False, True, "Champions", iState
```

Submit

Submits a process unit.

To attach documents when submitting, use [Submit2](#).

Syntax

```
<HsvProcessFlow>.Submit lScenario, lYear, lPeriod, lEntity, lParent,  
lValue, vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

Argument	Description
<i>lScenario</i>	The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	The member ID of the process unit's Period dimension member.
<i>lEntity</i>	The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are submitted. Pass TRUE to submit related process units, FALSE to submit only the process unit for the <i>lValue</i> member.
<i>bstrAnnotation</i>	The comment for the submission.
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_SUBMITTED, which is the level constant for the Submitted level.

This example submits a process unit. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Submit`.

```
Dim lScen As Long, lYear As Long, lPer As Long  
Dim lEnt As Long, lPar As Long, lVal As Long  
Dim cHsvProcessFlow As HsvProcessFlow, iState As Integer  
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")  
lYear = GetMemberID(DIMENSIONYEAR, "2000")  
lPer = GetMemberID(DIMENSIONPERIOD, "October")  
lEnt = GetMemberID(DIMENSIONENTITY, "Connecticut")  
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")  
lVal = GetMemberID(DIMENSIONVALUE, "USD")  
'g_cSession is an HsvSession object reference  
Set cHsvProcessFlow = g_cSession.ProcessFlow  
cHsvProcessFlow.Submit lScen, lYear, lPer, lEnt, lPar, lVal, _  
    False, "", iState
```

Submit2

Submits a process unit and provides the option to attach one or more documents.

Syntax

```
<HsvProcessFlow>.Submit2 lScenario, lYear, lPeriod, lEntity, lParent,  
lValue, vbUseAllValueMembers, bstrAnnotation, varabstrPaths,  
varabstrFiles, psNewProcessState
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the process unit's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the process unit's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the process unit's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the process unit's child Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the process unit's parent Entity dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the process unit's Value dimension member.
<i>vbUseAllValueMembers</i>	Boolean (ByVal). Determines whether process units for Value dimension members related to the <i>lValue</i> member are submitted. Pass TRUE to submit related process units, FALSE to submit only the process unit for the <i>lValue</i> member.
<i>bstrAnnotation</i>	String (ByVal). The comment for the submission.
<i>varabstrPaths</i>	String array (ByVal). 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.
<i>varabstrFiles</i>	String array (ByVal). 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.
<i>psNewProcessState</i>	Integer. Returns PROCESS_FLOW_STATE_SUBMITTED, which is the level constant for the Submitted level.

Example

This example submits a process unit and attaches documents. The calls to the user-defined `GetMemberID` function get the process unit's member IDs; for information on `GetMemberID`, see the [Examples](#) for `GetItemID`. These member IDs are then passed to `Submit2`.

```
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
Dim cHsvProcessFlow As HsvProcessFlow, iState As Integer
Dim saPaths(1) As String, saNames(1) As String
lScen = GetMemberID(DIMENSIONSCENARIO, "Budget")
lYear = GetMemberID(DIMENSIONYEAR, "2004")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "NewYork")
lPar = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lVal = GetMemberID(DIMENSIONVALUE, "USD")
saPaths(0) = "docs"
saPaths(1) = "docs"
saNames(0) = "Audit.txt"
saNames(1) = "Comments.txt"
'g_cSession is an HsvSession object reference
Set cHsvProcessFlow = g_cSession.ProcessFlow
cHsvProcessFlow.Submit2 lScen, lYear, lPer, lEnt, lPar, lVal, _
```


False, "see attachments", saPaths, saNames, iState

This chapter describes the methods of the HsvReports type library. You can use these methods to create and delete reports and to get report information.

Note: The HsvReports methods apply only to reports on an application server, not to reports saved on client PCs.

To use the HsvReports type library, you must reference `HsvReports.dll` in your project.

The HsvReports type library contains one object—the HsvReports object. The HsvReports object’s methods are summarized in [Table 30 on page 92](#), and are described in detail in the following topics.

Assign HsvReports object references with the `Reports` property of the HsvSession object. For an example, see [“HsvReports Type Library Overview” on page 91](#).

CheckReportSecurityClass

Indicates whether the connected user has either Read or All access rights to a report’s security class. If the user does not have access rights, a non-zero error number is thrown as an HRESULT. If the user has access rights, 0 is thrown as an HRESULT.

Tip: In Visual Basic, check for the return value with `Err.Number`.

To check whether the user has a specific level of access rights, use [CheckReportSecurityClass2](#).

Syntax

```
<HsvReports>.CheckReportSecurityClass bstrName, nReportFileType,  
nReportType
```

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

<code>bstrName</code>	String (ByVal). The name of the report.
-----------------------	---

<code>nReportFileType</code>	Integer (ByVal). Identifies the file type of the report. Pass one of the constants listed in “Document File Type Constants” on page 869 .
------------------------------	---

Argument	Description
<i>nReportType</i>	Integer (ByVal). Identifies the report type. Pass one of the constants listed in “Document Type Constants” on page 868 that represent report types.

CheckReportSecurityClass2

Indicates whether the connected user has a given level of access rights to a report’s security class. If the user does not have access rights, a non-zero error number is thrown as an HRESULT. If the user has access rights, zero is thrown as an HRESULT.

CheckReportSecurityClass2 also checks the user’s access rights to the document’s folder. For example, if the user has access rights to the current document but not to the parent folder, CheckReportSecurityClass2 returns an error.

Tip: In Visual Basic, check for the return value with `Err.Number`.

Syntax

```
<HsvReports>.CheckReportSecurityClass2 bstrPath, bstrFile, nReportType, nReportFileType, lAccessLevel
```

Argument	Description
<i>bstrPath</i>	String (ByVal). The path of the folder containing the report. The folders in the path are delimited by backslashes (\).
<i>bstrFile</i>	String (ByVal). The name of the report.
<i>nReportType</i>	Integer (ByVal). Identifies the report type. Pass one of the constants listed in “Document Type Constants” on page 868 that represent report types.
<i>nReportFileType</i>	Integer (ByVal). Identifies the file type of the report. Pass one of the constants listed in “Document File Type Constants” on page 869 .
<i>lAccessLevel</i>	Long (ByVal). The access rights for which to check. Use one of the following constants. <ul style="list-style-type: none"> ● <code>HFM_ACCESS_RIGHTS_READONLY</code>: Read access rights. ● <code>HFM_ACCESS_RIGHTS_ALL</code>: All access rights. <p>Note: These constants are members of the <code>HFMConstants</code> type library enumeration described in “Access Rights Constants” on page 856.</p>

CheckSecurityRole

Indicates whether the connected user is assigned to the security role that enables users to save reports of a given report type. If the user is not assigned to this role, a non-zero error number is thrown as an HRESULT; if the user is assigned to this role, zero is thrown as an HRESULT.

Tip: In Visual Basic, check for the return value with `Err.Number`.

Syntax

```
<HsvReports>.CheckSecurityRole nReportType
```

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

<i>nReportType</i>	Integer (ByVal). Identifies the report type. Pass one of the constants listed in “Document Type Constants” on page 868 that represent report types.
--------------------	---

DeleteDocuments

Deletes documents from the application server.

Syntax

```
<HsvReports>.DeleteDocuments varabstrPaths, varabstrNames, lDocumentType,  
lDocumentFileType, vbIncludeSubFolders
```

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

<i>varabstrPaths</i>	String array (ByVal). An array of strings that represents the hierarchical paths to the documents. The array contains an item for each path to a document.
----------------------	--

The members in these strings are delimited by backslashes (\), as shown in the following examples for a document named “Connecticut”:

```
    \Regional\UnitedStates\Connecticut
```

```
    \Management\Imbler\Connecticut
```

<i>varabstrNames</i>	String array (ByVal). The names of the documents to delete.
----------------------	---

<i>lDocumentType</i>	Long (ByVal). Identifies the type of document to be deleted. For valid values see “Document Type Constants” on page 868 .
----------------------	---

<i>lDocumentFileType</i>	Long (ByVal). Identifies the file types of the documents to be deleted. For valid values see “Document File Type Constants” on page 869 .
--------------------------	---

<i>vbIncludeSubFolders</i>	Boolean (ByVal). Determines if subdirectories are included.
----------------------------	---

DeleteReport

Deletes a report from the application server.

Syntax

```
<HsvReports>.DeleteReport bstrName, nReportFileType, nReportType
```

Argument	Description
<i>bstrName</i>	String (ByVal). The name of the report.
<i>nReportFileType</i>	Integer (ByVal). Identifies the file type of the report to be deleted. Pass one of the constants listed in “Document File Type Constants” on page 869 .
<i>nReportType</i>	Integer (ByVal). Identifies the type of report to be deleted. Pass one of the constants listed in “Document Type Constants” on page 868 that represent report types.

Example

DeleteReport is used in the [Example](#) for SetReport.

DeleteReports

Deletes one or more reports from the application server.

Syntax

```
<HsvReports>.DeleteReports varabstrFormNames, nReportFileType,
nReportType, pvaralRc
```

Argument	Description
<i>varabstrFormNames</i>	String array (ByVal). The names of the reports.
<i>nReportFileType</i>	Integer (ByVal). Identifies the file type of the report to be deleted. Pass one of the constants listed in “Document File Type Constants” on page 869 .
<i>nReportType</i>	Integer (ByVal). Identifies the type of report to be deleted. Pass one of the constants listed in “Document Type Constants” on page 868 that represent report types.
<i>pvaralRc</i>	Variant array. Returns any error numbers thrown by DeleteReports. The following hexadecimal error numbers are explicitly raised by Financial Management: <ul style="list-style-type: none"> ● 0x399. Indicates that the user does not have All security access. ● 0x7c6. Indicates that the user has not been assigned the proper role for deleting reports. Note that other error codes may be thrown if Financial Management has been improperly installed; these would be errors thrown by Windows or by the database.

Example

DeleteReports is used in the [Example](#) for EnumReports.

EnumDocuments

Returns the names, descriptions, timestamps, and security class IDs of documents that meet the search criteria.

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

Tip: `EnumDocumentsEx` is a similar method that returns additional information such as the documents' owners and document types. `EnumDocumentsEx` also enables you to filter for public or private documents.

Syntax

```
<HsvReports>.EnumDocuments(bstrPath, lDocumentType, lDocumentFileType,  
vbFilterByCurrentUserOnly, vbFilterByCreateTime, dStartTime, dStopTime,  
pvarabstrDescriptions, pvaradTimestamp, pvaralSecurityClass)
```

Argument	Description
<i>bstrPath</i>	String (ByVal). The path of the folder containing the documents. The folders in the path are delimited by backslashes (\).
<i>lDocumentType</i>	Long (ByVal). Identifies the type of document to return. Valid values are represented by the <code>HFMConstants</code> type library constants listed in “Document Type Constants” on page 868 .
<i>lDocumentFileType</i>	Long (ByVal). Identifies the file types of the documents to return. Valid values are represented by the <code>HFMConstants</code> type library constants listed in “Document File Type Constants” on page 869 .
<i>vbFilterByCurrentUserOnly</i>	Boolean (ByVal). Determines whether only documents created by the connected user are returned. Pass <code>TRUE</code> to return only documents created by the connected user, <code>FALSE</code> otherwise.
<i>vbFilterByCreateTime</i>	Boolean (ByVal). <i>For internal use.</i>
<i>dStartTime</i>	Double (ByVal). <i>For internal use.</i>
<i>dStopTime</i>	Double (ByVal). <i>For internal use.</i>
<i>pvarabstrDescriptions</i>	Variant array. Returns the descriptions of the documents. The array is returned as a String subtype.
<i>pvaradTimestamp</i>	Variant array. Returns the timestamps of the documents. The array is returned as a Double subtype; the array elements are formatted so that they can be cast to the Date data type.
<i>pvaralSecurityClass</i>	Variant array. Returns the documents' security class IDs. The array is returned as a Long subtype.

Return Value

Variant array. Returns the names of the documents that match the search criteria. The array is returned as a String subtype.

Example

The following function returns the names of all data forms in a given folder.

```
Function getWebFormNames(sPath As String) As Variant  
Dim vaRet, vaDescs, vaTimes, vaSec  
'm_cReports is an HsvReports object reference  
vaRet = m_cReports.EnumDocuments(sPath, WEBOM_DOCTYPE_WEBFORM, _
```

```

    WEBOM_DOCFILETYPE_FORMDEF, False, False, 0, 0, vaDescs, _
    vaTimes, vaSec)
getWebFormNames = vaRet
End Function

```

EnumDocumentsEx

Returns the names, descriptions, timestamps, security class IDs, privacy flags, folder content types, owners, file types, and document types of documents that meet the search criteria. You can also filter for public or private documents.

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

Tip: [EnumDocuments](#) is a similar method that returns only the documents' names, descriptions, timestamps, and security classes.

Syntax

```

<HsvReports>.EnumDocumentsEx(bstrPath, varalDocumentType,
varalDocumentFileType, vbFilterByCreateTime, dStartTime, dStopTime,
lShowPrivateDocs, pvarabstrDescriptions, pvaradTimestamp,
pvaralSecurityClass, pvarabIsPrivate, pvaralFolderContentType,
pvarabstrDocOwner, pvaralFileType, pvaralReportType)

```

Argument	Description
<i>bstrPath</i>	String (ByVal). The path of the folder containing the documents. The folders in the path are delimited by backslashes (\).
<i>varalDocumentType</i>	Variant (ByVal). Identifies the type of document to return. Valid values are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 868 .
<i>varalDocumentFileType</i>	Variant (ByVal). 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 869 .
<i>vbFilterByCreateTime</i>	Boolean (ByVal). <i>For internal use.</i>
<i>dStartTime</i>	Double (ByVal). <i>For internal use.</i>
<i>dStopTime</i>	Double (ByVal). <i>For internal use.</i>
<i>lShowPrivateDocs</i>	Long (ByVal). 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 878 .
<i>pvarabstrDescriptions</i>	Variant array. Returns the descriptions of the documents. The array is returned as a String subtype.
<i>pvaradTimestamp</i>	Variant array. Returns the timestamps of the documents. The array is returned as a Double subtype; the array elements are formatted so that they can be cast to the Date data type.

Argument	Description
<i>pvaralSecurityClass</i>	Variant array. Returns the documents' security class IDs. The array is returned as a Long subtype.
<i>pvarabIsPrivate</i>	Variant array. Indicates whether the documents are flagged as public or private. The array is returned as a Boolean subtype.
<i>pvaralFolderContentType</i>	Variant array. 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 for array items are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 868 . The array is returned as a Long subtype.
<i>pvarabstrDocOwner</i>	Variant array. Returns an array containing the usernames of the documents' owners. The array is returned as a String subtype.
<i>pvaralFileType</i>	Variant array. 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 869 . The array is returned as a Long subtype.
<i>pvaralReportType</i>	Variant array. 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 868 . The array is returned as a Long subtype.

Return Value

Variant array. Returns the names of the documents that match the search criteria. The array is returned as a String subtype.

Example

The following function returns a two-dimensional array containing the names and owners of the public data form in a given folder.

```
Function getPubWebFormNamesOwners(sPath As String) As Variant
Dim vaNames, vaDescs, vaTimes, vaSec, vaPrivate, vaContent
Dim vaOwners, vaFileType, vaReportType, vaRet()
'm_cReports is an HsvReports object reference
vaNames = m_cReports.EnumDocumentsEx(sPath, _
    WEBOM_DOCTYPE_WEBFORM, WEBOM_DOCFILETYPE_FORMDEF, False, 0, _
    0, ENUMSHOWPRIVATEDOCS_DONTSHOW, vaDescs, vaTimes, vaSec, _
    vaPrivate, vaContent, vaOwners, vaFileType, vaReportType)
ReDim vaRet(UBound(vaNames), 1)
For i = LBound(vaNames) To UBound(vaNames)
    vaRet(i, 0) = vaNames(i)
    vaRet(i, 1) = vaOwners(i)
Next
getPubWebFormNamesOwners = vaRet
End Function
```

EnumReports

Returns the names, descriptions, and timestamps of reports on the application server.

Syntax

```
<HsvReports>.EnumReports nReportFileType, vbFilterByCurrentUserOnly,  
vbFilterByCreateTime, dStartTime, dStopTime, vbFilterByReportType,  
nReportType, pvarabstrNames, pvarabstrDescriptions, pvaradTimestamp
```

Argument	Description
<i>nReportFileType</i>	Integer (ByVal). Identifies the file type of the report for which you want to return information. Pass one of the constants listed in “Document File Type Constants” on page 869 .
<i>vbFilterByCurrentUserOnly</i>	Boolean (ByVal). Determines whether only reports created by the connected user will be returned. Pass TRUE to only return reports created by the connected user, FALSE otherwise.
<i>vbFilterByCreateTime</i>	Boolean (ByVal). <i>For internal use.</i>
<i>dStartTime</i>	Double (ByVal). <i>For internal use.</i>
<i>dStopTime</i>	Double (ByVal). <i>For internal use.</i>
<i>vbFilterByReportType</i>	Boolean (ByVal). Determines whether reports of all types will be returned. Pass TRUE to return only the type of report identified by the <i>nReportType</i> argument, FALSE to return all types of reports.
<i>nReportType</i>	Integer (ByVal). If you pass TRUE for the <i>vbFilterByReportType</i> argument, this argument determines the type of report for which information will be returned. Pass one of the report type constants listed in “Document Type Constants” on page 868 that represent report types.
<i>pvarabstrNames</i>	Variant array. Returns the names of the reports that match the criteria specified in <i>EnumReports</i> ' arguments. The array is returned as a String subtype.
<i>pvarabstrDescriptions</i>	Variant array. Returns the descriptions of the reports. The array is returned as a String subtype.
<i>pvaradTimestamp</i>	Variant array. Returns the timestamps of the reports. The array is returned as a Double subtype; the array elements are formatted so that they can be cast to the Date data type.

Example

The following example deletes the reports that meet all of the following criteria:

- Reports created by the connected user
- Journal reports
- Report definition file type

EnumReports returns the names of all the reports that meet these criteria. The report names are then passed to *DeleteReports*.

```
Dim dOne As Double, dTwo As Double, vaNames, vaDescs
```

```

Dim vaTime, vaRet
'm_cReports is an HsvReports object reference
m_cReports.EnumReports WEBOM_DOCFILETYPE_RPTDEF, True, False, _
    dOne, dTwo, True, WEBOM_DOCTYPE_RPTJOURNAL, vaNames, _
    vaDescs, vaTime
m_cReports.DeleteReports vaNames, WEBOM_DOCFILETYPE_RPTDEF, _
    WEBOM_DOCTYPE_RPTJOURNAL, vaRet

```

GetDocument

Returns the definition of a document, as well as the document's security class and description.

Syntax

```

<HsvReports>.GetDocument(bstrPath, bstrName, lDocumentType,
lDocumentFileType, pbstrDescription, plSecurityClass)

```

Argument	Description
<i>bstrPath</i>	String (ByVal). The path to the document.
<i>bstrName</i>	String (ByVal). The name of the document.
<i>lDocumentType</i>	Long (ByVal). Identifies the type of document to return. Valid values are represented by the HFMConstants type library constants listed in "Document Type Constants" on page 868 .
<i>lDocumentFileType</i>	Long (ByVal). 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 869 .
<i>pbstrDescription</i>	String. Returns the description of the document.
<i>plSecurityClass</i>	Long. Returns the ID of the document's security class.

Return Value

Variant. Returns the document's definition.

GetDocumentEx

Returns the definition of a document, as well as other properties such as the document's type, file type, and security class.

Syntax

```

<HsvReports>.GetDocumentEx(bstrPath, bstrName, lDocumentType,
lDocumentFileType, pbstrDescription, plSecurityClass, pvbIsPrivate,
pbstrDocOwner, pdTimestamp, plFolderContentType)

```

Argument	Description
<i>bstrPath</i>	String (ByVal). The path to the document.

Argument	Description
<i>bstrName</i>	String (ByVal). The name of the document.
<i>IDocumentType</i>	Long (ByVal). Identifies the type of document to return. Valid values are represented by the HFMCConstants type library constants listed in “Document Type Constants” on page 868 .
<i>IDocumentFileType</i>	Long (ByVal). Identifies the file type of the documents to return. Valid values are represented by the HFMCConstants type library constants listed in “Document File Type Constants” on page 869 .
<i>pbstrDescription</i>	String. Returns the description of the document.
<i>plSecurityClass</i>	Long. Returns the ID of the document’s security class.
<i>pvblsPrivate</i>	Boolean. Indicates whether the document is public or private. TRUE indicates private, FALSE indicates public.
<i>pbstrDocOwner</i>	String. Returns the username of the document’s owner.
<i>pdTimestamp</i>	Double. Returns the document’s timestamp. The timestamp can be converted to a Date format; for example, in Visual Basic you can convert with <code>CDate</code> .
<i>plFolderContentType</i>	Long. 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 HFMCConstants type library constants listed in “Document Type Constants” on page 868 .

Return Value

Variant. Returns the document’s definition.

GetReport

Returns the definition of a report as well as the report’s security class ID and description. The report definition is returned as an array of bytes.

Syntax

```
<HsvReports>.GetReport bstrName, nReportFileType, nReportType,
plSecurityClass, pbstrDescription, pvarabyFile
```

Argument	Description
<i>bstrName</i>	String (ByVal). The name of the report. The report must exist on the application server.
<i>nReportFileType</i>	Integer (ByVal). Identifies the file type of the report. Pass one of the constants listed in “Document File Type Constants” on page 869 .
<i>nReportType</i>	Integer (ByVal). Identifies the type of report. Pass one of the constants listed in “Document Type Constants” on page 868 that represent report types.
<i>plSecurityClass</i>	Long. Returns the ID number of the report’s security class.

Argument	Description
	Tip: You can get the name of the security class by passing this ID to <code>GetSecurityClassLabel</code> . For details, see “GetSecurityClassLabel” on page 473 .

pbstrDescription String. Returns the description of the report.

pvarabyFile Variant array. Returns the report's definition. The array is returned as a Byte subtype.

Example

The following example prints the definition of a journal report named Monthly to Visual Basic's Immediate window.

```
Dim lSecID As Long, sDesc As String, vabyDef
'm_cReports is an HsvReports object reference
m_cReports.GetReport "Monthly", WEBOM_DOCFILETYPE_RPTDEF, _
    WEBOM_DOCTYPE_RPTJOURNAL, lSecID, sDesc, vabyDef
Debug.Print CStr(vabyDef)
```

SaveDocument

Saves a document on the application server.

Tip: [SaveDocumentEx](#) also saves a document, and provides additional options to specify the document's content type and privacy flag.

Syntax

```
<HsvReports>.SaveDocument bstrPath, bstrName, bstrDescription,
lDocumentType, lDocumentFileType, lSecurityClass, varabyDocument,
vbOverwriteExisting
```

Argument	Description
<i>bstrPath</i>	String (ByVal). The path of the folder containing the documents. The folders in the path are delimited by backslashes (\).
<i>bstrName</i>	String (ByVal). The name of the document.
<i>bstrDescription</i>	String (ByVal). The description of the document.
<i>lDocumentType</i>	Long (ByVal). The document type. Valid values are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 868 .
<i>lDocumentFileType</i>	Long (ByVal). The file type of the document. Valid values are represented by the HFMConstants type library constants listed in “Document File Type Constants” on page 869 .
<i>lSecurityClass</i>	Long (ByVal). The ID of the document's security class. Tip: You can get the ID from a security class name with GetSecurityClassID .
<i>varabyDocument</i>	Byte array (ByVal). The document's definition as an array of Bytes.

Argument	Description
<i>vbOverwriteExisting</i>	Boolean (ByVal). Determines whether to overwrite a document of the same name on the application server. Pass TRUE to overwrite existing documents, FALSE otherwise.

Example

The following subroutine saves the specified journal report in the .RPT format. Note that the subroutine uses `HsvSecurityAccess.GetSecurityClassID` to get the ID of the security class that is passed.

```
Sub saveJournalRpt(sPath As String, sName As String, _
    sDesc As String, sSecClass As String, baDoc() As Byte)
Dim cSecurityAccess As HsvSecurityAccess, lSecID As Long
'm_cSession is an HsvSecurityAccess object reference
Set cSecurityAccess = m_cSession.Security
'm_cReports is an HsvReports object reference
cSecurityAccess.GetSecurityClassID sSecClass, lSecID
m_cReports.SaveDocument sPath, sName, sDesc, _
    WEBOM_DOCTYPE_RPTJOURNAL, WEBOM_DOCFILETYPE_RPTDEF, lSecID, _
    baDoc, True
End Sub
```

SaveDocumentEx

Saves a document to the application server and specifies the document's content type and privacy flag.

Tip: To save a document without specifying its content type and privacy flag, use [SaveDocument](#).

Syntax

```
<HsvReports>.SaveDocumentEx bstrPath, bstrName, bstrDescription,
lDocumentType, lDocumentFileType, lSecurityClass, varabyDocument,
vbIsPrivate, lContentType, vbOverwriteExisting
```

Argument	Description
<i>bstrPath</i>	String (ByVal). The path of the folder containing the documents. The folders in the path are delimited by backslashes (\).
<i>bstrName</i>	String (ByVal). The name of the document.
<i>bstrDescription</i>	String (ByVal). The description of the document.
<i>lDocumentType</i>	Long (ByVal). The document type. Valid values are represented by the HFMConstants type library constants listed in " Document Type Constants " on page 868.
<i>lDocumentFileType</i>	Long (ByVal). The file type of the document. Valid values are represented by the HFMConstants type library constants listed in " Document File Type Constants " on page 869.

Argument	Description
<i>ISecurityClass</i>	Long (ByVal). The ID of the document's security class. Tip: You can get the ID from a security class name with GetSecurityClassID .
<i>varabyDocument</i>	Byte array (ByVal). The document's definition as an array of Bytes.
<i>vblsPrivate</i>	Boolean (ByVal). Specifies whether the folder is public or private. Pass TRUE for private, FALSE for public.
<i>IContentType</i>	Long (ByVal). 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 <code>SaveDocumentEx</code> will ignore this argument's value in that case. Valid values are represented by the <code>HFMConstants</code> type library constants listed in "Document Type Constants" on page 868 .
<i>vbOverwriteExisting</i>	Boolean (ByVal). Determines whether to overwrite a document of the same name on the application server. Pass TRUE to overwrite existing documents, FALSE otherwise.

SetReport

Saves a report on the application server.

Syntax

```
<HsvReports>.SetReport bstrName, nReportFileType, nReportType,
lSecurityClass, bstrDescription, varabyFile, vbOverwriteExisting
```

Argument	Description
<i>bstrName</i>	String (ByVal). The name of the report.
<i>nReportFileType</i>	Integer (ByVal). Identifies the file type of the report. Pass one of the constants listed in "Document File Type Constants" on page 869 .
<i>nReportType</i>	Integer (ByVal). Identifies the type of report. Pass one of the constants listed in "Document Type Constants" on page 868 that represent report types.
<i>ISecurityClass</i>	Long (ByVal). The ID number of the security class for the report. Tip: You can get a security class's ID number by passing its label to <code>GetSecurityClassID</code> . For details, see "GetSecurityClassID" on page 472 .
<i>bstrDescription</i>	String (ByVal). The description of the report.
<i>varabyFile</i>	Byte array (ByVal). The definition of the report. Pass the report definition as an array of bytes. You must pass a valid report definition. For details on report definitions, see the <i>Oracle Hyperion Financial Management, Fusion Edition User's Guide</i> .
<i>vbOverwriteExisting</i>	Boolean (ByVal). If a report with the name passed as the <i>bstrName</i> argument exists, this determines whether the existing report will be overwritten. Pass TRUE to overwrite an existing report, FALSE otherwise.

Example

The following example creates a subroutine that changes the security class of an existing report. The subroutine, named `setRptSecClass`, takes the existing report's name, file type, and report type, as well as the name of the desired security class. `GetReport` returns the report's information, and the report's current security class is then compared to the desired security class. If the security classes differ, the `If` structure deletes the report, then creates the new report with `SetReport`. Note that `SetReport` takes most of the report information returned by `GetReport`, with the only difference being that `SetReport` takes the ID of the security class passed to the subroutine.

```
Private Sub setRptSecClass(sName As String, lFile As Long, _
    lRpt As Long, sNewSec As String)
    Dim cSecurity As HsvSecurityAccess, lSecID As Long
    Dim sDesc As String, sSecName As String, vabDef, lNewID As Long
    Set cSecurity = m_cSession.Security
    'm_cReports is an HsvReports object reference
    m_cReports.GetReport sName, lFile, lRpt, lSecID, sDesc, vabDef
    cSecurity.GetSecurityClassLabel lSecID, sSecName
    If sSecName <> sNewSec Then
        cSecurity.GetSecurityClassID sNewSec, lNewID
        m_cReports.DeleteReport sName, lFile, lRpt
        m_cReports.SetReport sName, lFile, lRpt, lNewID, sDesc, _
            vabDef, True
    End If
End Sub
```

Note: The subroutine uses `HsvSecurityAccess` methods to get the name of the existing report's security class and to get the ID of the desired security class. For details on these methods, see [“GetSecurityClassLabel” on page 473](#) and [“GetSecurityClassID” on page 472](#).

In This Chapter

Types of Indexes	610
HsvMDDataBuffer Object Methods	610
HsvMDDataBufferLite Object Methods	639
IHsvMDDataBufferLite Interface	667
HsvTransactionData Object Methods	667
HsvMDIndexList Object Methods	673
HsvICTransactionsData Object Methods	674

This chapter describes the members of the HsvMDArrays type library. This type library provides objects that supplement the HsvData object by enabling you to manage data in arrays of cells, as well as an object that enables you to obtain the transaction data generated by statutory consolidations. This library also provides helper methods for intercompany transactions.

To use the HsvMDArrays type library, you must reference `HsvMDArrays.dll` in your project. The HsvMDArrays type library contains the following supported objects and interfaces:

- HsvMDDataBuffer object. Used to set, get, and enumerate cells, caching the cells' subcubes to RAM.
- HsvMDDataBufferLite object. Used to set, get, and enumerate cells, caching a minimum number of subcubes to RAM and the other subcubes to disk.
- IHsvMDDataBufferLite interface. Used to change the number of subcubes that are cached to RAM for the HsvMDDataBufferLite object.
- HsvTransactionData object. Returns the transaction data generated by statutory consolidations.
- HsvMDIndexList object. Contains an array of indexes that represent points-of-view in an HsvMDDataBuffer or HsvMDDataBufferLite object.
- HsvICTransactionsData. Provides helper methods for working with intercompany transactions.

Note: The type library's HsvCubeData, HsvMDCube, and HsvMetadataSecurityBuffer objects are not supported in this release.

Types of Indexes

Many of the `HsvMDDataBuffer` and `HsvMDDataBufferLite` methods take indexes of subcubes, periods, and cells. Obtain these indexes as follows:

- Subcube indexes:
 - To return a count of the subcubes in an `HsvMDDataBuffer` or `HsvMDDataBufferLite` object, use `BeginEnumeration`.
 - To return a subcube's index, use `GetCubeIndexFromPOV`.

Tip: For more information on subcubes, see [“About Subcubes” on page 43](#).

- Period indexes:
 - To return a count of periods, use `GetNumPeriodsInCubeForData`, `GetNumPeriodsInCubeForDescriptions`, or `GetNumPeriodsInCubeForLineItems`.
 - To return a period's index, use `GetPeriodIndexFromPOVForData`, `GetPeriodIndexFromPOVForDescriptions`, or `GetPeriodIndexFromPOVForLineItems`.
- Cell indexes: To return a count of periods, use `GetNumCellsForData`, `GetNumCellsForDescriptions`, or `GetNumCellsForLineItems`.

HsvMDDataBuffer Object Methods

An `HsvMDDataBuffer` object contains an array of cells. The object provides methods for setting, getting, and enumerating three types of cells:

- Cells with data
- Cells with descriptions
- Cells with line items

These cells are not stored data; in other words, the information in an `HsvMDDataBuffer` object's cells is not stored in an Financial Management application. You move data between an `HsvMDDataBuffer` object and an application with the following methods of the `HsvData` object:

- [UpdateDataUsingMDDataBuffer](#) inserts the data in an `HsvMDDataBuffer` object into an application's cells. Use this method after you have finished working with the cells in an `HsvMDDataBuffer` object.
- [AddDataToMDDataBuffer](#) adds the data stored in an application's cell to an `HsvMDDataBuffer` object. Use this method to build an `HsvMDDataBuffer` object from data stored in applications.

The `HsvMDDataBuffer` object caches all the subcubes to RAM. If you want similar functionality, but with only a minimum number of subcubes cached to RAM and the remaining subcubes cached to disk, use the `HsvMDDataBufferLite` object. For more information, see [“HsvMDDataBufferLite Object Methods” on page 639](#).

Assign `HsvMDDataBuffer` object references with the `Set` and `New` keywords. For an example, see “[HsvMDDataBuffer Object Overview](#)” on page 93.

The `HsvMDDataBuffer` object’s methods are summarized in [Table 31 on page 94](#), and are described in detail in the following topics.

BeginEnumeration

Locks an `HsvMDDataBuffer` object, and returns the number of subcubes contained by the object. You can then pass the number of subcubes to methods that require a subcube index.

Caution! Call `EndEnumeration` once you have finished working with an `HsvMDDataBuffer` object, otherwise the object will not be unlocked and errors could occur.

Syntax

```
<HsvMDDataBuffer>.BeginEnumeration p1NumCubes
```

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

<code>p1NumCubes</code>	Long. Returns the number of subcubes in the <code>HsvMDDataBuffer</code> object.
-------------------------	--

Example

`BeginEnumeration` is used in the [Example](#) for `GetDataAtIndex`.

CreateDataIndexList

Creates an `HsvMDIndexList` object based upon the items in an `HsvMDDataBuffer` object.

`CreateDataIndexList` takes arguments for the Financial Management dimensions; these arguments determine which items in the `HsvMDDataBuffer` object are added to the `HsvMDIndexList` object. For each dimension, pass one of the following:

- A member ID. Passing a member ID means that only items for this member will be added to the `HsvMDIndexList` object.
- `MEMBERALL` constant. Passing `MEMBERALL` means that all items containing the dimension member will be added to the `HsvMDIndexList` object.
- `MEMBERNOTUSED` constant. Passing `MEMBERNOTUSED` means that only those items containing `MEMBERNOTUSED` for the dimension member will be added to the `HsvMDIndexList` object.
- `MEMBERANYONE` constant. Passing `MEMBERANYONE` means that if the `HsvMDDataBuffer` object contains more than one item where all other dimension members are the same, only one instance will be added to the `HsvMDIndexList` object.

Tip: These constants are in the `HFMConstants` type library. For more information, see “[Dimension Member Constants](#)” on page 834.

Syntax

```
<HsvMDDataBuffer>.CreateDataIndexList lScenario As Long, lYear As Long,
lPeriod As Long, lView As Long, lEntity As Long, lParent As Long, lValue As
Long, lAccount As Long, lICP As Long, lCustom1 As Long, lCustom2 As Long,
lCustom3 As Long, lCustom4 As Long, ppIMDIndexListUnk As Unknown
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member, or one of the constants listed above in the method's description.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member, or one of the constants listed above in the method's description.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member, or one of the constants listed above in the method's description.
<i>lView</i>	Long (ByVal). The member ID of the View dimension member, or one of the constants listed above in the method's description.
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member, or one of the constants listed above in the method's description.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity, or one of the constants listed above in the method's description.
<i>lValue</i>	Long (ByVal). Long (ByVal). The member ID of the Value dimension member, or one of the constants listed above in the method's description.
<i>lAccount</i>	Long (ByVal). Long (ByVal). The member ID of the Account dimension member, or one of the constants listed above in the method's description.
<i>lICP</i>	Long (ByVal). Long (ByVal). The member ID of the Intercompany Partner dimension member, or one of the constants listed above in the method's description.
<i>lCustom1</i>	Long (ByVal). Long (ByVal). The member ID of the Custom 1 dimension member, or one of the constants listed above in the method's description.
<i>lCustom2</i>	Long (ByVal). The member ID of the Custom 2 dimension member, or one of the constants listed above in the method's description.
<i>lCustom3</i>	Long (ByVal). The member ID of the Custom 3 dimension member, or one of the constants listed above in the method's description.
<i>lCustom4</i>	Long (ByVal). The member ID of the Custom 4 dimension member, or one of the constants listed above in the method's description.
<i>ppIMDIndexListUnk</i>	HsvMDIndexList object. Returns an HsvMDIndexList object containing the items in the HsvMDDataBuffer object that match the criteria specified in the other arguments.

Example

The following example creates an `HsvMDIndexList` object that contains all of the items in the `HsvMDDataBuffer` object. The example then loops through the `HsvMDIndexList` object:

GetNumItems returns the number of items in the object, and GetItem is called for each item. After the call to GetItem, you could operate upon the item's dimension members by replacing the ellipsis (...) with code.

```
Dim lScen As Long, lYear As Long, lPer As Long, lView As Long
Dim lEnt As Long, lPar As Long, lVal As Long, lAcct As Long
Dim lICP As Long, lCust1 As Long, lCust2 As Long, lCust3 As Long
Dim lCust4 As Long, lItems As Long, lItem As Long
m_cMDDBuffer.CreateDataIndexList MEMBERALL, MEMBERALL, _
MEMBERALL, MEMBERALL, MEMBERALL, MEMBERALL, MEMBERALL, _
MEMBERALL, MEMBERALL, MEMBERALL, MEMBERALL, MEMBERALL, _
MEMBERALL, m_cMDIndexList
m_cMDIndexList.GetNumItems lItems - 1
For lItem = 0 To lItems
    m_cMDIndexList.GetItem lItem, lScen, lYear, lPer, lView, _
    lEnt, lPar, lVal, lAcct, lICP, lCust1, lCust2, lCust3, _
    lCust4
    ... 'Insert code here.
Next lItem
```

CreateDataIndexListEx

For internal use.

EndEnumeration

Unlocks an HsvMDDDataBuffer object that has been locked by BeginEnumeration.

Syntax

```
<HsvMDDDataBuffer>.EndEnumeration
```

Example

EndEnumeration is used in the [Example](#) for GetDataAtIndex.

EraseRecordFromPMBuffer

For internal use.

GetCheckLineItemDetailsForCaseInsensitiveDuplicates

Indicates whether case-insensitive duplicate checking is enabled for the HsvMDDDataBuffer instance. *acmeData* and *ACMEDATA* would be considered case-insensitive duplicate descriptions.

Tip: [SetCheckLineItemDetailsForCaseInsensitiveDuplicates](#) enables or disables whether the system should check for case-insensitive duplicate line item descriptions.

Syntax

```
<HsvMDDataBuffer>.GetCheckLineItemDetailsForCaseInsensitive  
Duplicates pvbEnabled
```

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

<i>pvbEnabled</i>	Boolean. Indicates whether case-insensitive duplicate checking is enabled. Returns TRUE if enabled, FALSE otherwise.
-------------------	--

GetCubeIndexFromPOV

Returns the index of a subcube in an HsvMDDataBuffer object, given the member IDs of the subcube's dimension members.

Syntax

```
<HsvMDDataBuffer>.GetCubeIndexFromPOV lScenario, lYear, lEntity, lParent,  
lValue, plCubeIndex
```

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

<i>lScenario</i>	Long (ByVal). The member ID of the subcube's Scenario dimension member.
------------------	---

<i>lYear</i>	Long (ByVal). The member ID of the subcube's Year dimension member.
--------------	---

<i>lEntity</i>	Long (ByVal). The member ID of the subcube's Entity dimension member.
----------------	---

<i>lParent</i>	Long (ByVal). The member ID of the subcube's Parent dimension member.
----------------	---

<i>lValue</i>	Long (ByVal). The member ID of the subcube's Value dimension member.
---------------	--

<i>plCubeIndex</i>	Long. The index of the subcube, or -1 if the HsvMDDataBuffer object does not contain the subcube.
--------------------	---

Example

GetCubeIndexFromPOV is used in the [Example](#) for GetPeriodIndexFromPOVForData.

GetCubePOVFromIndex

Returns the member IDs of a subcube's dimension members, given the subcube's index.

Syntax

```
<HsvMDDataBuffer>.GetCubePOVFromIndex lCubeIndex, plScenario, plYear,  
plEntity, plParent, plValue
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube.
-------------------	---

You can use `BeginEnumeration` to return the number of subcubes in an HsvMDDataBuffer object.

Argument Description

<i>pScenario</i>	Long. Returns the member ID of the subcube's Scenario dimension member.
<i>pYear</i>	Long. Returns the member ID of the subcube's Year dimension member.
<i>pEntity</i>	Long. Returns the member ID of the subcube's Entity dimension member.
<i>pParent</i>	Long. Returns the member ID of the <i>pEntity</i> argument's parent entity.
<i>pValue</i>	Long. Returns the member ID of the subcube's Value dimension member.

Example

The following example shows how to get the member IDs of the subcubes in an `HsvMDDataBuffer` object. `BeginEnumeration` returns the number of subcubes in the object. For each subcube, `GetCubePOVFromIndex` returns the member IDs. To use the member IDs in the loop, replace the ellipsis (...) with code.

```
Dim lNumCubes As Long, lScen As Long, lYear As Long
Dim lEnt As Long, lPar As Long, lVal As Long
m_cMDDBuffer.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDDBuffer.GetCubePOVFromIndex i, lScen, lYear, lEnt, _
        lPar, lVal
    ... 'Insert code here
Next i
m_cMDDBuffer.EndEnumeration
```

GetData

Returns a cell's data, given the member IDs that identify the cell. `GetData` also indicates whether the cell is null.

Tip: To get a cell's data by passing an index, use [GetDataAtIndex](#).

Syntax

```
<HsvMDDataBuffer>.GetData lScenario, lYear, lPeriod, lView, lEntity,
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
pdData, pvbIsNoData
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member. You must pass a member ID for a user-defined member. Do not pass the member ID for the system-defined member <Scenario View>.

Argument	Description
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). Pass the HFMConstants type library constant <code>MEMBERNOTUSED</code> .
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member. You must pass a member ID for a user-defined member. Do not pass a member ID for system-defined members such as <Entity Currency>.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pdData</i>	Double. Returns the cell's data.
<i>pvIsNoData</i>	Boolean. Indicates whether the cell is null. Returns TRUE if null, FALSE otherwise.

Example

The following example returns a cell's data in the `dRetData` variable. The example assumes the cell's dimension member IDs have previously been defined.

```
Dim dRetData As Double, bRetNoData As Boolean
cMDBuffer.GetData lScen, lYear, lPer, lView, lEnt, MEMBERNOTUSED, _
    lVal, lAcct, lIcp, lCust1, lCust2, lCust3, lCust4, dRetData, bRetNoData
```

GetDataAtIndex

Returns the data for a cell, given the indexes that identify the cell. `GetDataAtIndex` also indicates whether the cell is null, and returns the member IDs of the cell's View, Account, Intercompany Partner, and Custom dimension members.

Syntax

```
<HsvMDDataBuffer>.GetDataAtIndex lCubeIndex, lPeriodIndex, lCellIndex,
plView, plAccount, plICP, plCustom1, plCustom2, plCustom3, plCustom4,
pdData, pvIsNoData
```

Argument	Description
<i>lCubeIndex</i>	Long (ByVal). The index of the cell's subcube. Use the <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. Use the <code>GetNumPeriodsInCubeForData</code> or <code>GetPeriodIndexFromPOVForData</code> to determine valid index values.
<i>lCellIndex</i>	Long (ByVal). The index of the cell. Use <code>GetNumCellsForData</code> to determine valid index values.

Argument	Description
<i>plView</i>	Long. Returns the member ID of the cell's View dimension member.
<i>plAccount</i>	Long. Returns the member ID of the cell's Account dimension member.
<i>plICP</i>	Long. Returns the member ID of the cell's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the cell's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the cell's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the cell's Custom 3 dimension member.
<i>plCustom4</i>	Long. Returns the member ID of the cell's Custom 4 dimension member.
<i>pdData</i>	Double. Returns the cell's data.
<i>pvIsNoData</i>	Boolean. Indicates whether the cell is null. Returns TRUE if null, FALSE otherwise.

Example

The following example loops through the cells in an HsvMDDataBuffer object and prints each cell's data to Visual Basic's Immediate window. The example assumes that the object has previously been filled with data. `BeginEnumeration` locks the object and returns the number of subcubes in the object. The loops then works as follows:

- For each subcube, `GetNumPeriodsInCubeForData` gets the number of periods with cells that contain data.
- For each period in a subcube, `GetNumCellsForData` gets the number of cells that contain data.
- For each cell that contains data, `GetDataAtIndex` gets the data, which is then printed to the Immediate window.

`EndEnumeration` then unlocks the HsvMDDataBuffer object.

```
Dim lNumCubes As Long, lNumPers As Long, lNumCells As Long
Dim lVw As Long, lAccount As Long, lIntCo As Long
Dim lCus1 As Long, lCus2 As Long, lCus3 As Long, lCus4 As Long
Dim dMdData As Double, bMdNoData As Boolean
m_cMDBuffer.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDBuffer.GetNumPeriodsInCubeForData i, lNumPers
    For j = 0 To lNumPers - 1
        m_cMDBuffer.GetNumCellsForData i, j, lNumCells
        For k = 0 To lNumCells - 1
            m_cMDBuffer.GetDataAtIndex i, j, k, lVw, lAccount, _
                lIntCo, lCus1, lCus2, lCus3, lCus4, dMdData, bMdNoData
            Debug.Print dMdData
        Next k
    Next j
Next i
m_cMDBuffer.EndEnumeration
```

GetDescription

Returns a cell's description, given the member IDs that identify the cell.

Tip: To get a cell's description by passing an index, use `GetDescriptionAtIndex`. For details, see [“GetDescriptionAtIndex” on page 619](#).

Syntax

```
<HsvMDDataBuffer>.GetDescription lScenario, lYear, lPeriod, lEntity,  
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,  
pbstrDescription
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <code>lEntity</code> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pbstrDescription</i>	String. Returns the cell's description.

Example

The following example returns a cell's description. The example assumes that the cell's dimension member IDs have been previously defined.

```
Dim sCellDesc As String  
m_cMDBuffer.GetDescription m_lScen, m_lYear, m_lPer, _  
m_lEnt, m_lPar, m_lVal, m_lAcct, m_lICP, m_lCust1, _  
m_lCust2, m_lCust3, m_lCust4, sCellDesc
```

GetDescriptionAtIndex

Returns the description for a cell, given the indexes that identify the cell.

`GetDescriptionAtIndex` also returns the member IDs of the cell's Account, Intercompany Partner, and Custom dimension members.

Syntax

```
<HsvMDDataBuffer>.GetDescriptionAtIndex lCubeIndex, lPeriodIndex,  
lCellIndex, plAccount, plICP, plCustom1, plCustom2, plCustom3, plCustom4,  
pbstrDesc
```

Argument Description

<i>lCubeIndex</i>	Long (ByVal). The index of the cell's subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. Use <code>GetNumPeriodsInCubeForDescriptions</code> or <code>GetPeriodIndexFromPOVForDescriptions</code> to determine valid index values.
<i>lCellIndex</i>	Long (ByVal). The index of the cell. Use <code>GetNumCellsForDescriptions</code> to determine valid index values.
<i>plAccount</i>	Long. Returns the member ID of the cell's Account dimension member.
<i>plICP</i>	Long. Returns the member ID of the cell's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the cell's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the cell's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the cell's Custom 3 dimension member.
<i>plCustom4</i>	Long. Returns the member ID of the cell's Custom 4 dimension member.
<i>pbstrDesc</i>	String. Returns the cell's description.

Example

The following example loops through the cells in an `HsvMDDataBuffer` object and prints each cell's description to Visual Basic's Immediate window. The example assumes that the object has previously been filled with data. `BeginEnumeration` locks the object and returns the number of subcubes in the object. The loops then work as follows:

- For each subcube, `GetNumPeriodsInCubeForDescriptions` gets the number of periods with cells that contain descriptions.
- For each period in a subcube, `GetNumCellsForDescriptions` gets the number of cells that contain descriptions.
- For each cell that contains data, `GetDescriptionAtIndex` gets the description, which is then printed to the Immediate window.

`EndEnumeration` then unlocks the `HsvMDDataBuffer` object.

```
Dim lNumCubes As Long, lNumPers As Long, lNumCells As Long  
Dim lAccount As Long, lIntCo As Long, lCus1 As Long
```

```

Dim lCus2 As Long, lCus3 As Long, lCus4 As Long, sDesc As String
m_cMDBuffer.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDBuffer.GetNumPeriodsInCubeForDescriptions i, lNumPers
    For j = 0 To lNumPers - 1
        m_cMDBuffer.GetNumCellsForDescriptions i, j, lNumCells
        For k = 0 To lNumCells - 1
            m_cMDBuffer.GetDescriptionAtIndex i, j, k, lAccount, _
                lIntCo, lCus1, lCus2, lCus3, lCus4, sDesc
            Debug.Print sDesc
        Next k
    Next j
Next i
m_cMDBuffer.EndEnumeration

```

GetLineItems

Returns the data and descriptions for a cell's line items, given the member IDs that identify the cell's dimension members. `GetLineItems` also indicates whether the cell's data is saved as year-to-date or periodic.

Syntax

```

<HsvMDDataBuffer>.GetLineItems lScenario, lYear, lPeriod, lEntity, lParent,
lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
pvbSavedAsYTD, pvaradData, pvarabstrDescriptions

```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <code>lEntity</code> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pvbSavedAsYTD</i>	Boolean. Indicates how the cell's data is saved. Returns TRUE if saved as year-to-date, FALSE if saved as periodic.

Argument	Description
<i>pvaradData</i>	Variant array. Returns the data for the cell's line items. The array is returned as a Double subtype.
<i>pvarabstrDescriptions</i>	Variant array. Returns the descriptions of the cell's line items. The array is returned as a String subtype.

Example

The following example returns the data and descriptions of a cell's line items. The example assumes the cell's dimension member IDs have previously been defined.

```
Dim bYTD As Boolean, vdData, vsDescs
m_cMDBuffer.GetLineItems m_lScen, m_lYear, m_lPer, _
m_lEnt, m_lPar, m_lVal, m_lAcct, m_lICP, m_lCust1, _
m_lCust2, m_lCust3, m_lCust4, bYTD, vdData, vsDescs
```

GetLineItemsAtIndex

Returns the data and descriptions for a cell's line items, given the indexes that identify the cell. `GetLineItemsAtIndex` also returns the member IDs of the cell's Account, Intercompany Partner, and Custom dimension member, and indicates whether the line items are saved as year-to-date or periodic.

Syntax

```
<HsvMDDataBuffer>.GetLineItemsAtIndex lCubeIndex, lPeriodIndex,
lCellIndex, plAccount, plICP, plCustom1, plCustom2, plCustom3, plCustom4,
pvbSavedAsYTD, pvaradData, pvarabstrDescriptions
```

Argument	Description
<i>lCubeIndex</i>	Long (ByVal). The index of the cell's subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. Use <code>GetNumPeriodsInCubeForLineItems</code> or <code>GetPeriodIndexFromPOVForLineItems</code> to determine valid index values.
<i>lCellIndex</i>	Long (ByVal). The index of the cell. Use <code>GetNumCellsForLineItems</code> to determine valid index values.
<i>plAccount</i>	Long. Returns the member ID of the cell's Account dimension member.
<i>plICP</i>	Long. Returns the member ID of the cell's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the cell's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the cell's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the cell's Custom 3 dimension member.
<i>plCustom4</i>	Long. Returns the member ID of the cell's Custom 4 dimension member.

Argument	Description
<i>pvbSavedAsYTD</i>	Boolean. Indicates how the cell is saved. Returns TRUE if the cell is saved as year-to-date, FALSE if saved as periodic.
<i>pvaradData</i>	Variant array. Returns the data for the cell's line items. The array is returned as a Double subtype.
<i>pvarabstrDescriptions</i>	Variant array. Returns the descriptions of the cell's line items. The array is returned as a String subtype.

Example

The following example loops through the cells in an HsvMDDataBuffer object and prints each line item's data and description to Visual Basic's Immediate window. The example assumes that the object has previously been filled with data. `BeginEnumeration` locks the object and returns the number of subcubes in the object. The loops then work as follows:

- For each subcube, `GetNumPeriodsInCubeForLineItems` gets the number of periods with cells that contain line items.
- For each period in a subcube, `GetNumCellsForLineItems` gets the number of cells that contain line items.
- For each cell that contains line items, `GetLineItemsAtIndex` gets the line items' data and descriptions, which are then printed to the Immediate window.

`EndEnumeration` then unlocks the HsvMDDataBuffer object.

```
Dim lNumCubes As Long, lNumPers As Long, lNumCells As Long
Dim lAccount As Long, lIntCo As Long, lCus1 As Long
Dim lCus2 As Long, lCus3 As Long, lCus4 As Long
Dim bYTD As Boolean, vdMdData, vdMdDesc
m_cMDBuffer.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDBuffer.GetNumPeriodsInCubeForLineItems i, lNumPers
    For j = 0 To lNumPers - 1
        m_cMDBuffer.GetNumCellsForLineItems i, j, lNumCells
        For k = 0 To lNumCells - 1
            m_cMDBuffer.GetLineItemsAtIndex i, j, k, lAccount, _
                lIntCo, lCus1, lCus2, lCus3, lCus4, bYTD, _
                vdMdData, vdMdDesc
            For l = LBound(vdMdData) To UBound(vdMdData)
                Debug.Print vdMdData(l) & " " & vdMdDesc(l)
            Next l
        Next k
    Next j
Next i
m_cMDBuffer.EndEnumeration
```

GetNumCellsForData

Returns the number of cells that are in a period of a subcube and that contain data.

Syntax

```
<HsvMDDataBuffer>.GetNumCellsForData lCubeIndex, lPeriodIndex, plNumCells
```

Argument	Description
<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the period. Use <code>GetNumPeriodsInCubeForData</code> or <code>GetPeriodIndexFromPOVForData</code> to determine valid index values.
<i>plNumCells</i>	Long. Returns the number of cells that contain data.

Example

`GetNumCellsForData` is used in the [Example](#) for `GetDataAtIndex`.

GetNumCellsForDescriptions

Returns the number of cells that are in a period of a subcube and that contain descriptions.

Syntax

```
<HsvMDDataBuffer>.GetNumCellsForDescriptions lCubeIndex, lPeriodIndex, plNumCells
```

Argument	Description
<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the period. Use <code>GetNumPeriodsInCubeForDescriptions</code> or <code>GetPeriodIndexFromPOVForDescriptions</code> to determine valid index values.
<i>plNumCells</i>	Long. Returns the number of cells that contain descriptions.

Example

`GetNumCellsForDescriptions` is used in the [Example](#) for `GetDescriptionAtIndex`.

GetNumCellsForLineItems

Returns the number of cells that are in a period of a subcube and that contain line items.

Syntax

```
<HsvMDDataBuffer>.GetNumCellsForLineItems lCubeIndex, lPeriodIndex, plNumCells
```

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

<i>ICubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>IPeriodIndex</i>	Long (ByVal). The index of the period. Use <code>GetNumPeriodsInCubeForLineItems</code> or <code>GetPeriodIndexFromPOVForLineItems</code> to determine valid index values.
---------------------	--

<i>pNumCells</i>	Long. Returns the number of cells that contain line items.
------------------	--

Example

`GetNumCellsForLineItems` is used in the [Example](#) for `GetLineItemsAtIndex`.

GetNumPeriodsInCubeForData

Returns the number of periods that are in a subcube and that contain cells with data.

Syntax

```
<HsvMDDataBuffer>.GetNumPeriodsInCubeForData lCubeIndex, pNumPeriods
```

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

<i>ICubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>pNumPeriods</i>	Long. Returns the number of periods with cells that contain data.
--------------------	---

Example

`GetNumPeriodsInCubeForData` is used in the [Example](#) for `GetDataAtIndex`.

GetNumPeriodsInCubeForDescriptions

Returns the number of periods that are in a subcube and that contain cells with descriptions.

Syntax

```
<HsvMDDataBuffer>.GetNumPeriodsInCubeForDescriptions lCubeIndex, pNumPeriods
```

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

<i>ICubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>pNumPeriods</i>	Long. Returns the number of periods with cells that contain descriptions.
--------------------	---

Example

`GetNumPeriodsInCubeForDescriptions` is used in the [Example](#) for `GetDescriptionAtIndex`.

GetNumPeriodsInCubeForLineItems

Returns the number of periods that are in a subcube and that contain cells with line items.

Syntax

```
<HsvMDDataBuffer>.GetNumPeriodsInCubeForLineItems lCubeIndex, plNumPeriods
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>plNumPeriods</i>	Long. Returns the number of periods with cells that contain line items.
---------------------	---

Example

`GetNumPeriodsInCubeForLineItems` is used in the [Example](#) for `GetLineItemsAtIndex`.

GetPeriodIndexFromPOVForData

Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain data.

Syntax

```
<HsvMDDataBuffer>.GetPeriodIndexFromPOVForData lCubeIndex, lPeriod, plPeriodIndex
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
----------------	---

<i>plPeriodIndex</i>	Long. Returns the period's index, or -1 if the period does not contain cells with data.
----------------------	---

Example

The following example gets a subcube's index, the index of a period in the subcube, and the number of cells in the period that contain data, then prints the cells' data to Visual Basic's Immediate window. The example assumes that the member IDs of the subcube and period have been previously defined. `GetCubeIndexFromPOV` returns the index of the subcube.

`GetPeriodIndexFromPOVForData` then gets the period's index. `GetNumCellsForData` takes the subcube and period indexes and returns the number of cells that contain data. For each cell, `GetDataAtIndex` returns the cell's data, which is then printed to the Immediate window.

```
Dim lCubeIx As Long, lPerIx As Long, lNumCells As Long
Dim lNumCubes As Long, dCellData As Double, bIsNoData As Boolean
Dim lView As Long, lAcct As Long, lICP As Long, lCust1 As Long
Dim lCust2 As Long, lCust3 As Long, lCust4 As Long
m_cMDBuffer.BeginEnumeration lNumCubes
```

```

m_cMDBuffer.GetCubeIndexFromPOV m_lScen, m_lYear, m_lEnt, _
m_lPar, m_lVal, lCubeIx
' Exit if there are no subcubes
If lCubeIx = -1 Then
    m_cMDBuffer.EndEnumeration
    Exit Sub
Else
    m_cMDBuffer.GetPeriodIndexFromPOVForData lCubeIx, _
    m_lPer, lPerIx
    ' Exit if the period does not have cells with data
    If lPerIx = -1 Then
        m_cMDBuffer.EndEnumeration
        Exit Sub
    Else
        m_cMDBuffer.GetNumCellsForData lCubeIx, _
        lPerIx, lNumCells
        For i = 0 To lNumCells - 1
            m_cMDBuffer.GetDataAtIndex lCubeIx, lPerIx, i, _
            lView, lAcct, lICP, lCust1, lCust2, lCust3, _
            lCust4, dCellData, bIsNoData
            Debug.Print dCellData
        Next i
    End If
End If
m_cMDBuffer.EndEnumeration

```

GetPeriodIndexFromPOVForDescriptions

Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain descriptions.

Syntax

```

<HsvMDDataBuffer>.GetPeriodIndexFromPOVForDescriptions lCubeIndex,
lPeriod, plPeriodIndex

```

Argument Description

lCubeIndex Long (ByVal). The index of the subcube. Use `BeginEnumeration` or `GetCubeIndexFromPOV` to determine valid index values.

lPeriod Long (ByVal). The member ID of the Period dimension member.

plPeriodIndex Long. Returns the period's index, or -1 if the period does not contain cells with descriptions.

Example

The following example gets a subcube's index, the index of a period in the subcube, and the number of cells in the period that contain descriptions, then prints the cells' descriptions to Visual Basic's Immediate window. The example assumes that the member IDs of the subcube and period have been previously defined. `GetCubeIndexFromPOV` returns the index of the subcube. `GetPeriodIndexFromPOVForDescriptions` then gets the period's index. `GetNumCellsForDescriptions` takes the subcube and period indexes and returns the number

of cells that contain descriptions. For each cell, `GetDescriptionAtIndex` returns the cell's description, which is then printed to the Immediate window.

```
Dim lCubeIx As Long, lPerIx As Long, lNumCells As Long
Dim lNumCubes As Long, sCellDesc As String
Dim lAcct As Long, lICP As Long, lCust1 As Long
Dim lCust2 As Long, lCust3 As Long, lCust4 As Long
m_cMDBuffer.BeginEnumeration lNumCubes
m_cMDBuffer.GetCubeIndexFromPOV m_lScen, m_lYear, m_lEnt, _
m_lPar, m_lVal, lCubeIx
' Exit if there are no subcubes
If lCubeIx = -1 Then
    m_cMDBuffer.EndEnumeration
    Exit Sub
Else
    m_cMDBuffer.GetPeriodIndexFromPOVForDescriptions lCubeIx, _
    m_lPer, lPerIx
    ' Exit if the period does not have cells with descriptions
    If lPerIx = -1 Then
        m_cMDBuffer.EndEnumeration
        Exit Sub
    Else
        m_cMDBuffer.GetNumCellsForDescriptions lCubeIx, _
        lPerIx, lNumCells
        For i = 0 To lNumCells - 1
            m_cMDBuffer.GetDescriptionAtIndex lCubeIx, lPerIx, i, _
            lAcct, lICP, lCust1, lCust2, lCust3, _
            lCust4, sCellDesc
            Debug.Print sCellDesc
        Next i
    End If
End If
m_cMDBuffer.EndEnumeration
```

GetPeriodIndexFromPOVForLineItems

Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain line items.

Syntax

```
<HsvMDDataBuffer>.GetPeriodIndexFromPOVForLineItems lCubeIndex, lPeriod,
plPeriodIndex
```

Argument	Description
<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>plPeriodIndex</i>	Long. Returns the period's index, or -1 if the period does not contain cells with line items.

Example

The following example gets a subcube's index, the index of a period in the subcube, and the number of cells in the period that contain line items, then prints the line items' descriptions to Visual Basic's Immediate window. The example assumes that the member IDs of the subcube and period have been previously defined. `GetCubeIndexFromPOV` returns the index of the subcube. `GetPeriodIndexFromPOVForLineItems` then gets the period's index. `GetNumCellsForLineItems` takes the subcube and period indexes and returns the number of cells that contain line items. For each cell, `GetLineItemsAtIndex` returns the line items' data and descriptions, which are then printed to the Immediate window.

```
Dim lCubeIx As Long, lPerIx As Long, lNumCells As Long
Dim lNumCubes As Long, dCellData As Double, bYTD As Boolean
Dim vdCellData, vsDescs, lView As Long, lAcct As Long
Dim lICP As Long, lCust1 As Long, lCust2 As Long, lCust3 As Long
Dim lCust4 As Long
m_cMDBuffer.BeginEnumeration lNumCubes
m_cMDBuffer.GetCubeIndexFromPOV m_lScen, m_lYear, m_lEnt, _
m_lPar, m_lVal, lCubeIx
' Exit if there are no subcubes
If lCubeIx = -1 Then
    m_cMDBuffer.EndEnumeration
    Exit Sub
Else
    m_cMDBuffer.GetPeriodIndexFromPOVForLineItems lCubeIx, _
    m_lPer, lPerIx
    ' Exit if the period does not have cells with line items
    If lPerIx = -1 Then
        m_cMDBuffer.EndEnumeration
        Exit Sub
    Else
        m_cMDBuffer.GetNumCellsForLineItems lCubeIx, _
        lPerIx, lNumCells
        For i = 0 To lNumCells - 1
            m_cMDBuffer.GetLineItemsAtIndex lCubeIx, lPerIx, i, _
            lAcct, lICP, lCust1, lCust2, lCust3, _
            lCust4, bYTD, vdCellData, vsDescs
            For j = LBound(vdCellData) To UBound(vdCellData)
                Debug.Print vdCellData(j) & " " & vsDescs(j)
            Next j
        Next i
    End If
End If
m_cMDBuffer.EndEnumeration
```

GetPeriodPOVFromIndexForData

Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain data.

Syntax

```
<HsvMDDataBuffer>.GetPeriodPOVFromIndexForData lCubeIndex, lPeriodIndex, _
plPeriod
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. To determine valid index values, use <code>GetNumPeriodsInCubeForData</code> .
---------------------	--

<i>plPeriod</i>	Long. Returns the member ID of the period.
-----------------	--

GetPeriodPOVFromIndexForDescriptions

Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain descriptions.

Syntax

```
<HsvMDDataBuffer>.GetPeriodPOVFromIndexForDescriptions lCubeIndex,  
lPeriodIndex, plPeriod
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. To determine valid index values, use <code>GetNumPeriodsInCubeForDescriptions</code> .
---------------------	--

<i>plPeriod</i>	Long. Returns the member ID of the period.
-----------------	--

GetPeriodPOVFromIndexForLineItems

Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain line items.

Syntax

```
<HsvMDDataBuffer>.GetPeriodPOVFromIndexForLineItems lCubeIndex,  
lPeriodIndex, plPeriod
```

<i>Argument</i>	Description
-----------------	-------------

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. To determine valid index values, use <code>GetNumPeriodsInCubeForLineItems</code> .
---------------------	---

<i>plPeriod</i>	Long. Returns the member ID of the period.
-----------------	--

GetPMErrorRecordCount

For internal use.

GetPMRecordCount

For internal use.

GetRecordFromPMBuffer

For internal use.

GetRecordFromPMErrorBuffer

For internal use.

GetSortedNature

For internal use.

InsertDataAtBeginning

Inserts data for a cell at the top of an HsvMDDataBuffer object.

Syntax

```
<HsvMDDataBuffer>.InsertDataAtBeginning lScenario, lYear, lPeriod, lView,  
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,  
lCustom4, dData, vbIsNoData
```

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

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.

Argument **Description**

<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>dData</i>	Double (ByVal). The cell's data.
<i>vbIsNoData</i>	Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.

InsertDataAtEnd

Inserts data for a cell at the bottom of an *HsvMDDataBuffer* object.

Syntax

```
<HsvMDDataBuffer>.InsertDataAtEnd lScenario, lYear, lPeriod, lView,  
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,  
lCustom4, dData, vbIsNoData
```

Argument **Description**

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>dData</i>	Double (ByVal). The cell's data.
<i>vbIsNoData</i>	Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.

Example

The following example loops through an array of Account dimension member IDs and inserts data and descriptions for each account into a cell. The data and descriptions are taken from arrays of the same size. Note that the non-Account member IDs passed to `InsertDataAtEnd` and `InsertDescriptionAtEnd` are the same for each cell.

```
For i = LBound(vlIDs) To UBound(vlIDs)
' vlIDs() is a Long array of Account IDs, vdData() is a
' Double array of data.
m_cDataBuff.InsertDataAtEnd m_lScen, m_lYear, m_lPer, m_lView, _
m_lEnt, m_lPar, m_lVal, vlIDs(i), m_lICP, m_lCust1, _
m_lCust2, m_lCust3, m_lCust4, vdData(i), False
'vsDescs() is a String array of descriptions.
m_cDataBuff.InsertDescriptionAtEnd m_lScen, m_lYear, m_lPer, _
m_lEnt, m_lPar, m_lVal, vlIDs(i), m_lICP, _
m_lCust1, m_lCust2, m_lCust3, m_lCust4, vsDescs(i)
Next i
```

InsertDescriptionAtBeginning

Inserts a cell description at the top of an `HsvMDDataBuffer` object.

Syntax

```
<HsvMDDataBuffer>.InsertDescriptionAtBeginning lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, bstrDescription
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.

Argument	Description
<i>bstrDescription</i>	String (ByVal). The cell's description.

InsertDescriptionAtEnd

Inserts a cell description at the bottom of an HsvMDDataBuffer object.

Syntax

```
<HsvMDDataBuffer>.InsertDescriptionAtEnd lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, bstrDescription
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>bstrDescription</i>	String (ByVal). The cell's description.

Example

InsertDescriptionAtEnd is used in the [Example](#) for InsertDataAtEnd.

InsertLineItemsAtBeginning

Inserts line items for a cell at the top of an HsvMDDataBuffer object.

Syntax

```
<HsvMDDataBuffer>.InsertLineItemsAtBeginning lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, varadData, varabstrDescriptions
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>varadData</i>	Double array (ByVal). The data for the cell's line items.
<i>varabstrDescriptions</i>	String array (ByVal). The descriptions of the cell's line items.

InsertLineItemsAtEnd

Inserts line items for a cell at the bottom of an HsvMDDataBuffer object.

Syntax

```
<HsvMDDataBuffer>.InsertLineItemsAtEnd lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, vbSavedAsYTD, varadData, varabstrDescriptions
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.

Argument	Description
<i>IParent</i>	Long (ByVal). The member ID of the <code>lEntity</code> argument's parent entity.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>vbSavedAsYTD</i>	Boolean (ByVal). Determines how the cell's totals are saved. Pass TRUE if the cell's line items are saved as year-to-date, FALSE if saved as periodic.
<i>varadData</i>	Double array (ByVal). The data for the cell's line items.
<i>varabstrDescriptions</i>	String array (ByVal). The descriptions of the cell's line items.

InsertRecordIntoPMBuffer

For internal use.

InsertRecordIntoPMErrorsBuffer

For internal use.

RemoveAll

Clears all the cells from an `HsvMDDataBuffer` object.

Syntax

```
<HsvMDDataBuffer>.RemoveAll
```

SetCheckLineItemDetailsForCaseInsensitiveDuplicates

Specifies whether the system should check for case-insensitive duplicate line item descriptions. For example, *acmeData* and *ACMEDATA* would be considered case-insensitive duplicate descriptions.

If you enable checking for such duplicates, the system performs the check when the `HsvMDDataBuffer` instance's data is passed to the `HsvData` method

[UpdateDataUsingMDDataBuffer](#). With case-insensitive duplicate checking enabled, if the system detects a duplicate an error is thrown.

Syntax

```
<HsvMDDataBuffer>.SetCheckLineItemDetailsForCaseInsensitive  
Duplicates vbEnable
```

Argument Description

vbEnable Boolean (ByVal). Specifies whether to check for case-insensitive duplicates. Pass TRUE to check, FALSE otherwise.

SetData

Inserts a cell's data into an HsvMDDataBuffer object.

Syntax

```
<HsvMDDataBuffer>.SetData lScenario, lYear, lPeriod, lView, lEntity,  
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,  
dData, vbIsNoData, vbAddToExistingData, plNumElementsInDataUnit
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member. You must pass a member ID for a user-defined member. Do not pass the member ID for the system-defined member <Scenario View>.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). Pass the HFMConstants type library constant <code>MEMBERNOTUSED</code> .
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member. You must pass a member ID for a user-defined member. Do not pass a member ID for system-defined members such as <Entity Currency>.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.

Argument	Description
<i>dData</i>	Double (ByVal). The cell's data.
<i>vbIsNoData</i>	Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.
<i>vbAddToExistingData</i>	Boolean (ByVal). If the HsvMDDataBuffer object already contains data for the cell, this argument determines whether the existing data will be overwritten. Pass TRUE to add the <i>dData</i> argument's data to the existing data, FALSE to overwrite the existing data.
<i>pINumElementsInDataUnit</i>	Long. Returns the number of cells in the HsvMDDataBuffer object that are in the subcube to which the cell belongs. This number returned is calculated after <i>SetData</i> adds the cell.

Example

The following function inserts data into cells that differ by account but otherwise share the same dimension members.

```
Function setDataforAccts(lScen As Long, lYear As Long, lPer As Long, lView As Long, lEnt As Long, lVal As Long, _
    laAccts() As Long, lIcp As Long, lCust1 As Long, lCust2 As Long, _
    lCust3 As Long, lCust4 As Long, daData() As Double, saDescs() As String) As Long
Dim cMDBuffer As HsvMDDataBuffer, lNumElems As Long, cData As HsvData
Set cMDBuffer = New HSVMDARRAYSLib.HsvMDDataBuffer
'g_cSession is an HsvSession object reference
Set cData = g_cSession.Data
For i = LBound(laAccts) To UBound(laAccts)
cMDBuffer.SetData lScen, lYear, lPer, lView, lEnt, MEMBERNOTUSED, lVal, _
    laAccts(i), lIcp, lCust1, lCust2, lCust3, lCust4, daData(i), _
    False, False, lNumElems
Next i
cData.UpdateDataUsingMDDataBuffer cMDBuffer, HSV_DATA_UPDATE_ACCUMULATE, _
    False
setDataforAccts = lNumElems
End Function
```

SetDescription

Inserts a cell's description into an HsvMDDataBuffer object.

Syntax

```
<HsvMDDataBuffer>.SetDescription lScenario, lYear, lPeriod, lEntity,
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
bstrDescription
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.

Argument	Description
<i>IPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>bstrDescription</i>	String (ByVal). The cell's description.

SetGrowByAmount

Creates additional memory for an HsvMDDataBuffer object.

Syntax

```
<HsvMDDataBuffer>.SetGrowByAmount lGrowByAmount
```

<i>Argument</i>	Description
-----------------	-------------

<i>lGrowByAmount</i>	Long (ByVal). The number of cells for which you want to create memory.
----------------------	--

SetLineItems

Inserts a cell's line items into an HsvMDDataBuffer object.

Syntax

```
<HsvMDDataBuffer>.SetLineItems lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, vbSavedAsYTD, varadData, varabstrDescriptions
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.

Argument	Description
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>vbSavedAsYTD</i>	Boolean (ByVal). Determines how the cell's totals are saved. Pass TRUE if the cell is saved as year-to-date, FALSE if saved as periodic.
<i>varadData</i>	Double array (ByVal). The cell's line item data.
<i>varabstrDescriptions</i>	String array (ByVal). The cell's line item descriptions.

Example

The following example loops through a Double array of data, inserting a line item for each array item. For each item, a description from an equally sized String array is also inserted.

```

For i = LBound(vdData) To UBound(vdData)
' vdData() is a Double array of data, vsDescs() is a
' String array of descriptions.
m_cDataBuff.SetLineItems m_lScen, m_lYear, m_lPer, _
m_lEnt, m_lPar, m_lVal, m_lAcct, m_lICP, m_lCust1, _
m_lCust2, m_lCust3, m_lCust4, True, vdData, vsDescs
Next i

```

SetSortedNature

For internal use.

Sort

For internal use.

HsvMDDataBufferLite Object Methods

An HsvMDDataBufferLite object contains an array of cells. The object provides methods for setting, getting, and enumerating three types of cells:

- Cells with data
- Cells with descriptions
- Cells with line items

These cells are not stored data; in other words, the information in an `HsvMDDataBufferLite` object's cells is not stored in an Financial Management application. You move data between an `HsvMDDataBufferLite` object and an application with the following methods of the `HsvData` object:

- `UpdateDataUsingMDDataBuffer` inserts the data in an `HsvMDDataBufferLite` object into an application's cells. Use this method after you have finished working with the cells in an `HsvMDDataBufferLite` object. For details, see [“UpdateDataUsingMDDataBuffer” on page 374](#).
- `AddDataToMDDataBuffer` adds the data stored in an application's cell to an `HsvMDDataBufferLite` object. Use this method to build an `HsvMDDataBufferLite` object from data stored in applications. For details, see [“AddDataToMDDataBuffer” on page 277](#).

The `HsvMDDataBufferLite` object caches a minimum number of subcubes to RAM and the remaining subcubes to disk. By default, the `HsvMDDataBufferLite` object caches a minimum of one subcube to RAM. The system might cache additional subcubes to RAM, but a minimum of one subcube will always be cached to RAM. You can change the minimum number of subcubes that will be cached with [SetMinCubesInCache](#).

If you want similar functionality, but with all the subcubes cached to RAM, use the `HsvMDDataBuffer` object. For more information, see [“HsvMDDataBuffer Object Methods” on page 610](#).

Assign `HsvMDDataBufferLite` object references with the `Set` and `New` keywords. For an example, see [“HsvMDDataBufferLite Object Overview” on page 96](#).

The `HsvMDDataBufferLite` object's methods are summarized in [Table 32 on page 97](#), and are described in detail in the following topics.

BeginEnumeration

Locks an `HsvMDDataBufferLite` object, and returns the number of subcubes contained by the object. You can then pass the number of subcubes to methods that require a subcube index.

Caution! Call `EndEnumeration` once you have finished working with an `HsvMDDataBufferLite` object, otherwise the object will not be unlocked and errors could occur.

Syntax

```
<HsvMDDataBufferLite>.BeginEnumeration plNumCubes
```


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

<i>pINumCubes</i>	Long. Returns the number of subcubes in the HsvMDDataBufferLite object.
-------------------	---

Example

BeginEnumeration is used in the [Example](#) for GetDataAtIndex.

CreateDataIndexList

Creates an HsvMDIndexList object based upon the items in an HsvMDDataBufferLite object.

CreateDataIndexList takes arguments for the Financial Management dimensions; these arguments determine which items in the HsvMDDataBufferLite object are added to the HsvMDIndexList object. For each dimension, pass one of the following:

- A member ID. Passing a member ID means that only items for this member will be added to the HsvMDIndexList object.
- MEMBERALL constant. Passing MEMBERALL means that all items containing the dimension member will be added to the HsvMDIndexList object.
- MEMBERNOTUSED constant. Passing MEMBERNOTUSED means that only those items containing MEMBERNOTUSED for the dimension member will be added to the HsvMDIndexList object.
- MEMBERANYONE constant. Passing MEMBERANYONE means that if the HsvMDDataBufferLite object contains more than one item where all other dimension members are the same, only one instance will be added to the HsvMDIndexList object.

Tip: These constants are in the HFMConstants type library. For more information, see [“Dimension Member Constants” on page 834](#).

Syntax

```
<HsvMDDataBufferLite>.CreateDataIndexList lScenario As Long, lYear As Long, lPeriod As Long, lView As Long, lEntity As Long, lParent As Long, lValue As Long, lAccount As Long, lICP As Long, lCustom1 As Long, lCustom2 As Long, lCustom3 As Long, lCustom4 As Long, ppIMDIndexListUnk As Unknown
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member, or one of the constants listed above in the method's description.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member, or one of the constants listed above in the method's description.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member, or one of the constants listed above in the method's description.
<i>lView</i>	Long (ByVal). The member ID of the View dimension member, or one of the constants listed above in the method's description.

Argument	Description
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member, or one of the constants listed above in the method's description.
<i>lParent</i>	Long (ByVal). The member ID of the parent of the <i>lEntity</i> argument's entity, or one of the constants listed above in the method's description.
<i>lValue</i>	Long (ByVal). Long (ByVal). The member ID of the Value dimension member, or one of the constants listed above in the method's description.
<i>lAccount</i>	Long (ByVal). Long (ByVal). The member ID of the Account dimension member, or one of the constants listed above in the method's description.
<i>lICP</i>	Long (ByVal). Long (ByVal). The member ID of the Intercompany Partner dimension member, or one of the constants listed above in the method's description.
<i>lCustom1</i>	Long (ByVal). Long (ByVal). The member ID of the Custom 1 dimension member, or one of the constants listed above in the method's description.
<i>lCustom2</i>	Long (ByVal). The member ID of the Custom 2 dimension member, or one of the constants listed above in the method's description.
<i>lCustom3</i>	Long (ByVal). The member ID of the Custom 3 dimension member, or one of the constants listed above in the method's description.
<i>lCustom4</i>	Long (ByVal). The member ID of the Custom 4 dimension member, or one of the constants listed above in the method's description.
<i>ppMDIndexListUnk</i>	HsvMDIndexList object. Returns an HsvMDIndexList object containing the items in the HsvMDDataBufferLite object that match the criteria specified in the other arguments.

Example

The following example creates an HsvMDIndexList object that contains all of the items in the HsvMDDataBufferLite object. The example then loops through the HsvMDIndexList object: `GetNumItems` returns the number of items in the object, and `GetItem` is called for each item. After the call to `GetItem`, you could operate upon the item's dimension members by replacing the ellipsis (...) with code.

```
Dim lScen As Long, lYear As Long, lPer As Long, lView As Long
Dim lEnt As Long, lPar As Long, lVal As Long, lAcct As Long
Dim lICP As Long, lCust1 As Long, lCust2 As Long, lCust3 As Long
Dim lCust4 As Long, lItems As Long, lItem As Long
m_cMDBufferLite.CreateDataIndexList MEMBERALL, MEMBERALL, _
MEMBERALL, MEMBERALL, MEMBERALL, MEMBERALL, MEMBERALL, _
MEMBERALL, MEMBERALL, MEMBERALL, MEMBERALL, MEMBERALL, _
MEMBERALL, m_cMDIndexList
m_cMDIndexList.GetNumItems lItems - 1
For lItem = 0 To lItems
    m_cMDIndexList.GetItem lItem, lScen, lYear, lPer, lView, _
    lEnt, lPar, lVal, lAcct, lICP, lCust1, lCust2, lCust3, _
    lCust4
    ... 'Insert code here.
Next lItem
```

EraseRecordFromPMBuffer

For internal use.

EndEnumeration

Unlocks an HsvMDDataBufferLite object that has been locked by `BeginEnumeration`.

Syntax

```
<HsvMDDataBufferLite>.EndEnumeration
```

Example

`EndEnumeration` is used in the [Example](#) for `GetDataAtIndex`.

GetCheckLineItemDetailsForCaseInsensitiveDuplicates

Indicates whether case-insensitive duplicate checking is enabled for the HsvMDDataBufferLite instance. *acmeData* and *ACMEDATA* would be considered case-insensitive duplicate descriptions.

Tip: `SetCheckLineItemDetailsForCaseInsensitiveDuplicates` enables or disables whether the system should check for case-insensitive duplicate line item descriptions.

Syntax

```
<HsvMDDataBufferLite>.GetCheckLineItemDetailsForCaseInsensitive  
Duplicates pvbEnabled
```

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

<i>pvbEnabled</i>	Boolean. Indicates whether case-insensitive duplicate checking is enabled. Returns TRUE if enabled, FALSE otherwise.
-------------------	--

GetCubeIndexFromPOV

Returns the index of a subcube in an HsvMDDataBufferLite object, given the member IDs of the subcube's dimension members.

Syntax

```
<HsvMDDataBufferLite>.GetCubeIndexFromPOV lScenario, lYear, lEntity,  
lParent, lValue, p1CubeIndex
```

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

<i>lScenario</i>	Long (ByVal). The member ID of the subcube's Scenario dimension member.
------------------	---

Argument **Description**

<i>Year</i>	Long (ByVal). The member ID of the subcube's Year dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the subcube's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the subcube's Parent dimension member.
<i>IValue</i>	Long (ByVal). The member ID of the subcube's Value dimension member.
<i>plCubeIndex</i>	Long. The index of the subcube, or -1 if the HsvMDDataBufferLite object does not contain the subcube.

Example

GetCubeIndexFromPOV is used in the [Example](#) for GetPeriodIndexFromPOVForData.

GetCubePOVFromIndex

Returns the member IDs of a subcube's dimension members, given the subcube's index.

Syntax

```
<HsvMDDataBufferLite>.GetCubePOVFromIndex lCubeIndex, plScenario, plYear,
plEntity, plParent, plValue
```

Argument **Description**

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. You can use <code>BeginEnumeration</code> to return the number of subcubes in an <code>HsvMDDataBufferLite</code> object.
<i>plScenario</i>	Long. Returns the member ID of the subcube's Scenario dimension member.
<i>plYear</i>	Long. Returns the member ID of the subcube's Year dimension member.
<i>plEntity</i>	Long. Returns the member ID of the subcube's Entity dimension member.
<i>plParent</i>	Long. Returns the member ID of the <code>plEntity</code> argument's parent entity.
<i>plValue</i>	Long. Returns the member ID of the subcube's Value dimension member.

Example

The following example shows how to get the member IDs of the subcubes in an `HsvMDDataBufferLite` object. `BeginEnumeration` returns the number of subcubes in the object. For each subcube, `GetCubePOVFromIndex` returns the member IDs. To use the member IDs in the loop, replace the ellipsis (...) with code.

```
Dim lNumCubes As Long, lScen As Long, lYear As Long
Dim lEnt As Long, lPar As Long, lVal As Long
m_cMDDataBufferLite.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDDataBufferLite.GetCubePOVFromIndex i, lScen, lYear, lEnt, _
    lPar, lVal
    ... 'Insert code here
```

Next `i`
`m_cMDBufferLite.EndEnumeration`

GetData

Returns a cell's data, given the member IDs that identify the cell. `GetData` also indicates whether the cell is null.

Tip: To get a cell's data by passing an index, use `GetDataAtIndex`. For details, see [“GetDataAtIndex” on page 646](#).

Syntax

```
<HsvMDDataBufferLite>.GetData lScenario, lYear, lPeriod, lView, lEntity,  
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,  
pdData, pvbIsNoData
```

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

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member. You must pass a member ID for a user-defined member. Do not pass the member ID for the system-defined member <Scenario View>.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). Pass the HFMConstants type library constant <code>MEMBERNOTUSED</code> .
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member. You must pass a member ID for a user-defined member. Do not pass a member ID for system-defined members such as <Entity Currency>.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pdData</i>	Double. Returns the cell's data.
<i>pvbIsNoData</i>	Boolean. Indicates whether the cell is null. Returns TRUE if null, FALSE otherwise.

Example

The following example returns a cell's data in the `dDataCell` variable. The example assumes the cell's dimension member IDs have previously been defined.

```
Dim dDataCell As Double, bIsNoData As Boolean
m_cMDDBufferLite.GetData m_lScen, m_lYear, m_lPer, m_lView, _
m_lEnt, m_lPar, m_lVal, m_lAcct, m_lICP, m_lCust1, _
m_lCust2, m_lCust3, m_lCust4, dDataCell, bIsNoData
```

GetDataAtIndex

Returns the data for a cell, given the indexes that identify the cell. `GetDataAtIndex` also indicates whether the cell is null, and returns the member IDs of the cell's View, Account, Intercompany Partner, and Custom dimension members.

Syntax

```
<HsvMDDataBufferLite>.GetDataAtIndex lCubeIndex, lPeriodIndex, lCellIndex,
plView, plAccount, plICP, plCustom1, plCustom2, plCustom3, plCustom4,
pdData, pvbIsNoData
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the cell's subcube. Use the <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. Use the <code>GetNumPeriodsInCubeForData</code> or <code>GetPeriodIndexFromPOVForData</code> to determine valid index values.
<i>lCellIndex</i>	Long (ByVal). The index of the cell. Use <code>GetNumCellsForData</code> to determine valid index values.
<i>plView</i>	Long. Returns the member ID of the cell's View dimension member.
<i>plAccount</i>	Long. Returns the member ID of the cell's Account dimension member.
<i>plICP</i>	Long. Returns the member ID of the cell's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the cell's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the cell's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the cell's Custom 3 dimension member.
<i>plCustom4</i>	Long. Returns the member ID of the cell's Custom 4 dimension member.
<i>pdData</i>	Double. Returns the cell's data.
<i>pvbIsNoData</i>	Boolean. Indicates whether the cell is null. Returns TRUE if null, FALSE otherwise.

Example

The following example loops through the cells in an `HsvMDDataBufferLite` object and prints each cell's data to Visual Basic's Immediate window. The example assumes that the object has

previously been filled with data. `BeginEnumeration` locks the object and returns the number of subcubes in the object. The loops then works as follows:

- For each subcube, `GetNumPeriodsInCubeForData` gets the number of periods with cells that contain data.
- For each period in a subcube, `GetNumCellsForData` gets the number of cells that contain data.
- For each cell that contains data, `GetDataAtIndex` gets the data, which is then printed to the Immediate window.

`EndEnumeration` then unlocks the `HsvMDDataBufferLite` object.

```
Dim lNumCubes As Long, lNumPers As Long, lNumCells As Long
Dim lVw As Long, lAccount As Long, lIntCo As Long
Dim lCus1 As Long, lCus2 As Long, lCus3 As Long, lCus4 As Long
Dim dMdData As Double, bMdNoData As Boolean
m_cMDDataBufferLite.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDDataBufferLite.GetNumPeriodsInCubeForData i, lNumPers
    For j = 0 To lNumPers - 1
        m_cMDDataBufferLite.GetNumCellsForData i, j, lNumCells
        For k = 0 To lNumCells - 1
            m_cMDDataBufferLite.GetDataAtIndex i, j, k, lVw, lAccount, _
                lIntCo, lCus1, lCus2, lCus3, lCus4, dMdData, bMdNoData
            Debug.Print dMdData
        Next k
    Next j
Next i
m_cMDDataBufferLite.EndEnumeration
```

GetDescription

Returns a cell's description, given the member IDs that identify the cell.

Tip: To get a cell's description by passing an index, use `GetDescriptionAtIndex`. For details, see [“GetDescriptionAtIndex” on page 648](#).

Syntax

```
<HsvMDDataBufferLite>.GetDescription lScenario, lYear, lPeriod, lEntity,
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
pbstrDescription
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.

Argument	Description
<i>IParent</i>	Long (ByVal). The member ID of the <code>lEntity</code> argument's parent entity.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pbstrDescription</i>	String. Returns the cell's description.

Example

The following example returns a cell's description. The example assumes that the cell's dimension member IDs have been previously defined.

```
Dim sCellDesc As String
m_cMDBufferLite.GetDescription m_lScen, m_lYear, m_lPer, _
m_lEnt, m_lPar, m_lVal, m_lAcct, m_lICP, m_lCust1, _
m_lCust2, m_lCust3, m_lCust4, sCellDesc
```

GetDescriptionAtIndex

Returns the description for a cell, given the indexes that identify the cell.

`GetDescriptionAtIndex` also returns the member IDs of the cell's Account, Intercompany Partner, and Custom dimension members.

Syntax

```
<HsvMDDataBufferLite>.GetDescriptionAtIndex lCubeIndex, lPeriodIndex,
lCellIndex, plAccount, plICP, plCustom1, plCustom2, plCustom3, plCustom4,
pbstrDesc
```

Argument	Description
<i>lCubeIndex</i>	Long (ByVal). The index of the cell's subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. Use <code>GetNumPeriodsInCubeForDescriptions</code> or <code>GetPeriodIndexFromPOVForDescriptions</code> to determine valid index values.
<i>lCellIndex</i>	Long (ByVal). The index of the cell. Use <code>GetNumCellsForDescriptions</code> to determine valid index values.
<i>plAccount</i>	Long. Returns the member ID of the cell's Account dimension member.

Argument	Description
<i>plICP</i>	Long. Returns the member ID of the cell's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the cell's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the cell's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the cell's Custom 3 dimension member.
<i>plCustom4</i>	Long. Returns the member ID of the cell's Custom 4 dimension member.
<i>pbstrDesc</i>	String. Returns the cell's description.

Example

The following example loops through the cells in an `HsvMDDataBufferLite` object and prints each cell's description to Visual Basic's Immediate window. The example assumes that the object has previously been filled with data. `BeginEnumeration` locks the object and returns the number of subcubes in the object. The loops then work as follows:

- For each subcube, `GetNumPeriodsInCubeForDescriptions` gets the number of periods with cells that contain descriptions.
- For each period in a subcube, `GetNumCellsForDescriptions` gets the number of cells that contain descriptions.
- For each cell that contains data, `GetDescriptionAtIndex` gets the description, which is then printed to the Immediate window.

`EndEnumeration` then unlocks the `HsvMDDataBufferLite` object.

```
Dim lNumCubes As Long, lNumPers As Long, lNumCells As Long
Dim lAccount As Long, lIntCo As Long, lCus1 As Long
Dim lCus2 As Long, lCus3 As Long, lCus4 As Long, sDesc As String
m_cMDDataBufferLite.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDDataBufferLite.GetNumPeriodsInCubeForDescriptions i, _
    lNumPers
    For j = 0 To lNumPers - 1
        m_cMDDataBufferLite.GetNumCellsForDescriptions i, j, lNumCells
        For k = 0 To lNumCells - 1
            m_cMDDataBufferLite.GetDescriptionAtIndex i, j, k, lAccount, _
            lIntCo, lCus1, lCus2, lCus3, lCus4, sDesc
            Debug.Print sDesc
        Next k
    Next j
Next i
m_cMDDataBufferLite.EndEnumeration
```

GetLineItems

Returns the data and descriptions for a cell's line items, given the member IDs that identify the cell's dimension members. `GetLineItems` also indicates whether the cell's data is saved as year-to-date or periodic.

Syntax

```
<HsvMDDataBufferLite>.GetLineItems lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, pvbSavedAsYTD, pvaradData, pvarabstrDescriptions
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <code>lEntity</code> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pvbSavedAsYTD</i>	Boolean. Indicates how the cell's data is saved. Returns TRUE if saved as year-to-date, FALSE if saved as periodic.
<i>pvaradData</i>	VARIANT array. Returns the data for the cell's line items. The array is returned as a Double subtype.
<i>pvarabstrDescriptions</i>	VARIANT array. Returns the descriptions of the cell's line items. The array is returned as a String subtype.

Example

The following example returns the data and descriptions of a cell's line items. The example assumes the cell's dimension member IDs have previously been defined.

```
Dim bYTD As Boolean, vdData, vsDescs  
m_cMDDataBufferLite.GetLineItems m_lScen, m_lYear, m_lPer, _  
m_lEnt, m_lPar, m_lVal, m_lAcct, m_lICP, m_lCust1, _  
m_lCust2, m_lCust3, m_lCust4, bYTD, vdData, vsDescs
```

GetLineItemsAtIndex

Returns the data and descriptions for a cell's line items, given the indexes that identify the cell. `GetLineItemsAtIndex` also returns the member IDs of the cell's Account, Intercompany Partner, and Custom dimension member, and indicates whether the line items are saved as year-to-date or periodic.

Syntax

```
<HsvMDDataBufferLite>.GetLineItemsAtIndex lCubeIndex, lPeriodIndex,  
lCellIndex, plAccount, plICP, plCustom1, plCustom2, plCustom3, plCustom4,  
pvbSavedAsYTD, pvaradData, pvarabstrDescriptions
```

Argument	Description
<i>lCubeIndex</i>	Long (ByVal). The index of the cell's subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. Use <code>GetNumPeriodsInCubeForLineItems</code> or <code>GetPeriodIndexFromPOVForLineItems</code> to determine valid index values.
<i>lCellIndex</i>	Long (ByVal). The index of the cell. Use <code>GetNumCellsForLineItems</code> to determine valid index values.
<i>plAccount</i>	Long. Returns the member ID of the cell's Account dimension member.
<i>plICP</i>	Long. Returns the member ID of the cell's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the cell's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the cell's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the cell's Custom 3 dimension member.
<i>plCustom4</i>	Long. Returns the member ID of the cell's Custom 4 dimension member.
<i>pvbSavedAsYTD</i>	Boolean. Indicates how the cell is saved. Returns TRUE if the cell is saved as year-to-date, FALSE if saved as periodic.
<i>pvaradData</i>	VARIANT array. Returns the data for the cell's line items. The array is returned as a Double subtype.
<i>pvarabstrDescriptions</i>	VARIANT array. Returns the descriptions of the cell's line items. The array is returned as a String subtype.

Example

The following example loops through the cells in an `HsvMDDataBufferLite` object and prints each line item's data and description to Visual Basic's Immediate window. The example assumes that the object has previously been filled with data. `BeginEnumeration` locks the object and returns the number of subcubes in the object. The loops then work as follows:

- For each subcube, `GetNumPeriodsInCubeForLineItems` gets the number of periods with cells that contain line items.
- For each period in a subcube, `GetNumCellsForLineItems` gets the number of cells that contain line items.
- For each cell that contains line items, `GetLineItemsAtIndex` gets the line items' data and descriptions, which are then printed to the Immediate window.

`EndEnumeration` then unlocks the `HsvMDDataBufferLite` object.

```
Dim lNumCubes As Long, lNumPers As Long, lNumCells As Long  
Dim lAccount As Long, lIntCo As Long, lCus1 As Long  
Dim lCus2 As Long, lCus3 As Long, lCus4 As Long
```

```

Dim bYTD As Boolean, vdMdData, vdMdDesc
m_cMDBufferLite.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDBufferLite.GetNumPeriodsInCubeForLineItems i, lNumPers
    For j = 0 To lNumPers - 1
        m_cMDBufferLite.GetNumCellsForLineItems i, j, lNumCells
        For k = 0 To lNumCells - 1
            m_cMDBufferLite.GetLineItemsAtIndex i, j, k, lAccount, _
                lIntCo, lCus1, lCus2, lCus3, lCus4, bYTD, _
                vdMdData, vdMdDesc
            For l = LBound(vdMdData) To UBound(vdMdData)
                Debug.Print vdMdData(l) & " " & vdMdDesc(l)
            Next l
        Next k
    Next j
Next i
m_cMDBufferLite.EndEnumeration

```

GetNumCellsForData

Returns the number of cells that are in a period of a subcube and that contain data.

Syntax

```

<HsvMDDataBufferLite>.GetNumCellsForData lCubeIndex, lPeriodIndex,
pINumCells

```

Argument Description

lCubeIndex Long (ByVal). The index of the subcube. Use `BeginEnumeration` or `GetCubeIndexFromPOV` to determine valid index values.

lPeriodIndex Long (ByVal). The index of the period. Use `GetNumPeriodsInCubeForData` or `GetPeriodIndexFromPOVForData` to determine valid index values.

pINumCells Long. Returns the number of cells that contain data.

Example

`GetNumCellsForData` is used in the [Example](#) for `GetDataAtIndex`.

GetNumCellsForDescriptions

Returns the number of cells that are in a period of a subcube and that contain descriptions.

Syntax

```

<HsvMDDataBufferLite>.GetNumCellsForDescriptions lCubeIndex, lPeriodIndex,
pINumCells

```

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

<i>ICubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>IPeriodIndex</i>	Long (ByVal). The index of the period. Use <code>GetNumPeriodsInCubeForDescriptions</code> or <code>GetPeriodIndexFromPOVForDescriptions</code> to determine valid index values.
---------------------	--

<i>pINumCells</i>	Long. Returns the number of cells that contain descriptions.
-------------------	--

Example

`GetNumCellsForDescriptions` is used in the [Example](#) for `GetDescriptionAtIndex`.

GetNumCellsForLineItems

Returns the number of cells that are in a period of a subcube and that contain line items.

Syntax

```
<HsvMDDataBufferLite>.GetNumCellsForLineItems lCubeIndex, lPeriodIndex, pINumCells
```

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

<i>ICubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>IPeriodIndex</i>	Long (ByVal). The index of the period. Use <code>GetNumPeriodsInCubeForLineItems</code> or <code>GetPeriodIndexFromPOVForLineItems</code> to determine valid index values.
---------------------	--

<i>pINumCells</i>	Long. Returns the number of cells that contain line items.
-------------------	--

Example

`GetNumCellsForLineItems` is used in the [Example](#) for `GetLineItemsAtIndex`.

GetNumPeriodsInCubeForData

Returns the number of periods that are in a subcube and that contain cells with data.

Syntax

```
<HsvMDDataBufferLite>.GetNumPeriodsInCubeForData lCubeIndex, pINumPeriods
```

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

<i>ICubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>pINumPeriods</i>	Long. Returns the number of periods with cells that contain data.
---------------------	---

Example

`GetNumPeriodsInCubeForData` is used in the [Example](#) for `GetDataAtIndex`.

GetNumPeriodsInCubeForDescriptions

Returns the number of periods that are in a subcube and that contain cells with descriptions.

Syntax

```
<HsvMDDataBufferLite>.GetNumPeriodsInCubeForDescriptions lCubeIndex,  
plNumPeriods
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

plNumPeriods Long. Returns the number of periods with cells that contain descriptions.

Example

`GetNumPeriodsInCubeForDescriptions` is used in the [Example](#) for `GetDescriptionAtIndex`.

GetNumPeriodsInCubeForLineItems

Returns the number of periods that are in a subcube and that contain cells with line items.

Syntax

```
<HsvMDDataBufferLite>.GetNumPeriodsInCubeForLineItems lCubeIndex,  
plNumPeriods
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

plNumPeriods Long. Returns the number of periods with cells that contain line items.

Example

`GetNumPeriodsInCubeForLineItems` is used in the [Example](#) for `GetLineItemsAtIndex`.

GetPeriodIndexFromPOVForData

Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain data.

Syntax

```
<HsvMDDataBufferLite>.GetPeriodIndexFromPOVForData lCubeIndex, lPeriod,  
plPeriodIndex
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
----------------	---

<i>plPeriodIndex</i>	Long. Returns the period's index, or -1 if the period does not contain cells with data.
----------------------	---

Example

The following example gets a subcube's index, the index of a period in the subcube, and the number of cells in the period that contain data, then prints the cells' data to Visual Basic's Immediate window. The example assumes that the member IDs of the subcube and period have been previously defined. `GetCubeIndexFromPOV` returns the index of the subcube.

`GetPeriodIndexFromPOVForData` then gets the period's index. `GetNumCellsForData` takes the subcube and period indexes and returns the number of cells that contain data. For each cell, `GetDataAtIndex` returns the cell's data, which is then printed to the Immediate window.

```
Dim lCubeIx As Long, lPerIx As Long, lNumCells As Long  
Dim lNumCubes As Long, dCellData As Double, bIsNoData As Boolean  
Dim lView As Long, lAcct As Long, lICP As Long, lCust1 As Long  
Dim lCust2 As Long, lCust3 As Long, lCust4 As Long  
m_cMDDataBufferLite.BeginEnumeration lNumCubes  
m_cMDDataBufferLite.GetCubeIndexFromPOV m_lScen, m_lYear, m_lEnt, _  
m_lPar, m_lVal, lCubeIx  
' Exit if there are no subcubes  
If lCubeIx = -1 Then  
    m_cMDDataBufferLite.EndEnumeration  
    Exit Sub  
Else  
    m_cMDDataBufferLite.GetPeriodIndexFromPOVForData lCubeIx, _  
    m_lPer, lPerIx  
    ' Exit if the period does not have cells with data  
    If lPerIx = -1 Then  
        m_cMDDataBufferLite.EndEnumeration  
        Exit Sub  
    Else  
        m_cMDDataBufferLite.GetNumCellsForData lCubeIx, _  
        lPerIx, lNumCells  
        For i = 0 To lNumCells - 1  
            m_cMDDataBufferLite.GetDataAtIndex lCubeIx, lPerIx, i, _  
            lView, lAcct, lICP, lCust1, lCust2, lCust3, _  
            lCust4, dCellData, bIsNoData  
            Debug.Print dCellData  
        Next i  
    End If  
End If  
m_cMDDataBufferLite.EndEnumeration
```

GetPeriodIndexFromPOVForDescriptions

Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain descriptions.

Syntax

```
<HsvMDDataBufferLite>.GetPeriodIndexFromPOVForDescriptions lCubeIndex,  
lPeriod, plPeriodIndex
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
----------------	---

<i>plPeriodIndex</i>	Long. Returns the period's index, or -1 if the period does not contain cells with descriptions.
----------------------	---

Example

The following example gets a subcube's index, the index of a period in the subcube, and the number of cells in the period that contain descriptions, then prints the cells' descriptions to Visual Basic's Immediate window. The example assumes that the member IDs of the subcube and period have been previously defined. `GetCubeIndexFromPOV` returns the index of the subcube. `GetPeriodIndexFromPOVForDescriptions` then gets the period's index. `GetNumCellsForDescriptions` takes the subcube and period indexes and returns the number of cells that contain descriptions. For each cell, `GetDescriptionAtIndex` returns the cell's description, which is then printed to the Immediate window.

```
Dim lCubeIx As Long, lPerIx As Long, lNumCells As Long  
Dim lNumCubes As Long, sCellDesc As String  
Dim lAcct As Long, lICP As Long, lCust1 As Long  
Dim lCust2 As Long, lCust3 As Long, lCust4 As Long  
m_cMDDataBufferLite.BeginEnumeration lNumCubes  
m_cMDDataBufferLite.GetCubeIndexFromPOV m_lScen, m_lYear, m_lEnt, _  
m_lPar, m_lVal, lCubeIx  
' Exit if there are no subcubes  
If lCubeIx = -1 Then  
    m_cMDDataBufferLite.EndEnumeration  
    Exit Sub  
Else  
    m_cMDDataBufferLite.GetPeriodIndexFromPOVForDescriptions _  
    lCubeIx, m_lPer, lPerIx  
    ' Exit if the period does not have cells with descriptions  
    If lPerIx = -1 Then  
        m_cMDDataBufferLite.EndEnumeration  
        Exit Sub  
    Else  
        m_cMDDataBufferLite.GetNumCellsForDescriptions lCubeIx, _  
        lPerIx, lNumCells  
        For i = 0 To lNumCells - 1  
            m_cMDDataBufferLite.GetDescriptionAtIndex lCubeIx, lPerIx, _  
            i, lAcct, lICP, lCust1, lCust2, lCust3, _  
            lCust4, sCellDesc
```



```

        Debug.Print sCellDesc
    Next i
End If
End If
m_cMDBufferLite.EndEnumeration

```

GetPeriodIndexFromPOVForLineItems

Returns the index of a period within a subcube, given the subcube's index and the period's member ID. An index is returned only for periods with cells that contain line items.

Syntax

```

<HsvMDDataBufferLite>.GetPeriodIndexFromPOVForLineItems lCubeIndex,
lPeriod, plPeriodIndex

```

Argument Description

lCubeIndex Long (ByVal). The index of the subcube. Use `BeginEnumeration` or `GetCubeIndexFromPOV` to determine valid index values.

lPeriod Long (ByVal). The member ID of the Period dimension member.

plPeriodIndex Long. Returns the period's index, or -1 if the period does not contain cells with line items.

Example

The following example gets a subcube's index, the index of a period in the subcube, and the number of cells in the period that contain line items, then prints the line items' descriptions to Visual Basic's Immediate window. The example assumes that the member IDs of the subcube and period have been previously defined. `GetCubeIndexFromPOV` returns the index of the subcube. `GetPeriodIndexFromPOVForLineItems` then gets the period's index. `GetNumCellsForLineItems` takes the subcube and period indexes and returns the number of cells that contain line items. For each cell, `GetLineItemsAtIndex` returns the line items' data and descriptions, which are then printed to the Immediate window.

```

Dim lCubeIx As Long, lPerIx As Long, lNumCells As Long
Dim lNumCubes As Long, dCellData As Double, bYTD As Boolean
Dim vdCellData, vsDescs, lView As Long, lAcct As Long
Dim lICP As Long, lCust1 As Long, lCust2 As Long, lCust3 As Long
Dim lCust4 As Long
m_cMDBufferLite.BeginEnumeration lNumCubes
m_cMDBufferLite.GetCubeIndexFromPOV m_lScen, m_lYear, m_lEnt, _
m_lPar, m_lVal, lCubeIx
' Exit if there are no subcubes
If lCubeIx = -1 Then
    m_cMDBufferLite.EndEnumeration
    Exit Sub
Else
    m_cMDBufferLite.GetPeriodIndexFromPOVForLineItems lCubeIx, _
    m_lPer, lPerIx
    ' Exit if the period does not have cells with line items
    If lPerIx = -1 Then
        m_cMDBufferLite.EndEnumeration
    End If

```

```

Exit Sub
Else
m_cMDBufferLite.GetNumCellsForLineItems lCubeIx, _
lPerIx, lNumCells
For i = 0 To lNumCells - 1
m_cMDBufferLite.GetLineItemsAtIndex lCubeIx, lPerIx, i, _
lAcct, lICP, lCust1, lCust2, lCust3, _
lCust4, bYTD, vdCellData, vsDescs
For j = LBound(vdCellData) To UBound(vdCellData)
Debug.Print vdCellData(j) & " " & vsDescs(j)
Next j
Next i
End If
End If
m_cMDBufferLite.EndEnumeration

```

GetPeriodPOVFromIndexForData

Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain data.

Syntax

```

<HsvMDDataBufferLite>.GetPeriodPOVFromIndexForData lCubeIndex,
lPeriodIndex, plPeriod

```

Argument Description

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. To determine valid index values, use <code>GetNumPeriodsInCubeForData</code> .
<i>plPeriod</i>	Long. Returns the member ID of the period.

GetPeriodPOVFromIndexForDescriptions

Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain descriptions.

Syntax

```

<HsvMDDataBufferLite>.GetPeriodPOVFromIndexForDescriptions lCubeIndex,
lPeriodIndex, plPeriod

```

Argument Description

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. To determine valid index values, use <code>GetNumPeriodsInCubeForDescriptions</code> .

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

<i>pPeriod</i>	Long. Returns the member ID of the period.
----------------	--

GetPeriodPOVFromIndexForLineItems

Returns the member ID of a period, given a subcube index and a period index. IDs are returned only for periods with cells that contain line items.

Syntax

```
<HsvMDDataBufferLite>.GetPeriodPOVFromIndexForLineItems lCubeIndex,  
lPeriodIndex, pPeriod
```

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

<i>lCubeIndex</i>	Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.
-------------------	--

<i>lPeriodIndex</i>	Long (ByVal). The index of the cell's period. To determine valid index values, use <code>GetNumPeriodsInCubeForLineItems</code> .
---------------------	---

<i>pPeriod</i>	Long. Returns the member ID of the period.
----------------	--

GetPMErrorRecordCount

For internal use.

GetPMRecordCount

For internal use.

GetRecordFromPMBuffer

For internal use.

GetRecordFromPMErrorBuffer

For internal use.

GetSortedNature

For internal use.

InsertDataAtEnd

Inserts data for a cell at the bottom of an `HsvMDDataBufferLite` object.

Syntax

```
<HsvMDDataBufferLite>.InsertDataAtEnd lScenario, lYear, lPeriod, lView,  
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,  
lCustom4, dData, vbIsNoData
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>dData</i>	Double (ByVal). The cell's data.
<i>vbIsNoData</i>	Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.

Example

The following example loops through an array of Account dimension member IDs and inserts data and descriptions for each account into a cell. The data and descriptions are taken from arrays of the same size. Note that the non-Account member IDs passed to `InsertDataAtEnd` and `InsertDescriptionAtEnd` are the same for each cell.

```
For i = LBound(vlIDs) To UBound(vlIDs)  
' vlIDs() is a Long array of Account IDs, vdData() is a  
' Double array of data.  
m_cMDDataBufferLite.InsertDataAtEnd m_lScen, m_lYear, m_lPer, _  
m_lView, m_lEnt, m_lPar, m_lVal, vlIDs(i), m_lICP, m_lCust1, _  
m_lCust2, m_lCust3, m_lCust4, vdData(i), False  
'vsDescs() is a String array of descriptions.
```

```
m_cMDBufferLite.InsertDescriptionAtEnd m_lScen, m_lYear, _
m_lPer, m_lEnt, m_lPar, m_lVal, vlIDs(i), m_lICP, _
m_lCust1, m_lCust2, m_lCust3, m_lCust4, vsDescs(i)
Next i
```

InsertDescriptionAtEnd

Inserts a cell description at the bottom of an HsvMDDataBufferLite object.

Syntax

```
<HsvMDDataBufferLite>.InsertDescriptionAtEnd lScenario, lYear, lPeriod,
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,
lCustom4, bstrDescription
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>bstrDescription</i>	String (ByVal). The cell's description.

Example

InsertDescriptionAtEnd is used in the [Example](#) for InsertDataAtEnd.

InsertLineItemsAtEnd

Inserts line items for a cell at the bottom of an HsvMDDataBufferLite object.

Syntax

```
<HsvMDDataBufferLite>.InsertLineItemsAtEnd lScenario, lYear, lPeriod,  
lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3,  
lCustom4, vbSavedAsYTD, varadData, varabstrDescriptions
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>vbSavedAsYTD</i>	Boolean (ByVal). Determines how the cell's totals are saved. Pass TRUE if the cell's line items are saved as year-to-date, FALSE if saved as periodic.
<i>varadData</i>	Double array (ByVal). The data for the cell's line items.
<i>varabstrDescriptions</i>	String array (ByVal). The descriptions of the cell's line items.

InsertRecordIntoPMBuffer

For internal use.

InsertRecordIntoPMErrorsBuffer

For internal use.

RemoveAll

Clears all the cells from an HsvMDDataBufferLite object.

Syntax

```
<HsvMDDataBufferLite>.RemoveAll
```

SetCheckLineItemDetailsForCaseInsensitiveDuplicates

Specifies whether the system should check for case-insensitive duplicate line item descriptions. For example, *acmeData* and *ACMEDATA* would be considered case-insensitive duplicate descriptions.

If you enable checking for such duplicates, the system performs the check when the *HsvMDDataBufferLite* instance's data is passed to the *HsvData* method [UpdateDataUsingMDDataBuffer](#). With case-insensitive duplicate checking enabled, if the system detects a duplicate an error is thrown.

Syntax

```
<HsvMDDataBufferLite>.SetCheckLineItemDetailsForCaseInsensitive  
Duplicates vbEnable
```

Argument Description

vbEnable Boolean (ByVal). Specifies whether to check for case-insensitive duplicates. Pass TRUE to check, FALSE otherwise.

SetData

Inserts a cell's data into an *HsvMDDataBufferLite* object.

Syntax

```
<HsvMDDataBufferLite>.SetData lScenario, lYear, lPeriod, lView, lEntity,  
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,  
dData, vbIsNoData, vbAddToExistingData, plNumElementsInDataUnit
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member. You must pass a member ID for a user-defined member. Do not pass the member ID for the system-defined member <Scenario View>.
<i>lEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>lParent</i>	Long (ByVal). Pass the HFMConstants type library constant <code>MEMBERNOTUSED</code> .
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member. You must pass a member ID for a user-defined member. Do not pass a member ID for system-defined members such as <Entity Currency>.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.

Argument	Description
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>lCustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>lCustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>dData</i>	Double (ByVal). The cell's data.
<i>vbIsNoData</i>	Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.
<i>vbAddToExistingData</i>	Boolean (ByVal). If the HsvMDDataBufferLite object already contains data for the cell, this argument determines whether the existing data will be overwritten. Pass TRUE to add the <i>dData</i> argument's data to the existing data, FALSE to overwrite the existing data.
<i>plNumElementsInDataUnit</i>	Long. Returns the number of cells in the HsvMDDataBufferLite object that are in the subcube to which the cell belongs. This number returned is calculated after <i>SetData</i> adds the cell.

Example

The following example loops through an array of Account dimension member IDs and inserts data and descriptions for each account into a cell. The data and descriptions are taken from arrays of the same size. Note that the non-Account member IDs passed to *InsertDataAtEnd* and *InsertDescriptionAtEnd* are the same for each cell.

```

For i = LBound(laIDs) To UBound(laIDs)
' laIDs() is a Long array of Account IDs, daData() is a
' Double array of data.
m_cMDDataBufferLite.SetData m_lScen, m_lYear, m_lPer, m_lView, _
m_lEnt, m_lPar, m_lVal, laIDs(i), m_lICP, m_lCust1, _
m_lCust2, m_lCust3, m_lCust4, daData(i), False, False, _
lNumElems
' saDescs() is a String array of descriptions.
m_cMDDataBufferLite.SetDescription m_lScen, m_lYear, m_lPer, _
m_lEnt, m_lPar, m_lVal, laIDs(i), m_lICP, _
m_lCust1, m_lCust2, m_lCust3, m_lCust4, saDescs(i)
Next i

```

SetDescription

Inserts a cell's description into an HsvMDDataBufferLite object.

Syntax

```

<HsvMDDataBufferLite>.SetDescription lScenario, lYear, lPeriod, lEntity,
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
bstrDescription

```


Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>bstrDescription</i>	String (ByVal). The cell's description.

Example

`SetDescription` is used in the [Example](#) for `SetData`.

SetGrowByAmount

Creates additional memory for an `HsvMDDataBufferLite` object.

Syntax

```
<HsvMDDataBufferLite>.SetGrowByAmount lGrowByAmount
```

<i>Argument</i>	Description
<i>lGrowByAmount</i>	Long (ByVal). The number of cells for which you want to create memory.

SetLineItems

Inserts a cell's line items into an `HsvMDDataBufferLite` object.

Syntax

```
<HsvMDDataBufferLite>.SetLineItems lScenario, lYear, lPeriod, lEntity,
lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4,
vbSavedAsYTD, varadData, varabstrDescriptions
```

Argument	Description
<i>IScenario</i>	Long (ByVal). The member ID of the cell's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The member ID of the cell's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the cell's Entity dimension member.
<i>IParent</i>	Long (ByVal). The member ID of the <i>lEntity</i> argument's parent entity.
<i>IValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>vbSavedAsYTD</i>	Boolean (ByVal). Determines how the cell's totals are saved. Pass TRUE if the cell is saved as year-to-date, FALSE if saved as periodic.
<i>varadData</i>	Double array (ByVal). The cell's line item data.
<i>varabstrDescriptions</i>	String array (ByVal). The cell's line item descriptions.

Example

The following example loops through a Double array of data, inserting a line item for each array item. For each item, a description from an equally sized String array is also inserted.

```

For i = LBound(vdData) To UBound(vdData)
' vdData() is a Double array of data, vsDescs() is a
' String array of descriptions.
m_cMDBufferLite.SetLineItems m_lScen, m_lYear, m_lPer, _
m_lEnt, m_lPar, m_lVal, m_lAcct, m_lICP, m_lCust1, _
m_lCust2, m_lCust3, m_lCust4, True, vdData, vsDescs
Next i

```

SetSortedNature

For internal use.

Sort

For internal use.

IHsvMDDataBufferLite Interface

The IHsvMDDataBufferLite interface provides a method that changes the minimum number of subcubes cached to RAM for the HsvMDDataBufferLite object. For information on caching and the HsvMDDataBufferLite object, see [“HsvMDDataBufferLite Object Methods” on page 639](#).

Assign IHsvMDDataBufferLite object references with the `Set` and `New` keywords as shown below:

```
Dim cDataCache As IHsvMDDataBufferLite
Set cDataCache = New HSVMDARRAYSLib.HsvMDDataBufferLite
```

SetMinCubesInCache

Sets the number of subcubes that will be cached to RAM for the HsvMDDataBufferLite object.

Note: By default, the HsvMDDataBufferLite object caches one subcube to RAM. `SetMinCubesInCache` overrides this default.

Syntax

```
<IHsvMDDataBufferLite>.SetMinCubesInCache nMinCubesInCache
```

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

<code>nMinCubesInCache</code>	Integer (ByVal). The number of subcubes to cache.
-------------------------------	---

Example

The following example sets the number of cached subcubes to 2.

```
Dim cMDDataBufferCache As IHsvMDDataBufferLite
Set cMDDataBufferCache = New HSVMDARRAYSLib.HsvMDDataBufferLite
cMDDataBufferCache.SetMinCubesInCache 2
```

HsvTransactionData Object Methods

The HsvTransactionData object returns the transaction data generated by statutory consolidations. To return transaction data, you specify dimension members as selection criteria, then obtain the data for the specified members.

► The following steps provide an overview of how to return transaction data:

- 1 Set an HsvTransactionData object reference using the `Set` and `New` keywords as shown below:

```
Dim cHsvTransData As HsvTransactionData
Set cHsvTransData = New HSVMDARRAYSLib.HsvTransactionData
```

- 2 Call `Initialize` to set the current `Scenario` and `Year` dimension members of the transaction data to be returned.
- 3 Specify the dimension members to be used as selection criteria by calling `SetQueryItem` once per dimension member. For each call, pass one of the `HFMConstants` type library constants listed in “[Transaction Dimension Constants](#)” on page 845 along with the applicable member ID. For example, the following lines specify the current entity, parent, and period as selection criteria:

```
cHsvTransData.SetQueryItem _
TRANSACTION_DIMENSIONS_CUR_ENTITY, lEnt
cHsvTransData.SetQueryItem _
TRANSACTION_DIMENSIONS_CUR_PARENT, lPar
cHsvTransData.SetQueryItem _
TRANSACTION_DIMENSIONS_CUR_PERIOD, lPer
```

- 4 To have the `HsvTransactionData` object populated with an array of the transaction data, pass the `HsvTransactionData` object to the `HsvData` object’s `GetTransactionData` method.
- 5 Begin the enumeration of the array with which `HsvData.GetTransactionData` has populated the `HsvTransactionData` object by calling `BeginDataEnum`, which returns the number of items in the array.
- 6 To get the data, call the `HsvTransactionData` object’s `GetTransactionData` method. You can use the count of items returned by `BeginDataEnum` to loop through the object’s array.

Caution! Do not confuse this method with the `HsvData` object’s `GetTransactionData` method.

- 7 After finishing with the data, end the enumeration by calling `EndDataEnum`.

Tip: For an example that illustrates these steps, see [Example](#).

The `HsvTransactionData` object’s methods are summarized in “[HsvTransactionData Object Overview](#)” on page 99, and are described in detail in the following topics.

BeginDataEnum

Begins an enumeration of an `HsvTransactionData` object, and returns a count of the items with which `HsvData.GetTransactionData` has populated the object.

Note: After working with the enumerated data, end the enumeration by calling [EndDataEnum](#).

Syntax

```
<HsvTransactionData>.BeginDataEnum p1NumItems
```

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

<code>p1NumItems</code>	Long. Returns a count of the items in the <code>HsvTransactionData</code> object’s array.
-------------------------	---

Example

For an example that uses this method, see the [Example](#) for `GetTransactionData`.

BeginQueryEnum

For internal use.

EndDataEnum

Ends an `HsvTransactionData` object enumeration. You should call `EndDataEnum` after you have finished working with an enumeration that was started with `BeginDataEnum`.

Syntax

```
<HsvTransactionData>.EndDataEnum
```

Example

For an example that uses this method, see the [Example](#) for `GetTransactionData`.

EndQueryEnum

For internal use.

GetFixedDimensionMembers

For internal use.

GetQueryItem

For internal use.

GetTransactionData

Returns source and destination data for a transaction, given the index of the transaction within the `HsvTransactionData` object's array of transactions. In addition to the data, the member IDs of the transaction's dimension members are returned.

Syntax

```
<HsvTransactionData>.GetTransactionData lItem, plCurrentPeriod,  
plCurrentParent, plCurrentEntity, plDestEntity, plDestAccount,  
plDestValue, plDestICP, plDestCustom1, plDestCustom2, plDestCustom3,  
plDestCustom4, plSourceScenario, plSourceYear, plSourcePeriod,  
plSourceParent, plSourceEntity, plSourceValue, plSourceAccount,  
plSourceICP, plSourceView, plSourceCustom1, plSourceCustom2,
```

plSourceCustom3, plSourceCustom4, pdDestData, pdSourceData, pdFactor, pbstrNature

Argument	Description
<i>Item</i>	Long (ByVal). The index of the transaction in the HsvTransactionData object's array. This is a zero-based index. Tip: BeginDataEnum returns a count of the items in the HsvTransactionData object. For more information, see "BeginDataEnum" on page 668 .
<i>plCurrentPeriod</i>	Long. Returns the member ID of the transaction's current period.
<i>plCurrentParent</i>	Long. Returns the member ID of the transaction's current parent entity.
<i>plCurrentEntity</i>	Long. Returns the member ID of the transaction's current child entity.
<i>plDestEntity</i>	Long. Returns the member ID of the transaction's destination child entity.
<i>plDestAccount</i>	Long. Returns the member ID of the transaction's destination account.
<i>plDestValue</i>	Long. Returns the member ID of the transaction's destination Value dimension member.
<i>plDestICP</i>	Long. Returns the member ID of the transaction's destination Intercompany Partner dimension member.
<i>plDestCustom1</i>	Long. Returns the member ID of the transaction's destination Custom 1 dimension member.
<i>plDestCustom2</i>	Long. Returns the member ID of the transaction's destination Custom 2 dimension member.
<i>plDestCustom3</i>	Long. Returns the member ID of the transaction's destination Custom 3 dimension member.
<i>plDestCustom4</i>	Long. Returns the member ID of the transaction's destination Custom 4 dimension member.
<i>plSourceScenario</i>	Long. Returns the member ID of the transaction's source scenario.
<i>plSourceYear</i>	Long. Returns the member ID of the transaction's source year.
<i>plSourcePeriod</i>	Long. Returns the member ID of the transaction's source period.
<i>plSourceParent</i>	Long. Returns the member ID of the transaction's source parent entity.
<i>plSourceEntity</i>	Long. Returns the member ID of the transaction's source child entity.
<i>plSourceValue</i>	Long. Returns the member ID of the transaction's source Value dimension member.
<i>plSourceAccount</i>	Long. Returns the member ID of the transaction's source account.
<i>plSourceICP</i>	Long. Returns the member ID of the transaction's source Intercompany Partner dimension member.
<i>plSourceView</i>	Long. Returns the member ID of the transaction's source View dimension member.
<i>plSourceCustom1</i>	Long. Returns the member ID of the transaction's source Custom 1 dimension member.
<i>plSourceCustom2</i>	Long. Returns the member ID of the transaction's source Custom 2 dimension member.
<i>plSourceCustom3</i>	Long. Returns the member ID of the transaction's source Custom 3 dimension member.
<i>plSourceCustom4</i>	Long. Returns the member ID of the transaction's source Custom 4 dimension member.

Argument	Description
<code>pdDestData</code>	Double. Returns the transaction's destination data.
<code>pdSourceData</code>	Double. Returns the transaction's source data.
<code>pdFactor</code>	Double. Returns the factor that was passed in the <code>HS.Con</code> rules function. Note: For details on rules functions, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i> .
<code>pbstrNature</code>	String. Returns the nature string passed in the <code>HS.Con</code> rules function.

Example

The following example shows how to loop through transaction data using the current period, parent entity, and child entity as search criteria. The example illustrates the steps in [“HsvTransactionData Object Methods” on page 667](#) that show how to use the `HsvTransactionData` object.

```
Dim cHsvTransData As HsvTransactionData, cHsvData As HsvData
Dim lNumItems As Long, lIndex As Long, lCurPeriod As Long
Dim lCurParent As Long, lCurEnt As Long, lDestEnt As Long
Dim lDestAcct As Long, lDestVal As Long, lDestICP As Long
Dim lDestCust1 As Long, lDestCust2 As Long, lDestCust3 As Long
Dim lDestCust4 As Long, lSrcScen As Long, lSrcYear As Long
Dim lSrcPer As Long, lSrcParent As Long, lSrcEnt As Long
Dim lSrcVal As Long, lSrcAcct As Long, lSrcICP As Long
Dim lSrcView As Long, lSrcCust1 As Long, lSrcCust2 As Long
Dim lSrcCust3 As Long, lSrcCust4 As Long, dDestData As Double
Dim dSrcData As Double, dFactor As Double, sNature As String
Set cHsvData = m_cHsvSession.Data
Set cHsvTransData = New HSVMDARRAYSlib.HsvTransactionData
cHsvTransData.Initialize m_lScen, m_lYear
cHsvTransData.SetQueryItem TRANSACTION_DIMENSIONS_CUR_ENTITY, _
m_lEnt
cHsvTransData.SetQueryItem TRANSACTION_DIMENSIONS_CUR_PARENT, _
m_lPar
cHsvTransData.SetQueryItem TRANSACTION_DIMENSIONS_CUR_PERIOD, _
m_lPer
cHsvData.GetTransactionData cHsvTransData
cHsvTransData.BeginDataEnum lNumItems
For lIndex = 1 To lNumItems
    cHsvTransData.GetTransactionData lIndex, lCurPeriod, _
lCurParent, lCurEnt, lDestEnt, lDestAcct, lDestVal, _
lDestICP, lDestCust1, lDestCust2, lDestCust3, lDestCust4, _
lSrcScen, lSrcYear, lSrcPer, lSrcParent, lSrcEnt, _
lSrcVal, lSrcAcct, lSrcICP, lSrcView, lSrcCust1, _
lSrcCust2, lSrcCust3, lSrcCust4, dDestData, dSrcData, _
dFactor, sNature
    '... Insert code that uses the transactions' data here
Next lIndex
cHsvTransData.EndDataEnum
```

Initialize

Specifies the Scenario and Year dimension members for the HsvTransactionData object's transaction data. You must call `Initialize` before calling the other HsvTransactionData object methods.

Syntax

```
<HsvTransactionData>.Initialize lCurrentScenario, lCurrentYear
```

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

<i>lCurrentScenario</i>	Long (ByVal). The member ID of the transactions' current scenario.
-------------------------	--

<i>lCurrentYear</i>	Long (ByVal). The member ID of the transactions' current year.
---------------------	--

Example

For an example that uses this method, see the [Example](#) for `GetTransactionData`.

SetAccessRight

For internal use.

SetQueryItem

Specifies a transaction dimension member as a selection criterion for an HsvTransactionData object. To set multiple selection criteria for an HsvTransactionData object, make one `SetQueryItem` call per selection criterion.

Syntax

```
<HsvTransactionData>.SetQueryItem nTransactionDimension, lValue
```

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

<i>nTransactionDimension</i>	Integer (ByVal). The transaction dimension for the selection criterion. Pass one of the HFMConstants type library constants listed in "Transaction Dimension Constants" on page 845 .
------------------------------	---

<i>lValue</i>	Long (ByVal). The member ID of the dimension member to be set as a selection criterion. This ID must identify a member of the dimension specified with the <i>nTransactionDimension</i> argument.
---------------	---

Example

For an example that uses this method, see the [Example](#) for `GetTransactionData`.

SetTransactionData

For internal use.

HsvMDIndexList Object Methods

An HsvMDIndexList object contains a list of indexes that represent points-of-view in an HsvMDDataBuffer or HsvMDDataBufferLite object.

Create HsvMDIndexList objects with `CreateDataIndexList`, which is available to both the HsvMDDataBuffer and HsvMDDataBufferLite objects. `CreateDataIndexList` enables you to specify the items that will be added to the HsvMDIndexList object.

The HsvMDIndexList object's methods are summarized in [Table 34 on page 100](#), and are described in detail in the following topics.

GetItem

Returns the member IDs of the dimension members for an item in an HsvMDIndexList object.

Syntax

```
<HsvMDIndexList>.GetItem lItem, plScenario, plYear, plPeriod, plView,  
plEntity, plParent, plValue, plAccount, plICP, plCustom1, plCustom2,  
plCustom3, plCustom4
```

Argument Description

<i>lItem</i>	Long (ByVal). The index for which the member IDs will be returned. Use GetNumItems to get the number of indexes in the HsvMDIndexList object.
<i>plScenario</i>	Long. Returns the member ID of the index's Scenario dimension member.
<i>plYear</i>	Long. Returns the member ID of the index's Year dimension member.
<i>plPeriod</i>	Long. Returns the member ID of the index's Period dimension member.
<i>plView</i>	Long. Returns the member ID of the index's View dimension member.
<i>plEntity</i>	Long. Returns the member ID of the index's Scenario dimension member.
<i>plParent</i>	Long. Returns the member ID of the <i>plEntity</i> argument's entity.
<i>plValue</i>	Long. Returns the member ID of the index's Value dimension member.
<i>plAccount</i>	Long. Returns the member ID of the index's Account dimension member.
<i>plICP</i>	Long. Returns the member ID of the index's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the index's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the index's Custom 2 dimension member.

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

<i>plCustom3</i>	Long. Returns the member ID of the index's Custom 3 dimension member.
------------------	---

<i>plCustom4</i>	Long. Returns the member ID of the index's Custom 4 dimension member.
------------------	---

Example

GetItem is used in the [Example](#) for CreateDataIndexList.

GetNumItems

Returns the number of indexes in an HsvMDIndexList object. Use this to loop with [GetItem](#).

Syntax

```
<HsvMDIndexList>.GetNumItems plNumItems
```

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

<i>plNumItems</i>	Long. Returns the number of indexes contained by the object.
-------------------	--

Example

GetNumItems is used in the [Example](#) for CreateDataIndexList.

HsvICTransactionsData Object Methods

The HsvICTransactionsData object provides methods for working with intercompany transactions. An HsvICTransactionsData instance contains an array of intercompany transactions for a given scenario, year, and period, and enables you to access specific transactions and to process all of the array's transactions.

The HsvICTransactionsData object supplements the intercompany transaction features exposed by the HsvICM object. For example, you can use the HsvICTransactionsData object to get the details of an existing intercompany transaction.

To use the HsvICTransactionsData object, you must call certain methods in the sequence described in the following steps.

► To use the HsvICTransactionsData object:

- 1 Set an HsvICTransactionsData object reference using the `Set` and `New` keywords as shown below:

```
Dim cIcTransData As HsvICTransactionsData  
Set cIcTransData = New HsvICTransactionsData
```

- 2 Call `Initialize` to specify the scenario, year, and period for the transactions.
- 3 Perform one of the following steps:

- a. If you are processing all transactions for a scenario, year, and period, pass the object reference to `HsvICM.ProcessAllICTrans`.
- b. If you are working with given transactions, pass the `HsvICTransactionsData` object reference to `HsvICM.GetICTransactions` and then take the following steps:
 - i. Begin the enumeration of the array by calling `BeginDataEnum`, which returns the number of transactions added to the array.
 - ii. Work with the transactions, which are identified by index. This is a zero-based index. You can obtain the upper bounds of the index by subtracting one from the count returned by `BeginDataEnum`.
 - iii. After you have finished with the transactions, clean up by calling `EndDataEnum`.

For an code snippet that demonstrates these steps, see the examples for the `HsvICM` methods `HsvICM.ProcessAllICTrans` and `GetICTransactionData`.

The `HsvICTransactionsData` object's methods are summarized in “[HsvICTransactionsData Object Overview](#)” on page 100, and are described in detail in the following topics.

AddAccountCustomCombination

For internal use.

AddICTransactionData

For internal use.

AddQueryDimensionMemberID

For internal use.

AddQueryField

For internal use.

AddQueryFieldItem

For internal use.

AddQueryOrderField

For internal use.

BeginDataEnum

Begins the enumeration of an `HsvICTransactionsData` instance's array of transactions, and returns a count of the transactions that the array contains.

Call this method after calling [Initialize](#) and `HsvICM.GetICTransactions` and before calling the other `HsvICTransactionsData` methods.

Note: Once you have finished working with the data in the `HsvICTransactionsData` instance, clean up by calling [EndDataEnum](#).

Syntax

```
<HsvICTransactionsData>.BeginDataEnum plNumItems
```

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

<code>plNumItems</code>	Long. Returns a count of the transactions contained by the <code>HsvICTransactionsData</code> instance.
-------------------------	---

Example

`BeginDataEnum` is used in the example for [Initialize](#).

BeginQueryFieldEnum

For internal use.

BeginQueryOrderEnum

For internal use.

EndDataEnum

Cleans up an `HsvICTransactionsData` instance; you should always call `EndDataEnum` after you finish working with an enumeration of transactions for which you have called [BeginDataEnum](#).

Syntax

```
<HsvICTransactionsData>.EndDataEnum
```

Example

`EndDataEnum` is used in the example for [Initialize](#).

EndQueryFieldEnum

For internal use.

EndQueryOrderEnum

For internal use.

GetAccessRights

Gets the user's read and write access rights to an intercompany transaction.

Syntax

```
<HsvICTransactionsData>.GetAccessRights lItem, pvbReadAccess, plWriteAccess
```

Argument	Description
<i>lItem</i>	Long (ByVal). The index of the transaction in the HsvICTransactionsData instance's array of intercompany transactions. You can get the upper bounds of the index by subtracting one from the count of transactions returned by BeginDataEnum .
<i>pvbReadAccess</i>	Boolean. Returns the user's read access rights. TRUE indicates that the user has read access.
<i>plWriteAccess</i>	Long. Returns the user's write access rights. The following list describes the valid values: <ul style="list-style-type: none">● 0 = Read-only access.● 1 = User is allowed to specify reason codes for the Intercompany Partner transactions even if the user lacks write access to those transactions.● 2 = Write access.

GetAccountCustomCombination

For internal use.

GetAccountCustomCombinationTotal

For internal use.

GetEntityPartnerOption

For internal use.

GetErrorStatus

Returns the HRESULT associated with a given intercompany transaction.

Syntax

```
<HsvICTransactionsData>.GetErrorStatus lItemIndex, plHRStatus
```

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

<i>ItemIndex</i>	Long (ByVal). The index of the transaction in the HsvICTransactionsData instance's array of intercompany transactions.
------------------	--

Tip: You can get the upper bounds of the index by subtracting one from the count of transactions returned by [BeginDataEnum](#).

<i>plHRStatus</i>	Long. Returns the HRESULT. If there's no error, zero (0) is returned, otherwise a non-zero error number is returned.
-------------------	--

Tip: For information on Financial Management error numbers, see [Chapter 22, "Error Handling and the HsvResourceManager Type Library."](#)

Example

GetErrorStatus is used in the example for [GetICTransactionData](#).

GetFilterOptions

For internal use.

GetFixedDimensionMembers

Returns the member IDs of the Scenario, Year, and Period dimension members for an HsvICTransactionsData instance's transactions.

Note: To get the member IDs of the other dimension members for a given transaction, use [GetICTransactionCell](#).

Syntax

```
<HsvICTransactionsData>.GetFixedDimensionMembers plScenario, plYear,  
plPeriod
```

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

<i>plScenario</i>	Long. Returns the member ID of the Scenario dimension member.
-------------------	---

<i>plYear</i>	Long. Returns the member ID of the Year dimension member.
---------------	---

<i>plPeriod</i>	Long. Returns the member ID of the Period dimension member.
-----------------	---

GetICTransactionCell

Returns the member IDs of an intercompany transaction's Entity, Intercompany Partner, Account, and Custom dimension members.

Note: To get the member IDs of the Scenario, Year, and Period dimension members for an `HsvICTransactionsData` instance's transactions, use [GetFixedDimensionMembers](#). To get details on a given transaction, use [GetICTransactionData](#).

Syntax

```
<HsvICTransactionsData>.GetICTransactionCell lItem, plEntity, plICP,  
plAccount, plCustom1, plCustom2, plCustom3, plCustom4
```

Argument Description

lItem Long (ByVal). The index of the transaction in the `HsvICTransactionsData` instance's array of intercompany transactions.

Tip: You can get the upper bounds of the index by subtracting one from the count of transactions returned by [BeginDataEnum](#).

plEntity Long. Returns the member ID of the Entity dimension member.

plICP Long. Returns the member ID of the Intercompany Partner dimension member.

plAccount Long. Returns the member ID of the Account dimension member.

plCustom1 Long. Returns the member ID of the Custom 1 dimension member.

plCustom2 Long. Returns the member ID of the Custom 2 dimension member.

plCustom3 Long. Returns the member ID of the Custom 3 dimension member.

plCustom4 Long. Returns the member ID of the Custom 4 dimension member.

GetICTransactionData

Returns an intercompany transaction's details, including the transaction's amounts, currency, dimension members, and so on.

Syntax

```
<HsvICTransactionsData>.GetICTransactionData lItem, plSequenceId,  
plEntity, plICP, plAccount, plCustom1, plCustom2, plCustom3, plCustom4,  
plTransactionCurrency, plPostStatus, plMatchStatus, plReasonCode, plType,  
pdModifiedDate, pdTransactionDate, pdAmount, pdLocalAmount, pdRate,  
pbstrUser, pbstrId, pbstrSubId, pbstrReferenceId, pbstrMatchCode,  
pbstrComment1, pbstrComment2
```

Argument	Description
<i>Item</i>	<p>Long (ByVal). The index of the transaction in the HsvICTransactionsData instance's array of intercompany transactions.</p> <p>Tip: You can get the upper bounds of the index by subtracting one from the count of transactions returned by BeginDataEnum.</p>
<i>plSequenceld</i>	<p>Long. Returns the transaction's sequence ID.</p> <p>Note: Sequence IDs are internal unique identifiers of transactions.</p>
<i>plEntity</i>	Long. Returns the member ID of the transaction's Entity dimension member.
<i>plICP</i>	Long. Returns the member ID of the transaction's Intercompany Partner dimension member.
<i>plAccount</i>	Long. Returns the member ID of the transaction's Account dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the transaction's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the transaction's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the transaction's Custom 3 dimension member.
<i>plCustom4</i>	Long. Returns the member ID of the transaction's Custom 4 dimension member.
<i>plTransactionCurrency</i>	<p>Long. Returns the currency ID of the transaction's currency.</p> <p>Tip: You can get the currency's label by passing this ID to the HsvCurrencies method GetCurrencyLabel.</p>
<i>plPostStatus</i>	Long. Returns the transaction's posting status. Valid values are represented by the HFMConstants type library constants listed in " Posting Status Constants " on page 875.
<i>plMatchStatus</i>	Long. Returns the transaction's matching status. Valid values are represented by the HFMConstants type library constants listed in " Matching Status Constants " on page 875.
<i>plReasonCode</i>	<p>Long. Returns the ID of the transaction's reason code.</p> <p>Tip: To get the reason code, pass this ID to the HsvICM method GetICReasonCodeLabel.</p>
<i>plType</i>	Long. <i>For internal use.</i>
<i>pdModifiedDate</i>	Double. Returns a timestamp that identifies when the transaction was last modified. The timestamp is formatted as a Double.
<i>pdTransactionDate</i>	Double. Returns the transaction date formatted as a Double.
<i>pdAmount</i>	Double. Returns the transaction amount.
<i>pdLocalAmount</i>	Double. Returns the entity currency amount.
<i>pdRate</i>	Double. Returns the conversion rate, which represents the transaction amount divided by the entity currency amount.
<i>pbstrUser</i>	String. Returns the username of the user who last updated the transaction.
<i>pbstrId</i>	String. Returns the Transaction ID.
<i>pbstrSubId</i>	String. Returns the transaction's Sub ID.

Argument	Description
<i>pbstrReferenceId</i>	String. Returns the transaction's Reference ID.
<i>pbstrMatchCode</i>	String. Returns the transaction's match code.
<i>pbstrComment1</i>	String. Returns the transaction's first comment.
<i>pbstrComment2</i>	String. Returns the transaction's second comment.

Example

The following subroutine prints to Visual Basic's Immediate window the Transaction ID, Sub ID, and hexadecimal HRESULT of the transactions contained by an `HsvICTransactionsData` instance. The subroutine's arguments specify the transactions' scenario, year, and period.

`GetErrorStatus` obtains each transaction's HRESULT.

```
Sub printTransErrs(lScen As Long, lYear As Long, lPer As Long)
Dim cIcTransData As HsvICTransactionsData, cIcm As HsvICM, lItems As Long
Dim lSeqId As Long, lEnt As Long, lIcp As Long, lAcct As Long
Dim lCust1 As Long, lCust2 As Long, lCust3 As Long, lCust4 As Long
Dim lCurr As Long, lPostStat As Long, lMatchStat As Long, lRCode As Long
Dim lType As Long, dModDate As Double, dTrDate As Double, dAmt As Double
Dim dLocAmt As Double, dRate As Double, sUser As String, lErr As Long
Dim sTranId As String, sSubId As String, sRefId As String
Dim sMatchCode As String, sComm1 As String, sComm2 As String
Set cIcTransData = New HsvICTransactionsData
'g_cSession is an HsvSession object reference
Set cIcm = g_cSession.ICM
cIcTransData.Initialize lScen, lYear, lPer
cIcm.GetICTransactions cIcTransData, True
cIcTransData.BeginDataEnum lItems
For i = 0 To lItems - 1
    cIcTransData.GetErrorStatus i, lErr
    cIcTransData.GetICTransactionData i, lSeqId, lEnt, lIcp, lAcct, _
        lCust1, lCust2, lCust3, lCust4, lCurr, lPostStat, lMatchStat, _
        lRCode, lType, dModDate, dTrDate, dAmt, dLocAmt, dRate, sUser, _
        sTranId, sSubId, sRefId, sMatchCode, sComm1, sComm2
    Debug.Print sTranId & " - " & sSubId & ": " & Hex(lErr)
Next i
cIcTransData.EndDataEnum
End Sub
```

GetMatchCode

For internal use.

GetNumTransactionsCached

For internal use.

GetPagingOption

For internal use.

GetPartnerAsEntityList

For internal use.

GetPartnerQueryDimensionMemberIDs

For internal use.

GetQueryDimensionMemberIDs

For internal use.

GetQueryFieldInformation

For internal use.

GetQueryFieldItem

For internal use.

GetQueryOrderField

For internal use.

GetTotalTransactions

For internal use.

GetTransGroupType

For internal use.

Initialize

Specifies the scenario, year, and period of the transactions that the `HsvICTransactionsData` instance will contain. You must call `Initialize` before calling the `HsvICM` methods

[GetICTransactions](#) and [ProcessAllICTrans](#) and before using the other [HsvICTransactionsData](#) methods.

Syntax

```
<HsvICTransactionsData>.Initialize lScenario, lYear, lPeriod
```

Argument Description

lScenario Long (ByVal). The member ID of the transactions' Scenario dimension member.

lYear Long (ByVal). The member ID of the transactions' Year dimension member.

lPeriod Long (ByVal). The member ID of the transactions' Period dimension member.

Example

The following subroutine shows how to loop through intercompany transactions.

[HsvICM.GetICTransactions](#) populates the object with the transactions for the specified scenario, year, and period. [BeginDataEnum](#) gets the number of transactions, and [GetICTransactionData](#) is called for each transaction, with the Transaction ID and Sub ID printed to Visual Basic's Immediate window.

```
Sub printTransactionIds(lScen As Long, lYear As Long, lPer As Long)
Dim cIcTransData As HsvICTransactionsData, cIcm As HsvICM, lItems As Long
Dim lSeqId As Long, lEnt As Long, lIcp As Long, lAcct As Long
Dim lCust1 As Long, lCust2 As Long, lCust3 As Long, lCust4 As Long
Dim lCurr As Long, lPostStat As Long, lMatchStat As Long, lRCode As Long
Dim lType As Long, dModDate As Double, dTrDate As Double, dAmt As Double
Dim dLocAmt As Double, dRate As Double, sUser As String
Dim sTranId As String, sSubId As String, sRefId As String
Dim sMatchCode As String, sComm1 As String, sComm2 As String
Set cIcTransData = New HsvICTransactionsData
'g_cSession is an HsvSession object reference
Set cIcm = g_cSession.ICM
cIcTransData.Initialize lScen, lYear, lPer
cIcm.GetICTransactions cIcTransData, True
cIcTransData.BeginDataEnum lItems
For i = 0 To lItems - 1
    cIcTransData.GetICTransactionData i, lSeqId, lEnt, lIcp, lAcct, _
        lCust1, lCust2, lCust3, lCust4, lCurr, lPostStat, lMatchStat, _
        lRCode, lType, dModDate, dTrDate, dAmt, dLocAmt, dRate, sUser, _
        sTranId, sSubId, sRefId, sMatchCode, sComm1, sComm2
    Debug.Print sTranId & " - " & sSubId
Next i
cIcTransData.EndDataEnum
End Sub
```

InitializeSequenceIdMap

For internal use.

IsEntityInPartnerAsEntityList

For internal use.

IsICTransactionValid

For internal use.

RemoveQueryOrder

For internal use.

SetAccessRights

Sets the user's read and write access rights to an intercompany transaction.

Note: If you remove a user's access rights to a transaction, the transaction remains in the HsvICTransactionsData instance's array of transactions. However, attempting to access the transaction would cause an error.

Syntax

```
<HsvICTransactionsData>.SetAccessRights lItem, vbReadAccess, lWriteAccess
```

Argument	Description
<i>lItem</i>	Long (ByVal). The index of the transaction in the HsvICTransactionsData instance's array of intercompany transactions. You can get the upper bounds of the index by subtracting one from the count of transactions returned by BeginDataEnum .
<i>vbReadAccess</i>	Boolean (ByVal). A flag that specifies the user's read access rights. Pass TRUE to grant read access rights, FALSE to remove them.
<i>lWriteAccess</i>	Long (ByVal). A flag that specifies the user's write access rights. Pass one of the following values: <ul style="list-style-type: none">● 0 = Read-only access.● 1 = Allow the user to specify reason codes for the Intercompany Partner transactions even if the user lacks write access to those transactions.● 2 = Write access.

SetEntityPartnerOption

For internal use.

SetErrorStatus

For internal use.

SetFilterOptions

For internal use.

SetICTransactionData

For internal use.

SetPagingOption

For internal use.

SetPartnerAsEntityList

For internal use.

SetPartnerQueryDimensionMemberIDs

For internal use.

SetQueryDimensionMemberIDs

For internal use.

SetTotalTransactions

For internal use.

SortByCell

For internal use.

Uninitialize

For internal use.

In This Chapter

About Subcube Items.....	687
HsvCurrencyCube Object Methods	688
HsvNodeCube Object Methods	693

This chapter describes the members of the HsvDataCubes type library. The objects and methods of this type library are used to access data and information at the subcube level.

To use the HsvDataCubes type library, you must reference `HsvDataCubes.dll` in your project. The HsvDataCubes type library contains the following objects:

- The HsvCurrencyCube object provides access to currency subcubes.
- The HsvNodeCube object provides access to node subcubes.

For descriptions of currency and node subcubes, see “[About Subcubes](#)” on page 43.

About Subcube Items

Some methods apply to the *items* in a subcube. A subcube’s items are the valid Value>Account>Intercompany Partner>Custom dimension intersections for the subcube. Each intersection applies to multiple Period and View dimension members, meaning that items and cells have a one-to-many relationship.

To access cells by using a subcube’s items, you first return a count of a subcube’s items with `BeginEnumerationOfStoredData`, which takes the member ID of the items’ Value dimension member. You can then access information for an item’s cell by passing the item’s index number and the member IDs of the cell’s Period and View dimension members to `GetOneCellFromStoredItem`.

Caution! `BeginEnumerationOfStoredData` locks a subcube’s cells. To remove the lock, call `EndEnumerationOfStoredData`.

You can also get the member IDs of an item’s Value, Account, Intercompany Partner, and Custom dimension members with `GetPOVFromStoredItem`.

HsvCurrencyCube Object Methods

The HsvCurrencyCube object provides access to currency subcubes, which are described in “About Subcubes” on page 43. The HsvCurrencyCube’s object’s methods are summarized in “HsvCurrencyCube Object Overview” on page 103, and are described in detail in the following topics.

HsvCurrencyCube object references are obtained with the HsvData object’s [GetCurrencyCube](#) method. [GetCurrencyCube](#) takes member IDs of the desired subcube’s Scenario, Year, Entity, and Value dimension members, and returns an object reference that provides access to the subcube.

BeginEnumerationOfStoredData

Returns a count of the subcube items for a Value dimension member, while also locking the subcube’s cells to prevent users from changing data. [BeginEnumerationOfStoredData](#) takes the member ID of the Value dimension member. (For details on items, see “About Subcube Items” on page 687.)

Tip: Remove the lock by calling [EndEnumerationOfStoredData](#).

Syntax

```
<HsvCurrencyCube>.BeginEnumerationOfStoredData lValue, plHandle,  
plTotalNumStoredItems
```

Argument	Description
<i>lValue</i>	Long (ByVal). The member ID of the subcube items’ Value dimension member.
<i>plHandle</i>	Long. Returns a handle. You must pass this handle to other HsvCurrencyCube methods that have a “handle” argument.
<i>plTotalNumStoredItems</i>	Long. Returns a count of the subcube items for the <i>lValue</i> argument’s Value dimension member. Use this count to determine the valid range of index numbers for GetOneCellFromStoredItem or GetPOVFromStoredItem .

Example

[BeginEnumerationOfStoredData](#) is used in the [Example](#) for [GetOneCellFromStoredItem](#).

EndEnumerationOfStoredData

Unlocks a subcube that was locked by [BeginEnumerationOfStoredData](#).

Tip: Whenever `BeginEnumerationOfStoredData` is called, use `EndEnumerationOfStoredData` in any subsequent error handling to make sure that the subcube is unlocked.

Syntax

```
<HsvCurrencyCube>.EndEnumerationOfStoredData lHandle
```

Argument Description

lHandle Long (ByVal). The item's handle. Pass the handle that was returned by [BeginEnumerationOfStoredData](#).

Example

`EndEnumerationOfStoredData` is used in the [Example](#) for `GetOneCellFromStoredItem`.

GetCell

Returns the data in and transaction status of a cell in a subcube.

Syntax

```
<HsvCurrencyCube>.GetCell lPeriod, lView, lValue, lAccount, lICP, lCustom1,  
lCustom2, lCustom3, lCustom4, pdData, pbyTransType
```

Argument Description

lPeriod Long (ByVal). The member ID of the cell's Period dimension member.

lView Long (ByVal). The member ID of the cell's View dimension member.

lValue Long (ByVal). The member ID of the cell's Value dimension member.

lAccount Long (ByVal). The member ID of the cell's Account dimension member.

lICP Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.

lCustom1 Long (ByVal). The member ID of the cell's Custom 1 dimension member.

lCustom2 Long (ByVal). The member ID of the cell's Custom 2 dimension member.

lCustom3 Long (ByVal). The member ID of the cell's Custom 3 dimension member.

lCustom4 Long (ByVal). The member ID of the cell's Custom 4 dimension member.

pdData Double. Returns the cell's data.

pbyTransType Byte. Returns the cell's transaction status. For a list of `HFMConstants` type library constants that represent the valid statuses, see ["Cell Transaction Type Constants" on page 846](#).

Example

The following example assigns a cell's data and transaction status to the `dData` and `bytStatus` variables. The example assumes that the member IDs passed to `GetCurrencyCube` and `GetCell` are obtained from another procedure.

```
Dim cHsvCurrencyCube As HsvCurrencyCube
Dim dData As Double, bytStatus As Byte
m_cHsvData.GetCurrencyCube m_lScen, m_lYear, m_lEnt, m_lVal, _
cHsvCurrencyCube
cHsvCurrencyCube.GetCell m_lPer, m_lView, m_lVal, m_lAcct, _
m_lICP, m_lCust1, m_lCust2, m_lCust3, m_lCust4, dData, _
bytStatus
```

GetFixedDimensionMembers

Returns the member IDs of a subcube's Scenario, Year, and Entity dimension members, as well as the member ID of the subcube's input Value dimension member.

Syntax

```
<HsvCurrencyCube>.GetFixedDimensionMembers plScenario, plYear, plEntity,
plInputValueID
```

Argument	Description
<i>plScenario</i>	Long. Returns the member ID of the subcube's Scenario dimension member.
<i>plYear</i>	Long. Returns the member ID of the subcube's Year dimension member.
<i>plEntity</i>	Long. Returns the member ID of the subcube's Entity dimension member.
<i>plInputValueID</i>	Long. Returns the member ID of the subcube's input Value dimension member. For example, if the <code>HsvCurrencyCube</code> object has been initialized with a Value dimension member named USD Total , this argument would return the member ID of the Value dimension member named USD .

GetOneCellFromStoredItem

Returns cell information such as a cell's data and member IDs. You identify the cell with the index number of the subcube item and the member IDs of the cell's Period and View dimension members. (For details on items, see [“About Subcube Items” on page 687](#).)

Syntax

```
<HsvCurrencyCube>.GetOneCellFromStoredItem lHandle, lItem, lPeriod, lView,
plValue, plAccount, plICP, plCustom1, plCustom2, plCustom3, plCustom4,
pvbDimensionMembersAreValid, pdData, pbyTransType
```

Argument	Description
<i>lHandle</i>	Long (ByVal). The item's handle. You must pass the handle that was returned by BeginEnumerationOfStoredData .

Argument	Description
<i>lItem</i>	Long (ByVal). The index of the item within the count of items returned by <code>BeginEnumerationOfStoredData</code> . Note: This is a zero-based index. For example, if <code>BeginEnumerationOfStoredData</code> returns a count of 5 items, the valid index numbers are 0 through 4.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>plValue</i>	Long. Returns the member ID of the cell's Value dimension member.
<i>plAccount</i>	Long. Returns the member ID of the cell's Account dimension member.
<i>plICP</i>	Long. Returns the member ID of the cell's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the cell's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the cell's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the cell's Custom 3 dimension member.
<i>plCustom4</i>	Long. Returns the member ID of the cell's Custom 4 dimension member.
<i>pvbDimensionMembersAreValid</i>	Boolean. Indicates whether the cell is valid or obsolete. Returns TRUE if valid, FALSE if obsolete. An obsolete cell can exist in cases where metadata has been changed and data has been loaded in merge mode. For performance reasons, Financial Management does not delete obsolete cells from the database. Tip: To perform operations with only data for valid cells, test this argument for TRUE.
<i>pdData</i>	Double. Returns the cell's data.
<i>pbyTransType</i>	Byte. Returns the cell's transaction status. For a list of HFMConstants type library constants that represent the valid statuses, see "Cell Transaction Type Constants" on page 846 .

Example

The following example prints to Visual Basic's Immediate window the Account labels of and data in valid cells. `BeginEnumerationOfStoredData` returns the number of items, and the example then loops through these items with `GetOneCellFromStoredItem`. If `GetOneCellFromStoredItem`'s *pvbDimensionMembersAreValid* argument returns TRUE, the cell is valid and the example prints the information to the Immediate window. Once the loop ends, `EndEnumerationOfStoredData` unlocks the subcube.

Note: The example assumes that the member IDs passed to `GetCurrencyCube` and `GetOneCellFromStoredItem` are obtained from another procedure.

```
Dim cHsvCurrencyCube As HsvCurrencyCube, lItems As Long
Dim lAcctID As Long, lIcpID As Long, lCust1ID As Long
```

```

Dim lCust2ID As Long, lCust3ID As Long, lCust4ID As Long
Dim dData As Double, bytTrans As Byte, lHandle As Long
Dim cIHsvTreeInfo As IHsvTreeInfo, sAcctLabel As String
Dim lValID As Long, bValid As Boolean
m_cHsvData.GetCurrencyCube m_lScen, m_lYear, m_lEnt, m_lVal, _
cHsvCurrencyCube
cHsvCurrencyCube.BeginEnumerationOfStoredData m_lVal, lHandle, _
lItems
Set cIHsvTreeInfo = m_cHsvMetadata.Accounts
For i = 0 To lItems - 1
    cHsvCurrencyCube.GetOneCellFromStoredItem lHandle, i, _
    m_lPer, m_lView, lValID, lAcctID, lIcpID, lCust1ID, _
    lCust2ID, lCust3ID, lCust4ID, bValid, dData, bytTrans
    If bValid = True Then
        cIHsvTreeInfo.GetLabel lAcctID, sAcctLabel
        Debug.Print sAcctLabel & ": " & CStr(dData)
    End If
Next i
cHsvCurrencyCube.EndEnumerationOfStoredData lHandle

```

GetPOVFromStoredItem

Returns the member IDs of the Value, Account, Intercompany Partner, and Custom dimension members of a subcube item. You identify the item with its index number. (For details on items, see [“About Subcube Items” on page 687.](#))

Syntax

```

<HsvCurrencyCube>.GetPOVFromStoredItem lHandle, lItem, plValue, plAccount,
plICP, plCustom1, plCustom2, plCustom3, plCustom4,
pvbDimensionMembersAreValid

```

Argument	Description
<i>lHandle</i>	The item's handle. You must pass the handle that was returned by BeginEnumerationOfStoredData .
<i>lItem</i>	Long (ByVal). The index of the item within the count of items returned by BeginEnumerationOfStoredData . Note: This is a zero-based index. For example, if BeginEnumerationOfStoredData returns a count of 5 items, the valid index numbers are 0 through 4.
<i>plValue</i>	Long. Returns the member ID of the cell's Value dimension member.
<i>plAccount</i>	Long. Returns the member ID of the item's Account dimension member.
<i>plICP</i>	Long. Returns the member ID of the item's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the item's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the item's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the item's Custom 3 dimension member.

Argument	Description
<i>plCustom4</i>	Long. Returns the member ID of the item's Custom 4 dimension member.
<i>pvbDimensionMembersAreValid</i>	Boolean. Indicates whether the cell is valid or obsolete. Returns TRUE if valid, FALSE if obsolete. An obsolete cell can exist in cases where metadata has been changed and data has been loaded in merge mode. For performance reasons, Financial Management does not delete obsolete cells from the database. Note: To perform operations with only data for valid cells, test this argument for TRUE.

HsvNodeCube Object Methods

The HsvNodeCube object provides access to node subcubes, which are described in [“About Subcubes” on page 43](#). The HsvNodeCube object's methods are summarized in [“HsvNodeCube Object Overview” on page 103](#), and are described in detail in the following topics.

HsvNodeCube object references are obtained with the HsvData object's [GetNodeCube](#) method. [GetNodeCube](#) takes member IDs of the desired subcube's Scenario, Year, and parent and child Entity dimension members, and returns a reference that provides access to the subcube.

BeginEnumerationOfStoredData

Returns a count of the subcube items for a Value dimension member, while also locking the subcube's cells to prevent users from changing data. [BeginEnumerationOfStoredData](#) takes the member ID of the Value dimension member. (For details on items, see [“About Subcube Items” on page 687](#).)

Tip: Remove the lock by calling [EndEnumerationOfStoredData](#).

Syntax

```
<HsvNodeCube>.BeginEnumerationOfStoredData lValue, plHandle, plTotalNumStoredItems
```

Argument	Description
<i>lValue</i>	Long (ByVal). The member ID of the subcube items' Value dimension member.
<i>plHandle</i>	Long (x86 platform) or Double (x64 platform). Returns a handle. You must pass this handle to other HsvNodeCube methods that have a “handle” argument.
<i>plTotalNumStoredItems</i>	Long. Returns a count of the subcube items for the <i>lValue</i> argument's Value dimension member. Use this count to determine the valid range of index numbers when using GetOneCellFromStoredItem or GetPOVFromStoredItem .

Example

`BeginEnumerationOfStoredData` is used in the [Example](#) for `GetOneCellFromStoredItem`.

EndEnumerationOfStoredData

Unlocks a subcube that was locked by `BeginEnumerationOfStoredData`.

Tip: Whenever `BeginEnumerationOfStoredData` is called, use `EndEnumerationOfStoredData` in any subsequent error handling to make sure that the subcube is unlocked.

Syntax

```
<HsvNodeCube>.EndEnumerationOfStoredData lHandle
```

Argument Description

<i>lHandle</i>	Long (ByVal) (x86 platform) or Double (ByVal) (x64 platform). The item's handle. Pass the handle that was returned by BeginEnumerationOfStoredData .
----------------	--

Example

`EndEnumerationOfStoredData` is used in the [Example](#) for `GetOneCellFromStoredItem`.

GetCell

Returns the data in and transaction status of a cell in a subcube.

Syntax

```
<HsvNodeCube>.GetCell lPeriod, lView, lValue, lAccount, lICP, lCustom1,  
lCustom2, lCustom3, lCustom4, pdData, pbyTransType
```

Argument Description

<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>lValue</i>	Long (ByVal). The member ID of the cell's Value dimension member.
<i>lAccount</i>	Long (ByVal). The member ID of the cell's Account dimension member.
<i>lICP</i>	Long (ByVal). The member ID of the cell's Intercompany Partner dimension member.
<i>lCustom1</i>	Long (ByVal). The member ID of the cell's Custom 1 dimension member.
<i>lCustom2</i>	Long (ByVal). The member ID of the cell's Custom 2 dimension member.

Argument	Description
<i>ICustom3</i>	Long (ByVal). The member ID of the cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The member ID of the cell's Custom 4 dimension member.
<i>pdData</i>	Double. Returns the cell's data.
<i>pbyTransType</i>	Byte. Returns the cell's transaction status. For a list of HFMConstants type library constants that represent the valid statuses, see “Cell Transaction Type Constants” on page 846 .

Example

The following example assigns a cell's data and transaction status to the *dData* and *bytStatus* variables. The example assumes that the member IDs passed to *GetNodeCube* and *GetCell* are obtained from another procedure.

```
Dim cHsvNodeCube As HsvNodeCube
Dim dData As Double, bytStatus As Byte
m_cHsvData.GetNodeCube m_lScen, m_lYear, m_lEnt, m_lPar, _
cHsvNodeCube
cHsvNodeCube.GetCell m_lPer, m_lView, m_lVal, m_lAcct, _
m_lICP, m_lCust1, m_lCust2, m_lCust3, m_lCust4, dData, _
bytStatus
```

GetFixedDimensionMembers

Returns the member IDs of a subcube's Scenario, Year, and parent and child Entity dimension members.

Syntax

```
<HsvNodeCube>.GetFixedDimensionMembers plScenario, plYear, plEntity,
plParent
```

Argument Description

<i>plScenario</i>	Long. Returns the member ID of the subcube's Scenario dimension member.
<i>plYear</i>	Long. Returns the member ID of the subcube's Year dimension member.
<i>plEntity</i>	Long. Returns the member ID of the subcube's Entity dimension member.
<i>plParent</i>	Long. Returns the member ID of the parent of the <i>plEntity</i> argument's entity.

GetOneCellFromStoredItem

Returns cell information such as a cell's data and member IDs. You identify the cell with the index number of the subcube item and the member IDs of the cell's Period and View dimension members. (For details on items, see [“About Subcube Items” on page 687](#).)

Syntax

```
<HsvNodeCube>.GetOneCellFromStoredItem lHandle, lItem, lPeriod, lView,  
plValue, plAccount, plICP, plCustom1, plCustom2, plCustom3, plCustom4,  
pvbDimensionMembersAreValid, pdData, pbyTransType
```

Argument	Description
<i>lHandle</i>	Long (ByVal) (x86 platform) or Double (x64 platform). The item's handle. You must pass the handle that was returned by BeginEnumerationOfStoredData .
<i>lItem</i>	Long (ByVal). The index of the item within the count of items returned by BeginEnumerationOfStoredData . Caution! This is a zero-based index. For example, if BeginEnumerationOfStoredData returns a count of 5 items, the valid index numbers are 0 through 4.
<i>lPeriod</i>	Long (ByVal). The member ID of the cell's Period dimension member.
<i>lView</i>	Long (ByVal). The member ID of the cell's View dimension member.
<i>plValue</i>	Long (ByVal). Returns the member ID of the cell's Value dimension member.
<i>plAccount</i>	Long. Returns the member ID of the cell's Account dimension member.
<i>plICP</i>	Long. Returns the member ID of the cell's Intercompany Partner dimension member.
<i>plCustom1</i>	Long. Returns the member ID of the cell's Custom 1 dimension member.
<i>plCustom2</i>	Long. Returns the member ID of the cell's Custom 2 dimension member.
<i>plCustom3</i>	Long. Returns the member ID of the cell's Custom 3 dimension member.
<i>plCustom4</i>	Long. Returns the member ID of the cell's Custom 4 dimension member.
<i>pvbDimensionMembersAreValid</i>	Boolean. Indicates whether the cell is valid or obsolete. Returns TRUE if valid, FALSE if obsolete. An obsolete cell can exist in cases where metadata has been changed and data has been loaded in merge mode. For performance reasons, Financial Management does not delete obsolete cells from the database. Tip: To perform operations with only data for valid cells, test this argument for TRUE.
<i>pdData</i>	Double. Returns the cell's data.
<i>pbyTransType</i>	Byte. Returns the cell's transaction status. For a list of HFMConstants type library constants that represent the valid statuses, see " Cell Transaction Type Constants " on page 846 .

Example

The following example prints the Account labels and data of valid cells to Visual Basic's Immediate window. [BeginEnumerationOfStoredData](#) returns the number of items, and the example then loops through these items with [GetOneCellFromStoredItem](#). If [GetOneCellFromStoredItem](#)'s *pvbDimensionMembersAreValid* argument returns TRUE,

the cell is valid and the example prints the information to the Immediate window. Once the loop ends, `EndEnumerationOfStoredData` unlocks the subcube.

Note: The example assumes an x86 platform, and that the member IDs passed to `GetNodeCube` and `GetOneCellFromStoredItem` are obtained from another procedure.

```
Dim cHsvNodeCube As HsvNodeCube, lItems As Long
Dim lAcctID As Long, lIcpID As Long, lCust1ID As Long
Dim lCust2ID As Long, lCust3ID As Long, lCust4ID As Long
Dim dData As Double, bytTrans As Byte
Dim cIHsvTreeInfo As IHsvTreeInfo, sAcctLabel As String
Dim lHandle As Long, lValID As Long, bValid As Boolean
m_cHsvData.GetNodeCube m_lScen, m_lYear, m_lEnt, m_lPar, _
cHsvNodeCube
cHsvNodeCube.BeginEnumerationOfStoredData m_lVal, lHandle, _
lItems
Set cIHsvTreeInfo = m_cHsvMetadata.Accounts
For i = 0 To lItems - 1
    cHsvNodeCube.GetOneCellFromStoredItem lHandle, i, m_lPer, _
    m_lView, lValID, lAcctID, lIcpID, lCust1ID, lCust2ID, _
    lCust3ID, lCust4ID, bValid, dData, bytTrans
    If bValid = True Then
        cIHsvTreeInfo.GetLabel lAcctID, sAcctLabel
        Debug.Print sAcctLabel & ": " & CStr(dData)
    End If
Next i
cHsvNodeCube.EndEnumerationOfStoredData lHandle
```

GetPOVFromStoredItem

Returns the member IDs of the Value, Account, Intercompany Partner, and Custom dimension members of a subcube item. You identify the item with its index number. (For details on items, see “[About Subcube Items](#)” on page 687.)

Syntax

```
<HsvNodeCube>.GetPOVFromStoredItem lHandle, lItem, plValue, plAccount,
plICP, plCustom1, plCustom2, plCustom3, plCustom4,
pvbDimensionMembersAreValid
```

Argument	Description
<i>lHandle</i>	Long (ByVal) (x86 platform) or Double (x64 platform). The item's handle. You must pass the handle that was returned by BeginEnumerationOfStoredData .
<i>lItem</i>	Long (ByVal). The index of the item within the count of items returned by <code>BeginEnumerationOfStoredData</code> . Note: This is a zero-based index. For example, if <code>BeginEnumerationOfStoredData</code> returns a count of 5 items, the valid index numbers are 0 through 4.

Argument	Description
<i>pIValue</i>	Long. Returns the member ID of the cell's Value dimension member.
<i>pIAccount</i>	Long. Returns the member ID of the item's Account dimension member.
<i>pIICP</i>	Long. Returns the member ID of the item's Intercompany Partner dimension member.
<i>pICustom1</i>	Long. Returns the member ID of the item's Custom 1 dimension member.
<i>pICustom2</i>	Long. Returns the member ID of the item's Custom 2 dimension member.
<i>pICustom3</i>	Long. Returns the member ID of the item's Custom 3 dimension member.
<i>pICustom4</i>	Long. Returns the member ID of the item's Custom 4 dimension member.
<i>pvbDimensionMembersAreValid</i>	<p>Boolean. Indicates whether the cell is valid or obsolete. Returns TRUE if valid, FALSE if obsolete.</p> <p>An obsolete cell can exist in cases where metadata has been changed and data has been loaded in merge mode. For performance reasons, Financial Management does not delete obsolete cells from the database.</p> <p>Note: To perform operations with only data for valid cells, test this argument for TRUE.</p>

In This Chapter

HsvStarSchemaACM Object Methods	699
IHsvStarSchemaTemplates Interface	706

This chapter describes the HsvStarSchemaACM type library, which exposes Financial Management's Extended Analytics features.

Note: For Extended Analytics configuration instructions and feature information, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

The type library contains the HsvStarSchemaACM object and the IHsvStarSchemaTemplates interface. The HsvStarSchemaACM object exposes Extended Analytics data extractions and is described in “[HsvStarSchemaACM Object Methods](#)” on page 699. The IHsvStarSchemaTemplates interface exposes Extended Analytics templates and is described in “[IHsvStarSchemaTemplates Interface](#)” on page 706.

The HsvStarSchemaACM object also contains the following enumerations:

- `SS_PUSH_OPTIONS`. For more information, see the description of [CreateStarSchema](#)'s `ssPushType` argument.
- `EA_EXTRACT_TYPE_FLAGS`. For more information, see the description of [CreateStarSchema](#)'s `eaExtractType` argument.
- `EA_TASK_STATUS_FLAGS`. For more information, see the description of [GetAsynchronousTaskStatus](#)'s `plCurrentTask` argument.

To use the HsvStarSchemaACM type library, you must reference `HsvStarSchemaACM.dll` in your project.

HsvStarSchemaACM Object Methods

The HsvStarSchemaACM object enables you to work with Extended Analytics data extractions. For example, you can create and delete star schemas, return the Extended Analytics Data Source Names registered on an application server, and obtain an extraction's log. This object's methods are summarized in [Table 38](#) on page 104, and are described in detail in the following topics.

HsvStarSchemaACM is a server-side object. To set an HsvStarSchemaACM object reference, create the object on the application server with `HsvSession.CreateObject`, then set the object reference with `SetSession`. The example for `EnumRegisteredDSNs` shows how to do this.

CreateStarSchema

Creates or updates a star schema by exporting data for all cells that intersect the specified dimension members. `CreateStarSchema` launches an asynchronous thread.

`CreateStarSchema` provides arguments to specify the dimension members for the extraction. You can specify members using any of the following techniques:

- Member IDs. You can specify either one member or an array of members.
- Member labels. You can specify either one member or an array of members.
- Member lists. Enclose member list names in braces ({ }).
- The ALL keyword to specify all members of the dimension.

Note: You can mix these techniques in a call to `CreateStarSchema`. For example, you can specify both member IDs and labels within an array for a given dimension.

To export data with `CreateStarSchema`, the connected user must be assigned to the Application Administrator role. To check whether the user is assigned to this role, use the `HsvSecurityAccess` method `IsApplicationAdministrator`.

Syntax

```
<HsvStarSchemaACM>.CreateStarSchema bstrDSN, bstrTablePrefix, ssPushType, eaExtractType, vbExcludeDynamicAccts, varaScenarios, varaYears, varaPeriods, varaViews, varaEntities, varaParents, varaValues, varaAccounts, varaICPs, varaCustom1, varaCustom2, varaCustom3, varaCustom4
```

Argument	Description
<i>bstrDSN</i>	String (ByVal). The Extended Analytics Data Source Name that points to the database for the star schema. Tip: To get the Extended Analytics Data Source Names that have been registered on an application server, use <code>EnumRegisteredDSNs</code> .
<i>bstrTablePrefix</i>	String (ByVal). The Relational Table Prefix, which is the prefix that identifies the star schema's tables. If a star schema identified by this prefix does not exist, <code>CreateStarSchema</code> creates a new set of tables. If the star schema exists, the tables are updated; the <i>ssPushType</i> argument determines whether all data is deleted from the <PREFIX>_FACT table.
<i>ssPushType</i>	SS_PUSH_OPTIONS (ByVal). A flag that specifies whether to create a new star schema or update an existing star schema. Valid values are represented by the following constants (from Enum SS_PUSH_OPTIONS): <ul style="list-style-type: none">• <code>ssCREATE</code> - Create a star schema• <code>ssUPDATE</code> - Update a star schema

Argument	Description
<code>eaExtractType</code>	<p>= 7, = 7 } EA_EXTRACT_TYPE_FLAGS;</p> <p>EA_EXTRACT_TYPE_FLAGS (ByVal). A flag that specifies the aggregation to use. Valid values are represented by the following constants (from Enum EA_EXTRACT_TYPE_FLAGS):</p> <ul style="list-style-type: none"> ● EA_EXTRACT_TYPE_MIN ● EA_EXTRACT_TYPE_STANDARD - Standard ● EA_EXTRACT_TYPE_METADATA_ALL - Metadata Only ● EA_EXTRACT_TYPE_METADATA_SELECTED - Selected Metadata Only ● EA_EXTRACT_TYPE_SQL_AGG - <i>For internal use.</i> ● EA_EXTRACT_TYPE_ESSBASE - Oracle Essbase ● EA_EXTRACT_TYPE_WAREHOUSE - Data Warehouse ● EA_EXTRACT_TYPE_FLATFILE ● EA_EXTRACT_TYPE_MAX
<code>vbExcludeDynamicAccts</code>	<p>Boolean (ByVal). A flag that specifies whether to exclude dynamic accounts. Pass TRUE to exclude these accounts, FALSE to include them.</p>
<code>varaScenarios</code>	<p>Variant (ByVal). The Scenario dimension members for which to extract.</p>
<code>varaYears</code>	<p>Variant (ByVal). The Year dimension members for which to extract.</p>
<code>varaPeriods</code>	<p>Variant (ByVal). The Period dimension members for which to extract.</p>
<code>varaViews</code>	<p>Variant (ByVal). The View dimension members for which to extract.</p>
<code>varaEntities</code>	<p>Variant (ByVal). The Entity dimension members for which to extract. For each specified member, you must specify a corresponding parent member in the <code>varaParents</code> argument.</p>
<code>varaParents</code>	<p>Variant (ByVal). The parents of the <code>varaEntities</code> argument's entities.</p>
<code>varaValues</code>	<p>Variant (ByVal). The Value dimension members for which to extract.</p>
<code>varaAccounts</code>	<p>Variant (ByVal). The Account dimension members for which to extract.</p>
<code>varaICPs</code>	<p>Variant (ByVal). The Intercompany Partner dimension members for which to extract.</p>
<code>varaCustom1</code>	<p>Variant (ByVal). The Custom 1 dimension members for which to extract.</p>
<code>varaCustom2</code>	<p>Variant (ByVal). The Custom 2 dimension members for which to extract.</p>
<code>varaCustom3</code>	<p>Variant (ByVal). The Custom 3 dimension members for which to extract.</p>
<code>varaCustom4</code>	<p>Variant (ByVal). The Custom 4 dimension members for which to extract.</p>

Example

The following subroutine uses the dimension members selected on an HsvPOVSelection control to export data to a Standard star schema. `HsvPOVSelection.GetCheckedItems` is used to build an array of selected members and a second array of parent Entity members. These arrays are passed to `CreateStarSchema` along with the DSN and Relational Table Prefix.

```
Sub ExtractStarSchema(sDSN As String, sPrefix As String)
```

```

Dim i As Long, vaMems(12), vaPars, vTemp
' for each of the 12 dimensions...
For i = 0 To 11
    'For the entity dimension, get the parents
    If i = DIMENSIONENTITY Then
        'cFormPOV represents an initialized HsvPOVSelection control
        cFormPOV.GetCheckedItems i, vaMems(i), vaPars
    Else ' else, ignore parents
        cFormPOV.GetCheckedItems i, vaMems(i), vTemp
    End If
Next
' g_cStarSchema is a previously set HsvStarSchemaACM instance
g_cStarSchema.CreateStarSchema sDSN, sPrefix, _
    ssCREATE, EA_EXTRACT_TYPE_STANDARD, True, _
    vaMems (DIMENSIONSCENARIO), vaMems (DIMENSIONYEAR), _
    vaMems (DIMENSIONPERIOD), vaMems (DIMENSIONVIEW), _
    vaMems (DIMENSIONENTITY), vaPars, vaMems (DIMENSIONVALUE), _
    vaMems (DIMENSIONACCOUNT), vaMems (DIMENSIONICP), _
    vaMems (DIMENSIONCUSTOM1), vaMems (DIMENSIONCUSTOM2), _
    vaMems (DIMENSIONCUSTOM3), vaMems (DIMENSIONCUSTOM4)
End Sub

```

DeleteStarSchema

Deletes a star schema from a given database.

To delete a star schema, the connected user must be assigned to the Application Administrator role. To check whether the user is assigned to this role, use the `HsvSecurityAccess` method [IsApplicationAdministrator](#).

Syntax

```
<HsvStarSchemaACM>.DeleteStarSchema bstrDSN, bstrTablePrefix
```

Argument	Description
<i>bstrDSN</i>	String (ByVal). The Extended Analytics Data Source Name for the database that contains the star schema to be deleted. Tip: To get the Extended Analytics Data Source Names that have been registered on an application server, use EnumRegisteredDSNs .
<i>bstrTablePrefix</i>	String (ByVal). The Relational Table prefix, which is the prefix that identifies the star schema's tables. All tables with names that begin with this prefix are deleted.

Example

The following example deletes a star schema named Widgets from a data source named ExtendedAnalytics.

```
'cStarSchema is a previously set HsvStarSchemaACM object
cStarSchema.DeleteStarSchema "ExtendedAnalytics", "Widgets"
```

EnumApplicationStarSchemas

For internal use.

EnumRegisteredDSNs

Returns an array of the Extended Analytics Data Source Names that have been registered on the application server.

Note: To add Data Source Names for Extended Analytics, use the Financial Management Configuration Utility. For instructions, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

Syntax

```
<HsvStarSchemaACM>.EnumRegisteredDSNs()
```

Return Value

Variant. Returns an array of Strings containing the Data Source Names. The array is 1-based.

Example

The following example populates a combo box with the Data Source Names that have been registered on an application server. The HsvStarSchemaACM instance is created on the application server by HsvSession.CreateObject. The application server is identified by the HsvSession object reference passed to SetSession.

```
Dim cStarSchema As HsvStarSchemaACM, vaDSNs
Set cStarSchema = cSession.CreateObject _
    ("Hyperion.HsvStarSchemaACM")
'cSession is a previously set HsvSession object
cStarSchema.SetSession cSession
vaDSNs = cStarSchema.EnumRegisteredDSNs
For i = LBound(vaDSNs) To UBound(vaDSNs)
    'cboDSNs is the name of the combo box
    cboDSNs.AddItem vaDSNs(i)
Next
'If there are registered DSNs, display the first one
If cboDSNs.ListCount > 0 Then cboDSNs.Text = cboDSNs.List(0)
```

GetAsynchronousTaskStatus

Returns status information for the asynchronous thread launched by [CreateStarSchema](#). GetAsynchronousTaskStatus returns the identity of the current task, the number of items to be processed, currently processed number of items, and whether the thread is still running. Once the thread has finished, the method also returns the HRESULT that represents the thread's success or failure to extract.

Tip: You can use `GetAsynchronousTaskStatus` to display the status of the thread. For example, you can periodically update a progress bar to display the number of items that have been processed.

Syntax

```
<HsvStarSchemaACM>.GetAsynchronousTaskStatus plCurrentTask, pdNumRecords,  
pdNumCompletedRecords, pvbIsRunning, plErrorCode
```

Argument	Description
<i>plCurrentTask</i>	<p>EA_TASK_STATUS_FLAGS. Returns the identity of the task currently being processed by the <code>CreateStarSchema</code> thread. Valid values represented by the following constants (from Enum EA_TASK_STATUS_FLAGS):</p> <ul style="list-style-type: none">EA_TASK_STATUS_BLOCKED - The task is waiting to be queued.EA_TASK_STATUS_CANCELLED - The task has been cancelled.EA_TASK_STATUS_COMPLETE - The task is complete.EA_TASK_STATUS_COMPLETE_W_ERRORS - The task is complete, but an error has occurred. The error number is returned by the <i>plErrorCode</i> argument.EA_TASK_STATUS_DATA - The task is extracting data.EA_TASK_STATUS_ESSBASE_AGG - Aggregation is occurring externally in Essbase.EA_TASK_STATUS_INITIALIZING - The task is initializing.EA_TASK_STATUS_METADATA - The task is extracting metadata.EA_TASK_STATUS_QUEUED - The task is queued.EA_TASK_STATUS_SQL_AGG - <i>For internal use.</i> <p>Note: Enum EA_TASK_STATUS_FLAGS also contains the EA_TASK_STATUS_MIN and EA_TASK_STATUS_MAX constants, which represent the lower and upper bounds of the enumeration.</p>
<i>pdNumRecords</i>	<p>Double. Returns the total number of items to be processed for the task. This does not apply to the Initializing task.</p> <p>Note: The number of items processed differs from the number of records added to the database. For data extracts, the number of items represents the potential cell intersections. For metadata extracts, the number of items represents the combination of specified members.</p>
<i>pdNumCompletedRecords</i>	Double. Returns the total number of items that have been processed for the task.
<i>pvbIsRunning</i>	Boolean. Indicates whether the <code>CreateStarSchema</code> thread is still running. Returns TRUE if the thread is running, FALSE if it has finished.
<i>plErrorCode</i>	Long. Returns the HRESULT for the thread's final processing status. This value is valid only when the thread has finished running.

GetExtractLogData

Returns a string that provides a log of the `HsvStarSchemaACM` instance's most recent call to [CreateStarSchema](#).

Note: Financial Management periodically deletes information on tasks performed. If the log information or an extraction has been deleted, `GetExtractLogData` will not return log information. The `pvbHadData` argument indicates whether `GetExtractLogData` could find log data.

Syntax

```
<HsvStarSchemaACM>.GetExtractLogData pbstrLogData, pvbHadData
```

Argument	Description
<code>pbstrLogData</code>	String. Returns a log of the most recent call to <code>CreateStarSchema</code> .
<code>pvbHadData</code>	Boolean. Returns a flag that indicates whether <code>GetExtractLogData</code> could find the log information. Returns TRUE if the log information is found, FALSE otherwise.

GetPersistedSettings

For internal use.

QuitAsynchronousTask

Terminates the thread launched by `CreateStarSchema`.

Syntax

```
<HsvStarSchemaACM>.QuitAsynchronousTask
```

SetDefaultTablePrefix

For internal use.

SetPersistedSettings

For internal use.

SetSession

Points to the `HsvSession` object that represents the connection to the application. You must call `SetSession` before using the other `HsvStarSchemaACM` methods.

Syntax

```
<HsvStarSchemaACM>.SetSession pIHsvSessionUnk
```

Argument	Description
<i>pHsvSessionUnk</i>	HsvSession object (ByVal). The HsvSession object returned by HsxClient.OpenApplication or HsxClientUI.OpenApplication when the application was opened.

Example

SetSession is used in the example for [EnumRegisteredDSNs](#).

TestSQLConnection

For internal use.

IHsvStarSchemaTemplates Interface

The IHsvStarSchemaTemplates interface enables you to work with Extended Analytics templates. For example, you can use this interface to create and delete templates.

To set an IHsvStarSchemaTemplates object reference, set the object variable to a previously set HsvStarSchemaACM object reference as shown in the following example:

```
Dim cStarSchemaTemplates As IHsvStarSchemaTemplates
'cStarSchema is an HsvStarSchemaACM object reference
Set cStarSchemaTemplates = cStarSchema
```

The interface's methods are summarized in [“IHsvStarSchemaTemplates Interface Overview” on page 105](#), and are described in detail in the following topics.

DeleteTemplate

Deletes a given Extended Analytics template.

Syntax

```
<IHsvStarSchemaTemplates>.DeleteTemplate bstrTplName
```

Argument	Description
<i>bstrTplName</i>	String (ByVal). The name of the template to delete.

EnumTemplates

Returns the names of the application's Extended Analytics templates.

Syntax

```
<IHsvStarSchemaTemplates>.EnumTemplates()
```

Return Value

Variant. Returns an array of strings that contain the templates' names.

GetTemplate

Returns an XML string that contains a given Extended Analytics template's definition.

Syntax

```
<IHsvStarSchemaTemplates>.GetTemplate(bstrTplName)
```

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

<i>bstrTplName</i>	String (ByVal). The name of the template.
--------------------	---

Return Value

String. The XML string that contains the template's definition. The string's schema is described in ["XML String Schema" on page 707](#).

SetTemplate

Creates an Extended Analytics template, using an XML string that contains the template definition.

Syntax

```
<IHsvStarSchemaTemplates>.SetTemplate bstrTplName, bstrTplContents,  
vbOverwrite
```

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

<i>bstrTplName</i>	String (ByVal). The name of the new template.
--------------------	---

<i>bstrTplContents</i>	String (ByVal). The XML string containing the template definition. The string's schema is described below.
------------------------	--

<i>vbOverwrite</i>	Boolean (ByVal). A flag that specifies whether to overwrite an existing template of the same name. Pass TRUE to overwrite, FALSE otherwise.
--------------------	---

XML String Schema

Following is the schema for the XML string passed to the *bstrTplContents* argument and returned by [GetTemplate](#).

```
<povTemplate>  
  <povEA></povEA>  
  <options>  
    <tablePrefix></tablePrefix>  
    <exportOption></exportOption>
```

```

    <selectedDSN></selectedDSN>
    <excludeDynAccts></excludeDynAccts>
  </options>
</povTemplate>

```

The following table describes the XML string's schema:

Element	Description
<i>povTemplate</i>	The root element.
<i>povEA</i>	<p>Specifies the dimension members for the template. Members for each dimension are preceded by the following characters:</p> <ul style="list-style-type: none"> ● A# = Account ● I# = Intercompany Partner ● C1# = Custom1 ● C2# = Custom2 ● C3# = Custom3 ● C4# = Custom4 ● S# = Scenario ● Y# = Year ● P# = Period ● W# = View ● E# = Entity <p>Note: The Entity dimension must include both the parent and child members, delimited by a period.</p> <ul style="list-style-type: none"> ● V# = Value <p>Dimensions are delimited with periods. Multiple members for a dimension are delimited by semicolons. Member list labels are enclosed with braces ({ }).</p>
<i>options</i>	Contains the elements that define the template's extract options.
<i>tablePrefix</i>	The Relational Table Prefix, which is the prefix that identifies the star schema's tables.
<i>exportOption</i>	<p>The aggregation to use. Valid values are listed below:</p> <ul style="list-style-type: none"> ● 0 = Standard ● 1 = Metadata Only ● 2 = Selected Metadata Only ● 3 = <i>For internal use</i> ● 4 = Essbase ● 5 = Data Warehouse
<i>selectedDSN</i>	The Extended Analytics Data Source Name that points to the database for the star schema.
<i>excludeDynAccts</i>	A flag that specifies whether to exclude dynamic accounts. Zero (0) indicates to include dynamic accounts, a non-zero value indicates to exclude them.

Example

The following example creates a template for a Standard aggregation type with dynamic accounts included. The example specifies multiple members for the Entity dimension and a member list for the Period dimension.

```
Dim StarSchemaTemplates As IHsvStarSchemaTemplates
'cStarSchema is an HsvStarSchemaACM object reference
Set StarSchemaTemplates = cStarSchema
StarSchemaTemplates.SetTemplate "mytemplate", "<povTemplate>" _
& "<povEA>S#Actual.Y#2004.P{Months}.w#&lt;Scenario View&gt;" _
& ".E#UnitedStates.NewYork; UnitedStates.Virginia." _
& "V#&lt;Entity Currency&gt;.A#Sales.I#[ICP None].C1#[None]" _
& ".C2#[None].C3#[None].C4#[None]</povEA><options>" _
& " <tablePrefix>myPrefix</tablePrefix><exportOption>0" _
& "</exportOption><selectedDSN>myDSN</selectedDSN><excludeDynAccts>0" _
& "</excludeDynAccts></options></povTemplate>", True
```


In This Chapter

HsvICM Object Methods	711
IHsvAdminICM Interface Methods	732

This chapter describes the HsvICM type library, which exposes Financial Management's intercompany transaction features.

The type library contains the HsvICM object and the IHsvAdminICM interface. The HsvICM object exposes transaction processing and is described in “[HsvICM Object Methods](#)” on page 711. The IHsvAdminICM interface exposes the administration of intercompany transactions and is described in “[IHsvAdminICM Interface Methods](#)” on page 732.

HsvICM Object Methods

The HsvICM object enables you to work with intercompany transactions. For example, you can use this object to create, match, and unmatched transactions.

To set an HsvICM object reference, use the HsvSession object's `ICM` property as shown in the following example:

```
Dim cICM As HsvICM
'g_cSession is an HsvSession object reference
Set cICM = g_cSession.ICM
```

The interface's methods are summarized in “[HsvICM Object Overview](#)” on page 105, and are described in detail in the following topics.

Note: The HsvMDArrays type library contains the HsvICTransactionsData object, which provides helper methods for working with intercompany transactions. For more information, see “[HsvICTransactionsData Object Methods](#)” on page 674.

AIMVTransPosted

Indicates whether all transactions for a given scenario, year, period, and entity that are in a matched state have been posted.

A transaction is in a matched state if it is matched or is mismatched and contains a valid reason code. You can check a transaction's matching status and reason code with `HsvICTransactionsData.GetICTransactionData`.

Syntax

```
<HsvICM>.AllMVTransPosted(lScenario, lYear, lPeriod, lEntity)
```

Argument Description

lScenario Long (ByVal). The member ID of the Scenario dimension member.

lYear Long (ByVal). The member ID of the Year dimension member.

lPeriod Long (ByVal). The member ID of the Period dimension member.

lEntity Long (ByVal). The member ID of the Entity dimension member.

Return Value

Boolean. Returns TRUE if the matched state transactions have been posted, FALSE otherwise.

CheckReportSecurity

For internal use.

CreateICTransaction

Creates an intercompany transaction for a given cell.

Tip: To test whether a cell is valid for intercompany transactions, use [DoesCellSupportICTransactionDetail](#). To update an existing transaction, use [SaveICTransaction](#).

Syntax

```
<HsvICM>.CreateICTransaction vbOverwrite, lScenario, lYear, lPeriod,  
lEntity, lICP, lAccount, lC1, lC2, lC3, lC4, lTRCur, lReason, dTRAMt,  
dTRLAMt, dTRRate, dTRDate, bstrID, bstrSubID, bstrRefID, bstrComment1,  
bstrComment2
```

Argument Description

vbOverwrite Boolean (ByVal). A flag that specifies whether to overwrite existing transactions that have the same Transaction ID and Sub ID. Pass TRUE to overwrite, FALSE otherwise.

lScenario Long (ByVal). The member ID of the transaction's Scenario dimension member.

lYear Long (ByVal). The member ID of the transaction's Year dimension member.

Argument	Description
<i>IPeriod</i>	Long (ByVal). The member ID of the transaction's Period dimension member.
<i>IEntity</i>	Long (ByVal). The member ID of the transaction's Entity dimension member.
<i>IICP</i>	Long (ByVal). The member ID of the transaction's Intercompany Partner dimension member.
<i>IAccount</i>	Long (ByVal). The member ID of the transaction's Account dimension member.
<i>IC1</i>	Long (ByVal). The member ID of the transaction's Custom 1 dimension member.
<i>IC2</i>	Long (ByVal). The member ID of the transaction's Custom 2 dimension member.
<i>IC3</i>	Long (ByVal). The member ID of the transaction's Custom 3 dimension member.
<i>IC4</i>	Long (ByVal). The member ID of the transaction's Custom 4 dimension member.
<i>ITRCur</i>	Long (ByVal). The ID of the transaction's currency. Tip: You can obtain the ID of an Entity dimension member's currency with GetEntityCurrencyID . You also can obtain a currency ID with GetTransCurrencyID .
<i>IReason</i>	Long (ByVal). The ID of the transaction's reason code. Tip: You can get a reason code ID with GetICReasonCodeID .
<i>dTRAmt</i>	Double (ByVal). The transaction amount.
<i>dTRLAmt</i>	Double (ByVal). The entity currency amount.
<i>dTRRate</i>	Double (ByVal). The conversion rate.
<i>dTRDate</i>	Double (ByVal). The transaction date formatted as a Double.
<i>bstrID</i>	String (ByVal). The Transaction ID.
<i>bstrSubID</i>	String (ByVal). The transaction's Sub ID.
<i>bstrRefID</i>	String (ByVal). The transaction's Reference ID.
<i>bstrComment1</i>	String (ByVal). The first comment for the transaction. Note: You can specify a maximum of 256 characters.
<i>bstrComment2</i>	String (ByVal). The second comment for the transaction. Note: You can specify a maximum of 256 characters.

Example

The following function creates an intercompany transaction with the current date as the transaction date and the specified entity's default currency as the transaction currency, and returns whether the specified cell is valid for intercompany transactions. The function uses `HsvValues.GetValueIDFromCurrencyID` to get the member ID of the Value dimension member that corresponds to the entity's default currency, and passes this ID to `DoesCellSupportICTransactionDetail`.

```
Function createIcTransEntCurrToday(1Scen As Long, 1Year As Long, 1Per _
```

```

    As Long, lEnt As Long, lIcp As Long, lAcct As Long, lCust1 As Long, _
    lCust2 As Long, lCust3 As Long, lCust4 As Long, sReason As String, _
    dTranAmt As Double, dEntCurrAmt As Double, sTranId As String, _
    sSubId As String) As Boolean
Dim cICM As HsvICM, lCurr As Long, lReason As Long, bSupport As Boolean
Dim lVal As Long, cValues As HsvValues
'g_cSession is an HsvSession object reference
Set cICM = g_cSession.ICM
lCurr = cICM.GetEntityCurrencyID(lEnt)
'g_cMetadata is an HsvMetadata object reference
Set cValues = g_cMetadata.Values
cValues.GetValueIDFromCurrencyID lCurr, lVal
bSupport = cICM.DoesCellSupportICTransactionDetail(lScen, lYear, lPer, _
    lEnt, lVal, lAcct, lIcp, lCust1, lCust2, lCust3, lCust4)
If bSupport = True Then
    lReason = cICM.GetICReasonCodeID(sReason)
    cICM.CreateICTransaction True, lScen, lYear, lPer, lEnt, lIcp, _
        lAcct, lCust1, lCust2, lCust3, lCust4, lCurr, lReason, dTranAmt, _
        dEntCurrAmt, 0, Now, sTranId, sSubId, "", "", ""
    createIcTransEntCurrToday = True
Else
    createIcTransEntCurrToday = False
End If
End Function

```

DeleteICReasonCode

Deletes a reason code.

Syntax

```
<HsvICM>.DeleteICReasonCode bstrReasonCode
```

Argument Description

bstrReasonCode String (ByVal). The reason code to be deleted.

DoesCellSupportICTransactionDetail

Indicates whether a cell supports intercompany transactions.

Syntax

```
<HsvICM>.DoesCellSupportICTransactionDetail(lScenario, lYear, lPeriod,
lEntity, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4)
```

Argument Description

lScenario Long (ByVal). The cell's Scenario dimension member.

lYear Long (ByVal). The cell's Year dimension member.

lPeriod Long (ByVal). The cell's Period dimension member.

Argument Description

<i>IEntity</i>	Long (ByVal). The cell's Entity dimension member.
<i>IValue</i>	Long (ByVal). The cell's Value dimension member.
<i>IAccount</i>	Long (ByVal). The cell's Account dimension member.
<i>IICP</i>	Long (ByVal). The cell's Intercompany Partner dimension member.
<i>ICustom1</i>	Long (ByVal). The cell's Custom 1 dimension member.
<i>ICustom2</i>	Long (ByVal). The cell's Custom 2 dimension member.
<i>ICustom3</i>	Long (ByVal). The cell's Custom 3 dimension member.
<i>ICustom4</i>	Long (ByVal). The cell's Custom 4 dimension member.

Return Value

Boolean. Returns TRUE if the cell supports intercompany transactions, FALSE otherwise.

Example

`DoesCellSupportICTransactionDetail` is used in the example for [CreateICTransaction](#).

GetColumnFilter

For internal use.

GetCurrencyInfo

Returns a currency's label, translation operator, and scale.

Syntax

```
<HsvICM>.GetCurrencyInfo lCurID, pbstrCurrencyLabel, pbstrCurrencyOp, psCurrencyScale
```

Argument	Description
<i>lCurID</i>	Long (ByVal). The currency ID of the currency.
<i>pbstrCurrencyLabel</i>	String. Returns the currency's label.
<i>pbstrCurrencyOp</i>	String. Returns the currency's translation operator.
<i>psCurrencyScale</i>	Integer. Returns the currency's scale.

GetEntitiesContacts

Returns the usernames and security identifiers of the users to be alerted for a given scenario and intercompany transaction-related event for the specified entities.

Syntax

```
<HsvICM>.GetEntitiesContacts lEventType, lScenario, varalEntities, pvara2DUsersInfo
```

Argument	Description
<i>lEventType</i>	Long (ByVal). The ID of the intercompany transaction-related event. Valid values are represented by the HFMConstants enumeration tagICEVENTYPE, which is described in “Event Constants” on page 877 .
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>varalEntities</i>	Long array (ByVal). The member IDs of the Entity dimension members.
<i>pvara2DUsersInfo</i>	Variant. Returns a two-dimensional array containing the usernames and security identifiers of the users to be alerted. The first dimension corresponds to the Entity dimension members passed in the <i>varalEntities</i> argument. The second dimension contains the user information and is indexed from 0 to 1: <ul style="list-style-type: none">0 = The security identifiers of the users. If an entity has multiple users to be alerted, the security identifiers are delimited by semicolons.1 = The usernames of the users. If an entity has multiple users to be alerted, the usernames are delimited by semicolons.

GetEntityCurrencyID

Returns the currency ID of an entity’s default currency.

Syntax

```
<HsvICM>.GetEntityCurrencyID(lEntityID)
```

Argument Description

lEntityID Long (ByVal). The member ID of the Entity dimension member.

Return Value

Long. Returns the currency’s ID.

Example

GetEntityCurrencyID is used in the example for [CreateICTransaction](#).

GetICReasonCodeID

Gets the ID of a reason code.

Syntax

```
<HsvICM>.GetICReasonCodeID(bstrReasonCodeLabel)
```

Argument	Description
<i>bstrReasonCodeLabel</i>	String (ByVal). The reason code.

Return Value

Long. Returns the reason code's ID.

Example

GetICReasonCodeID is used in the example for [CreateICTransaction](#).

GetICReasonCodeLabel

Returns a reason code from the code's internal ID.

Syntax

```
<HsvICM>.GetICReasonCodeLabel (lReasonCodeID)
```

Argument	Description
<i>lReasonCodeID</i>	Long (ByVal). The reason code's ID.

Return Value

String. Returns the reason code.

GetICReasonCodes

Returns an application's reason codes and their corresponding IDs and descriptions.

Syntax

```
<HsvICM>.GetICReasonCodes ()
```

Return Value

Variant. Returns a multidimensional array of the reason codes, IDs, and descriptions. The first dimension contains three items that represent the reason code information, and the second dimension contains one item for each reason code in the application. The following table describes the first dimension:

Index	Description
0	Long. Returns the IDs of the reason codes.
1	String. Returns the reason codes.
2	String. Returns the descriptions of the reason codes.

Example

The following example prints an application's reason codes and descriptions to Visual Basic's Immediate window.

```
Dim cICM As HsvICM, vaRet As Variant
'g_cSession is an HsvSession object reference
Set cICM = g_cSession.ICM
vaRet = cICM.GetICReasonCodes
For i = LBound(vaRet, 2) To UBound(vaRet, 2)
    Debug.Print vaRet(1, i) & ": " & vaRet(2, i)
Next i
```

GetICTransactions

Populates an `HsvICTransactionsData` object reference with intercompany transactions for the scenario, year, and period specified with `HsvICTransactionsData.Initialize`. You can filter transactions containing dimension members for which the connected user does not have security rights.

Syntax

```
<HsvICM>.GetICTransactions piUnkICTransactionsData, vbSkipDimSecurity
```

Argument	Description
<i>piUnkICTransactionsData</i>	Object (ByVal). An initialized <code>HsvICTransactionsData</code> object reference. <code>GetICTransactions</code> populates the object with the intercompany transactions for the specified scenario, year, and period.
<i>vbSkipDimSecurity</i>	Boolean (ByVal). A flag that specifies whether to filter transactions containing dimension members for which the connected user does not have security rights. Pass <code>TRUE</code> to return transactions regardless of the user's security rights, <code>FALSE</code> to return only those transactions containing members for which the user has security rights.

Example

`GetICTransactions` is used in the example for `HsvICTransactionsData.Initialize`.

GetMonitorICDetails

Returns counts of intercompany transactions that have various posting and matching statuses for the specified Scenario, Year, Period, and Entity dimension members.

Syntax

```
<HsvICM>.GetMonitorICDetails lScenario, lYear, lPeriod, lEntity,  
plUnpostedUnMatched, plUnpostedMisMatched, plUnpostedMatched,  
plUnpostedReasonMisMatched, plPostedUnMatched, plPostedMisMatched,  
plPostedMatched, plPostedReasonMisMatched
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member.
<i>plUnpostedUnMatched</i>	Long. Returns the number of intercompany transactions that are unposted and unmatched.
<i>plUnpostedMisMatched</i>	Long. Returns the number of intercompany transactions that are unposted and mismatched without a reason code.
<i>plUnpostedMatched</i>	Long. Returns the number of intercompany transactions that are unposted and matched.
<i>plUnpostedReasonMisMatched</i>	Long. Returns the number of intercompany transactions that are unposted and mismatched with a reason code.
<i>plPostedUnMatched</i>	Long. Returns the number of intercompany transactions that are posted and unmatched.
<i>plPostedMisMatched</i>	Long. Returns the number of intercompany transactions that are posted and mismatched without a reason code.
<i>plPostedMatched</i>	Long. Returns the number of intercompany transactions that are posted and matched.
<i>plPostedReasonMisMatched</i>	Long. Returns the number of intercompany transactions that are posted and mismatched with a reason code.

GetMonitorICSummary

Returns the number of Entity dimension members that have various combinations of locking and process statuses for the given Scenario, Year, Period, and Entity dimension members.

Syntax

```
<HsvICM>.GetMonitorICSummary lScenario, lYear, lPeriod, varalEntitiesIDs,  
plUnlockedNotStarted, plLockedNotStarted, plUnlockedStarted,  
plLockedStarted
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.

Argument	Description
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>varalEntitiesIDs</i>	Long array (ByVal). The member IDs of the Entity dimension members.
<i>plUnlockedNotStarted</i>	Long. Returns the number of Entity dimension members that have an Unlocked locking status and a Not Started process status.
<i>plLockedNotStarted</i>	Long. Returns the number of Entity dimension members that have a Locked locking status and a Not Started process status.
<i>plUnlockedStarted</i>	Long. Returns the number of Entity dimension members that have an Unlocked locking status and a Started process status.
<i>plLockedStarted</i>	Long. Returns the number of Entity dimension members that have a Locked locking status and a Started process status.

GetMonitorICTransactions

Indicates whether one or more Entity dimension members have intercompany transactions for given Scenario, Year, and Period dimension members. For the entities that have intercompany transactions, `GetMonitorICTransactions` returns an array that consists of the entities' member IDs, process and locking statuses, and usernames and timestamps for the most recently modified transactions.

Syntax

```
<HsvICM>.GetMonitorICTransactions lScenario, lYear, lPeriod,
varalEntities, lSortFilterOptions, lFromRec, lMaxRows, plTotalEntities,
pvara2DMonitor
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>varalEntities</i>	Long array (ByVal). The member IDs of the Entity dimension members.
<i>lSortFilterOptions</i>	<p>Long (ByVal). Specifies filtering and sorting options for the information returned by the <i>pvara2DMonitor</i> argument. Valid values are represented by the HFMConstants enumeration <code>tagICM_MONITOR_FILTER_SORT_FLAGS</code>, which is described in “Filtering and Sorting Options” on page 876.</p> <p>You can use the addition operator (+) with the enumeration's constants to specify combinations of filtering and sorting options. The following example filters for both Started and Not Started process statuses:</p> <pre>ICM_MONITOR_FILTER_STARTED + ICM_MONITOR_FILTER_NOTSTARTED</pre>
<i>lFromRec</i>	Long (ByVal). The index of the first entity within the set of entities that match the criteria to include in the array returned by the <i>pvara2DMonitor</i> argument.

Argument	Description
<i>lMaxRows</i>	Long (ByVal). The maximum number of transactions to include in the array returned by the <i>pvara2DMonitor</i> argument.
<i>pITotalEntities</i>	Long. Returns the number of Entity dimension members that match the criteria.
<i>pvara2DMonitor</i>	Variant. Returns a two-dimensional array containing information for the entities that have intercompany transactions. The first dimension consists of five elements, and the second dimension contains one element for each entity that has intercompany transactions. The following list describes the elements in the first dimension. <ul style="list-style-type: none"> ● Element 0 returns the member IDs of the entities. ● Element 1 returns the process statuses for the entities. Valid values are represented by the HFMCConstants enumeration tagICMENTITYPROCESSSTATUS, which is described in “Process Status Constants” on page 877. ● Element 2 returns the process statuses for the entities. Valid values are represented by the HFMCConstants enumeration tagICMENTITYSTATUS, which is described in “Lock Status Constants” on page 874. ● Element 3 returns the usernames of the users who most recently modified transactions for the entities. ● Element 4 returns the timestamps of the most recently modified transactions for the entities.

GetRowFilter

For internal use.

GetTransCurrencyID

Returns the currency ID of a given currency.

Syntax

```
<HsvICM>.GetTransCurrencyID(bstrTRCurr)
```

Argument Description

bstrTRCurr String (ByVal). The currency's label.

Return Value

Long. Returns the currency ID.

GetUnMatchTransactions

Populates an HsvICTransactionsData object reference with unmatched intercompany transactions for the scenario, year, and period specified with HsvICTransactionsData.[Initialize](#).

Syntax

```
<HsvICM>.GetUnMatchTransactions pIUnkICTransactionsData
```

Argument

Description

pIUnkICTransactionsData Object (ByVal). An HsvICTransactionsData object reference containing unmatched intercompany transactions for the specified scenario, year, and period.

IsOneSideOfTransactionGroupWriteable

Indicates whether the user has write access to either the entity or Intercompany Partner cell of all transactions in a given range of an HsvICTransactionsData object reference's transactions.

For example, suppose that you are unmatching the transactions listed in the following table:

Entity	Intercompany Partner (Entity)	Account	Amount
A	B	Receivable	100
A	B	Sales	100
B	A	Payable	200

If the user has write access to entity A and to all three accounts, but not to entity B, the user has access to A's cells in all the transactions, and `IsOneSideOfTransactionGroupWriteable` returns TRUE. However, if the user has write access to entity A and the Sales and Payable accounts, but not to entity B and the Receivable account, the user lacks write access to both cells in the first transaction listed, and `IsOneSideOfTransactionGroupWriteable` returns FALSE.

Syntax

```
<HsvICM>.IsOneSideOfTransactionGroupWriteable(pIUnkICTransactionsData, lScenario, lYear, lPeriod, lFromIndex, lToIndex)
```

Argument

Description

pIUnkICTransactionsData Object (ByVal). An HsvICTransactionsData object reference.

Tip: To populate an HsvICTransactionsData object reference with groups of unmatched transactions, use [GetUnMatchTransactions](#).

lScenario Long (ByVal). The member ID of the Scenario dimension member for the transactions.

lYear Long (ByVal). The member ID of the Year dimension member for the transactions.

lPeriod Long (ByVal). The member ID of the Period dimension member for the transactions.

lFromIndex Long (ByVal). The index of the first transaction in the range of the HsvICTransactionsData object reference's transactions to be tested.

Argument	Description
<i>IToIndex</i>	Long (ByVal). The index of the last transaction in the range of the HsvICTransactionsData object reference's transactions to be tested.

Return Value

Boolean. Returns TRUE if the user has write access to at least one cell in all of the specified transactions.

MatchAutoAccounts

Matches intercompany transactions by account. You can optionally restrict matching to only those transactions that include specified Custom dimension members.

Syntax

```
<HsvICM>.MatchAutoAccounts lScenario, lYear, lPeriod, varalEntities,
varalICPs, varabstrAcctsPOV, varabstrMACctsPOV, lMatchOption,
dMatchTolerance, bstrMatchCurr
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member for which to match intercompany transactions.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member for which to match intercompany transactions.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member for which to match intercompany transactions.
<i>varalEntities</i>	Long array (ByVal). The member IDs of the Entity dimension members for which to match intercompany transactions.
<i>varalICPs</i>	Long array (ByVal). The member IDs of the Intercompany Partner dimension members for which to match intercompany transactions.
<i>varabstrAcctsPOV</i>	String array (ByVal). The strings that specify the Account and Custom dimension members for which to match intercompany transactions. Use the following rules and syntax when defining the array's strings: <ul style="list-style-type: none"> ● You must specify an Account dimension member. ● You optionally can specify members for any of the Custom dimensions. ● Dimensions are delimited with periods. ● Members for each dimension are preceded by the following characters: <ul style="list-style-type: none"> ○ A# = Account ○ C1# = Custom1 ○ C2# = Custom2 ○ C3# = Custom3 ○ C4# = Custom4

The following example specifies an Account dimension member and a Custom 1 member:

Argument	Description
	"A#RecltIC.C1#Decreases"
<i>varabstrMAcctsPOV</i>	String array (ByVal). The strings that specify the matching Account and Custom dimension members. Use the same rules and syntax as is described for the <i>varabstrAcctsPOV</i> argument.
<i>lMatchOption</i>	Long (ByVal). This argument is reserved for future use. However, you must pass a valid Long, such as zero (0).
<i>dMatchTolerance</i>	Double (ByVal). The matching tolerance.
<i>bstrMatchCurr</i>	String (ByVal). The transaction currency by which to match. To match for all transaction currencies, pass a blank string.

MatchAutoIDs

Matches intercompany transactions by Transaction ID or Reference ID.

Syntax

```
<HsvICM>.MatchAutoIDs lScenario, lYear, lPeriod, varalEntities, varalICPs, lMatchOption, bstrMatchCurr, bstrMatchIDs
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member for which to match intercompany transactions.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member for which to match intercompany transactions.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member for which to match intercompany transactions.
<i>varalEntities</i>	Long array (ByVal). The member IDs of the Entity dimension members for which to match intercompany transactions.
<i>varalICPs</i>	Long array (ByVal). The member IDs of the Intercompany Partner dimension members for which to match intercompany transactions.
<i>lMatchOption</i>	Long (ByVal). A flag that specifies the type of ID by which to match. Valid values are represented by the following constants, which are members of the HFMConstants enumeration tagICMMATCHOPTIONSENUM: <ul style="list-style-type: none"> ● ICT_MATCH_REFERENCE ● ICT_MATCH_TRANSACTIONID For descriptions of these constants, see “Match Option Constants” on page 874 .
<i>bstrMatchCurr</i>	String (ByVal). The transaction currency for which to match. To match for all transaction currencies, pass a blank string.
<i>bstrMatchIDs</i>	String (ByVal). A comma-delimited list of the Transaction IDs or the Reference IDs that identify the transactions to match. <p>You can use the percentage symbol (%) as a wildcard character. For example, the following string matches all IDs that begin with either “ref” or “acct”:</p>

Argument	Description
	"ref%,acct%"

Example

The following example matches all intercompany transactions for the specified dimension members that have reference IDs beginning with either “ref” or “acct”.

```
Sub matchRefIdsPattern(lScen As Long, lYear As Long, lPer As Long, _
    laEnts() As Long, laICPs() As Long)
Dim cICM As HsvICM
'g_cSession is an HsvSession object reference
Set cICM = g_cSession.ICM
cICM.MatchAutoIDs lScen, lYear, lPer, laEnts, laICPs, _
    ICT_MATCH_REFERENCE, "", "ref%,acct%"
End Sub
```

NoTransMatchedOrPosted

Indicates whether matched or posted transactions exist for a given scenario, year, and period.

Syntax

```
<HsvICM>.NoTransMatchedOrPosted(lScenario, lYear, lPeriod)
```

Argument Description

lScenario Long (ByVal). The member ID of the Scenario dimension member.

lYear Long (ByVal). The member ID of the Year dimension member.

lPeriod Long (ByVal). The member ID of the Period dimension member.

Return Value

Boolean. Returns TRUE if matched or posted transactions exist for the scenario, year, and period, FALSE otherwise.

ProcessAllICTrans

Deletes, posts, unposts, or unmatches all transactions for a given scenario, year, and period.

Tip: To process only specified transactions for a given scenario, year, and period, use [ProcessICTransactions](#).

Syntax

```
<HsvICM>.ProcessAllICTrans(lICAction, piUnkICTransactionsData)
```

Argument	Description
<i>ICAction</i>	<p>Long (ByVal). A flag that specifies the processing to perform. Pass one of the following constants, which are members of the HFMConstants type library enumeration tagICMTRANSPROCESSACTION:</p> <ul style="list-style-type: none"> ● ICM_DELETETRANS_ALL ● ICM_POSTTRANS_ALL ● ICM_UNMATCHTRANS_ALL ● ICM_UNPOSTTRANS_ALL <p>For descriptions of these constants, see “Processing Action Constants” on page 875.</p>
<i>pUnkICTransactionsData</i>	<p>HsvICTransactionsData object (ByVal). An initialized HsvICTransactionsData object reference.</p> <p>Note: The HsvICTransactionsData method Initialize specifies the scenario, year, and period of the transactions to process.</p>

Return Value

String. Returns an XML string that indicates whether an error occurred. The string is of the following format, with the status attribute containing zero if no error occurred, or a non-zero error number otherwise:

```
<processtransactionsresponse status="HRESULT">
</processtransactionsresponse>
```

Example

The following function deletes all transactions for the specified scenario, year, and period.

```
Function deleteTrans(lScen As Long, lYear As Long, lPer As Long) _
    As String
Dim cIcTransData As HsvICTransactionsData, cICM As HsvICM, sRet As String
Set cIcTransData = New HsvICTransactionsData
'g_cSession is an HsvSession object reference
Set cICM = g_cSession.ICM
cIcTransData.Initialize lScen, lYear, lPer
sRet = cICM.ProcessAllICTrans(ICM_DELETETRANS_ALL, cIcTransData)
deleteTrans = sRet
End Function
```

ProcessICTransactions

Deletes, posts, unposts, or unmatched the specified intercompany transactions for a given scenario, year, and period. Transactions are identified by arrays of the transactions' sequence IDs and the Entity, Intercompany Partner, Account, and Custom dimension member IDs. These arrays must have a one-to-one correspondence.

Tip: To process all transactions for a given scenario, year, and period, use [ProcessAllICTrans](#).

Syntax

```
<HsvICM>.ProcessICTransactions(lICAction, lScenario, lYear, lPeriod,  
varalSeqIDs, varalEntities, varalICPs, varalAccounts, varalC1s, varalC2s,  
varalC3s, varalC4s)
```

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

<i>lICAction</i>	Long (ByVal). A flag that specifies the processing to perform. Pass one of the following constants, which are members of the HFMConstants type library enumeration tagICMTRANSPROCESSACTION:
------------------	--

- ICM_POSTTRANS
- ICM_UNPOSTTRANS
- ICM_DELETETRANS
- ICM_MANUALMATCHTRANS
- ICM_UNMATCHTRANS

For descriptions of these constants, see [“Processing Action Constants” on page 875](#).

<i>lScenario</i>	Long (ByVal). The member ID of the transactions’ Scenario dimension members.
------------------	--

<i>lYear</i>	Long (ByVal). The member ID of the transactions’ Year dimension members.
--------------	--

<i>lPeriod</i>	Long (ByVal). The member ID of the transactions’ Period dimension members.
----------------	--

<i>varalSeqIDs</i>	Long array (ByVal). The transactions’ sequence IDs. Sequence IDs are internal unique identifiers of transactions. You can get a transaction’s sequence ID with the HsvICTransactionsData method GetICTransactionData .
--------------------	---

<i>varalEntities</i>	Long array (ByVal). The member IDs of the cells’ Entity dimension members.
----------------------	--

<i>varalICPs</i>	Long array (ByVal). The member IDs of the cells’ Intercompany Partner dimension members.
------------------	--

<i>varalAccounts</i>	Long array (ByVal). The member IDs of the cells’ Account dimension members.
----------------------	---

<i>varalC1s</i>	Long array (ByVal). The member IDs of the cells’ Custom 1 dimension members.
-----------------	--

<i>varalC2s</i>	Long array (ByVal). The member IDs of the cells’ Custom 2 dimension members.
-----------------	--

<i>varalC3s</i>	Long array (ByVal). The member IDs of the cells’ Custom 3 dimension members.
-----------------	--

<i>varalC4s</i>	Long array (ByVal). The member IDs of the cells’ Custom 4 dimension members.
-----------------	--

Return Value

String. Returns an XML string, with the format varying depending upon the type of processing performed:

- For posting, unposting, and deleting, the XML structure is as follows, with the `<ictransactions>` tag containing one `<transaction>` tag for each transaction that `ProcessICTransactions` could not process. Each `<transaction>` tag contains information about the transaction, along with an HRESULT that indicates why the transaction could not be processed:

```
<?xml version="1.0" ?><ictransactions><transaction><entity />
<partner /><account /><transid /><transsubid /> <hresult /></
transaction></ictransactions>
```

- For matching and unmatching, the XML structure is as follows, with the status attribute containing an HRESULT that indicates why the transaction could not be processed:

```
<processtransactionsresponse status="HRESULT"> </
processtransactionsresponse>
```

Tip: To get the description of an HRESULT, pass the HRESULT to the HsvResourceManager method [GetResourceStringFromHR](#).

SaveColumnFilter

For internal use.

SaveICReasonCode

Creates a reason code.

Syntax

```
<HsvICM>.SaveICReasonCode bstrReasonCode, bstrReasonDescription
```

Argument	Description
<i>bstrReasonCode</i>	String (ByVal). The reason code.
<i>bstrReasonDescription</i>	String (ByVal). The description of the reason code.

SaveICTransaction

Updates an existing transaction.

Tip: To test whether a cell is valid for intercompany transactions, use [DoesCellSupportICTransactionDetail](#). To create a new transaction, use [CreateICTransaction](#). To get a transaction's details, use [HsvICTransactionsData.GetICTransactionData](#).

Syntax

```
<HsvICM>.SaveICTransaction vbAmountChanged, vbOverwrite, lSeqID,
lScenario, lYear, lPeriod, lEntity, lICP, lAccount, lC1, lC2, lC3, lC4,
lTRCur, lReason, dTRAmt, dTRLAmt, dTRRate, dTRDate, bstrID, bstrSubID,
bstrRefID, bstrComment1, bstrComment2
```


Argument	Description
<i>vbAmountChanged</i>	Boolean (ByVal). A flag that specifies whether the transaction amount is being changed. Pass TRUE to change the amount, FALSE otherwise.
<i>vbOverwrite</i>	Boolean (ByVal). A flag that specifies whether to overwrite existing transactions that have the same Transaction ID and Sub ID. Pass TRUE to overwrite, FALSE otherwise.
<i>ISeqID</i>	Long (ByVal). The transaction's sequence ID. Sequence IDs are internal unique identifiers of transactions. You can get a transaction's sequence ID with the HsvICTransactionsData method GetICTransactionData .
<i>IScenario</i>	Long (ByVal). The transaction's Scenario dimension member.
<i>IYear</i>	Long (ByVal). The transaction's Year dimension member.
<i>IPeriod</i>	Long (ByVal). The transaction's Period dimension member.
<i>IEntity</i>	Long (ByVal). The transaction's Entity dimension member.
<i>IICP</i>	Long (ByVal). The transaction's Intercompany Partner dimension member.
<i>IAccount</i>	Long (ByVal). The transaction's Account dimension member.
<i>IC1</i>	Long (ByVal). The transaction's Custom 1 dimension member.
<i>IC2</i>	Long (ByVal). The transaction's Custom 2 dimension member.
<i>IC3</i>	Long (ByVal). The transaction's Custom 3 dimension member.
<i>IC4</i>	Long (ByVal). The transaction's Custom 4 dimension member.
<i>ITRCur</i>	Long (ByVal). The ID of the transaction's currency. Tip: You can obtain the ID of an Entity dimension member's currency with GetEntityCurrencyID . You also can obtain a currency ID with GetTransCurrencyID .
<i>IReason</i>	Long (ByVal). The ID of the transaction's reason code. Tip: You can get a reason code ID with GetICReasonCodeID .
<i>dTRAmt</i>	Double (ByVal). The transaction amount.
<i>dTRLAmt</i>	Double (ByVal). The entity currency amount.
<i>dTRRate</i>	Double (ByVal). The conversion rate.
<i>dTRDate</i>	Double (ByVal). The transaction date formatted as a Double.
<i>bstrID</i>	String (ByVal). The Transaction ID.
<i>bstrSubID</i>	String (ByVal). The transaction's Sub ID.
<i>bstrRefID</i>	String (ByVal). The transaction's Reference ID.
<i>bstrComment1</i>	String (ByVal). The first comment for the transaction. Note: You can specify a maximum of 256 characters.

Argument	Description
<i>bstrComment2</i>	String (ByVal). The second comment for the transaction. Note: You can specify a maximum of 256 characters.

SaveRowFilter

For internal use.

SetReasonCodeToICTransactions

Sets the reason code for one or more intercompany transactions.

The intercompany transactions are identified by sequence IDs. The sequence IDs and the transactions' members of the Entity and Intercompany Partner dimensions are passed in arrays that have a one-to-one correspondence.

Tip: To obtain transactions' sequence IDs and dimension members, use the HsvICTransactionsData object's [GetICTransactionData](#) method.

Syntax

```
<HsvICM>.SetReasonCodeToICTransactions(lScenario, lYear, lPeriod, lReasonCodeID, varalSeqIDs, varalEntities, varalICPs)
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the transactions' Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the transactions' Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the transactions' Period dimension member.
<i>lReasonCodeID</i>	Long (ByVal). The ID of the reason code to set.
<i>varalSeqIDs</i>	Long array (ByVal). The transactions' sequence IDs.
<i>varalEntities</i>	Long array (ByVal). The member IDs of the transactions' Entity dimension members.
<i>varalICPs</i>	Long array (ByVal). The member IDs of the transactions' Intercompany Partner dimension members.

Return Value

String. Returns an XML string that describes the transactions for which the reason code was set. The following table describes the XML string.

Table 73 SetReasonCodeToICTransactions Return Value XML String

Element	Description
ictransactions	Root element, and contains one <transaction> per intercompany transaction.
transaction	Represents an intercompany transaction, and contains the dimension member, ID, and HRESULT elements described in the following rows.
entity	The label of the transaction's Entity dimension member.
partner	The label of the transaction's Intercompany Partner dimension member.
account	The label of the transaction's Account dimension member.
transid	The transaction's Transaction ID.
transsubid	The transaction's Sub ID.
hresult	The HRESULT associated with updating the transaction's reason code.

UnMatchICTransactions

Unmatches transactions to which the specified match codes have been assigned.

Syntax

```
<HsvICM>.UnMatchICTransactions(lScenario, lYear, lPeriod,
varabstrMatchCodes)
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The Scenario dimension member for which to unmatched transactions.
<i>lYear</i>	Long (ByVal). The Year dimension member for which to unmatched transactions.
<i>lPeriod</i>	Long (ByVal). The Period dimension member for which to unmatched transactions.
<i>varabstrMatchCodes</i>	String array (ByVal). The match codes for which to unmatched transactions.

Return Value

String. Returns an XML string that indicates whether an error occurred. The string is of the following format, with the status attribute containing zero if no error occurred, or a non-zero error number otherwise:

```
<processtransactionsresponse status="HRESULT">
</processtransactionsresponse>
```

IHsvAdminICM Interface Methods

The IHsvAdminICM interface enables you to programmatically administer intercompany transactions. For example, you can use this interface to open and close periods, lock and unlock entities, and set and get period settings.

To set an IHsvAdminICM object reference, use the HsvSession object's [ICM](#) property as shown in the following example:

```
Dim cAdminICM As IHsvAdminICM
'g_cSession is an HsvSession object reference
Set cAdminICM = g_cSession.ICM
```

The interface's methods are summarized in [“IHsvAdminICM Interface Overview”](#) on page 107, and are described in detail in the following topics.

CheckSecurityForICExtract

Indicates whether the connected user has security rights for extracting intercompany transactions. If the user does not have security rights, a non-zero error number is thrown as an HRESULT. If the user has security rights, 0 is thrown as an HRESULT.

Tip: In Visual Basic, check for the return value with `Err.Number`.

Syntax

```
<HsvAdminICM>.CheckSecurityForICExtract
```

CloseICPeriod

Closes a given period for intercompany transactions.

Syntax

```
<IHsvAdminICM>.CloseICPeriod lScenario, lYear, lPeriod
```

Argument Description

lScenario Long (ByVal). The member ID of the Scenario dimension member for which to close the period.

lYear Long (ByVal). The member ID of the Year dimension member for which to close the period.

lPeriod Long (ByVal). The member ID of the Period dimension member to close.

Example

The following subroutine closes a period. The subroutine takes the labels of the Scenario, Year, and Period dimension members, and obtains the corresponding member IDs with `IHsvTreeInfo.GetItemID`.

```

Sub closePeriodLabels(sScen As String, sYear As String, sPer As String)
    Dim lScen As Long, lYear As Long, lPer As Long
    Dim cAdminICM As IHsvAdminICM, cTreeInfo As IHsvTreeInfo
    'g_cSession is an HsvSession object reference
    Set cAdminICM = g_cSession.ICM
    'g_cMetadata is an HsvMetadata object reference
    Set cTreeInfo = g_cMetadata.Scenarios
    lScen = cTreeInfo.GetItemID(sScen)
    Set cTreeInfo = g_cMetadata.Years
    lYear = cTreeInfo.GetItemID(sYear)
    Set cTreeInfo = g_cMetadata.Periods
    lPer = cTreeInfo.GetItemID(sPer)
    cAdminICM.CloseICPeriod lScen, lYear, lPer
End Sub

```

GetICEntitiesLockStatus

Indicates whether any entities are locked for a given scenario, year, and period.

Note: To return an Integer that indicates whether a given entity is locked or unlocked, use [GetLockStatusICEntity](#).

Syntax

```
<HsvAdminICM>.GetICEntitiesLockStatus(lScenario, lYear, lPeriod)
```

Argument Description

lScenario Long (ByVal). The member ID of the Scenario dimension member.

lYear Long (ByVal). The member ID of the Year dimension member.

lPeriod Long (ByVal). The member ID of the Period dimension member.

Return Value

Variant. Returns either an empty variant or a two-dimensional array. If no entities are locked, an empty variant is returned. If one or more entities are locked, a two-dimensional array is returned; the first dimension contains the member IDs of the Entity dimension members, and the second dimension contains the entities' locking statuses, which are represented by the HFMConstants type library constants listed in [“Lock Status Constants” on page 874](#).

GetLockStatusICEntities

Deprecated - use [GetICEntitiesLockStatus](#).

GetLockStatusICEntity

Indicates whether an entity is locked or unlocked for a given scenario, period, and year.

Note: To test whether any entities are locked or unlocked for a scenario, year, and period, use [GetICEntitiesLockStatus](#).

Syntax

```
<IHsvAdminICM>.GetLockStatusICEntity(lScenario, lYear, lPeriod, lEntity)
```

Argument Description

lScenario Long (ByVal). The member ID of the Scenario dimension member.

lYear Long (ByVal). The member ID of the Year dimension member.

lPeriod Long (ByVal). The member ID of the Period dimension member.

lEntity Long (ByVal). The member ID of the Entity dimension member.

Return Value

Integer. Indicates whether the entity is locked. Valid values are represented by the HFMCConstants type library constants listed in [“Lock Status Constants” on page 874](#).

Example

The following function takes Scenario, Year, Period, and Entity dimension member IDs and returns a Boolean that indicates whether the entity is locked.

```
Function isEntityLocked(lScen As Long, lYear As Long, lPer As Long, _  
    lEnt As Long) As Boolean  
    Dim cAdminICM As IHsvAdminICM, iStatus As Integer  
    'g_cSession is an HsvSession object reference  
    Set cAdminICM = g_cSession.ICM  
    iStatus = cAdminICM.GetLockStatusICEntity(lScen, lYear, lPer, lEnt)  
    If iStatus = ICM_LOCKED Then  
        isEntityLocked = True  
    Else  
        isEntityLocked = False  
    End If  
End Function
```

GetSettingsICPeriod

Returns the period status, matching tolerance, and Match/Validate Before Post settings for a scenario, year, and period.

Note: To return these settings for multiple periods, use [GetSettingsICPeriods](#).

Syntax

```
<IHsvAdminICM>.GetSettingsICPeriod lScenario, lYear, lPeriod, psStatus,  
psMVBP, pdMatchTolerance
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>psStatus</i>	Integer. Returns the period's status. Valid values are represented by the HFMCConstants type library constants listed in "Period Status Constants" on page 874 .
<i>psMVBP</i>	Integer. Returns the period's Match/Validate Before Post setting. Valid values are represented by the HFMCConstants type library constants listed in "Match/Validate Before Post Constants" on page 874 .
<i>pdMatchTolerance</i>	Double. Returns the matching tolerance specified for the period.

Example

The following function takes Scenario, Year, Period, and Entity dimension member IDs and returns a Boolean that indicates whether the period's Match/Validate Before Post setting is on.

```
Function isMatchValidateOn(lScen As Long, lYear As Long, lPer As Long) _
    As Boolean
    Dim cAdminICM As IHsvAdminICM, iStat As Integer, iMatch As Integer
    Dim dTol As Double
    'g_cSession is an HsvSession object reference
    Set cAdminICM = g_cSession.ICM
    cAdminICM.GetSettingsICPeriod lScen, lYear, lPer, iStat, iMatch, dTol
    If iMatch = ICM_MVBP_ON Then
        isMatchValidateOn = True
    Else
        isMatchValidateOn = False
    End If
End Function
```

GetSettingsICPeriods

Returns the period status, matching tolerance, and Match/Validate Before Post settings of multiple periods for a scenario and year.

Note: To return these settings for a single period, use [GetSettingsICPeriod](#).

Syntax

```
<IHsvAdminICM>.GetSettingsICPeriods(lScenario, lYear, varalPeriods)
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.

Argument Description

varalPeriods Long array (ByVal). The member IDs of the Period dimension members.

Return Value

Variant. Returns a multidimensional array of the settings for each period. The first dimension contains four items that represent the period settings, and the second dimension contains one item for each specified period. The following table describes the first dimension:

Index	Description
0	Long. Returns the member ID of the Period dimension member to which the settings apply.
1	Byte. Returns the period's status. Valid values are represented by the HFMConstants type library constants listed in "Period Status Constants" on page 874 .
2	Byte. Returns the period's Match/Validate Before Post setting. Valid values are represented by the HFMConstants type library constants listed in "Match/Validate Before Post Constants" on page 874 .
3	Double. Returns the matching tolerance specified for the period.

Example

The following snippet populates a user-defined type with the settings returned by `GetSettingsICPeriods`. The period labels, statuses, and Match/Validate Before Post settings are converted to strings before being added to the type. Following is the type declaration:

```
Public Type perSettings
    sPeriod As String
    sStatus As String
    sMatchValidate As String
    dTolerance As Double
End Type
```

The example assumes that the variables containing the Scenario, Year, and Period member IDs have been populated.

```
Dim cAdminICM As IHsvAdminICM, cTreeInfo As IHsvTreeInfo
Dim vaRet As Variant, udtSetting() As perSettings
'laPers() is a previously-defined array of Period member IDs
ReDim udtSetting(UBound(laPers))
'g_cSession is an HsvSession object reference
Set cAdminICM = g_cSession.ICM
'g_cMetadata is an HsvMetadata object reference
Set cTreeInfo = g_cMetadata.Periods
'dimension member IDs have already been defined
vaRet = cAdminICM.GetSettingsICPeriods(lScen, lYear, laPers)
For i = LBound(vaRet, 2) To UBound(vaRet, 2)
    cTreeInfo.GetLabel vaRet(0, i), udtSetting(i).sPeriod
    Select Case vaRet(1, i)
        Case ICM_CLOSED
            udtSetting(i).sStatus = "Closed"
        Case ICM_OPENED
            udtSetting(i).sStatus = "Open"
```



```

    Case ICM_UNOPENED
        udtSetting(i).sStatus = "Unopened"
    End Select
    If vaRet(2, i) = ICM_MVBP_OFF Then
        udtSetting(i).sMatchValidate = "Off"
    Else
        udtSetting(i).sMatchValidate = "On"
    End If
    udtSetting(i).dTolerance = vaRet(3, i)
Next i

```

LoadICTransactions

Loads or scans intercompany transactions from arrays that specify the transactions' dimension members and data.

Syntax

```

<IHsvAdminICM>.LoadICTransactions var2dlsaTranDim, var2dbstrsaTranDetails,
bstrDelimiter, lNumTrans, lLoadOptions, dStartLoadTime, pvaralResults

```

Argument	Description
<i>var2dlsaTranDim</i>	<p>Long array (ByVal). A two-dimensional array that specifies the dimension members for which to load transactions. The first dimension contains one item per transaction, and has a one-to-one correspondence with the first dimension of the <i>var2dbstrsaTranDetails</i> argument's array.</p> <p>The second dimension specifies the member IDs of the transactions' dimension members, contains 10 items, and is indexed as follows:</p> <ul style="list-style-type: none"> ● 0 = Scenario ● 1 = Year ● 2 = Period ● 3 = Entity ● 4 = Intercompany Partner ● 5 = Account ● 6 = Custom 1 ● 7 = Custom 2 ● 8 = Custom 3 ● 9 = Custom 4
<i>var2dbstrsaTranDetails</i>	<p>String array (ByVal). A two-dimensional array that specifies the transaction data. The first dimension contains one item per transactions, and has a one-to-one correspondence with the first dimension of the <i>var2dlsaTranDim</i> argument's array.</p> <p>The second dimension specifies the transactions' data, contains 10 items, and is indexed as follows. All items are strings except when otherwise noted:</p> <ul style="list-style-type: none"> ● 0 = Transaction ID. ● 1 = Sub ID. ● 2 = Transaction Date. This must be a date cast to a Double, as in the following example: <pre style="text-align: center;"> Cdbl(CDate("11/13/2004")) </pre>

Argument	Description
	<ul style="list-style-type: none"> ● 3 = Transaction currency label. ● 4 = Transaction amount. ● 5 = Entity currency amount. ● 6 = Conversion rate. ● 7 = Reference ID. ● 8 = Comment 1. ● 9 = Comment 2.
<i>bstrDelimiter</i>	String (ByVal). A delimiter character that the system can use to process the transactions. This character cannot be contained in any of the strings in the <i>var2dbstrsaTranDetails</i> argument's array.
<i>lNumTrans</i>	Long (ByVal). The number of transactions to load. If this number is less than the indexes of the first dimensions of the <i>var2dlsaTranDim</i> and <i>var2dbstrsaTranDetails</i> arguments' arrays, then only the number of transactions specified here are loaded.
<i>lLoadOptions</i>	Long (ByVal). The load option, which specifies whether to load or scan the data and whether to merge or replace existing transactions. Valid values are represented by the HFMConstants type library constants listed in "Transaction Load Mode Constants" on page 876 .
<i>dStartLoadTime</i>	Double (ByVal). The timestamp indicating when to start the loading task. The timestamp must be a Date cast to a Double.
<i>pvaralResults</i>	VARIANT. Returns an array of HRESULTS indicating the success or failure of the loading. Each item corresponds to the first dimension in the <i>var2dbstrsaTranDetails</i> argument's array of transactions.

Example

The following example loads an intercompany transaction.

```

Dim cAdminICM As IHsvAdminICM, laDims(0, 9) As Long
Dim saData(0, 9) As String, vaRet
Set cAdminICM = g_cSession.ICM
' GetMemberID is a user-defined function that returns a member ID from
' a member label
laDims(0, 0) = GetMemberID(DIMENSIONSCENARIO, "ActMon")
laDims(0, 1) = GetMemberID(DIMENSIONYEAR, "2004")
laDims(0, 2) = GetMemberID(DIMENSIONPERIOD, "February")
laDims(0, 3) = GetMemberID(DIMENSIONENTITY, "C")
laDims(0, 4) = GetMemberID(DIMENSIONICP, "A")
laDims(0, 5) = GetMemberID(DIMENSIONACCOUNT, "PayltIC")
laDims(0, 6) = GetMemberID(DIMENSIONCUSTOM1, "Increases")
laDims(0, 7) = GetMemberID(DIMENSIONCUSTOM2, "[None]")
laDims(0, 8) = GetMemberID(DIMENSIONCUSTOM3, "[None]")
laDims(0, 9) = GetMemberID(DIMENSIONCUSTOM4, "[None]")
saData(0, 0) = "Tran001"
saData(0, 1) = "Sub001"
saData(0, 2) = CDbl(CDate("2/13/2004"))
saData(0, 3) = "EUR"
saData(0, 4) = "100"
saData(0, 5) = "100"
saData(0, 6) = "1.0"
saData(0, 7) = "Ref001"

```

```

saData(0, 8) = "comment 1"
saData(0, 9) = "comment 2"
cAdminICM.LoadICTransactions lADims, saData, ";", 1, ICT_LOAD_REPLACE, _
    Now, vaRet

```

LockICEntity

Locks an entity for a given scenario, year, and period.

Syntax

```
<IHsvAdminICM>.LockICEntity lScenario, lYear, lPeriod, lEntity, vbCheckMVBP
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>lEntity</i>	Long (ByVal). The member ID of the Entity dimension member.
<i>vbCheckMVBP</i>	Boolean (ByVal). A flag that specifies whether to lock the entity only if matched transactions and mismatched transactions with a reason code are posted. Pass TRUE to lock only if such transactions are posted, FALSE otherwise. If you pass TRUE, and one or more of such transactions are not posted, the entity is not locked and an error is thrown.

OpenICPeriod

Opens a period for a given scenario and year, and specifies the period's matching tolerance and Match/Validate Before Post settings.

Syntax

```
<IHsvAdminICM>.OpenICPeriod lScenario, lYear, lPeriod, lMVBP,
dMatchTolerance
```

Argument	Description
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>lMVBP</i>	Integer (ByVal). The Match/Validate Before Post setting for the period. Valid values are represented by the HFMConstants type library constants listed in "Match/Validate Before Post Constants" on page 874 .
<i>dMatchTolerance</i>	Double (ByVal). The matching tolerance setting for the period.

OpenICPeriod2

Opens a period for a given scenario and year, and specifies the period's Match/Validate Before Post settings and absolute or percentage matching tolerance.

Syntax

```
<HsvAdminICM>.OpenICPeriod2 lScenario, lYear, lPeriod, lMVBP, dAcctTol, dTIDTol, dManTol, dAcctPct, dTIDPct, dManPct
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.
<i>lMVBP</i>	Integer (ByVal). The Match/Validate Before Post setting for the period. Valid values are represented by the HFMCConstants type library constants listed in "Match/Validate Before Post Constants" on page 874
<i>dAcctTol</i>	Double (ByVal). Absolute matching tolerance for match by accounts.
<i>dTIDTol</i>	Double (ByVal). Absolute matching tolerance for match by ID.
<i>dManTol</i>	Double (ByVal). Absolute matching tolerance for manual match.
<i>dAcctPct</i>	Double (ByVal). Percentage matching tolerance for match by accounts.
<i>dTIDPct</i>	Double (ByVal). Percentage matching tolerance for match by IDs.
<i>dManPct</i>	Double (ByVal). Percentage matching tolerance for manual match.

SavePeriodsSettings

For internal use.

UnlockICEntity

Unlocks an entity for a given scenario, year, and period.

Syntax

```
<IHsvAdminICM>.UnlockICEntity lScenario, lYear, lPeriod, lEntity
```

Argument Description

<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.

Argument Description

IEntity Long (ByVal). The member ID of the Entity dimension member.

UpdatePeriodSettings

Updates an open period's matching tolerance and Match/Validate Before Post settings for a given scenario and year.

Caution! If you specify a period that is not open, an error occurs.

Syntax

```
<IHsvAdminICM>.UpdatePeriodSettings lScenario, lYear, lPeriod, sMVBP,
dMatchTolerance
```

Argument Description

IScenario Long (ByVal). The member ID of the Scenario dimension member.

IYear Long (ByVal). The member ID of the Year dimension member.

IPeriod Long (ByVal). The member ID of the Period dimension member.

sMVBP Integer (ByVal). The Match/Validate Before Post setting for the period. Valid values are represented by the HFMCConstants type library constants listed in ["Match/Validate Before Post Constants" on page 874](#).

dMatchTolerance Double (ByVal). The matching tolerance setting for the period.

UpdatePeriodSettings2

Updates an open period's Match/Validate Before Post settings and absolute or percentage matching tolerance for a given scenario and year.

Syntax

```
<HsvAdminICM>.UpdatePeriodSettings2 lScenario, lYear, lPeriod, sMVBP,
dAcctTol, dTIDTol, dManTol, dAcctPct, dTIDPct, dManPct
```

Argument Description

IScenario Long (ByVal). The member ID of the Scenario dimension member.

IYear Long (ByVal). The member ID of the Year dimension member.

IPeriod Long (ByVal). The member ID of the Period dimension member.

sMVBP Integer (ByVal). The Match/Validate Before Post setting for the period. Valid values are represented by the HFMCConstants type library constants listed in ["Match/Validate Before Post Constants" on page 874](#)

Argument Description

<i>dAcctTol</i>	Double (ByVal). Absolute matching tolerance for match by accounts.
<i>dTIDTol</i>	Double (ByVal). Absolute matching tolerance for match by ID.
<i>dManTol</i>	Double (ByVal). Absolute matching tolerance for manual match.
<i>dAcctPct</i>	Double (ByVal). Percentage matching tolerance for match by accounts.
<i>dTIDPct</i>	Double (ByVal). Percentage matching tolerance for match by IDs.
<i>dManPct</i>	Double (ByVal). Percentage matching tolerance for manual match.

In This Chapter

Load and Extract Option Interfaces.....	743
Common Properties and Methods.....	747
HsvSecurityLoadACV Type Library	752
HsvMetadataLoadACV Type Library.....	758
HsvRulesLoadACV Type Library	768
HsvcDataLoad Type Library	775
HsvJournalLoadACV Type Library	784

Financial Management includes type libraries for loading and extracting security information, metadata, rules and member lists, data, and journals. These type libraries are all client-side application components (ACVs), enabling you to programmatically load and extract information from and to text files on client computers.

Load and Extract Option Interfaces

In Financial Management, the load and extract operations for security information, metadata, data, and journals allow users to specify various load and extract options. For example, when users extract metadata, they can specify the dimensions that they want to extract.

The same principle applies to programmatically loading and extracting security information, metadata, data, and journals. Use the following objects to set load and extract options:

- For a given load or extract operation, each load or extract option is represented by an instance of the *IHsvLoadExtractOption* interface. For example, when extracting metadata there is an *IHsvLoadExtractOption* instance that specifies whether scenarios are extracted, another *IHsvLoadExtractOption* instance that specifies whether years are extracted, and so on. The *IHsvLoadExtractOption* interface provides the `CurrentValue` property, which enables you to specify a load or extract option's value, as well as other properties.
- The set of *IHsvLoadExtractOption* instances for a given type of load or extract operation is represented by an instance of the *IHsvLoadExtractOptions* interface. An *IHsvLoadExtractOptions* instance provides a read-only collection of the *IHsvLoadExtractOption* instances for a load or extract operation. The *IHsvLoadExtractOptions* interface includes `Item` and `Count` properties that enable you to access and loop through the available *IHsvLoadExtractOption* instances.

You obtain instances of the `IHsvLoadExtractOptions` interface with the `LoadOptions` and `ExtractOptions` properties of the `HsvSecurityLoadACV`, `HsvcDataLoad`, `HsvJournalLoadACV`, and `HsvMetadataLoadACV` objects. You then obtain instances of the `IHsvLoadExtractOption` interface with the `Item` property of the `IHsvLoadExtractOptions` interface. See “[IHsvLoadExtractOptions Interface Properties](#)” on page 744 and “[IHsvLoadExtractOption Interface Properties](#)” on page 745.

IHsvLoadExtractOptions Interface Properties

The `IHsvLoadExtractOptions` interface provides two properties: `Item` and `Count`. Use `Item` to obtain `IHsvLoadExtractOption` object references, and use `Count` to return a count of the available `IHsvLoadExtractOption` instances. The following topics provide more details on these properties.

The way in which you assign `IHsvLoadExtractOptions` object references depends upon whether the object is being used to load or extract information:

- For loading, use the `LoadOptions` property.
- For extracting, use the `ExtractOptions` property.

Count

Returns a count of the `IHsvLoadExtractOption` instances available for setting load or extract options. In other words, this property returns a count of the available load or extract options.

Data Type

Long.

Item

Returns object references to instances of the `IHsvLoadExtractOption` interface. In other words, this property provides access to the available load or extract options.

`Item` takes a `Variant` parameter that specifies the load or extract option that you want to use. You can pass the following types of information as `Item`'s parameter:

- The constant that represents the load or extract option.
- The name of the load or extract option. Valid option names are returned by `IHsvLoadExtractOption.Name`.
- The numeric ID of the load or extract option. Valid option IDs are returned by `IHsvLoadExtractOption.OptionID`.

For example, the `Year` extract option for the `HsvcDataLoad` object is identified by the constant `HSV_DATAEXTRACT_OPT_YEAR_SUBSET`, the name `Year`, and the option ID number 8. This means that *all* of the following calls to `Item` have the same result:

```
Set cOpt = cOptions.Item _  
(HSV_DATAEXTRACT_OPT_YEAR_SUBSET)
```



```
Set cOpt = cOptions.Item("Year")
Set cOpt = cOptions.Item(8)
```

For details on `Item`'s valid parameters, see the following topics:

- [“Security Load Options” on page 755](#)
- [“Security Extract Options” on page 756](#)
- [“Metadata Load Options” on page 761](#)
- [“Metadata Extract Options” on page 765](#)
- [“Data Load Options” on page 778](#)
- [“Data Extract Options” on page 781](#)
- [“Journal Load Options” on page 789](#)
- [“Journal Extract Options \(Unfiltered\)” on page 789](#)

Data Type

IHsvLoadExtractOption interface.

Example

`Item` is used in several examples, including the following:

- [“Example for Loading Security” on page 754](#)
- [“Example for Extracting Metadata” on page 761](#)
- [“Example for Extracting Data” on page 777](#)
- [“Example for Loading Journals” on page 785](#)

Note: For simplicity's sake, this document follows a convention of passing constants to `Item`. However, in all explanations and examples, the corresponding `Name` or `OptionID` property could be substituted for the constant.

IHsvLoadExtractOption Interface Properties

Use the `IHsvLoadExtractOption` interface's properties to get and set the values of load and extract options, and to get various types of option-related information. These properties are summarized in [Table 44 on page 110](#), and are described in the following topics.

To assign `IHsvLoadExtractOption` object references, use the `IHsvLoadExtractOptions` interface's `Item` property. For more information, see [“Item” on page 744](#).

CurrentValue

Sets or returns the current value of a load or extract option.

Data Type

Variant; the subtype depends upon the load or extract option. For details on `CurrentValue`'s valid values, see the following topics:

- [“Security Load Options” on page 755](#)
- [“Security Extract Options” on page 756](#)
- [“Metadata Load Options” on page 761](#)
- [“Metadata Extract Options” on page 765](#)
- [“Data Load Options” on page 778](#)
- [“Data Extract Options” on page 781](#)
- [“Journal Load Options” on page 789](#)
- [“Journal Extract Options \(Unfiltered\)” on page 789](#)

Example

`CurrentValue` is used in several examples, including the following examples:

- [“Example for Loading Security” on page 754](#)
- [“Example for Loading Metadata” on page 760](#)
- [“Example for Extracting Metadata” on page 761](#)
- [“Example for Extracting Data” on page 777](#)
- [“Example for Extracting Journals Without Filters” on page 787](#)

DefaultValue

Returns the default value of a load or extract option.

Data Type

Variant.

MaxValue

For some load or extract options, `MaxValue` returns the option's maximum valid value. Note that `MaxValue` is not supported by all load or extract options.

Data Type

Long.

MinValue

For some load or extract options, `MinValue` returns the option's minimum valid value. Note that `MinValue` is not supported by all load or extract options.

Data Type

Long.

Name

Returns the name of a load or extract option. You can pass this name to `IHsvLoadExtractOptions.Item`; for more information, see [“Item” on page 744](#).

Data Type

String.

OptionID

Returns the numeric ID of a load or extract option. You can pass this ID to `IHsvLoadExtractOptions.Item`; for more information, see [“Item” on page 744](#).

Data Type

Long.

ValidationList

For some load and extract options, `ValidationList` returns the options' valid values. For example, Financial Management supports only certain delimiter characters in load and extract files. For delimiter load and extract options, `ValidationList` returns a `String` that contains the valid delimiters.

Data Type

String.

Tip: If no valid values for an option are exposed through this property, `ValidationList` returns a blank `String`.

Common Properties and Methods

The `HsvSecurityLoadACV`, `HsvMetadataLoadACV`, `HsvcDataLoad`, and `HsvJournalLoadACV` objects share various properties and methods. All of these objects implement the `SetSession` method, which points to the application for which information is being loaded or extracted.

In addition, the `HsvSecurityLoadACV`, `HsvMetadataLoadACV`, `HsvcDataLoad`, and `HsvJournalLoadACV` objects all implement the following properties and methods:

- `ExtractOptions` property
- `LoadOptions` property
- `Extract` method

- `Load` method

The properties are described in [“Common Load and Extract Properties”](#) on page 748, and the methods are described in [“Common Load and Extract Methods”](#) on page 749.

Common Load and Extract Properties

The `HsvSecurityLoadACV`, `HsvMetadataLoadACV`, `HsvcDataLoad`, and `HsvJournalLoadACV` objects all provide the `ExtractOptions` and `LoadOptions` properties, which are used to assign `IHsvLoadExtractOptions` object references. These properties are described in the following topics.

ExtractOptions

Returns an object reference to an instance of the `IHsvLoadExtractOptions` interface for an extraction operation. You *must* set this property before calling the `Extract` method for any of the applicable objects.

Use the object reference returned by `ExtractOptions` to access extract options with the `Item` property.

Applicable Objects

- `HsvcDataLoad`
- `HsvJournalLoadACV`
- `HsvMetadataLoadACV`
- `HsvSecurityLoadACV`

Data Type

`IHsvLoadExtractOptions` interface.

Example

`ExtractOptions` is used in the following examples:

- [“Example for Extracting Security”](#) on page 755
- [“Example for Extracting Metadata”](#) on page 761
- [“Example for Extracting Data”](#) on page 777
- [“Example for Extracting Journals Without Filters”](#) on page 787

LoadOptions

Returns an object reference to an instance of the `IHsvLoadExtractOptions` interface for a load operation. You *must* set this property before calling the `Load` method for any of the applicable objects.

Use the object reference returned by `LoadOptions` to access load options with the `Item` property.

Applicable Objects

- `HsvcDataLoad`
- `HsvJournalLoadACV`
- `HsvMetadataLoadACV`
- `HsvSecurityLoadACV`

Data Type

`IHsvLoadExtractOptions` interface.

Example

`LoadOptions` is used in the following examples:

- “[Example for Loading Security](#)” on page 754
- “[Example for Loading Metadata](#)” on page 760
- “[Example for Loading Data](#)” on page 776
- “[Example for Loading Journals](#)” on page 785

Common Load and Extract Methods

The `HsvSecurityLoadACV`, `HsvMetadataLoadACV`, `HsvcDataLoad`, and `HsvJournalLoadACV` objects all implement the `Extract` and `Load` methods, which load and extract metadata, data, and journals. In addition, these objects all implement the `SetSession` method, which specifies the application for which information is being loaded and extracted. These methods are described in the following topics.

Extract

Extracts security information, metadata, data, or journals from an application into a text file on the client computer.

You must call `SetSession` and set the `ExtractOptions` property before calling `Extract`. The various extract operations may have other prerequisites for calling `Extract`. For information on how `Extract` is used with its applicable objects, see the following topics:

- “[Extracting Security Information](#)” on page 754
- “[Extracting Metadata](#)” on page 760
- “[Extracting Data](#)” on page 777
- “[Extracting Journals](#)” on page 786

Applicable Objects

- HsvcDataLoad
- HsvJournalLoadACV
- HsvMetadataLoadACV
- HsvSecurityLoadACV

Syntax

```
<objectRef>.Extract bstrClientFilename, bstrClientLogFileName
```

Argument	Description
<i>bstrClientFilename</i>	String (ByVal). The name and path of the file into which the security information, metadata, data, or journals will be extracted. The path must exist on the client computer. For details on the extracted file's format, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i> .

bstrClientLogFileName String (ByVal). The name and path of the log file for the extraction operation.

Example

Extract is used in the following examples:

- [“Example for Extracting Security” on page 755](#)
- [“Example for Extracting Metadata” on page 761](#)
- [“Example for Loading Data” on page 776](#)
- [“Example for Extracting Journals Without Filters” on page 787](#)

Load

Loads metadata, data, or journals from a text file into an application, with the text file located on a client computer. For metadata, Load is supported only for Classic applications.

Caution! If you attempt to load metadata into a Performance Management Architect application, Load will fail.

You must call `SetSession` and set the `LoadOptions` property before calling `Load`. The various load operations may have other prerequisites for calling `Load`. For information on how `Load` is used with its applicable objects, see the following topics:

- [“Loading Metadata” on page 759](#)
- [“Loading Data” on page 776](#)
- [“Loading Journals” on page 785](#)

Applicable Objects

- HsvcDataLoad
- HsvJournalLoadACV
- HsvMetadataLoadACV

Note: The HsvSecurityLoadACV object uses a slightly different `Load` method; for details, see [“Load” on page 758](#). In addition, the HsvcDataLoad object offers an alternate method for loading data. This method returns a flag that indicates whether errors were logged; for details, see [“Load2” on page 783](#).

Syntax

```
<objectRef>.Load bstrClientFilename, bstrClientLogFileName
```

Argument	Description
<i>bstrClientFilename</i>	<p>String (ByVal). The name and path of the file to be loaded. The path must be valid for the client computer.</p> <p>For details on the file's required format, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i>.</p> <p>Tip: The metadata load file format has changed since the original release of Financial Management. If the metadata load file is in an obsolete file format, an error is thrown.</p>
<i>bstrClientLogFileName</i>	String (ByVal). The name and path of the log file for the load operation.

Example

`Load` is used in the following examples:

- [“Example for Loading Metadata” on page 760](#)
- [“Example for Loading Data” on page 776](#)
- [“Example for Loading Journals” on page 785](#)

SetSession

Points to the HsvSession object for the application in which security information, metadata, data, or journals are being loaded or extracted.

Applicable Objects

- HsvcDataLoad
- HsvJournalLoadACV
- HsvMetadataLoadACV
- HsvSecurityLoadACV

Caution! You must call `SetSession` before calling any of the other properties or methods for the objects listed above, otherwise an error will occur. This rule applies to

the common properties; for example, you must call `SetSession` before setting the `ExtractOptions` property.

Syntax

```
<objectRef>.SetSession pIHsvSessionUnk
```

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

<i>pIHsvSessionUnk</i>	HsvSession object (ByVal). The HsvSession object that was returned by <code>HsxClient.OpenApplication</code> or <code>HsxClientUI.OpenApplication</code> when the application was opened.
------------------------	---

For information on `HsxClient.OpenApplication`, see [“OpenApplication” on page 140](#); for information on `HsxClientUI.OpenApplication`, see [“OpenApplication” on page 149](#).

Example

`SetSession` is used in several examples, including the following examples:

- [“Example for Extracting Security” on page 755](#)
- [“Example for Loading Metadata” on page 760](#)
- [“Example for Extracting Data” on page 777](#)
- [“Example for Loading Journals” on page 785](#)

HsvSecurityLoadACV Type Library

To use the HsvSecurityLoadACV type library, you must reference `HsvSecurityLoadACV.dll` in your project.

Use the HsvSecurityLoadACV type library to load and extract an application’s security information. This type library exposes the HsvSecurityLoadACV object, and also exposes the IHsvLoadExtractOptions and IHsvLoadExtractOption interfaces.

The following topics show you how to use the HsvSecurityLoadACV type library:

- [“IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvSecurityLoadACV” on page 753](#)
- [“Loading Security Information” on page 753](#)
- [“Extracting Security Information” on page 754](#)
- [“Security Load Options” on page 755](#)
- [“Security Extract Options” on page 756](#)
- [“HsvSecurityLoadACV Object Properties” on page 757](#)
- [“HsvSecurityLoadACV Object Methods” on page 757](#)

IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvSecurityLoadACV

The HsvSecurityLoadACV type library exposes the IHsvLoadExtractOptions and IHsvLoadExtractOption interfaces; these interfaces provide access to security load and extract options. For information on these interfaces, see [“Load and Extract Option Interfaces” on page 743](#).

- To set a security load or extract option:
 - 1 **Set an IHsvLoadExtractOption object reference for the option with** `IHsvLoadExtractOptions.Item`.
 - 2 **Set the option’s value with the** `IHsvLoadExtractOption.CurrentValue` **property.**

Tip: The options available for loading data are described in [“Security Load Options” on page 755](#), and the options available for extracting data are described in [“Security Extract Options” on page 756](#). These topics include the valid values for `Item` and `CurrentValue`.

Loading Security Information

The following steps provide an overview of how to load security information. These steps assume that an application has previously been opened with one of the `OpenApplication` methods.

Caution! If you attempt to load security classes into a Performance Management Architect application, `Load` will fail.

Tip: For an example that illustrates these steps, see [“Example for Loading Security” on page 754](#).

- To load security:
 - 1 **Set an object reference to the** `HsvSecurityLoadACV` **object.**
 - 2 **Point to the application into which data is being loaded by calling** `HsvSecurityLoadACV.SetSession`. **For** `SetSession`’s **argument, pass the** `HsvSession` **object reference that was returned by** `HsxClient.OpenApplication` **or** `HsxClientUI.OpenApplication`.
 - 3 **Set an object reference to the** `IHsvLoadExtractOptions` **interface with the** `HsvSecurityLoadACV.LoadOptions` **property.**
 - 4 **Optional.** By default, users and groups, security classes, role access definitions, and security class access definitions are loaded. To override the defaults for these or the other load options, specify the values for these options. The available load options are listed in [“Security Load Options” on page 755](#).
 - 5 **Load the security information by calling** `HsvSecurityLoadACV.Load`. `Load` **takes the file names and paths of the data load file and of the log file.**

Example for Loading Security

The following subroutine loads users and groups, security classes, role access definitions, and security class access definitions. The `ValidateUsers` load option is set to `TRUE` so that only valid Windows usernames in the load file will be loaded. Note how the example uses the Boolean returned by `Load`'s third argument to display a warning if the load file contains one or more invalid usernames.

```
Sub LoadSecACV(sFile As String, sLog As String)
Dim cSecurityLoadACV As HsvSecurityLoadACV
Dim cOptions As IHsvLoadExtractOptions
Dim cOpt As IHsvLoadExtractOption, vWarningSet
Set cSecurityLoadACV = New HsvSecurityLoadACV
'g_cSession is an HsvSession object reference
cSecurityLoadACV.SetSession g_cSession
'Initialize the IHsvLoadExtractOptions interface.
Set cOptions = cSecurityLoadACV.LoadOptions
'Set ValidateUsers to TRUE before loading.
Set cOpt = cOptions.Item _
    (HSV_SECURITYLOAD_OPT_AUTHENTICATE_USERS)
cOpt.CurrentValue = True
cSecurityLoadACV.Load sFile, sLog, vWarning
If vWarning = True Then
    MsgBox "The load file contained invalid user names." & _
        vbCrLf & "Check the log file for details."
End If
End Sub
```

Extracting Security Information

The following steps provide an overview of how to extract security information from an application. These steps assume that an application has previously been opened with one of the `OpenApplication` methods.

Tip: For an example that illustrates these steps, see [“Example for Extracting Security” on page 755](#).

➤ To extract security information:

- 1** Set an object reference to the `HsvSecurityLoadACV` object.
- 2** Point to the application from which data is being extracted by calling `HsvSecurityLoadACV.SetSession`. For `SetSession`'s argument, pass the `HsvSession` object reference that was returned by `HsxClient.OpenApplication` or `HsxClientUI.OpenApplication`.
- 3** Set an object reference to the `IHsvLoadExtractOptions` interface with the `HsvSecurityLoadACV.ExtractOptions` property.
- 4** **Optional.** By default, users and groups, security classes, role access definitions, and security class access definitions are extracted. To override the defaults for these or the other extract options, specify the values for these options. The available extract options are listed in [“Security Extract Options” on page 756](#).

- 5 Extract the security information by calling `HsvSecurityLoadACV.Extract`. `Extract` takes the file names and paths of the extract file and of the log file.

Example for Extracting Security

The following example extracts role access definitions and security class access definitions from an application.

```
Dim cSecurityLoadACV As HsvSecurityLoadACV
Dim cOptions As IHsvLoadExtractOptions
Dim cOpt As IHsvLoadExtractOption
Set cSecurityLoadACV = New HsvSecurityLoadACV
'Specify the HsvSession object for the application.
cSecurityLoadACV.SetSession g_cHsvSession
'Initialize the IHsvLoadExtractOptions interface.
Set cOptions = cSecurityLoadACV.ExtractOptions
Set cOpt = cOptions.Item(HSV_SECURITYEXTRACT_OPT_SECURITY_CLASSES)
cOpt.CurrentValue = False
Set cOpt = cOptions.Item(HSV_SECURITYEXTRACT_OPT_USERS)
cOpt.CurrentValue = False
cSecurityLoadACV.Extract "c:\Acme\SecurityAccess.sec", _
    "c:\Acme\SecurityAccess.log"
```

Security Load Options

For each security load option, the following table lists the constant that identifies the option and the corresponding `Name` property — either the constant or the name can be passed to `IHsvLoadExtractOptions.Item`. The table also describes each option and the type of information to be passed to `IHsvLoadExtractOption.CurrentValue`.

Tip: For information on `Item`, see “[Item](#)” on page 744. For information on `CurrentValue`, see “[CurrentValue](#)” on page 745.

Table 74 Security Load Options

Constant	Load Option Information
HSV_SECURITYLOAD_OPT_AUTHENTICATE_USERS	<p>Name property: <code>ValidateUsers</code></p> <p>Usage: Determines whether the usernames and user groups specified in a load file will be authenticated as valid Windows users.</p> <p>Pass to <code>CurrentValue</code>: Boolean — TRUE to validate users, FALSE to allow users to be loaded without validation.</p> <p>Tip: If the <code>ValidateUsers</code> option is set to TRUE and the load file contains invalid usernames or user groups, <code>Load</code>'s <code>pvbWarnings</code> argument returns TRUE and the invalid user details are noted in the log file. For details, see “Load” on page 758.</p>
HSV_SECURITYLOAD_OPT_CLEAR_ALL	<p>Name property: <code>ClearAll</code></p> <p>Usage: Determines whether existing security information will be cleared before the file is loaded.</p>

Constant	Load Option Information
	Pass to CurrentValue: Boolean – TRUE to clear existing information, FALSE otherwise.
HSV_SECURITYLOAD_OPT_DELIMITER_CHAR	<p>Name property: Delimiter</p> <p>Usage: Specifies a load file's delimiter.</p> <p>Pass to CurrentValue: String – a valid delimiter character. The <code>ValidationList</code> property returns the valid delimiters.</p>
HSV_SECURITYLOAD_OPT_ROLE_ACCESS	<p>Name property: RoleAccess</p> <p>Usage: Determines whether role access definitions will be loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load role access definitions, FALSE otherwise.</p>
HSV_SECURITYLOAD_OPT_SECURITY_CLASS_ACCESS	<p>Name property: SecurityClassAccess</p> <p>Usage: Determines whether security class access definitions will be loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load security access definitions, FALSE otherwise.</p>
HSV_SECURITYLOAD_OPT_SECURITY_CLASSES	<p>Name property: SecurityClasses</p> <p>Usage: Determines whether security classes will be loaded. This option is supported only for Classic applications.</p> <p>Pass to CurrentValue: Boolean – TRUE to load security classes, FALSE otherwise.</p> <p>Caution! If you attempt to load security classes into a Performance Management Architect application, <code>Load</code> will fail.</p>
HSV_SECURITYLOAD_OPT_USERS	<p>Name property: Users</p> <p>Usage: Determines whether usernames and user groups will be loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load usernames and user groups, FALSE otherwise.</p>

Security Extract Options

For each security extract option, the following table lists the constant that identifies the option and the corresponding Name property — either the constant or the name can be passed to `IHsvLoadExtractOptions.Item`. The table also describes each option and the type of information to be passed to `IHsvLoadExtractOption.CurrentValue`.

Tip: For information on `Item`, see “Item” on page 744. For information on `CurrentValue`, see “CurrentValue” on page 745.

Table 75 Security Extract Options

Constant	Extract Option Information
HSV_SECURITYEXTRACT_OPT_DELIMITER_CHAR	<p>Name property: Delimiter</p> <p>Usage: Specifies an extract file's delimiter.</p>

Constant	Extract Option Information
	Pass to CurrentValue: String – a valid delimiter character. The <code>ValidationList</code> property returns the valid delimiters.
HSV_SECURITYEXTRACT_OPT_ROLE_ACCESS	Name property: <code>RoleAccess</code> Usage: Determines whether role access definitions will be extracted. Pass to CurrentValue: Boolean – TRUE to extract role access definitions, FALSE otherwise.
HSV_SECURITYEXTRACT_OPT_SECURITY_CLASS_ACCESS	Name property: <code>SecurityClassAccess</code> Usage: Determines whether security class access definitions will be extracted. Pass to CurrentValue: Boolean – TRUE to extract security access definitions, FALSE otherwise.
HSV_SECURITYEXTRACT_OPT_SECURITY_CLASSES	Name property: <code>SecurityClasses</code> Usage: Determines whether security classes will be extracted. Pass to CurrentValue: Boolean – TRUE to extract security classes, FALSE otherwise.
HSV_SECURITYEXTRACT_OPT_USERS	Name property: <code>Users</code> Usage: Determines whether usernames and user groups will be extracted. Pass to CurrentValue: Boolean – TRUE to extract usernames and user groups, FALSE otherwise.

HsvSecurityLoadACV Object Properties

The `HsvSecurityLoadACV` object provides the following properties:

- `ExtractOptions`. See [“ExtractOptions” on page 748](#).
- `LoadOptions`. See [“LoadOptions” on page 748](#).

These properties are members of several objects. The `HsvSecurityLoadACV` object has no additional properties.

Note: Use the `Set` keyword to assign `HsvSecurityLoadACV` object references.

HsvSecurityLoadACV Object Methods

The `HsvSecurityLoadACV` object provides the following methods:

- `Extract`. See [“Extract” on page 749](#).
- `Load`. See the following topic.
- `SetSession`. See [“SetSession” on page 751](#).

The `Extract` and `SetSession` methods are members of several objects, while the `Load` method is slightly different than the `Load` method for the other load- and extract-related objects.

Load

Loads security information from a text file into an application, with the text file being located on the client computer.

Caution! If you attempt to load security classes into a Performance Management Architect application, `Load` will fail.

You must call `SetSession` and set the `LoadOptions` property before calling `Load`. For information on how `Load` is used, see [“Loading Security Information” on page 753](#).

Syntax

```
<HsvSecurityLoadACV>.Load bstrClientFilename, bstrClientLogFileName,  
pvbWarnings
```

Argument	Description
<code>bstrClientFilename</code>	String (ByVal). The name and path of the file containing the security information to be loaded. The path must be valid for the client computer. For details on the file's required format, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i> .
<code>bstrClientLogFileName</code>	String (ByVal). The name and path of the log file for the load operation.
<code>pvbWarnings</code>	Boolean. Indicates whether warning messages were generated in the log file for a successful load. Returns TRUE if warnings were generated, FALSE otherwise. Note that if this returns TRUE, <code>Load</code> does not generate a trappable error. Tip: Warning messages are generated if the <code>ValidateUsers</code> load option is set to TRUE and the load file contains invalid usernames or user groups.

Example

`Load` is used in the [“Example for Loading Security” on page 754](#).

HsvMetadataLoadACV Type Library

To use the `HsvMetadataLoadACV` type library, you must reference `HsvMetadataLoadACV.dll` in your project.

Use the `HsvMetadataLoadACV` type library to load and extract metadata. This type library exposes the `HsvMetadataLoadACV` object, and also exposes the `IHsvLoadExtractOptions` and `IHsvLoadExtractOption` interfaces.

The following topics show you how to use the `HsvMetadataLoadACV` type library:

- [“IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvMetadataLoadACV” on page 759](#)
- [“Loading Metadata” on page 759](#)

- [“Extracting Metadata ” on page 760](#)
- [“Metadata Load Options” on page 761](#)
- [“Metadata Extract Options” on page 765](#)
- [“HsvMetadataLoadACV Object Properties” on page 768](#)
- [“HsvMetadataLoadACV Object Methods” on page 768](#)

Tip: To load metadata from and extract data to application servers, use the HsvMetadata type library; for details, see [“Load” on page 180](#) and [“Extract” on page 165](#). Note that the HsvMetadataLoadACV library provides properties and methods that simplify handling of the metadata load and extract options, while the HsvMetadata library exposes these options as arrays.

IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvMetadataLoadACV

The HsvMetadataLoadACV type library exposes the IHsvLoadExtractOptions and IHsvLoadExtractOption interfaces; these interfaces provide access to metadata load and extract options. For information on these interfaces, see [“Load and Extract Option Interfaces” on page 743](#).

► To set a metadata load or extract option:

- 1 **Set an IHsvLoadExtractOption object reference for the option with** `IHsvLoadExtractOptions.Item`.
- 2 **Set the option’s value with the** `IHsvLoadExtractOption.CurrentValue` **property.**

Tip: The options available for loading metadata are described in [“Metadata Load Options” on page 761](#), and the options available for extracting metadata are described in [“Metadata Extract Options” on page 765](#). These topics include the valid values for `Item` and `CurrentValue`.

Loading Metadata

The following steps provide an overview of how to load metadata into a Classic application. These steps assume that an application has previously been opened with one of the `OpenApplication` methods.

Caution! If you attempt to load metadata into a Performance Management Architect application, `Load` will fail.

For an example that illustrates these steps, see [“Example for Loading Metadata” on page 760](#).

- To load metadata into a Classic application:
 - 1 **Set an object reference to the HsvMetadataLoadACV object.**
 - 2 **Point to the application into which metadata is being loaded by calling `HsvMetadataLoadACV.SetSession`. For `SetSession`'s argument, pass the `HsvSession` object reference that was returned by `HsxClient.OpenApplication` or `HsxClientUI.OpenApplication`.**
 - 3 **Set an object reference to the `IHsvLoadExtractOptions` interface with the `HsvMetadataLoadACV.LoadOptions` property.**
 - 4 **Optional. All dimensions are loaded by default. To exclude a dimension from being loaded, set the `CurrentValue` property of the `IHsvLoadExtractOption` interface instance for the dimension to `FALSE`.**
 - 5 **Optional. To override the defaults for other load options, specify the values for these options. The available load options are listed in “[Metadata Load Options](#)” on page 761.**
 - 6 **Load the data by calling `HsvMetadataLoadACV.Load`. `Load` takes the file names and paths of the metadata load file and of the log file.**

Example for Loading Metadata

The following subroutine loads metadata from an XML file. The example loads metadata for all dimensions, replacing any existing metadata.

```
Sub LoadMetaACV(sFile As String, sLog As String)
Dim cMetadataLoadACV As HsvMetadataLoadACV
Dim cOptions As IHsvLoadExtractOptions
Dim cOpt As IHsvLoadExtractOption
Set cMetadataLoadACV = New HsvMetadataLoadACV
'g_cSession is an HsvSession object reference
cMetadataLoadACV.SetSession g_cSession
'Initialize the IHsvLoadExtractOptions interface.
Set cOptions = cMetadataLoadACV.LoadOptions
'Turn on Replace mode
Set cOpt = cOptions.Item(HSV_METALOAD_OPT_USE_REPLACE_MODE)
cOpt.CurrentValue = False
'Load from an XML file
Set cOpt = cOptions.Item(HSV_METALOAD_OPT_FILE_FORMAT)
cOpt.CurrentValue = HSV_METALOAD_EX_FORMAT_XML
'Load the Metadata
cMetadataLoadACV.Load sFile, sLog
End Sub
```

Extracting Metadata

The following steps provide an overview of how to extract metadata. These steps assume that an application has previously been opened with one of the `OpenApplication` methods.

Tip: For an example that illustrates these steps, see “[Example for Extracting Metadata](#)” on page 761.

- To extract metadata:
 - 1 **Set an object reference to the HsvMetadataLoadACV object.**
 - 2 **Point to the application from which data is being extracted by calling `HsvMetadataLoadACV.SetSession`. For `SetSession`'s argument, pass the `HsvSession` object reference that was returned by `HsxClient.OpenApplication` or `HsxClientUI.OpenApplication`.**
 - 3 **Set an object reference to the `IHsvLoadExtractOptions` interface with the `HsvMetadataLoadACV.ExtractOptions` property.**
 - 4 **Optional. All dimensions are extracted by default. To exclude a dimension from being extracted, set the `CurrentValue` property of the `IHsvLoadExtractOption` interface instance for the dimension to `FALSE`.**
 - 5 **Optional. To override the defaults for other extract options, specify the values for these options. The available extract options are listed in [“Metadata Extract Options” on page 765](#).**
 - 6 **Extract the metadata by calling `HsvMetadataLoadACV.Extract`. `Extract` takes the file names and paths of the metadata extract file and of the log file.**

Example for Extracting Metadata

The following example extracts metadata into an XML file. Note that application settings are not extracted.

```
Dim cMetadataLoadACV As HsvMetadataLoadACV
Dim cOptions As IHsvLoadExtractOptions
Dim cOpt As IHsvLoadExtractOption
Set cMetadataLoadACV = New HsvMetadataLoadACV
'Specify the HsvSession object for the application.
cMetadataLoadACV.SetSession g_cHsvSession
'Initialize the IHsvLoadExtractOptions interface.
Set cOptions = cMetadataLoadACV.ExtractOptions
'Exclude application settings
Set cOpt = cOptions.Item(HSV_METAEXTRACT_OPT_APP_SETTINGS)
cOpt.CurrentValue = False
'Extract as an XML file
Set cOpt = cOptions.Item(HSV_METAEXTRACT_OPT_FILE_FORMAT)
cOpt.CurrentValue = HSV_METAEXTRACT_FORMAT_XML
'Extract the Metadata
cMetadataLoadACV.Extract "c:\Acme\myApp.xml", _
    "c:\Acme\myApp.log"
```

Metadata Load Options

For each metadata load option, the following table lists the constant that identifies the option and the corresponding `Name` property — either the constant or the name can be passed to `IHsvLoadExtractOptions.Item`. The table also describes each option and the type of information to be passed to `IHsvLoadExtractOption.CurrentValue`. (See [“Item” on page 744](#) and [“CurrentValue” on page 745](#).)

Tip: The options for loading dimension members default to TRUE, and the options for clearing previously loaded members default to FALSE.

Table 76 Metadata Load Options

Constant	Load Option Information
HSV_METALOAD_ OPT_ACCOUNTS	<p>Name property: Accounts</p> <p>Usage: Specifies whether Account dimension members are loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load accounts, otherwise FALSE.</p>
HSV_METALOAD_ OPT_ACCOUNTS_ SYSTEM	<p>Name property: SystemAccounts</p> <p>Usage: Specifies whether system accounts are loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load system accounts, otherwise FALSE. These are system members, so if this option is set to TRUE you must also set the LoadSystemMembers option to TRUE.</p> <p>By default, this option is set to FALSE.</p>
HSV_METALOAD_ OPT_APP_SETTINGS	<p>Name property: AppSettings</p> <p>Usage: Specifies whether application settings are loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load application settings, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CHECK_ INTEGRITY	<p>Name property: CheckIntegrity</p> <p>Usage: Specifies whether to validate the integrity of the metadata file against the metadata in the current application.</p> <p>Note: If integrity errors occur, they are noted in the log file and no portion of the file is loaded into the application.</p> <p>Pass to CurrentValue: Boolean – TRUE to check integrity, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CLEAR_ ACCOUNTS	<p>Name property: ClearAccounts</p> <p>Usage: Specifies whether previously loaded Account dimension members are deleted before metadata is loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to delete previously loaded accounts, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CLEAR_CONSOL_ METHODS	<p>Name property: ClearConsolMethods</p> <p>Usage: Specifies whether existing consolidation methods are cleared before metadata is loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to delete previously loaded consolidation methods, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CLEAR_ CURRENCIES	<p>Name property: ClearCurrencies</p> <p>Usage: Specifies whether previously loaded currencies are deleted before metadata is loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to delete previously loaded currencies, otherwise FALSE.</p>

Constant	Load Option Information
HSV_METALOAD_ OPT_CLEAR_ CUSTOM1	<p>Name property: ClearCustom1</p> <p>Usage: Specifies whether previously loaded Custom 1 dimension members are deleted before metadata is loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to delete previously loaded Custom 1 members, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CLEAR_ CUSTOM2	<p>Name property: ClearCustom2</p> <p>Usage: Specifies whether previously loaded Custom 2 dimension members are deleted before metadata is loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to delete previously loaded Custom 2 members, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CLEAR_ CUSTOM3	<p>Name property: ClearCustom3</p> <p>Usage: Specifies whether previously loaded Custom 3 dimension members are deleted before metadata is loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to delete previously loaded Custom 3 members, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CLEAR_ CUSTOM4	<p>Name property: ClearCustom4</p> <p>Usage: Specifies whether previously loaded Custom 4 dimension members are deleted before metadata is loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to delete previously loaded Custom 4 members, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CLEAR_ENTITIES	<p>Name property: ClearEntities</p> <p>Usage: Specifies whether previously loaded Entity dimension members are deleted before metadata is loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to delete previously loaded entities, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CLEAR_ SCENARIOS	<p>Name property: ClearScenarios</p> <p>Usage: Specifies whether previously loaded Scenario dimension members are deleted before metadata is loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to delete previously loaded scenarios, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CONSOL_ METHODS	<p>Name property: ConsolMethods</p> <p>Usage: Specifies whether consolidation methods are loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load consolidation methods, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CURRENCIES	<p>Name property: Currencies</p> <p>Usage: Specifies whether currencies are loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load currencies, otherwise FALSE.</p>
HSV_METALOAD_ OPT_CUSTOM1	<p>Name property: Custom1</p> <p>Usage: Specifies whether Custom 1 dimension members are loaded.</p>

Constant	Load Option Information
	Pass to CurrentValue: Boolean – TRUE to load Custom 1 dimension members, otherwise FALSE.
HSV_METALOAD_ OPT_CUSTOM2	Name property: Custom2 Usage: Specifies whether Custom 2 dimension members are loaded. Pass to CurrentValue: Boolean – TRUE to load Custom 2 dimension members, otherwise FALSE.
HSV_METALOAD_ OPT_CUSTOM3	Name property: Custom3 Usage: Specifies whether Custom 3 dimension members are loaded. Pass to CurrentValue: Boolean – TRUE to load Custom 3 dimension members, otherwise FALSE.
HSV_METALOAD_ OPT_CUSTOM4	Name property: Custom4 Usage: Specifies whether Custom 4 dimension members are loaded. Pass to CurrentValue: Boolean – TRUE to load Custom 4 dimension members, otherwise FALSE.
HSV_METALOAD_ OPT_DELIMITER_CHAR	Name property: Delimiter Usage: Specifies a load file's delimiter. Pass to CurrentValue: String – a valid delimiter character. Note that the <code>ValidationList</code> property returns the valid delimiters.
HSV_METALOAD_ OPT_ENTITIES	Name property: Entities Usage: Specifies whether Entity dimension members are loaded. Pass to CurrentValue: Boolean – TRUE to load entities, otherwise FALSE.
HSV_METALOAD_ OPT_FILE_FORMAT	Name property: FileFormat Usage: Specifies whether the metadata load file is in an ASCII text or XML format. Pass to CurrentValue: Specify one of the constants in Table 78 on page 768 . By default, ASCII text files are loaded.
HSV_METALOAD_ OPT_ICP	Name property: ICPs Usage: Specifies whether Intercompany Partner dimension members are loaded. Pass to CurrentValue: Boolean – TRUE to load Intercompany Partner dimension members, otherwise FALSE. These are system members, so if this option is set to TRUE you must also set the LoadSystemMembers option to TRUE.
HSV_METALOAD_ OPT_LOAD_SYSTEM_ MEMBERS	Name property: LoadSystemMembers Usage: Specifies whether system members are loaded. Pass to CurrentValue: Boolean – TRUE to load system members, otherwise FALSE. Note: If this option is set to TRUE, you must also set to TRUE the options for the system members to be loaded.
HSV_METALOAD_ OPT_LOG_FILE_ APPEND	<i>For internal use.</i>

Constant	Load Option Information
HSV_METALOAD_OPT_MAX	<i>For internal use.</i>
HSV_METALOAD_OPT_MIN	<i>For internal use.</i>
HSV_METALOAD_OPT_PRESCAN	<p>Name property: Prescan</p> <p>Usage: Specifies whether a load file is loaded or is merely scanned for syntax accuracy when <code>Load</code> is called.</p> <p>Pass to CurrentValue: Boolean – TRUE to scan without loading, FALSE to load the metadata. By default, this option is set to FALSE.</p>
HSV_METALOAD_OPT_SCENARIOS	<p>Name property: Scenarios</p> <p>Usage: Specifies whether Scenario dimension members are loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load scenarios, otherwise FALSE.</p>
HSV_METALOAD_OPT_USE_REPLACE_MODE	<p>Name property: UseReplaceMode</p> <p>Usage: Specifies whether the metadata replaces or is merged with existing metadata.</p> <p>Pass to CurrentValue: Boolean – TRUE to replace existing metadata, FALSE to merge with existing metadata. Note that this option is set to FALSE by default.</p>
HSV_METALOAD_OPT_VALUE	<p>Name property: Values</p> <p>Usage: Specifies whether Value dimension members are loaded.</p> <p>Pass to CurrentValue: Boolean – TRUE to load Value dimension members, otherwise FALSE. These are system members, so if this option is set to TRUE you must also set the <code>LoadSystemMembers</code> option to TRUE.</p>

Metadata Extract Options

For each metadata extract option, the following table lists the constant that identifies the option and the corresponding Name property — either the constant or the name can be passed to `IHsvLoadExtractOptions.Item`. The table also describes each option and the type of information to be passed to `IHsvLoadExtractOption.CurrentValue`; see “Item” on page 744 and “CurrentValue” on page 745.

Tip: The options for extracting dimension members default to TRUE. An exception is the option for system-generated accounts, which defaults to FALSE.

Table 77 Metadata Extract Options

Constant	Extract Option Information
HSV_METAEXTRACT_OPT_ACCOUNTS	<p>Name property: Accounts</p> <p>Usage: Specifies whether Account dimension members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract accounts, otherwise FALSE.</p>

Constant	Extract Option Information
HSV_METAEXTRACT_OPT_ACCOUNTS_SYSTEM	<p>Name property: SystemAccounts</p> <p>Usage: Specifies whether system-generated Account dimension members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract system accounts, otherwise FALSE. If this option is set to TRUE you must also set the ExtractSystemMembers option to TRUE.</p>
HSV_METAEXTRACT_OPT_APP_SETTINGS	<p>Name property: AppSettings</p> <p>Usage: Specifies whether application settings are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract application settings, otherwise FALSE.</p>
HSV_METAEXTRACT_OPT_CONSOL_METHODS	<p>Name property: ConsolMethods</p> <p>Usage: Specifies whether consolidation methods are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract consolidation methods, otherwise FALSE.</p>
HSV_METAEXTRACT_OPT_CURRENCIES	<p>Name property: Currencies</p> <p>Usage: Specifies whether currencies are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract currencies, otherwise FALSE.</p>
HSV_METAEXTRACT_OPT_CUSTOM1	<p>Name property: Custom1</p> <p>Usage: Specifies whether Custom 1 dimension members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract Custom 1 members, otherwise FALSE.</p>
HSV_METAEXTRACT_OPT_CUSTOM2	<p>Name property: Custom2</p> <p>Usage: Specifies whether Custom 2 dimension members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract Custom 2 members, otherwise FALSE.</p>
HSV_METAEXTRACT_OPT_CUSTOM3	<p>Name property: Custom3</p> <p>Usage: Specifies whether Custom 3 dimension members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract Custom 3 members, otherwise FALSE.</p>
HSV_METAEXTRACT_OPT_CUSTOM4	<p>Name property: Custom4</p> <p>Usage: Specifies whether Custom 4 dimension members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract Custom 4 members, otherwise FALSE.</p>
HSV_METAEXTRACT_OPT_DELIMITER_CHAR	<p>Name property: Delimiter</p> <p>Usage: Specifies an extract file's delimiter.</p> <p>Pass to CurrentValue: String – a valid delimiter character. The <code>ValidationList</code> property returns the valid delimiters.</p>
HSV_METAEXTRACT_OPT_ENTITIES	<p>Name property: Entities</p> <p>Usage: Specifies whether Entity dimension members are extracted.</p>

Constant	Extract Option Information
	Pass to CurrentValue: Boolean – TRUE to extract entities, otherwise FALSE.
HSV_METAEXTRACT_ OPT_EXTRACT_SYSTEM_ MEMBERS	<p>Name property: ExtractSystemMembers</p> <p>Usage: Specifies whether system members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract system members, otherwise FALSE.</p> <p>Note: If this option is set to TRUE, you must also set to TRUE the options for the system members to be extracted.</p>
HSV_METAEXTRACT_ OPT_FILE_FORMAT	<p>Name property: FileFormat</p> <p>Usage: Specifies whether the metadata extract file is in an ASCII text or XML format.</p> <p>Pass to CurrentValue: Specify one of the constants in Table 78 on page 768. By default, metadata is extracted to ASCII text files.</p>
HSV_METAEXTRACT_ OPT_ICP	<p>Name property: ICPs</p> <p>Usage: Specifies whether Intercompany Partner dimension members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract Intercompany Partner dimension members, otherwise FALSE. These are system members, so if this option is set to TRUE you must also set the ExtractSystemMembers option to TRUE.</p>
HSV_METAEXTRACT_ OPT_MAX	<i>For internal use.</i>
HSV_METAEXTRACT_ OPT_MIN	<i>For internal use.</i>
HSV_METAEXTRACT_ OPT_PERIODS	<p>Name property: Periods</p> <p><i>For internal use.</i></p>
HSV_METAEXTRACT_ OPT_SCENARIOS	<p>Name property: Scenarios</p> <p>Usage: Specifies whether Scenario dimension members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract scenarios, otherwise FALSE.</p>
HSV_METAEXTRACT_ OPT_VALUE	<p>Name property: Values</p> <p>Usage: Specifies whether Value dimension members are extracted.</p> <p>Pass to CurrentValue: Boolean – TRUE to extract Value dimension members, otherwise FALSE. These are system members, so if this option is set to TRUE you must also set the ExtractSystemMembers option to TRUE.</p>
HSV_METAEXTRACT_ OPT_VIEWS	<p>Name property: Views</p> <p><i>For internal use.</i></p>
HSV_METAEXTRACT_ OPT_YEARS	<p>Name property: Years</p> <p><i>For internal use.</i></p>

Metadata File Format Constants

To specify the format of a metadata load or extract file, set the `CurrentValue` property of the “FileFormat” load or extract option to one of the constants listed in the following table.

Table 78 Metadata File Format Constants

Constant	Description
HSV_METALOADEX_FORMAT_NATIVE	Metadata is to be loaded or extracted in an ASCII text file.
HSV_METALOADEX_FORMAT_XML	Metadata is to be loaded or extracted in an XML text file.
HSV_METALOADEX_FORMAT_XML_COMMONSHEMA	<i>For internal use.</i>

For examples that use the HSV_METALOADEX_FORMAT_XML constant, see “[Example for Loading Metadata](#)” on page 760 and “[Example for Extracting Metadata](#)” on page 761.

HsvMetadataLoadACV Object Properties

The HsvMetadataLoadACV object provides the following properties:

- [ExtractOptions](#)
- [LoadOptions](#)

These properties are members of several objects. The HsvMetadataLoadACV object has no additional properties.

Note: Use the `Set` keyword to assign HsvMetadataLoadACV object references.

HsvMetadataLoadACV Object Methods

The HsvMetadataLoadACV object provides the following methods:

- [Extract](#)
- [Load](#) (Classic applications only)

Caution! If you attempt to load metadata into a Performance Management Architect application, `Load` will fail.

- [SetSession](#)

These methods are members of several objects. The HsvMetadataLoadACV object has no additional methods.

HsvRulesLoadACV Type Library

To use the HsvRulesLoadACV type library, you must reference `HsvRulesLoadACV.dll` in your project.

The HsvRulesLoadACV type library contains one object—the HsvRulesLoadACV object.

HsvRulesLoadACV Object Methods

The HsvRulesLoadACV object provides methods for loading and extracting rules and member lists. These methods are summarized in [Table 49 on page 113](#), and are described in detail in the following topics.

Note: Use the `Set` keyword to assign HsvRulesLoadACV object references.

ExtractCalcRules

Extracts rules from an application into a text file on the client computer.

Note: You must call [SetSession](#) before calling `ExtractCalcRules`. `SetSession` points to the application from which rules are being extracted.

Syntax

```
<HsvRulesLoadACV>.ExtractCalcRules bstrClientFilename, bstrLogFilename
```

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

<code>bstrClientFilename</code>	String (ByVal). The name and path of the file into which the rules will be extracted. The path must exist on the client computer.
---------------------------------	---

For details on the extracted file's format, see the *Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide*.

<code>bstrLogFilename</code>	String (ByVal). The name and path of the log file for the extraction operation.
------------------------------	---

Example

The following example extracts rules from an application.

```
Dim cHsvRulesLoadACV As HsvRulesLoadACV, bSession As Boolean
Set cHsvRulesLoadACV = New HsvRulesLoadACV
bSession = cHsvRulesLoadACV.SetSession(m_cHsvSession, _
HFM_LANGUAGE_INSTALLED)
If bSession = True Then
    cHsvRulesLoadACV.ExtractCalcRules "c:\Acme\myApp.rle", _
    "c:\Acme\myApp.log"
End If
```

ExtractMemberListRules

Extracts member lists from an application into a text file on the client computer.

Note: You must call [SetSession](#) before calling `ExtractMemberListRules`. `SetSession` points to the application from which member lists are being extracted.

Syntax

```
<HsvRulesLoadACV>.ExtractMemberListRules bstrClientFilename,  
bstrLogFilename
```

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

<i>bstrClientFilename</i>	String (ByVal). The name and path of the file into which the member lists will be extracted. The path must exist on the client computer. For details on the extracted file's format, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i> .
---------------------------	---

<i>bstrLogFilename</i>	String (ByVal). The name and path of the log file for the extraction operation.
------------------------	---

Example

The following example extracts member lists from an application.

```
Dim cHsvRulesLoadACV As HsvRulesLoadACV, bSession As Boolean  
Set cHsvRulesLoadACV = New HsvRulesLoadACV  
bSession = cHsvRulesLoadACV.SetSession(m_cHsvSession, _  
HFM_LANGUAGE_INSTALLED)  
If bSession = True Then  
    cHsvRulesLoadACV.ExtractMemberListRules _  
        "c:\Acme\myApp.lst", "c:\Acme\myApp.log"  
End If
```

GetCalcRulesType

For internal use.

LoadCalcRules

Scans or loads a rules file from a client computer.

LoadCalcRules returns Booleans that indicate whether validation errors, validation warnings, and validation information were included in the log file.

To validate whether rules in a file violate the referential integrity of any intercompany transactions, use [LoadCalcRules2](#).

Note: You must call [SetSession](#) before calling LoadCalcRules. SetSession points to the application for which rules are being loaded.

Syntax

```
<HsvRulesLoadACV>.LoadCalcRules bstrClientFilename, bstrLogFilename,  
vbScanOnly, pvbErrorsWereFound, pvbWarningsWereFound, pvbInfoWasProvided
```

Argument	Description
<i>bstrClientFilename</i>	String (ByVal). The name and path of the file containing the rules to be loaded or scanned. The path must be valid for the client computer. For details on the file's required format, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i> .
<i>bstrLogFilename</i>	String (ByVal). The name and path of the log file for the load or scan operation.
<i>vbScanOnly</i>	Boolean (ByVal). A flag that specifies whether to load or scan the rules file. Pass TRUE to scan, FALSE to load.
<i>pvbErrorsWereFound</i>	Boolean. Indicates whether <code>LoadCalcRules</code> included any validation errors in the log file. Returns TRUE if errors were found, FALSE otherwise.
<i>pvbWarningsWereFound</i>	Boolean. Indicates whether <code>LoadCalcRules</code> included any validation warnings in the log file. Returns TRUE if warnings were found, FALSE otherwise.
<i>pvbInfoWasProvided</i>	Boolean. Indicates whether <code>LoadCalcRules</code> included any information not classified as errors or warnings in the log file. Returns TRUE if non-error and non-warning information was included, FALSE otherwise.

Example

The following example loads rules into an application.

```
Dim cHsvRulesLoadACV As HsvRulesLoadACV, bErr As Boolean
Dim bSession As Boolean, bWarn As Boolean, bInfo As Boolean
Set cHsvRulesLoadACV = New HsvRulesLoadACV
bSession = cHsvRulesLoadACV.SetSession(m_chsvSession, _
HFM_LANGUAGE_INSTALLED)
If bSession = True Then
    cHsvRulesLoadACV.LoadCalcRules "c:\Acme\myApp.rle", _
    "c:\Acme\myApp.log", False, bErr, bWarn, bInfo
End If
```

LoadCalcRules2

Scans or loads a rules file from a client computer, optionally validating whether the rules violate the referential integrity of any intercompany transactions.

`LoadCalcRules2` returns Booleans that indicate whether validation errors, validation warnings, and validation information were included in the log file.

Note: You must call `SetSession` before calling `LoadCalcRules2`. `SetSession` points to the application for which rules are being loaded.

Syntax

```
<HsvRulesLoadACV>.LoadCalcRules2 bstrClientFilename, bstrLogFilename,
vbScanOnly, vbCheckRefInteg, pvbErrorsWereFound, pvbWarningsWereFound,
pvbInfoWasProvided
```

Argument	Description
<i>bstrClientFilename</i>	String (ByVal). The name and path of the file containing the rules to be loaded or scanned. The path must be valid for the client computer. For details on the file's required format, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i> .
<i>bstrLogFilename</i>	String (ByVal). The name and path of the log file for the load or scan operation.
<i>vbScanOnly</i>	Boolean (ByVal). A flag that specifies whether to load or scan the rules file. Pass TRUE to scan, FALSE to load.
<i>vbCheckRefInteg</i>	Boolean (ByVal). A flag that specifies whether to validate the referential integrity of intercompany transactions. Pass TRUE to validate intercompany transactions, FALSE otherwise.
<i>pbErrorsWereFound</i>	Boolean. Indicates whether <code>LoadCalcRules2</code> included any validation errors in the log file. Returns TRUE if errors were found, FALSE otherwise.
<i>pbWarningsWereFound</i>	Boolean. Indicates whether <code>LoadCalcRules2</code> included any validation warnings in the log file. Returns TRUE if warnings were found, FALSE otherwise.
<i>pvbInfoWasProvided</i>	Boolean. Indicates whether <code>LoadCalcRules2</code> included any information not classified as errors or warnings in the log file. Returns TRUE if non-error and non-warning information was included, FALSE otherwise.

LoadMemberListRules

Validates and loads a member lists file from a client computer. A flag determines whether `LoadMemberListRules` loads after validation or validates without loading.

`LoadMemberListRules` returns Booleans that indicate whether validation errors, validation warnings, and validation information were included in the log file.

Note: You must call `SetSession` before calling `LoadMemberListRules`. `SetSession` points to the application for which member lists are being loaded.

Syntax

```
<HsvRulesLoadACV>.LoadMemberListRules bstrClientFilename, bstrLogFilename,
vbScanOnly, pvbErrorsWereFound, pvbWarningsWereFound, pvbInfoWasProvided
```

Argument	Description
<i>bstrClientFilename</i>	String (ByVal). The name and path of the file containing the member lists to be loaded. The path must be valid for the client computer. For details on the file's required format, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i> .
<i>bstrLogFilename</i>	String (ByVal). The name and path of the log file for the load or validate operation.
<i>vbScanOnly</i>	Boolean (ByVal). Determines whether <code>LoadMemberListRules</code> loads after validating. Pass TRUE to load after validation, FALSE to validate without loading.

Argument	Description
<i>pvbErrorsWereFound</i>	Boolean. Indicates whether <code>LoadMemberListRules</code> included any validation errors in the log file. Returns TRUE if errors were found, FALSE otherwise.
<i>pvbWarningsWereFound</i>	Boolean. Indicates whether <code>LoadMemberListRules</code> included any validation warnings in the log file. Returns TRUE if warnings were found, FALSE otherwise.
<i>pvbInfoWasProvided</i>	Boolean. Indicates whether <code>LoadMemberListRules</code> included any information not classified as errors or warnings in the log file. Returns TRUE if non-error and non-warning information was included, FALSE otherwise.

Example

The following example loads member lists into an application.

```
Dim cHsvRulesLoadACV As HsvRulesLoadACV, bErr As Boolean
Dim bSession As Boolean, bWarn As Boolean, bInfo As Boolean
Set cHsvRulesLoadACV = New HsvRulesLoadACV
bSession = cHsvRulesLoadACV.SetSession(m_cHsvSession, _
HFM_LANGUAGE_INSTALLED)
If bSession = True Then
    cHsvRulesLoadACV.LoadMemberListRules "c:\Acme\myApp.lst", _
    "c:\Acme\myApp.log", False, bErr, bWarn, bInfo
End If
```

ScriptableLoadCalcRules

For internal use.

ScriptableLoadCalcRules2

For internal use.

ScriptableLoadMemberListRules

For internal use.

SetSession

Points to the `HsvSession` object for the application in which rules or member lists are being loaded or extracted.

Caution! You must call `SetSession` before calling any of the other `HsvRulesLoadACV` object's properties or methods, otherwise an error will occur.

Syntax

```
<HsvRulesLoadACV>.SetSession(pIHsvSessionUnk, lLanguageID)
```

Argument	Description
<i>pHsvSessionUnk</i>	HsvSession object (ByVal). The HsvSession object that was returned by <code>HsxClient.OpenApplication</code> or <code>HsxClientUI.OpenApplication</code> when the application was opened. For information on <code>HsxClient.OpenApplication</code> , see “OpenApplication” on page 140 ; for information on <code>HsxClientUI.OpenApplication</code> , see “OpenApplication” on page 149 .
<i>lLanguageID</i>	Long (ByVal). Identifies the language in which any error messages should be displayed. Pass one of the HFMConstants type library constants listed in “Supported Language Constants” on page 833 . Tip: To have error messages display in the installed language, pass the <code>HFM_LANGUAGE_INSTALLED</code> constant.

Return Value

Boolean. Returns TRUE if `SetSession` succeeds, FALSE otherwise.

Example

`SetSession` is used in the [Example](#) for `LoadMemberListRules`.

SetSessionAndResource

Specifies a reference to a `HsvResourceManager` object.

Syntax

```
<HsvRulesLoadACV>.SetSessionAndResource (pIHsvSessionUnk,
pIUnkResourceManager, lLanguageID)
```

Argument	Description
<i>pHsvSessionUnk</i>	HsvSession object (ByVal). The HsvSession object that was returned by <code>HsxClient.OpenApplication</code> or <code>HsxClientUI.OpenApplication</code> when the application was opened. For information on <code>HsxClient.OpenApplication</code> , see “OpenApplication” on page 140 ; for information on <code>HsxClientUI.OpenApplication</code> , see “OpenApplication” on page 149 .
<i>pIUnkResourceManager</i>	Object (ByVal). The <code>HsvResourceManager</code> object to reference.
<i>lLanguageID</i>	Long (ByVal). Specifies the language in which any error messages are displayed. Pass one of the HFMConstants type library constants listed in “Supported Language Constants” on page 833 . Tip: To display error messages in the installed language, pass the <code>HFM_LANGUAGE_INSTALLED</code> constant.

Return Value

Boolean. Returns TRUE if `SetSessionAndResource` succeeds, FALSE otherwise.

HsvcDataLoad Type Library

To use the HsvcDataLoad type library, you must reference `HsvcDataLoad.dll` in your project.

Use the HsvcDataLoad type library to load from and extract data to client computers. This type library exposes the HsvcDataLoad object, and also exposes the IHsvLoadExtractOptions and IHsvLoadExtractOption interfaces.

The following topics show you how to use the HsvcDataLoad type library:

- [“IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvcDataLoad” on page 775](#)
- [“Loading Data” on page 776](#)
- [“Extracting Data” on page 777](#)
- [“Data Load Options” on page 778](#)
- [“Data Extract Options” on page 781](#)
- [“HsvcDataLoad Object Properties” on page 783](#)
- [“HsvcDataLoad Object Methods” on page 783](#)

Tip: To load data from and extract data to application servers, use the HsvData type library; for details, see [“Load” on page 353](#) and [“Extract” on page 295](#). Note that the HsvcDataLoad library provides properties and methods that simplify handling of the data load and extract options, while the HsvData library exposes these options as arrays.

IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvcDataLoad

The HsvcDataLoad type library exposes the IHsvLoadExtractOptions and IHsvLoadExtractOption interfaces; these interfaces provide access to data load and extract options. For information on these interfaces, see [“Load and Extract Option Interfaces” on page 743](#).

► To set a data load or extract option:

- 1 **Set an IHsvLoadExtractOption object reference for the option with `IHsvLoadExtractOptions.Item`.**
- 2 **Set the option’s value with the `IHsvLoadExtractOption.CurrentValue` property.**

Tip: The options available for loading data are described in [“Data Load Options” on page 778](#), and the options available for extracting data are described in [“Data Extract Options” on page 781](#). These topics include the valid values for `Item` and `CurrentValue`.

Loading Data

The following steps provide an overview of how to load data. These steps assume that an application has previously been opened with one of the `OpenApplication` methods.

Tip: For an example that illustrates these steps, see [“Example for Loading Data” on page 776](#).

► To load data:

- 1 Set an object reference to the `HsvcDataLoad` object.
- 2 Point to the application into which data is being loaded by calling `HsvcDataLoad.SetSession`. For `SetSession`'s argument, pass the `HsvSession` object reference that was returned by `HsxClient.OpenApplication` or `HsxClientUI.OpenApplication`.
- 3 Set an object reference to the `IHsvLoadExtractOptions` interface with the `HsvcDataLoad.LoadOptions` property.
- 4 To load data, you must override the `Mode` load option's default setting; by default, data load files are scanned for syntax errors without being loaded. To override this default setting, perform the following steps:
 - a. Pass the `HSV_DATALOAD_OPT_MODE` constant to `Item` to return an `IHsvLoadExtractOption` object reference for the `Mode` load option.
 - b. Set the `IHsvLoadExtractOption` interface instance's `CurrentValue` property to `HSV_DATALOAD_LOAD`.
- 5 **Optional.** To override the defaults for other load options, specify the values for these options. The available load options are listed in [Table 79 on page 779](#).
- 6 Load the data by calling `HsvcDataLoad.Load` or `HsvcDataLoad.Load2`. Both methods take the file names and paths of the data load file and of the log file; `Load2` also returns a flag that indicates whether loading errors were logged.

Example for Loading Data

The following example loads data with `HsvcDataLoad.Load`, with the `Duplicates` option set to `accumulate`.

```
Dim cHsvcDataLoad As HsvcDataLoad, cOptions As IHsvLoadExtractOptions
Dim cOpt As IHsvLoadExtractOption
Set cHsvcDataLoad = New HsvcDataLoad
'Specify the HsvSession object for the application.
cHsvcDataLoad.SetSession g_chsvSession
'Initialize the IHsvLoadExtractOptions interface.
Set cOptions = cHsvcDataLoad.LoadOptions
'Set the option for loading data.
Set cOpt = cOptions.Item(HSV_DATALOAD_OPT_MODE)
cOpt.CurrentValue = HSV_DATALOAD_LOAD
'Set the option for accumulate mode.
Set cOpt = cOptions.Item(HSV_DATALOAD_OPT_DUPLICATES)
cOpt.CurrentValue = HSV_DATALOAD_ACCUMULATE
'Load the data
cHsvcDataLoad.Load "c:\Acme\myApp.dat", "c:\Acme\myApp.log"
```


Extracting Data

The following steps provide an overview of how to extract data. These steps assume that an application has previously been opened with one of the `OpenApplication` methods.

Tip: For an example that illustrates these steps, see [“Example for Extracting Data” on page 777](#).

► To extract data:

- 1 **Set an object reference to the `HsvcDataLoad` object.**
- 2 **Point to the application from which data is being extracted by calling `HsvcDataLoad.SetSession`.** For `SetSession`'s argument, pass the `HsvSession` object reference that was returned by `HsxClient.OpenApplication` or `HsxClientUI.OpenApplication`.
- 3 **Set an object reference to the `IHsvLoadExtractOptions` interface with the `HsvcDataLoad.ExtractOptions` property.**
- 4 **Specify the dimension members for which data is being extracted by setting the following extract options. You are *required* to specify these options:**
 - The dimension member ID of the scenario.
 - The dimension member ID of the year.
 - The dimension member IDs of the periods.
 - The dimension member IDs of the accounts.
 - The dimension member IDs of the child entities.
- 5 **Optional.** To override the defaults for other extract options, specify the values for these options. The available extract options are listed in [Table 80 on page 781](#).
- 6 **Extract the data by calling `HsvcDataLoad.Extract`.** `Extract` takes the file names and paths of the data extract file and of the log file.

Example for Extracting Data

The following example extracts data for the following dimension members:

- Scenario = Actual
- Year = 2000
- Periods = July, August, September
- Account = Sales
- Child entity = NewYork
- Parent entity = UnitedStates

Note: The example uses a user-defined function named `GetMemberID` to obtain the IDs of these members. For information on this user-defined function, see the [Examples](#) for `GetItemID`.

```
Dim cHsvcDataLoad As HsvcDataLoad, cOptions As IHsvLoadExtractOptions
Dim cOpt As IHsvLoadExtractOption, lScen As Long, lYear As Long
Dim lPer(2) As Long, lAcct(0) As Long, lPar(0) As Long, lEnt(0) As Long
Set cHsvcDataLoad = New HsvcDataLoad
'Specify the HsvSession object for the application.
cHsvcDataLoad.SetSession g_cHsvSession
'Initialize the IHsvLoadExtractOptions interface.
Set cOptions = cHsvcDataLoad.ExtractOptions
'Set the scenario
Set cOpt = cOptions.Item(HSV_DATAEXTRACT_OPT_SCENARIO_SUBSET)
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
cOpt.CurrentValue = lScen
'Set the year
Set cOpt = cOptions.Item(HSV_DATAEXTRACT_OPT_YEAR_SUBSET)
lYear = GetMemberID(DIMENSIONYEAR, "2000")
cOpt.CurrentValue = lYear
'Set the periods
Set cOpt = cOptions.Item(HSV_DATAEXTRACT_OPT_PERIOD_SUBSET)
lPer(0) = GetMemberID(DIMENSIONPERIOD, "July")
lPer(1) = GetMemberID(DIMENSIONPERIOD, "August")
lPer(2) = GetMemberID(DIMENSIONPERIOD, "September")
cOpt.CurrentValue = lPer
'Set the account
Set cOpt = cOptions.Item(HSV_DATAEXTRACT_OPT_ACCOUNT_SUBSET)
lAcct(0) = GetMemberID(DIMENSIONACCOUNT, "Sales")
cOpt.CurrentValue = lAcct
'set the entities
Set cOpt = cOptions.Item(HSV_DATAEXTRACT_OPT_PARENT_SUBSET)
lPar(0) = GetMemberID(DIMENSIONENTITY, "UnitedStates")
cOpt.CurrentValue = lPar
Set cOpt = cOptions.Item(HSV_DATAEXTRACT_OPT_ENTITY_SUBSET)
lEnt(0) = GetMemberID(DIMENSIONENTITY, "NewYork")
cOpt.CurrentValue = lEnt
'Extract the data
cHsvcDataLoad.Extract "c:\Acme\myApp.dat", "c:\Acme\myApp.log"
```

Data Load Options

For each data load option, the following table lists the constant that identifies the option and the corresponding `Name` property — either the constant or the name can be passed to `IHsvLoadExtractOptions.Item`. The table also describes each option and the type of information to be passed to `IHsvLoadExtractOption.CurrentValue`.

Table 79 Data Load Options

Constant	Load Option Information
HSV_DATALOAD_ OPT_ACCUMULATE_ WITHIN_FILE	<p>Name property: Accumulate within file</p> <p>Usage: Specifies whether multiple values for the same cell within the load file should be accumulated or allowed to overwrite each other.</p> <p>If the load file does not contain multiple values for the same cell, then this option has no effect. In addition, this load option applies to line items and values, but not to descriptions – multiple descriptions always overwrite.</p> <p>Note: Data for system accounts never gets accumulated.</p> <p>Pass to CurrentValue: TRUE to accumulate, FALSE to overwrite.</p>
HSV_DATALOAD_ OPT_DELIMITER_ CHAR	<p>Name property: Delimiter</p> <p>Usage: Specifies a load file's delimiter.</p> <p>Pass to CurrentValue: String – a valid delimiter character. The <code>ValidationList</code> property returns the valid delimiters.</p>
HSV_DATALOAD_ OPT_DUPLICATES	<p>Name property: Duplicates</p> <p>Usage: Sets the data load mode to replace, merge, or accumulate.</p> <p>Tip: This option corresponds to the Load Mode option buttons in Financial Management's Load Data workspace frame. For details on the load modes, see the <i>Oracle Hyperion Financial Management, Fusion Edition User's Guide</i>.</p> <p>Pass to CurrentValue: One of the following constants:</p> <ul style="list-style-type: none"> ● HSV_DATALOAD_MERGE. Merge mode. ● HSV_DATALOAD_REPLACE. Replace mode. ● HSV_DATALOAD_ACCUMULATE. Accumulate mode. ● HSV_DATALOAD_REPLACEWITHSECURITY. Replace with Security mode. <p>Note: For descriptions of these modes, see the description of the <code>HsvData</code> type library's corresponding constants in "Update Mode Constants" on page 276. You should be especially aware of the security-related differences between the Replace and Replace with Security modes.</p>
HSV_DATALOAD_ OPT_FILE_CONTAINS_ SHARES	<p>Name property: Does the file contain shares data</p> <p>Usage: Specifies whether the load file contains shares data such as "shares outstanding" or "voting outstanding" or "owned."</p> <p>Pass to CurrentValue: TRUE if the file contains shares data, FALSE otherwise.</p>
HSV_DATALOAD_ OPT_FILE_FORMAT	<p>Name property: FileFormat</p> <p>Usage: Specifies the load file format. In this release, only the native file format is supported.</p> <p>Pass to CurrentValue: The following constant, which represents the only supported option:</p> <p style="text-align: center;">HSV_DATALOAD_FILE_FORMAT_NATIVE</p>
HSV_DATALOAD_ OPT_LOAD_CALC	<p><i>For internal use.</i></p>

Constant	Load Option Information
HSV_DATALOAD_ OPT_LOG_FILE_ APPEND	<p>Name property: Append to Log File</p> <p>Usage: Specifies whether log data is appended to or overwrites the existing log file.</p> <p>Pass to CurrentValue: Boolean – TRUE to append, FALSE to overwrite.</p>
HSV_DATALOAD_ OPT_FILE_CONTAINS_ SUBMISSIONPHASE	<p>Name property: Does the file contain submission phase data</p> <p>Usage: Specifies whether the load file contains data for phased submissions.</p> <p>Pass to CurrentValue: Boolean – TRUE if phased submissions data is being loaded.</p>
10	<p>Name property: DecimalChar</p> <p>Note: This option is not represented by a constant; 10 is the option's index in the array of data load options.</p> <p>Usage: Specifies the decimal character used in the load file. By default this option is set to an empty string, which indicates the load operation will use the decimal character set as the user's preference.</p> <p>Tip: To get the user preferences for the decimal and thousands separator characters, use the <code>HsvSystemInfo</code> method GetNumberFormattingUserParameters.</p> <p>Pass to CurrentValue: String – if the load file's decimal character differs from that specified as the user's preference, specify the load file's decimal character.</p>
11	<p>Name property: ThousandsChar</p> <p>Note: This option is not represented by a constant; 11 is the option's index in the array of data load options.</p> <p>Usage: Specifies the thousands separator character used in the load file. By default this option is set to an empty string, which indicates the load operation will use the thousands separator character set as the user's preference.</p> <p>Pass to CurrentValue: String – if the load file's thousands separator differs from that specified as the user's preference, specify the load file's thousands separator.</p>
HSV_DATALOAD_ OPT_MAX	<p>Usage: Points to the object that represents the load option with the highest index in the <code>IHsvLoadExtractOptions</code> collection.</p> <p>Pass to CurrentValue: A valid value for the option with the highest index.</p> <p>Note: There is no corresponding <code>Name</code> property. Calling the <code>Name</code> property for an <code>IHsvLoadExtractOption</code> object initialized with this constant returns the name of the option with the highest index.</p>
HSV_DATALOAD_ OPT_MIN	<p>Usage: Points to the object that represents the load option with the lowest index in the <code>IHsvLoadExtractOptions</code> collection.</p> <p>Pass to CurrentValue: A valid value for the option with the lowest index.</p> <p>Note: There is no corresponding <code>Name</code> property. Calling the <code>Name</code> property for an <code>IHsvLoadExtractOption</code> object initialized with this constant returns the name of the option with the lowest index.</p>
HSV_DATALOAD_ OPT_MODE	<p>Name property: Mode</p> <p>Usage: Specifies whether the data file is loaded or is merely scanned when <code>Load</code> is called.</p> <p>Pass to CurrentValue: One of the following constants:</p> <ul style="list-style-type: none"> ● <code>HSV_DATALOAD_LOAD</code>. Data will be loaded. ● <code>HSV_DATALOAD_SCAN</code>. Data will be scanned but not loaded.

Constant	Load Option Information
	The <code>Mode</code> option defaults to scanning, so you must set the option to <code>HSV_DATALOAD_LOAD</code> in order to load data.

Data Extract Options

For each data extract option, the following table lists the constant that identifies the option and the corresponding `Name` property — either the constant or the name can be passed to `IHsvLoadExtractOptions.Item`. The table also describes each option and the type of information to be passed to `IHsvLoadExtractOption.CurrentValue`.

Table 80 Data Extract Options

Constant	Extract Option Information
<code>HSV_DATAEXTRACT_OPT_ACCOUNT_SUBSET</code>	<p>Name property: Account Subset</p> <p>Usage: Specifies the Account dimension members for which data is being extracted.</p> <p>Pass to CurrentValue: A Long array containing Account dimension member IDs.</p>
<code>HSV_DATAEXTRACT_OPT_CUSTOM1_SUBSET</code>	<i>For internal use.</i>
<code>HSV_DATAEXTRACT_OPT_CUSTOM2_SUBSET</code>	<i>For internal use.</i>
<code>HSV_DATAEXTRACT_OPT_CUSTOM3_SUBSET</code>	<i>For internal use.</i>
<code>HSV_DATAEXTRACT_OPT_CUSTOM4_SUBSET</code>	<i>For internal use.</i>
<code>HSV_DATAEXTRACT_OPT_DELIMITER_CHAR</code>	<p>Name property: Delimiter</p> <p>Usage: Specifies an extract file's delimiter.</p> <p>Pass to CurrentValue: String – a valid delimiter character. The <code>ValidationList</code> property returns the valid delimiters.</p>
<code>HSV_DATAEXTRACT_OPT_ENTITY_SUBSET</code>	<p>Name property: Entity Subset</p> <p>Usage: Specifies the child Entity dimension members for which data is being extracted.</p> <p>Pass to CurrentValue: A Long array containing Entity dimension member IDs.</p>
<code>HSV_DATAEXTRACT_OPT_EXTRACT_ALL_DATA</code>	<i>For internal use.</i>
<code>HSV_DATAEXTRACT_OPT_EXTRACT_CALC</code>	<p>Name property: Extract Calculated</p> <p>Usage: Specifies whether to extract calculated data.</p> <p>Pass to CurrentValue: TRUE to extract calculated data, FALSE otherwise.</p>
<code>HSV_DATAEXTRACT_OPT_EXTRACT_FILE_APPEND</code>	<i>For internal use.</i>

Constant	Extract Option Information
HSV_DATAEXTRACT_OPT_ICP_SUBSET	<i>For internal use.</i>
HSV_DATAEXTRACT_OPT_LOG_FILE_APPEND	<p>Name property: Append to log file</p> <p>Usage: Specifies whether log data is appended to or overwrites the existing log file.</p> <p>Pass to CurrentValue: TRUE to append, FALSE to overwrite.</p>
HSV_DATAEXTRACT_OPT_LOG_FILE_NAME	<i>For internal use.</i>
HSV_DATAEXTRACT_OPT_MAX	<p>Usage: Points to the object that represents the extract option with the highest index in the IHsvLoadExtractOptions collection.</p> <p>Pass to CurrentValue: A valid value for the option with the highest index.</p> <p>Note: There is no corresponding Name property. Calling the Name property for an IHsvLoadExtractOption object initialized with this constant returns the name of the option with the highest index.</p>
HSV_DATAEXTRACT_OPT_MIN	<p>Usage: Points to the object that represents the extract option with the lowest index in the IHsvLoadExtractOptions collection.</p> <p>Pass to CurrentValue: A valid value for the option with the lowest index.</p> <p>Note: There is no corresponding Name property. Calling the Name property for an IHsvLoadExtractOption object initialized with this constant returns the name of the option with the lowest index.</p>
HSV_DATAEXTRACT_OPT_PARENT_SUBSET	<p>Name property: Parent Subset</p> <p>Usage: Specifies the parent Entity dimension members for which data is being extracted.</p> <p>Pass to CurrentValue: A Long array containing Entity dimension member IDs.</p>
HSV_DATAEXTRACT_OPT_PERIOD_SUBSET	<p>Name property: Period Subset</p> <p>Usage: Specifies the Period dimension members for which data is being extracted.</p> <p>Pass to CurrentValue: A Long array containing Period dimension member IDs.</p>
HSV_DATAEXTRACT_OPT_SCENARIO_SUBSET	<p>Name property: Scenario</p> <p>Usage: Specifies the Scenario dimension member for which data is being extracted.</p> <p>Pass to CurrentValue: A Long containing the Scenario dimension member ID.</p>
HSV_DATAEXTRACT_OPT_UNICODE	<i>For internal use.</i>
HSV_DATAEXTRACT_OPT_VALUE_SUBSET	<i>For internal use.</i>
HSV_DATAEXTRACT_OPT_VIEW	<p>Name property: View</p> <p>Usage: Specifies the View dimension member for which data is being extracted.</p> <p>Pass to CurrentValue: One of the following constants:</p> <ul style="list-style-type: none"> ● HSV_DATA_VIEW_PERIODIC . Data for the Periodic view will be extracted. ● HSV_DATA_VIEW_YTD . Data for the year-to-date view will be extracted.

Constant	Extract Option Information
	<ul style="list-style-type: none"> ● HSV_DATA_VIEW_SCENARIO. Data for the scenario's default View dimension member will be extracted. <p>Tip: These constants correspond to the Data View option buttons in the Extract Data workspace frame.</p>
HSV_DATAEXTRACT_OPT_YEAR_SUBSET	<p>Name property: Year</p> <p>Usage: Specifies the Year dimension member for which data is being extracted.</p> <p>Pass to CurrentValue: A Long containing the Year dimension member ID.</p>

HsvcDataLoad Object Properties

The HsvcDataLoad object provides the following properties:

- `ExtractOptions`. See [“ExtractOptions” on page 748](#).
- `LoadOptions`. See [“LoadOptions” on page 748](#).

These properties are members of several objects. The HsvcDataLoad object has no additional properties.

Note: Use the `Set` keyword to assign HsvcDataLoad object references.

HsvcDataLoad Object Methods

The HsvcDataLoad object provides the following methods:

- [Extract](#)
- [Load](#)
- [Load2](#)
- [SetSession](#)

`Extract`, `Load`, and `SetSession` are members of several objects. The HsvcDataLoad object has other methods that are documented in the following topics; however, other than `Load2`, those are only for internal use.

DMELoad

For internal use.

Load2

Loads data from a text file on a client computer and returns a flag indicating whether any errors were logged.

You must call [SetSession](#) and set the [LoadOptions](#) property before calling `Load2`.

Syntax

```
<HsvcDataLoad>.Load2(bstrClientFileName, bstrClientLogFileName)
```

Argument	Description
<i>bstrClientFileName</i>	String (ByVal). The name and path of the file containing the data to be loaded. The path must be valid for the client computer. For details on the file's required format, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i> .
<i>bstrClientLogFileName</i>	String (ByVal). The name and path of the log file for the load operation.

Return Value

Variant. Returns TRUE if errors were logged to the log file, FALSE otherwise.

SetFileForLoad

For internal use.

StartLoad

For internal use.

HsvJournalLoadACV Type Library

To use the HsvJournalLoadACV type library, you must reference HsvJournalLoadACV.dll in your project.

Use the HsvJournalLoadACV type library to load and extract journals and journal templates. This type library exposes the HsvJournalLoadACV object, and also exposes the IHsvLoadExtractOptions and IHsvLoadExtractOption interfaces.

The following topics show you how to use the HsvJournalLoadACV type library:

- [“IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvJournalLoadACV” on page 785](#)
- [“Loading Journals” on page 785](#)
- [“Extracting Journals” on page 786](#)
- [“Journal Load Options” on page 789](#)
- [“Journal Extract Options \(Unfiltered\)” on page 789](#)
- [“Journal Extract Options \(Filtered\)” on page 791](#)
- [“HsvJournalLoadACV Object Properties” on page 793](#)
- [“HsvJournalLoadACV Object Methods” on page 793](#)

IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for HsvJournalLoadACV

The HsvJournalLoadACV type library exposes the IHsvLoadExtractOptions and IHsvLoadExtractOption interfaces; these interfaces provide access to data load and extract options. For information on these interfaces, see [“Load and Extract Option Interfaces”](#) on page 743.

- To set a journal load or extract option:
 - 1 **Set an IHsvLoadExtractOption object reference for the option with** `IHsvLoadExtractOptions.Item`.
 - 2 **Set the option’s value with the** `IHsvLoadExtractOption.CurrentValue` **property.**

Tip: The options available for loading journals are described in [Table 81 on page 789](#), and the options available for extracting journals are described in [Table 82 on page 790](#). These tables include the valid values for `Item` and `CurrentValue`.

Loading Journals

The following steps provide an overview of how to load journals. These steps assume that an application has previously been opened with one of the `OpenApplication` methods.

Tip: For an example that illustrates these steps, see [“Example for Loading Journals”](#) on page 785.

- To load journals:
 - 1 **Set an object reference to the HsvJournalLoadACV object.**
 - 2 **Point to the application into which journals are being loaded by calling** `HsvJournalLoadACV.SetSession`. **For** `SetSession`’s **argument**, **pass the HsvSession object reference that was returned by** `HsxClient.OpenApplication` **or** `HsxClientUI.OpenApplication`.
 - 3 **Set an object reference to the IHsvLoadExtractOptions interface with the** `HsvJournalLoadACV.LoadOptions` **property.**
 - 4 **Optional.** **To change the delimiter from the default, set the Delimiter load option.** For information on this option, see [Table 81 on page 789](#). Note that this is the only load option available for loading journals.
 - 5 **Load the journals by calling** `HsvJournalLoadACV.Load`. `Load` **takes the file names and paths of the journal load file and of the log file.**

Note: If the load file contains templates, these will also be loaded.

Example for Loading Journals

The following example loads the journals and templates in a load file.

```

Dim cJournalLoadACV As HsvJournalLoadACV
Dim cOptions As IHsvLoadExtractOptions, cOpt As IHsvLoadExtractOption
Set cJournalLoadACV = New HsvJournalLoadACV
'Specify the HsvSession object for the application.
cJournalLoadACV.SetSession g_cHsvSession
'Initialize the IHsvLoadExtractOptions interface.
Set cOptions = cJournalLoadACV.LoadOptions
'Load the Journals
cJournalLoadACV.Load "c:\Acme\MyApp.jlf", "c:\Acme\MyApp.log"

```

Extracting Journals

The HsvJournalLoadACV type library provides two ways in which to extract journals:

- You can extract all journals for given Scenario, Year, and Period dimension members. For details, see [“Extracting Journals without Filters” on page 786](#).
- You can extract journals that meet various filtering criteria. For example, you can extract journals for multiple Period, Entity, and Value dimension members, and extract journals of given types and statuses. For details, see [“Extracting Journals with Filters” on page 787](#).

Extracting Journals without Filters

To extract journals without applying filters, set extract options with the [ExtractOptions](#) property, then call [Extract](#) as shown in the following steps. These steps assume that an application has previously been opened with one of the [OpenApplication](#) methods.

Note: For an example that illustrates these steps, see [“Example for Extracting Journals Without Filters” on page 787](#).

➤ To extract journals without filtering:

- 1 Set an object reference to the HsvJournalLoadACV object.
- 2 Point to the application from which journals are being extracted by calling `HsvJournalLoadACV.SetSession`. For `SetSession`'s argument, pass the HsvSession object reference that was returned by `HsxClient.OpenApplication` or `HsxClientUI.OpenApplication`.
- 3 Set an object reference to the IHsvLoadExtractOptions interface with the `HsvJournalLoadACV.ExtractOptions` property.
- 4 Specify the Scenario and Year dimension members for which journals are to be extracted. To specify these members, use the Scenario and Year extract options.

Note: For details on journal extract options, see [Table 82 on page 790](#).

- 5 **Optional.** To override the defaults for other extract options, specify the values for these options. If you do not override any defaults, journals will be extracted, and standard and recurring templates will not be extracted.

- 6 **Extract the journals by calling** `HsvJournalLoadACV.Extract`. `Extract` takes the file names and paths of the journal extract file and of the log file.

Example for Extracting Journals Without Filters

The following example extracts journals for the Actual scenario in the year 2000.

Note: The example uses a user-defined function named `GetMemberID` to obtain the IDs of these members. For information on this user-defined function, see the [Examples](#) for `GetItemID`.

```
Dim cJournalLoadACV As HsvJournalLoadACV
Dim cOptions As IHsvLoadExtractOptions
Dim cOpt As IHsvLoadExtractOption
Dim lScen As Long, lYear As Long
Set cJournalLoadACV = New HsvJournalLoadACV
'Specify the HsvSession object for the application.
cJournalLoadACV.SetSession g_cHsvSession
'Initialize the IHsvLoadExtractOptions interface.
Set cOptions = cJournalLoadACV.ExtractOptions
'Set the scenario to "Actual"
Set cOpt = cOptions.Item(HSV_JOURNALEXTRACT_OPT_SCENARIO)
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
cOpt.CurrentValue = lScen
'Set the year to "2000"
Set cOpt = cOptions.Item(HSV_JOURNALEXTRACT_OPT_YEAR)
lYear = GetMemberID(DIMENSIONYEAR, "2000")
cOpt.CurrentValue = lYear
'Extract the Journals
cJournalLoadACV.Extract "c:\Acme\Jnl.jlf", "c:\Acme\Jnl.log"
```

Extracting Journals with Filters

To extract journals by applying filters, set extract options with the `ExtractOptionsEx` property, then call `ExtractEx` method as described in the following steps.

Note: For an example that illustrates these steps, see “[Example for Filtered Journal Extractions](#)” on page 788.

- To extract journals by applying filtering criteria:
 - 1 **Set an object reference to the `HsvJournalLoadACV` object.**
 - 2 **Point to the application from which journals are being extracted by calling `HsvJournalLoadACV.SetSession`. For `SetSession`'s argument, pass the `HsvSession` object reference that was returned by `HsxClient.OpenApplication` or `HsxClientUI.OpenApplication`.**
 - 3 **Set an object reference to the `IHsvLoadExtractOptions` interface with the `HsvJournalLoadACV.ExtractOptionsEx` property.**

- 4 Specify the Scenario and Year dimension members for which journals are to be extracted. To specify these members, use the Scenario and Year extract options.
- 5 Specify the journal type filter by using the Type journal extract option.

Tip: For details on journal extract options, see [Table 83 on page 791](#).

- 6 **Optional.** To override the defaults for other extract options, specify the values for these options. If you do not override any defaults, journals will be extracted, standard and recurring templates will not be extracted, and no filtering criteria will be applied.

Note: The extraction uses only those filtering criteria for which you have set the corresponding extract option. For example, to avoid filtering by label and group, do not set the Label and Group extract options.

- 7 Extract the journals by calling `HsvJournalLoadACV.ExtractEx`. `ExtractEx` takes the file names and paths of the journal extract file and of the log file.

Example for Filtered Journal Extractions

The following example extracts journals for the Actual scenario in the year 2000 that have a Regular journal type and a status of Submitted or Posted.

Note: The example uses a user-defined function named `GetMemberID` to obtain the IDs of these members. For information on this user-defined function, see the [Examples](#) for `GetItemID`.

```
Dim cJournalLoadACV As HsvJournalLoadACV
Dim cOptions As IHsvLoadExtractOptions
Dim cOpt As IHsvLoadExtractOption, lScen As Long
Dim lYear As Long, laTypes(0) As Long, laStatus(1) As Long
Set cJournalLoadACV = New HsvJournalLoadACV
'Specify the HsvSession object for the application.
cJournalLoadACV.SetSession g_cHsvSession
'Initialize the IHsvLoadExtractOptions interface.
Set cOptions = cJournalLoadACV.ExtractOptionsEx
'Set the scenario to "Actual"
Set cOpt = cOptions.Item(HSV_JOURNALEXTRACT_EX_OPT_SCENARIO)
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
cOpt.CurrentValue = lScen
'Set the year to "2000"
Set cOpt = cOptions.Item(HSV_JOURNALEXTRACT_EX_OPT_YEAR)
lYear = GetMemberID(DIMENSIONYEAR, "2000")
cOpt.CurrentValue = lYear
'Set the Type to Regular
Set cOpt = cOptions.Item(HSV_JOURNALEXTRACT_EX_OPT_FILTER_TYPE)
laTypes(0) = JTF_REGULAR
cOpt.CurrentValue = laTypes
'Extract only Submitted and Posted journals
Set cOpt = _
    cOptions.Item(HSV_JOURNALEXTRACT_EX_OPT_FILTER_STATUS)
laStatus(0) = JSF_SUBMITTED
```

```

laStatus(1) = JSF_POSTED
cOpt.CurrentValue = laStatus
'Extract the Journals
cJournalLoadACV.ExtractEx "c:\Acme\Jnl.jlf", "c:\Acme\Jnl.log"

```

Journal Load Options

For each journal load option, the following table lists the constant that identifies the option and the corresponding `Name` property — either the constant or the name can be passed to `IHsvLoadExtractOptions.Item`. The table also describes the option and the type of information to be passed to `IHsvLoadExtractOption.CurrentValue`.

Table 81 Journal Load Options

Constant	Load Option Information
HSV_JOURNALLOAD_OPT_DELIMITER_CHAR	<p>Name property: Delimiter</p> <p>Usage: Specifies a load file's delimiter.</p> <p>Pass to CurrentValue: String — a valid delimiter character. The <code>ValidationList</code> property returns the valid delimiters.</p>
HSV_JOURNALLOAD_OPT_MAX	<p>Usage: Points to the object that represents the extract option with the highest index in the <code>IHsvLoadExtractOptions</code> collection.</p> <p>Pass to CurrentValue: A valid value for the option with the highest index.</p> <p>There is no corresponding <code>Name</code> property. Calling the <code>Name</code> property for an <code>IHsvLoadExtractOption</code> object initialized with this constant returns the name of the option with the highest index.</p>
HSV_JOURNALLOAD_OPT_MIN	<p>Usage: Points to the object that represents the extract option with the lowest index in the <code>IHsvLoadExtractOptions</code> collection.</p> <p>Pass to CurrentValue: A valid value for the option with the lowest index.</p> <p>There is no corresponding <code>Name</code> property. Calling the <code>Name</code> property for an <code>IHsvLoadExtractOption</code> object initialized with this constant returns the name of the option with the lowest index.</p>

Journal Extract Options (Unfiltered)

For each unfiltered journal extract option, the following table lists the constant that identifies the option and the corresponding `Name` property — either the constant or the name can be passed to `Item`. The table also describes each option and the type of information to be passed to `CurrentValue`.

By default, journals are extracted, and standard and recurring templates are not extracted. Use the Regular, Standard, and Recurring extract options to override these defaults.

Table 82 Journal Extract Options (Unfiltered)

Constant	Extract Option Information
HSV_JOURNALEXTRACT_OPT_DELIMITER_CHAR	<p>Name property: Delimiter</p> <p>Usage: Specifies an extract file's delimiter.</p> <p>Pass to CurrentValue: String – a valid delimiter character. The <code>ValidationList</code> property returns the valid delimiters.</p>
HSV_JOURNALEXTRACT_OPT_SCENARIO	<p>Name property: Scenario</p> <p>Usage: Specifies the Scenario dimension member for which journals are being extracted.</p> <p>Pass to CurrentValue: A Long containing the Scenario dimension member ID.</p>
HSV_JOURNALEXTRACT_OPT_PERIOD	<p>Name property: Period</p> <p>Usage: Specifies the Period dimension member for which journals are being extracted.</p> <p>Pass to CurrentValue: A Long containing the Period dimension member ID.</p>
HSV_JOURNALEXTRACT_OPT_YEAR	<p>Name property: Year</p> <p>Usage: Specifies the Year dimension member for which journals are being extracted.</p> <p>Pass to CurrentValue: A Long containing the Year dimension member ID.</p>
HSV_JOURNALEXTRACT_OPT_REGULAR	<p>Name property: Regular</p> <p>Usage: Specifies whether journals are being extracted.</p> <p>Pass to CurrentValue: Boolean - TRUE to extract journals, otherwise FALSE. This option defaults to TRUE.</p>
HSV_JOURNALEXTRACT_OPT_STANDARD	<p>Name property: Standard</p> <p>Usage: Specifies whether standard templates are being extracted.</p> <p>Pass to CurrentValue: Boolean - TRUE to extract standard templates, otherwise FALSE. This option defaults to FALSE.</p>
HSV_JOURNALEXTRACT_OPT_RECURRING	<p>Name property: Recurring</p> <p>Usage: Specifies whether recurring templates are being extracted.</p> <p>Pass to CurrentValue: Boolean - TRUE to extract recurring templates, otherwise FALSE. This option defaults to FALSE.</p>
HSV_JOURNALEXTRACT_OPT_MAX	<p>Usage: Points to the object that represents the extract option with the highest index in the <code>IHsvLoadExtractOptions</code> collection.</p> <p>Pass to CurrentValue: A valid value for the option with the highest index.</p> <p>There is no corresponding <code>Name</code> property. Calling the <code>Name</code> property for an <code>IHsvLoadExtractOption</code> object initialized with this constant returns the name of the option with the highest index.</p>
HSV_JOURNALEXTRACT_OPT_MIN	<p>Usage: Points to the object that represents the extract option with the lowest index in the <code>IHsvLoadExtractOptions</code> collection.</p> <p>Pass to CurrentValue: A valid value for the option with the lowest index.</p> <p>There is no corresponding <code>Name</code> property. Calling the <code>Name</code> property for an <code>IHsvLoadExtractOption</code> object initialized with this constant returns the name of the option with the lowest index.</p>

Journal Extract Options (Filtered)

For each filtered journal extract option, the following table lists the constant that identifies the option and the corresponding `Name` property — either the constant or the name can be passed to `Item`. The table also describes each option and the type of information to be passed to `CurrentValue`.

Caution! Use these options only with the `ExtractOptionsEx` property.

By default journals are extracted, and standard and recurring templates are not extracted. Use the Regular, Standard, and Recurring extract options to override these defaults.

Table 83 Journal Extract Options (Filtered)

Constant	Extract Option Information
HSV_JOURNALEXTRACT_EX_OPT_DELIMITER_CHAR	<p>Name property: Delimiter</p> <p>Usage: Specifies an extract file's delimiter.</p> <p>Pass to CurrentValue: String — a valid delimiter character. The <code>ValidationList</code> property returns the valid delimiters.</p>
HSV_JOURNALEXTRACT_EX_OPT_SCENARIO	<p>Name property: Scenario</p> <p>Usage: Specifies the Scenario dimension member for which journals are being extracted.</p> <p>Pass to CurrentValue: A Long containing the Scenario dimension member ID.</p>
HSV_JOURNALEXTRACT_EX_OPT_PERIOD	<p>Name property: Period</p> <p>Usage: Specifies the Period dimension members for which journals are being extracted.</p> <p>Pass to CurrentValue: A Long array that contains the IDs of the Period dimension members.</p>
HSV_JOURNALEXTRACT_EX_OPT_YEAR	<p>Name property: Year</p> <p>Usage: Specifies the Year dimension member for which journals are being extracted.</p> <p>Pass to CurrentValue: A Long containing the Year dimension member ID.</p>
HSV_JOURNALEXTRACT_EX_OPT_ENTITIES	<p>Name property: Entity</p> <p>Usage: Specifies the child Entity dimension members for which journals are being extracted.</p> <p>Pass to CurrentValue: A Long array that contains the IDs of the child Entity dimension members.</p>
HSV_JOURNALEXTRACT_EX_OPT_PARENTS	<p>Name property: Parent</p> <p>Usage: Specifies the parent Entity dimension members for which journals are being extracted.</p> <p>Pass to CurrentValue: A Long array that contains the IDs of the parent Entity dimension members. This array must have a one-to-one correspondence with the array for the child Entity dimension members.</p>

Constant	Extract Option Information
HSV_JOURNALEXTRACT_EX_OPT_VALUES	<p>Name property: Value</p> <p>Usage: Specifies the Value dimension members for which journals are being extracted.</p> <p>Pass to CurrentValue: A Long array that contains the IDs of the Value dimension members.</p>
HSV_JOURNALEXTRACT_EX_OPT_REGULAR	<p>Name property: Regular</p> <p>Usage: Specifies whether journals are being extracted.</p> <p>Pass to CurrentValue: Boolean - TRUE to extract journals, otherwise FALSE. This option defaults to TRUE.</p>
HSV_JOURNALEXTRACT_EX_OPT_STANDARD	<p>Name property: Standard</p> <p>Usage: Specifies whether standard templates are being extracted.</p> <p>Pass to CurrentValue: Boolean - TRUE to extract standard templates, otherwise FALSE. This option defaults to FALSE.</p>
HSV_JOURNALEXTRACT_EX_OPT_RECURRING	<p>Name property: Recurring</p> <p>Usage: Specifies whether recurring templates are being extracted.</p> <p>Pass to CurrentValue: Boolean - TRUE to extract recurring templates, otherwise FALSE. This option defaults to FALSE.</p>
HSV_JOURNALEXTRACT_EX_OPT_FILTER_BALANCE	<p>Name property: Balance Type</p> <p>Usage: Specifies the balance type filter for the extraction.</p> <p>Pass to CurrentValue: A Long array containing the balance types to use as filtering criteria. Valid values are represented by the HFMCconstants type library constants listed in "Balance Type Constants" on page 848.</p>
HSV_JOURNALEXTRACT_EX_OPT_FILTER_GROUP	<p>Name property: Group</p> <p>Usage: Specifies the journal group filter for the extraction.</p> <p>Pass to CurrentValue: A String array containing the journal group filtering criteria. You can use the pound character (#) as a wildcard.</p>
HSV_JOURNALEXTRACT_EX_OPT_FILTER_LABEL	<p>Name property: Label</p> <p>Usage: Specifies the journal label filter for the extraction.</p> <p>Pass to CurrentValue: A String array containing the journal label filtering criteria. You can use the pound character (#) as a wildcard.</p>
HSV_JOURNALEXTRACT_EX_OPT_FILTER_STATUS	<p>Name property: Status</p> <p>Usage: Specifies the journal status filter for the extraction.</p> <p>Pass to CurrentValue: A Long array containing the status filtering criteria. Valid values are represented by the HFMCconstants type library constants listed in "Journal Status Constants" on page 851.</p>
HSV_JOURNALEXTRACT_EX_OPT_FILTER_TYPE	<p>Name property: Type</p> <p>Usage: Specifies the journal type filter for the extraction.</p> <p>Pass to CurrentValue: A Long array containing the journal type filtering criteria. Valid values are represented by the HFMCconstants type library constants listed in "Journal Type Constants" on page 851.</p>

Constant	Extract Option Information
HSV_JOURNALEXTRACT_EX_OPT_MAX	<p>Usage: Points to the object that represents the extract option with the highest index in the IHsvLoadExtractOptions collection.</p> <p>Pass to CurrentValue: A valid value for the option with the highest index.</p> <p>There is no corresponding <code>Name</code> property. Calling the <code>Name</code> property for an IHsvLoadExtractOption object initialized with this constant returns the name of the option with the highest index.</p>
HSV_JOURNALEXTRACT_EX_OPT_MIN	<p>Usage: Points to the object that represents the extract option with the lowest index in the IHsvLoadExtractOptions collection.</p> <p>Pass to CurrentValue: A valid value for the option with the lowest index.</p> <p>There is no corresponding <code>Name</code> property. Calling the <code>Name</code> property for an IHsvLoadExtractOption object initialized with this constant returns the name of the option with the lowest index.</p>

HsvJournalLoadACV Object Properties

The HsvJournalLoadACV object provides the following properties:

- `ExtractOptions`. See “[ExtractOptions](#)” on page 748.
- `ExtractOptionsEx`.
- `LoadOptions`. See “[LoadOptions](#)” on page 748.

The `ExtractOptions` and `LoadOptions` properties are members of several objects.

Note: Use the `Set` keyword to assign HsvJournalLoadACV object references.

ExtractOptionsEx

Returns an object reference to an instance of the IHsvLoadExtractOptions interface for a filtered journal extraction. You *must* set this property before extracting journals with `ExtractEx`.

For information on filtered journal extractions, as well as an example, see “[Extracting Journals with Filters](#)” on page 787.

Data Type

IHsvLoadExtractOptions interface.

HsvJournalLoadACV Object Methods

The HsvJournalLoadACV object provides the following methods:

- `Extract`.
- `ExtractEx`.
- `Load`.

- [SetSession](#).

The `Extract`, `Load`, and `SetSession` methods are members of several objects.

ExtractEx

Extracts journals from an application into a text file on the client computer, applying the filtering criteria set with [ExtractOptionsEx](#).

You must call [SetSession](#) and set the `ExtractOptionsEx` property before calling `Extract`.

For information on filtered journal extractions, as well as an example, see “[Extracting Journals with Filters](#)” on page 787.

Note: To extract journals without applying filtering criteria, use [Extract](#).

Syntax

```
<HsvJournalLoadACV>.ExtractEx bstrServerFilename, bstrServerLogFilename
```

Argument	Description
<i>bstrServerFilename</i>	String. The name and path of the file into which the journals will be extracted. The path must exist on the client computer. For details on the extracted file's format, see the <i>Oracle Hyperion Financial Management, Fusion Edition Administrator's Guide</i> .
<i>bstrServerLogFilename</i>	String. The name and path of the log file for the extraction.

In This Chapter

HsvPOVSelection Object Methods.....	796
HsvPOVSelection Object Events	807

This chapter describes the members of the HsvPOVSelection type library. The type library contains the HsvPOVSelection object. This object exposes an insertable ActiveX control that provides a user interface for selecting dimension members. For example, you can use this control in programs where users specify members for a Point of View.

Note: This is the control used in Financial Management dialogs such as the Point of View dialog box.

The HsvPOVSelection control provides one tab per dimension. Each tab includes a button that displays the Select Member List dialog box. This dialog box enables users to select the member list and top member that define the members available for selection. For Organization by Period applications, the Select Member List dialog box also enables display of only active Entity dimension members.

To use the HsvPOVSelection control, you must reference `HsvPOVSelection.dll` in your project. This file is located in the `<Financial Management installation>\Client` directory

Tip: In Visual Basic 6, make this reference with the Controls tab of the Components dialog box. After making this reference, Visual Basic adds a button for the HsvPOVSelection control to the Toolbox.

Note: In addition to the methods and events documented in this chapter, the control exposes standard ActiveX methods, properties, and events. In Visual Basic, you can view these standard members with the Object Browser, and find information on them in the Visual Basic online help.

HsvPOVSelection Object Methods

The HsvPOVSelection methods enable you to define the appearance of a HsvPOVSelection control's instance, to select members, and to get the selected members. You can define the control's appearance in the following ways:

- Show tabs for some or all dimensions.
- Toggle between multi-select and single-select modes for a given dimension's tab. For more information on these selection modes, see [ShowCheckBoxes](#).

Tip: The selection mode determines how you identify the members selected by users. In multi-select mode, use [GetCheckedItems](#); in single-select mode, use [GetMember](#).

- Specify the member list that defines the members displayed for a dimension. You can also specify the list's top member.

Tip: Specifying the top member means that only a subset of a member list—the top member and those beneath it—are displayed.

- Specify that only active Entity members are displayed for Organization by Period applications. You also can specify the Scenario, Year, and Period dimension members that define which entities are considered active for display.

To display a control, you specify the dimension tabs to be displayed with [EnableDimension](#), then show the control with [Initialize](#). You must call [EnableDimension](#) before calling [Initialize](#). You can call the HsvPOVSelection methods for configuring the control either before or after calling [Initialize](#).

Note: HsvPOVSelection object references are identified by the control's Name property.

The HsvPOVSelection object's methods are summarized in [Table 54 on page 115](#), and are described in detail in the following topics.

CheckAllItems

For internal use.

CheckAllItemsUsingSelectedList

Selects or clears all the currently displayed check boxes for a given dimension when the tab is configured for multi-select mode.

The currently displayed members are determined by the member list and top member specified for the dimension. Member lists can be specified either programmatically or by users. To set a member list programmatically, use [SetListInfo](#).

Tip: To get the member IDs of selected members, use [GetCheckedItems](#).

Syntax

```
<HsvPOVSelection>.CheckAllItemsUsingSelectedList lDim, bCheck
```

Argument Description

<i>lDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 834 .
<i>bCheck</i>	Boolean (ByVal). A flag that determines whether all check boxes are selected or cleared. Pass TRUE to select, FALSE to clear.

Example

The following example clears all check boxes for the Entity dimension.

```
' cFormPOV represents an initialized HsvPOVSelection control  
cFormPOV.CheckAllItemsUsingSelectedList DIMENSIONENTITY, False
```

EnableDimension

Displays or hides the tab for a given dimension. You can only use `EnableDimension` before the control is displayed with [Initialize](#).

Tip: To specify the selection mode for a dimension, use [ShowCheckBoxes](#).

Syntax

```
<HsvPOVSelection>.EnableDimension lDim, bEnable
```

Argument Description

<i>lDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 834 .
<i>bEnable</i>	Boolean (ByVal). A flag that specifies whether the dimension’s tab is displayed or hidden. Pass TRUE to display, FALSE to hide.

Example

The following example configures the control to display the Scenario dimension’s tab and hide the other dimensions’ tabs. The example configures the tab for multi-select mode with [ShowCheckBoxes](#), then calls [Initialize](#) to display the control.

```
' cFormPOV represents an initialized HsvPOVSelection control  
For i = DIMENSION_LBOUND To DIMENSION_UBOUND  
  If i = DIMENSIONSCENARIO Then  
    cFormPOV.EnableDimension i, True  
    cFormPOV.ShowCheckBoxes i, True  
  Else  
    cFormPOV.EnableDimension i, False  
  End If  
Next
```

```
' cSession is a previously set HsvSession instance
cFormPOV.Initialize cSession
```

GetCheckedItems

Returns the member IDs of the selected members on a given dimension's tab when the tab is configured for multi-select mode. For the Entity dimension, `GetCheckedItems` also returns the member IDs of the selected members' parents.

Note: To return a count of the selected check boxes for a dimension, use `GetNumCheckedItems`. If a tab is configured for single-select mode, get the selected member with `GetMember`.

Syntax

```
<HsvPOVSelection>.GetCheckedItems lDim, pvaralItemIDs, pvaralParentIDs
```

Argument	Description
<i>IDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 834 .
<i>pvaralItemIDs</i>	Variant. Returns an array of member IDs for the selected members. The array items are Longs.
<i>pvaralParentIDs</i>	Variant. For the Entity dimension, returns an array of member IDs for the parents of the selected members. The array has a one-to-one correspondence with the <i>pvaralItemIDs</i> argument's array. The array items are Longs. Note: If an Entity member has no parent, the HFMConstants type library constant <code>MEMBERNOTUSED</code> is returned. For information on this constant, see "Dimension Member Constants" on page 834 . For the other dimensions, this argument returns an empty Variant.

Example

`GetCheckedItems` is used in the example for the `HsvStarSchemaACM` method [CreateStarSchema](#).

GetListInfo

Returns the ID of the member list that is currently selected for a given dimension, as well as the member ID of the specified top member.

Tip: To return whether only active Entity members are displayed for Organization by Period applications, use `GetOBPInfo`.

Syntax

```
<HsvPOVSelection>.GetListInfo lDim, plListID, plListTopMemberID
```

Argument	Description
<i>IDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 834 .
<i>plListID</i>	Long. Returns the ID of the member list. Tip: To get a member list name from its ID, use the IHsvTreeInfo method GetMemberListName .
<i>plListTopMemberID</i>	Long. Returns the member ID of the specified top member. If no top member is specified, the HFMConstants type library constant <code>MEMBERNOTUSED</code> is returned. For information on this constant, see “Dimension Member Constants” on page 834 .

Example

The following function returns the name of the member list that is selected on a given dimension’s tab. The name is obtained by passing the list ID to `IHsvTreeInfo.GetMemberListName`.

```
Function getListName(lDim As Long) As String
Dim lId As Long, lTopMemId As Long
Dim cTreeInfo As IHsvTreeInfo, sRet As String
'g_cMetadata is a previously set HsvMetadata instance
Set cTreeInfo = g_cMetadata.Dimension(lDim)
'cFormPOV represents an initialized HsvPOVSelection control
cFormPOV.GetListInfo lDim, lId, lTopMemId
cTreeInfo.GetMemberListName lId, sRet
getListName = sRet
End Function
```

GetMember

Returns the member ID of the currently selected member for a given dimension when the tab is configured for single-select mode.

Note: In multi-select mode, `GetMember` returns the ID of the selected member label regardless of whether that member’s check box is selected. If a tag is configured for multi-select mode, get the selected members with [GetCheckedItems](#).

Syntax

```
<HsvPOVSelection>.GetMember lDim, plItemID, plParentID
```

Argument Description

<i>IDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 834 .
<i>plItemID</i>	Long. Returns the member ID of the selected member. Tip: To get a member’s label, pass this ID to the IHsvTreeInfo method GetLabel .

Argument Description

pParentID Long. For the Entity dimension, returns the member ID of the selected member's parent.

If the member has no parent, or belongs to a dimension other than Entity, the HFMConstants type library constant `MEMBERNOTUSED` is returned. For information on this constant, see [“Dimension Member Constants” on page 834](#).

Example

The following function returns the name of the member that is selected on a given dimension's tab. The name is obtained by passing the member ID to `IHsvTreeInfo.GetLabel`.

```
Function getSelMemName(lDim As Long) As String
Dim lId As Long, lParId As Long, sMem As String
Dim cTreeInfo As IHsvTreeInfo, sPar As String
'g_cMetadata is a previously set HsvMetadata instance
Set cTreeInfo = g_cMetadata.Dimension(lDim)
'cFormPOV represents an initialized HsvPOVSelection control
cFormPOV.GetMember lDim, lId, lParId
cTreeInfo.GetLabel lId, sMem
If lParId > -1 Then
    'concatenate Parent name
    cTreeInfo.GetLabel lParId, sPar
    getSelMemName = sPar & "." & sMem
Else
    getSelMemName = sMem
End If
End Function
```

GetNumCheckedItems

Returns a count of the check boxes that are selected on a given dimension's tab.

Tip: To get the member IDs of members for which check boxes are selected, use [GetCheckedItems](#).

Syntax

```
<HsvPOVSelection>.GetNumCheckedItems lDim, pNumCheckedItems
```

Argument	Description
<i>lDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 834 .
<i>pNumCheckedItems</i>	Long. Returns the number of selected check boxes.

GetOBPInfo

Indicates whether the control is set to display only active Entity dimension members. If only active entities are displayed, `GetOBPInfo` also returns the member IDs of the Scenario, Year, and Period dimension members that define which entities are considered active for display.

For applications in which Organization by Period is enabled, a user can choose to view only active entities when selecting a member list for the Entity dimension. In addition, active entity viewing can be programmatically specified with `SetOBPInfo`.

Note: To test whether Organization by Period is enabled, use `HsvMetadata.IsOrgByPeriodApplication`.

Syntax

```
<HsvPOVSelection>.GetOBPInfo pvarbEnabled, plScenario, plYear, plPeriod
```

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

<i>pvarbEnabled</i>	Boolean. Indicates whether only active Entity members are displayed. Returns TRUE if only active entities are displayed, FALSE otherwise.
---------------------	---

<i>plScenario</i>	Long. Returns the member ID of the Scenario dimension member specified for active entity viewing. If active entity viewing is not enabled, the HFMConstants type library constant <code>MEMBERNOTUSED</code> is returned.
-------------------	---

Note: For information on `MEMBERNOTUSED`, see “Dimension Member Constants” on page 834.

<i>plYear</i>	Long. Returns the member ID of the Year dimension member that the user specified for active entity viewing. If active entity viewing is not enabled, the HFMConstants type library constant <code>MEMBERNOTUSED</code> is returned.
---------------	---

<i>plPeriod</i>	Long. Returns the member ID of the Period dimension member that the user specified for active entity viewing. If active entity viewing is not enabled, the HFMConstants type library constant <code>MEMBERNOTUSED</code> is returned.
-----------------	---

Initialize

Displays the `HsvPOVSelection` control. The control displays the dimension members for the application represented by the specified `HsvSession` instance.

Note: You must call `EnableDimension` before calling `Initialize`.

Syntax

```
<HsvPOVSelection>.Initialize plHsvSessionUnk
```

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

<i>plHsvSessionUnk</i>	<code>HsvSession</code> object (ByVal). The <code>HsvSession</code> instance for the application.
------------------------	---

Argument	Description
	Tip: HsvSession object references are returned by the <code>OpenApplication</code> methods of the <code>HsxClient</code> and <code>HsxClientUI</code> objects.

Example

`Initialize` is used in the example for [EnableDimension](#).

InitializeEx

For internal use.

InitializeHTTP

For internal use.

InitializeWithConnection

For internal use.

SelectDimension

Displays the tab for a given dimension.

Syntax

```
<HsvPOVSelection>.SelectDimension lDim
```

Argument	Description
<i>lDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the <code>HFMConstants</code> type library constants listed in "Dimension ID Constants" on page 834 .

Example

The following example displays the Entity dimension tab.

```
'cFormPOV represents an initialized HsvPOVSelection control
cFormPOV.SelectDimension DIMENSIONENTITY
```

SetCheckedItems

Selects or clears check boxes for the specified dimension members when the tab is configured for multi-select mode.

Tip: To return the member IDs of members for which check boxes are selected, use [GetCheckedItems](#).

Syntax

```
<HsvPOVSelection>.SetCheckedItems lDim, varalItemIDs, varalParentIDs,  
bCheck, bClearAllExistingCheckedItems
```

Argument	Description
<i>lDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 834 .
<i>varalItemIDs</i>	Long array (ByVal). The member IDs of the members for which check boxes are being selected or cleared.
<i>varalParentIDs</i>	Variant (ByVal). The member IDs for the parents of the <i>varalItemIDs</i> argument’s entities, if the <i>lDim</i> argument specifies the Entity dimension. For the Entity dimension, you must pass a Long array of parent member IDs. The array should have a one-to-one correspondence with the <i>varalItemIDs</i> argument’s array. Tip: If an Entity member does not have a parent, pass the HFMConstants type library constant MEMBERNOTUSED, which is described in “Dimension Member Constants” on page 834 . If the <i>lDim</i> argument specifies a dimension other than Entity, this argument is ignored. However, you must pass a value such as an empty Variant or a Long array.
<i>bCheck</i>	Boolean (ByVal). A flag that specifies whether to select or clear check boxes for the specified members. Pass TRUE to select the check boxes, FALSE to clear them.
<i>bClearAllExistingCheckedItems</i>	Boolean (ByVal). A flag that specifies whether to clear check boxes that had been selected before the call to <i>SetCheckedItems</i> . Pass TRUE to clear previously selected check boxes, FALSE otherwise.

Example

The following subroutine uses member labels to select check boxes for specified members. IHsvTreeInfo.[GetItemID](#) gets the member IDs of the labels passed to the subroutine, and these IDs are passed to *SetCheckedItems*.

```
Sub selectMembers(lDim As Long, vaMems, vaPars)  
Dim cTreeInfo As IHsvTreeInfo, laIds() As Long  
Dim laParIds() As Long  
'g_cMetadata is a previously set HsvMetadata instance  
Set cTreeInfo = g_cMetadata.Dimension(lDim)  
ReDim laIds(UBound(vaMems))  
For i = LBound(vaMems) To UBound(vaMems)  
    laIds(i) = cTreeInfo.GetItemID(CStr(vaMems(i)))  
Next  
If lDim = DIMENSIONENTITY Then  
    ReDim laParIds(UBound(vaPars))  
    For i = LBound(vaPars) To UBound(vaPars)  
        If Len(vaPars(i)) > 0 Then  
            laParIds(i) = cTreeInfo.GetItemID(vaPars(i))  
        Else
```

```

        laParIds(i) = MEMBERNOTUSED
    End If
Next
End If
'cFormPOV represents an initialized HsvPOVSelection control
cFormPOV.SetCheckedItems lDim, laIds, laParIds, True, True
End Sub

```

SetDimUnk

For internal use.

SetListInfo

Specifies the member list to display for a given dimension, as well as the top member of the list to display.

Syntax

```
<HsvPOVSelection>.SetListInfo lDim, lListID, lListTopMemberID
```

Argument	Description
<i>lDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 834 .
<i>lListID</i>	Long (ByVal). The ID of the member list to display.
<i>lListTopMemberID</i>	Long (ByVal). The member ID of the member to display. If you do not want to specify a top member, pass the HFMConstants type library constant MEMBERNOTUSED. For information on this constant, see “Dimension Member Constants” on page 834 .

Example

The following subroutine displays the specified member list and top member. The subroutine takes the list name and the member label. It then uses some IHsvTreeInfo methods to get the corresponding IDs, which are passed to SetListInfo.

```

Sub selectList(lDim As Long, sList As String, sTopMem As String)
Dim cTreeInfo As IHsvTreeInfo, lList As Long, lTopMem As Long
'g_cMetadata is a previously set HsvMetadata instance
Set cTreeInfo = g_cMetadata.Dimension(lDim)
cTreeInfo.GetMemberListID sList, lList
lTopMem = cTreeInfo.GetItemID(sTopMem)
cFormPOV.SetListInfo lDim, lList, lTopMem
End Sub

```

SetMember

Selects a member label on a given dimension's tab when the tab is configured for single-select mode.

Note: In tabs configured for multi-select mode, `SetMember` does not select the specified member's check box. To select check boxes, use `SetCheckedItems`.

To get the member ID of a dimension's currently selected member label, use `GetMember`.

Syntax

```
<HsvPOVSelection>.SetMember lDim, lItemID, lParentID
```

Argument Description

<i>lDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the <code>HFMConstants</code> type library constants listed in "Dimension ID Constants" on page 834.
<i>lItemID</i>	Long (ByVal). The member ID of the member to be selected.
<i>lParentID</i>	Long (ByVal). For the Entity dimension, the member ID of the <i>lItemID</i> argument's parent. This argument is ignored for dimensions other than Entity. However, you must pass a valid Long.

SetOBPInfo

Specifies whether the `HsvPOVSelection` control displays only active Entity dimension members. If you specify to display only active entities, you also can specify the Scenario, Year, and Period dimension members that determine which entities are considered active.

Note: To test whether Organization by Period is enabled, use `HsvMetadata.IsOrgByPeriodApplication`.

Syntax

```
<HsvPOVSelection>.SetOBPInfo varbEnabled, lScenario, lYear, lPeriod
```

Argument Description

<i>varbEnabled</i>	Boolean (ByVal). Specifies whether the control displays only active Entity members. Pass <code>TRUE</code> to display only active entities, <code>FALSE</code> otherwise. When <code>TRUE</code> is passed, the Active Entities Only check box is selected on the dialog box for selecting Entity dimension member lists; the specified Scenario, Year, and Period dimensions are also displayed.
<i>lScenario</i>	Long (ByVal). The member ID of the Scenario dimension member.
<i>lYear</i>	Long (ByVal). The member ID of the Year dimension member.
<i>lPeriod</i>	Long (ByVal). The member ID of the Period dimension member.

Example

The following function enables active Entity viewing for the specified Scenario, Year, and Period dimension members. The function uses `IHsvTreeInfo.GetItemID` to get the member IDs for the passed Scenario, Year, and Period member labels. These IDs are then passed to `SetOBPInfo`.

```
Sub setObpFromLabels(sScen As String, sYear As String, _
    sPer As String)
    Dim cTreeInfo As IHsvTreeInfo, lScen As Long
    Dim lYear As Long, lPer As Long
    'g_cMetadata is a previously set HsvMetadata instance
    Set cTreeInfo = g_cMetadata.Scenarios
    lScen = cTreeInfo.GetItemID(sScen)
    Set cTreeInfo = g_cMetadata.Years
    lYear = cTreeInfo.GetItemID(sYear)
    Set cTreeInfo = g_cMetadata.Periods
    lPer = cTreeInfo.GetItemID(sPer)
    'cFormPOV represents an initialized HsvPOVSelection control
    cFormPOV.SetOBPInfo True, lScen, lYear, lPer
End Sub
```

SetResourceAndLanguage

Sets the control to display in a given language.

You can use only languages into which the Financial Management release has been localized. If a release has not been localized, English is the default language.

Note: This method does not set the language in which member descriptions are displayed. To specify a language for member descriptions, use the `HsvSystemInfo` method [SetLanguageUserParameters](#).

Syntax

```
<HsvPOVSelection>.SetResourceAndLanguage pUnkHsvResourceManager,
lHFMLanguage
```

Argument	Description
<i>pUnkHsvResourceManager</i>	<code>HsvResourceManager</code> object (ByVal). An <code>HsvResourceManager</code> object reference. For more information, see “HsvResourceManager Object Methods” on page 812 .
<i>lHFMLanguage</i>	Long (ByVal). The ID of the language. To obtain IDs of the languages into which the release has been localized, use the <code>HsvResourceManager</code> method GetAvailableLanguages .

ShowCheckBoxes

Configures a given dimension’s tab for multi-select or single-select mode.

In multi-select mode, check boxes are displayed so that users can select multiple members. In single-select mode, users can select one member; no check boxes are displayed, and users select members by highlighting member labels.

In multi-select mode, get the selected members with [GetCheckedItems](#). In single-select mode, get the selected member with [GetMember](#).

Note: In multi-select mode, the dimension's tab also displays buttons for selecting and clearing all members.

Syntax

```
<HsvPOVSelection>.ShowCheckBoxes lDim, bShow
```

Argument Description

<i>lDim</i>	Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 834 .
<i>bShow</i>	Boolean (ByVal). A flag that specifies the tab's selection mode. Pass TRUE to specify multi-select mode, FALSE to specify single-select mode.

Example

ShowCheckBoxes is used in the example for [EnableDimension](#).

HsvPOVSelection Object Events

The HsvPOVSelection object's events are summarized in [Table 55 on page 117](#), and are described in detail in the following topics.

CheckBoxChanged

Fires when the user selects or clears a member's check box. The event procedure's parameters provide the member's dimension ID and member ID, as well as a flag indicating whether the check box was selected or cleared.

Note: This event fires only when the user selects one check box. If the user clicks the button for selecting or clearing all check boxes, the [MultiCheckBoxChanged](#) event fires.

Syntax

```
<HsvPOVSelection>_CheckBoxChanged(lDim As Long, lItemID As Long, lParentID As Long, bNewCheckState As Boolean)
```

Argument	Description
<i>IDim</i>	Long (ByVal). The ID of the member's dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 834 .
<i>ItemID</i>	Long (ByVal). The member ID of the member.
<i>ParentID</i>	Long (ByVal). The parent of the <i>ItemID</i> argument's member. For dimensions other than Entity, or for Entity members without parents, the system passes the HFMConstants type library constant <code>MEMBERNOTUSED</code> to this argument. For information on this constant, see "Dimension Member Constants" on page 834 .
<i>bNewCheckState</i>	Boolean (ByVal). A flag that indicates whether the check box was selected or cleared. TRUE indicates selected, FALSE indicates cleared.

Example

The following event procedure prints to Visual Basic's Immediate window the label of the member for which the check box was changed and whether the check box was selected or cleared. If the member has a parent, the parent's label is also printed. The `IHsvTreeInfo` method [GetLabel](#) gets the labels from the member IDs passed to the event procedure.

```
Private Sub cFormPOV_CheckBoxChanged(ByVal lDim As Long, _
    ByVal lItemID As Long, ByVal lParentID As Long, _
    ByVal bNewCheckState As Boolean)
    Dim cTreeInfo As IHsvTreeInfo, cMetadata As HsvMetadata
    Dim sMemLabel As String, sParLabel As String
    'cSession is a previously set HsvSession instance
    Set cMetadata = cSession.Metadata
    Set cTreeInfo = cMetadata.Dimension(lDim)
    cTreeInfo.GetLabel lItemID, sMemLabel
    'if the member has a parent, get the parent's label
    If lParentID <> MEMBERNOTUSED Then
        cTreeInfo.GetLabel lParentID, sParLabel
        sMemLabel = sParLabel & "." & sMemLabel
    End If
    Debug.Print sMemLabel & "=" & bNewCheckState
End Sub
```

DimensionWasSelected

Fires when the tab for a dimension is selected by either the user or a call to [SelectDimension](#). The event procedure's parameter provides the ID of the dimension that was selected.

Syntax

```
<HsvPOVSelection>_DimensionWasSelected(lDim)
```

Argument Description

<i>IDim</i>	Long (ByVal). The ID of the selected dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 834 .
-------------	---

MultiCheckBoxChanged

Fires when the user clicks either the button to select all check boxes or the button to clear them all. The event procedure's parameters provide the ID of the dimension and a flag that indicates which button was clicked.

Syntax

```
<HsvPOVSelection>_MultiCheckBoxChanged(lDim As Long, bNewCheckState As Boolean)
```

Argument	Description
<i>lDim</i>	Long (ByVal). The ID of the member's dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 834 .
<i>bNewCheckState</i>	Boolean (ByVal). A flag that indicates which button was clicked. TRUE indicates the button to select all check boxes, FALSE indicates the button to clear all check boxes.

SelectionChanged

Fires when the user selects a member's label. The event procedure's parameters provide the ID of the dimension and the member IDs of the previously selected and currently selected member labels.

Note: This event also fires the first time that a dimension's tab is selected.

Syntax

```
<HsvPOVSelection>_ SelectionChanged(lDim As Long, lPrevItemID As Long, lPrevParentID As Long, lNewItemID As Long, lNewParentID As Long)
```

Argument	Description
<i>lDim</i>	Long (ByVal). The ID of the member's dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 834 .
<i>lPrevItemID</i>	Long (ByVal). The member ID of the previously selected member.
<i>lPrevParentID</i>	Long (ByVal). The parent of the <i>lPrevItemID</i> argument's member. For dimensions other than Entity, or for Entity members without parents, the system passes the HFMConstants type library constant <code>MEMBERNOTUSED</code> to this argument. For information on this constant, see "Dimension Member Constants" on page 834 .
<i>lNewItemID</i>	Long (ByVal). The member ID of the currently selected member.
<i>lNewParentID</i>	Long (ByVal). The parent of the <i>lNewItemID</i> argument's member. For dimensions other than Entity, or for Entity members without parents, the system passes the HFMConstants type library constant <code>MEMBERNOTUSED</code> to this argument.

Example

The following event procedure prints the labels of the previously and currently selected members to Visual Basic's Immediate window. The IHsvTreeInfo method [GetItemID](#) is used to get the member labels from the member IDs passed to the event procedure.

```
Private Sub cFormPOV_SelectionChanged(ByVal lDim As Long, _
    ByVal lPrevItemID As Long, ByVal lPrevParentID As Long, _
    ByVal lNewItemID As Long, ByVal lNewParentID As Long)
    Dim cTreeInfo As IHsvTreeInfo, cMetadata As HsvMetadata
    Dim sPrevLabel As String, sNewLabel As String
    Dim sPrevParLabel As String, sNewParLabel As String
    'cSession is a previously set HsvSession instance
    Set cMetadata = cSession.Metadata
    Set cTreeInfo = cMetadata.Dimension(lDim)
    cTreeInfo.GetLabel lPrevItemID, sPrevLabel
    cTreeInfo.GetLabel lNewItemID, sNewLabel
    Debug.Print "Previous label: " & sPrevLabel
    Debug.Print "New label: " & sNewLabel
End Sub
```

Error Handling and the HsvResourceManager Type Library

In This Chapter

Error Handling with Financial Management.....	811
HsvResourceManager Object Methods.....	812
Error Message Lookup Utility.....	822

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 Visual Basic and 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 `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.

To use the `HsvResourceManager` type library, you must reference `HsvResourceManager.dll` in your project. The `HsvResourceManager` type library contains one object—the `HsvResourceManager` object.

To obtain an `HsvResourceManager` object reference in Visual Basic or VBScript, use the `New` keyword as shown in the following example:

```
Dim cResourceManager As HsvResourceManager
Set cResourceManager = New HsvResourceManager
```

Caution! In Visual Basic, do not use `CreateObject`. Using `CreateObject` causes an error.

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 Call `Initialize` to specify the Financial Management tier for which you want to use resources. You must call `Initialize` before calling the other `HsvResourceManager` methods.
- 2 Do one of the following:
 - Pass the XML string and the `HRESULT` to `GetFormattedError` or `GetFormattedErrorWithLineFeed`.
 - Pass the `HRESULT` to `GetStringFromHR`.

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 returns strings of technical information for system messages such as errors. These strings contains a uniquely identifying Error Reference Number, followed by various fields of information. These fields are delimited by semicolons, 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: GSZABO1; File:  
CHsxServerImpl.cpp; Line: 1842; Ver: 3.0.0.196;
```

The following table describes the fields:

Table 84 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
----------	-------------

<code>pvarastrLanguagesIDs</code>	Variant. Returns an array of language IDs that identify the languages for which resources are available.
-----------------------------------	--

Argument	Description
	Tip: The <code>HFMConstants</code> enumeration <code>tagHFM_LANGUAGES</code> represents language IDs that are valid for all releases; for more information, see “Supported Language Constants” on page 833 .
<code>pvarastrLanguages</code>	Variant. Returns an array of strings that describe the languages for which resources are available.

Example

`GetAvailableLanguages` is used in the example for [GetFormattedError](#).

GetCurrentHSSRegistrationVersion

For internal use.

GetCurrentVersion

Returns the version number of Financial Management that is displayed to end-users.

Note: To return the version number displayed to end-users, use [GetCurrentVersionInUserDisplayFormat](#).

Syntax

```
<HsvResourceManager>.GetCurrentVersion()
```

Return Value

Variant. Returns the version number as a String subtype.

GetCurrentVersionInUserDisplayFormat

Returns the version number of Financial Management that is displayed to end-users.

Note: To return the full version number, including the build number, use [GetCurrentVersion](#).

Syntax

```
<HsvResourceManager>.GetCurrentVersionInUserDisplayFormat()
```

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 878 .
--------------------	--

<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, dTimeStamp)
End Function
```

GetFormattedDateTimeForLanguage

Returns a string representation of a double-byte date/time value.

Syntax

```
<HsvResourceManager>.GetFormattedDateTimeForLanguage(lLanguageId,
dDateTime, bDateValueOnly, bTimeValueOnly)
```

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 833 .
--------------------	---

<i>dDateTime</i>	Double (ByVal). The timestamp.
------------------	--------------------------------

<i>bDateValueOnly</i>	Long (ByVal). TRUE to return only the date value.
-----------------------	---

<i>bTimeValueOnly</i>	Long (ByVal). TRUE to return only the time value.
-----------------------	---

Return Value

String

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 833 .
<i>hr</i>	Long (ByVal). The HRESULT that identifies the error. Tip: 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. Tip: 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>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 813 .

Example

The following function takes a language ID and a tier 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 getHFMError(lId As Long, iTier As Integer) As String
Dim cResourceManager As HsvResourceManager, vaIDs, vaNames
Dim lLanguageID As Long, vUserError, vTechError
Set cResourceManager = New HsvResourceManager
cResourceManager.Initialize iTier
cResourceManager.GetAvailableLanguages vaIDs, vaNames
' 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 i
cResourceManager.GetFormattedError lLanguageID, Err.Number, _
    Err.Description, "Unknown Error", vUserError, vTechError
getHFMError = 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 833 .
<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.

Argument	Description
<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 813 .

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(1LanguageId)
```

Argument Description

1LanguageId 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 833](#).

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>bstLanguages</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 833 .
--------------------	---

Return Value

String. Returns the language code.

GetLocaleIdFromLanguageId

Returns a Windows locale ID for the given Financial Management language ID.

Syntax

```
<HsvResourceManager>.GetLocaleIdFromLanguageId (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 833 .
--------------------	---

Return Value

Long

GetResourceString

For internal use.

GetResourceStringFromHR

Returns the resource string for an HRESULT in the specified language.

Syntax

```
<HsvResourceManager>.GetResourceStringFromHR (lLanguageId, hr)
```

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

<i>lLanguageId</i>	Long (ByVal). Identifies the language for which the error message string will be returned. Pass one of the HFMCConstants type library constants listed in “Supported Language Constants” on page 833 .
--------------------	--

<i>hr</i>	Long (ByVal). The HRESULT that identifies the error.
-----------	--

Tip: In Visual Basic and 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 878](#).

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 833](#).

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.
--------------------------	---

GetWindowsDateFormatForLocale

Returns the Windows date format for the given locale.

Syntax

```
<HsvResourceManager>.GetWindowsDateFormatForLocale lLocaleId,  
pvarbstrDateSeparator, pvarbstrDateOrder
```

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

<i>lLocaleId</i>	Long (ByVal). Windows locale ID. Can be obtained by calling <i>GetLocaleId</i> from <i>LanguageId</i> .
------------------	---

<i>pvarbstrDateSeparator</i>	Variant array (ByVal). Returns a string containing the date separator character.
------------------------------	--

<i>pvarbstrDateOrder</i>	Variant array (ByVal). 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 so you must call `Initialize` before calling any of the other `HsvResourceManager` methods.

Syntax

```
<HsvResourceManager>.Initialize sTier
```

Argument Description

sTier Integer (ByVal). Identifies the tier for which you want to use resources. Pass one of the HFMConstants type library constants listed in [“Tier Constants” on page 855](#).

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 HFMErrorsLookup.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.



Constants: The HFMCConstants Type Library

In This Appendix

Metadata-Related Constants	823
Cell Status Constants.....	841
Journal-Related Constants	847
Process Management Constants	853
Report Column Constants.....	854
Consolidation Type Constants	855
Tier Constants	855
Security Constants.....	856
Web Constants.....	863
Extracted File Encoding Constants	869
User Activity Constants	870
Task Status Constants.....	872
Log Severity Constants	873
Intercompany Transaction Constants	873
Miscellaneous Constants.....	877

This appendix lists constants for Financial Management. The DLL for the HFMCConstants type library is `HFMCConstants.dll`.

Note: The HFMCConstants 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 HFMCConstants type library includes numerous metadata-related constants. The following categories of metadata constants are provided:

- “Account Dimension Constants” on page 824
- “Custom Dimension Constants” on page 827
- “Consolidation Method Constants” on page 829
- “Currency Attribute Constants” on page 831

- [“Entity Dimension Constants” on page 831](#)
- [“View Dimension Constants” on page 832](#)
- [“Supported Language Constants” on page 833](#)
- [“Dimension-Related Constants” on page 833](#)
- [“Intercompany Partner Constants” on page 835](#)
- [“Period Dimension Constants” on page 836](#)
- [“Scenario Attribute Constants” on page 837](#)
- [“Member List Constants” on page 838](#)
- [“Value Dimension Constants” on page 838](#)
- [“IHsvTreeInfo Interface Constants” on page 840](#)
- [“Application Setting Attribute ID Constants” on page 840](#)

Account Dimension Constants

The HFMConstants type library provides the following types of constants for the Account dimension:

- [“Account Attribute Constants” on page 824](#)
- [“System Account Constants” on page 825](#)
- [“Account Member List Constants” on page 826](#)
- [“Account Type Constants” on page 826](#)

Account Attribute Constants

The following constants represent attributes of accounts.

Table 85 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.

Constant	Description
ATTRIB_ACCOUNT_ENABLE_CUSTOM4_AGGR	EnableCustom4Aggr attribute.
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 86 tagACCOUNTCONSTANTS Enumeration

Constant	Description
MEMBERACTIVESTATUS	Active account.
MEMBERCONSOL1	Consol1 account.
MEMBERCONSOL2	Consol2 account.
MEMBERCONSOL3	Consol3 account.
MEMBERCONSOLIDATIONMETHOD	Method account.

Constant	Description
MEMBERDIRECTPERCENTOWNERSHIP	DOWN account.
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 87 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 88 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 827](#)
- [“Custom 3 and 4 Dimension Member Lists” on page 828](#)
- [“Custom Dimension Attributes” on page 828](#)

Custom 1 and 2 Dimension Member Lists

The following constants represent system-defined member lists for the Custom 1 and Custom 2 dimensions.

Table 89 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 90 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 91 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 829](#)
- [“Consolidation Methods: Control Attribute Constants” on page 830](#)
- [“Consolidation Methods: ToPercentControlComp Attribute Constants” on page 830](#)
- [“Consolidation Methods: PercentConsol Attribute Constants” on page 830](#)

Consolidation Method Attribute Constants

The following constants represent the attributes of consolidation methods.

Table 92 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_TOPCTCTRLP	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 93 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 94 tagCONSOLMETHOD_TOPCTCTRLP Enumeration

Constant	Description
CONSOLMETHOD_TOPCTCTRLP_LESSTHAN	Less-than (<) operator.
CONSOLMETHOD_TOPCTCTRLP_LESSTHANOREQUAL	Less-than-or-equal-to (<=) operator.
CONSOLMETHOD_TOPCTCTRLP_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 95 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 Constants

The following constants represent currency attributes.

Table 96 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 831](#)
- [“Entity Member List Constants” on page 832](#)

Entity Attribute Constants

The following constants represent attributes of Entity dimension members.

Table 97 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 98 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 HFMCconstants type library provides the following types of constants for the View dimension:

- [“Frequencies Constants” on page 832](#)
- [“View Dimension Member Constants” on page 833](#)

Frequencies Constants

The following constants represent attributes of frequencies.

Table 99 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 100 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.
MEMBERSCENARIOVIEW	<Scenario View> member.
MEMBERYTD	YTD member.

Supported Language Constants

The following constants represent languages supported by Financial Management.

Table 101 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. This constant can also be used in cases where a given language is not applicable, such as with FindMatchingMembersFromHierarchyWildcard .

Dimension-Related Constants

The HFMConstants type library provides the following types of dimension-related constants:

- [“Dimension ID Constants”](#) on page 834

- [“Constant Representing All Dimensions” on page 834](#)
- [“Dimension Member Constants” on page 834](#)

Dimension ID Constants

The following constants represent internal IDs of Financial Management’s dimensions.

Table 102 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.
DIMENSIONSCENARIO	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 834](#), which represent specific dimensions).

Dimension Member Constants

The following constants can be useful when working with dimension members.

Table 103 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 HFMConstants type library provides the following types of constants for the Intercompany Partner dimension:

- [“Intercompany Partner Member Constants” on page 835](#)
- [“Intercompany Partner Member List Constants” on page 836](#)
- [“Intercompany Partner Attribute Constants” on page 836](#)

Intercompany Partner Member Constants

The following constants represent members of the Intercompany Partner dimension.

Table 104 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 105 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 Attribute Constants

The following table lists constants for Intercompany Partner dimension attributes.

Table 106 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 836](#)
- [“Period Member List Constants” on page 837](#)

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 107 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 108 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 109 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 HFMCConstants type library provides the following types of constants for the Value dimension:

- [“Value Dimension Member Constants” on page 838](#)
- [“Value Dimension Member List Constants” on page 839](#)

Value Dimension Member Constants

The following constants represent Value dimension members.

Table 110 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 111 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.

IHsvTreeInfo Interface Constants

The following constants can be used with the IHsvTreeInfo interface.

Table 112 tagIHSTREECONSTANTS Enumeration

Constant	Description
DEFAULT_PARENT_UNDEFINED	Represents a member with no defined default parent.
NUM_STANDARD_ITEM_COLUMNS	<i>For internal use.</i>
SIBLING_NONE	Represents no siblings.
TREE_POSITION_FIRST_SIBLING	Represents the first sibling.
TREE_POSITION_LAST_SIBLING	Represents the last sibling.
TREE_ROOT	Represents the root of the dimension hierarchy.

Application Setting Attribute ID Constants

The following constants represent IDs of application setting attributes.

Table 113 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.

Constant	Description
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.
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.

Cell Status Constants

The HFMCconstants type library provides the following types of status-related constants:

- [“Cell Calculation Status Constants” on page 842](#)
- [“Cell Status Constants” on page 842](#)
- [“Cell Metadata Status Constants” on page 843](#)

- “Subcube Period Calculation Status Constants” on page 843
- “Transaction Dimension Constants” on page 845
- “Cell Transaction Type Constants” on page 846
- “Calculation Status Types” on page 846
- “Additional Status Information Constant” on page 847

Cell Calculation Status Constants

The following constants represent calculation statuses of cells.

Table 114 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.
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 115 tagCALCSTATUSLOWBITS Enumeration

Constant	Description
CELLSTATUS_DERIVED	The cell's data is derived.
CELLSTATUS_HASICDETAILTRANS	The cell contains intercompany transactions.
CELLSTATUS_HASTRANSACTIONS	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 116 tagCALCSTATUSMIDDLEBITS Enumeration

Constant	Description
CELLSTATUS_DRILLABLE	The cell allows drill through to the source data.
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 117 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.
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>

Constant	Description
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 118 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.
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.

Constant	Description
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 Type Constants

The following constants represent transaction types for cells.

Table 119 tagTRANSACTIONTYPECONSTANTS Enumeration

Constant	Description
TRANSACTIONTYPE_AGGREGATED	The cell's data was calculated by aggregation.
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 Types

The following constants represent calculation statuses. For information on calculation statuses, see the *Oracle Hyperion Financial Management, Fusion Edition User's Guide*.

Table 120 tagCALCSTATUSSTATISTICS Enumeration

Constant	Description
CALCSTATUS_STATSCOL_ALL	All statuses.
CALCSTATUS_STATSCOL_CH	CH status.
CALCSTATUS_STATSCOL_CHND	CH ND status.
CALCSTATUS_STATSCOL_CN	CN status.

Constant	Description
CALCSTATUS_STATSCOL_CNND	CN ND status.
CALCSTATUS_STATSCOL_LOCKED	Locked status.
CALCSTATUS_STATSCOL_NODATA	NoData status.
CALCSTATUS_STATSCOL_OK	OK ND status.
CALCSTATUS_STATSCOL_OKND	OK status.
CALCSTATUS_STATSCOL_OKSC	OK SC status.
CALCSTATUS_STATSCOL_TR	TR status.
CALCSTATUS_STATSCOL_TRND	TR ND status.
CALCSTATUS_STATSCOL_MIN	Represents the lowest value that one of the status constants represents; use this and CALCSTATUS_STATSCOL_MAX to loop through the statuses.
CALCSTATUS_STATSCOL_MAX	Represents the highest value that one of the status constants represents.
CALCSTATUS_STATSCOL_UNKNOWN	Unknown calculation status.

Additional Status Information Constant

Additional cell status information, such as that returned by `HsvData.GetStatusEx`, is represented by the tag `CELLSTATUSEXBITS` enumeration. This enumeration currently contains the `CELLSTATUSEX_SUPPORTSICTRANS` constant, which represents a cell that supports intercompany transactions.

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 848
- “Balance Type Constants” on page 848
- “Journal Column Display Constants” on page 848
- “Debit/Credit Constants” on page 849
- “Period Status Constants” on page 849
- “Journal Report Display Option Constants” on page 850
- “Journal Report Sort Option Constants” on page 850
- “Journal Report Total Flag Constants” on page 850
- “Journal Status Constants” on page 851
- “Journal Type Constants” on page 851

- “Journal Web Session Parameter Constants” on page 851
- “Template Column Display Constants” on page 851
- “Template Type Constants” on page 852
- “Journal and Template Type Constants” on page 852

Journal Action Constants

The following constants represent journal actions.

Table 121 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 122 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 123 tagJOURNALCOLUMNSFORVIEWANDFILTER Enumeration

Constant	Description
COLUMN_JOURNALAPPROVEDBY	Approved By column.

Constant	Description
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 124 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 125 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 126 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 127 tagJOURNALREPORTSORTFLAGS Enumeration

Constant	Description
JOURNALREPORT_SORT_ASCENDING	Ascending sort order.
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 128 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 129 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 130 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.
JTF_REGULAR	Regular journal.
JTF_UNIT	<i>For internal use.</i>

Journal Web Session Parameter Constants

The following constants represent default settings for Web sessions.

Table 131 tagJOURNALWEBSSESSIONPARAMETERS 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 132 tagTEMPLATECOLUMNSFORVIEWANDFILTER Enumeration

Constant	Description
COLUMN_TEMPLATEBALANCEATTRIBUTE	Balance Type column.
COLUMN_TEMPLATEDESCRIPTION	Description column.
COLUMN_TEMPLATEGROUP	Group column.
COLUMN_TEMPLATELABEL	Label column.
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 133 tagTEMPLATETYPEFLAGS Enumeration

Constant	Description
TTF_RECURRING	Recurring template.
TTF_STANDARD	Standard template.

Journal and Template Type Constants

The following constants represent types of journals and templates. These constants apply to cases where a method can optionally extract journals and templates, such as `HsvJournals.EnumJournalIDsForExtractFilter`.

Table 134 tagJOURNALEXTRACTJOURNALTYPE Enumeration

Constant	Description
JOURNALEXTRACT_JOURNALTYPE_REGULAR	Journals.
JOURNALEXTRACT_JOURNALTYPE_STANDARD	Standard journal templates.
JOURNALEXTRACT_JOURNALTYPE_RECURRING	Recurring journal templates.

Process Management Constants

The HFMCconstants type library provides process management-related constants for actions and review levels.

Process Management Action Constants

The following constants represent process management actions.

Table 135 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 136 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.

Constant	Description
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.
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.

Report Column Constants

The following constants represent columns in reports.

Table 137 tagREPORTCOLUMNSFORVIEWANDFILTER 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.

Constant	Description
COLUMN_REPORTLINEITEMCUSTOM2	Custom 2 column.
COLUMN_REPORTLINEITEMCUSTOM3	Custom 3 column.
COLUMN_REPORTLINEITEMCUSTOM4	Custom 4 column.
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 138 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 139 tagHFM_TIERS Enumeration

Constant	Description
HFM_TIER1	Client tier.

Constant	Description
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 856](#)
- [“Role Constants” on page 856](#)
- [“Task Constants” on page 858](#)

Access Rights Constants

The following constants represent access rights.

Table 140 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 141 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

Constant	Description
HFM_ROLE_CONSOLIDATE_ALL	Consolidate All
HFM_ROLE_CONSOLIDATE_ALL_DATA	Consolidate
HFM_ROLE_CREATE_INTEGRATIONS	Create Integrations
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 IC 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

Constant	Description
HFM_ROLE_MANAGE_TEMPLATES	Manage Templates
HFM_ROLE_POST_JOURNALS	Post Journals
HFM_ROLE_PROCESS_FLOW_REVIEWER1	Reviewer 1
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 142 tagHFM_TASK_ENUM Enumeration

Constant	Description
HFM_NUM_TASKS	Represents the number of tasks for an application.
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

Constant	Description
HFM_TASK_DATA_EXPLORER_REFRESH_DATA	Refresh Data
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

Constant	Description
HFM_TASK_ICT_MANAGE_PERIODS	Intercompany Transaction - Manage Periods
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

Constant	Description
HFM_TASK_JOURNALS_SETUP_JOURNALS_GENERATE_RECURRING	Generate Recurring Journals
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

Constant	Description
HFM_TASK_TASK_AUDIT	Task Audit
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

Web Constants

The HFMConstants type library includes the following categories of constants for the Web components.

- [“Data Explorer Task Constants” on page 863](#)
- [“Data Explorer Process Management Constants” on page 864](#)
- [“Data Grid Definition Constants” on page 865](#)
- [“Data Grid Member Expansion Mode Constants” on page 865](#)
- [“Data Grid Dimension Expansion Mode Constants” on page 866](#)
- [“Data Grid Transaction Information Constants” on page 866](#)
- [“Data Information Display Constants” on page 866](#)
- [“Data Display Page Constant” on page 867](#)
- [“Member Display Constants” on page 867](#)
- [“Metadata Information Constants” on page 867](#)
- [“Document Type Constants” on page 868](#)
- [“Document File Type Constants” on page 869](#)

Note: There is also an enum named `WEBOM_POV_SELECTTYPES`, but this is only for internal use.

Data Explorer Task Constants

The following constants represent Data Explorer tasks.

Table 143 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 144 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 145 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_ROWDIRMS	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 146 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 147 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 148 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 149 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 150 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 and ways of sorting dimension members.

Table 151 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.

Constant	Description
WEBOM_METADATA_INFO_SORTBY_DESC	Sort members by 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 152 tagDOCUMENTTYPES Enumeration

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 workspace.
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.

Constant	Description
WEBOM_DOCTYPE_WORKSPACE	Workspace.

Document File Type Constants

The following constants represent the file types of documents.

Table 153 tagDOCUMENTFILETYPES Enumeration

Constant	Description
WEBOM_DOCFILETYPE_RPTDEF	Report definition file, <code>RPT</code> format.
WEBOM_DOCFILETYPE_RPTXML	Report definition file, <code>XML</code> format.
WEBOM_DOCFILETYPE_RPTHML	Report definition file, <code>HTML</code> 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 workspace.
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 154 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.

User Activity Constants

The following constants represent user activities.

Table 155 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.
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.

Constant	Activity
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.
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.

Constant	Activity
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.
USERACTIVITYCODE_SECURITY_SCAN	Security Scan.
USERACTIVITYCODE_TASK_AUDIT_PURGED	Purge Task Audit Records.
USERACTIVITYCODE_TRANSLATION	Translate.

Task Status Constants

The following constants represent task statuses.

Table 156 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.

Constant	Description
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.

Log Severity Constants

The following constants represent severity levels of logs.

Table 157 tagLOGTYPES Enumeration

Constant	Description
LOG_TYPE_ERROR	Error severity.
LOG_TYPE_ERROR_ACCUMULATE	Accumulate severity.
LOG_TYPE_INFORMATION	Informational severity.
LOG_TYPE_INFORMATION_ACCUMULATE	Informational-Accumulate severity.
LOG_TYPE_WARNING	Warning severity.
LOG_TYPE_WARNING_ACCUMULATE	Warning-Accumulate severity.
LOG_TYPE_TOTAL	Represents the total number of log types.

Intercompany Transaction Constants

The following enumerations contain constants for intercompany transactions:

- [“Lock Status Constants” on page 874](#)
- [“Period Status Constants” on page 874](#)
- [“Match/Validate Before Post Constants” on page 874](#)
- [“Match Option Constants” on page 874](#)
- [“Posting Status Constants” on page 875](#)
- [“Matching Status Constants” on page 875](#)
- [“Processing Action Constants” on page 875](#)
- [“Transaction Load Mode Constants” on page 876](#)
- [“Filtering and Sorting Options” on page 876](#)

- [“Process Status Constants” on page 877](#)
- [“Event Constants” on page 877](#)

Lock Status Constants

The following constants represent entity lock statuses for intercompany transactions.

Table 158 tagICMENTITYSTATUS Enumeration

Constant	Description
ICM_LOCKED	The entity is locked.
ICM_UNLOCKED	The entity is unlocked.
ICM_LOCKABLE	The entity can be locked.
ICM_UNLOCKABLE	The entity can be unlocked.

Period Status Constants

The following constants represent period statuses for intercompany transactions.

Table 159 tagICMPERIODSTATUS Enumeration

Constant	Description
ICM_CLOSED	The period is closed for intercompany transactions.
ICM_OPENED	The period is open for intercompany transactions.
ICM_UNOPENED	The period is has not been opened for intercompany transactions.

Match/Validate Before Post Constants

The following constants represent the valid Match/Validate Before Post settings for a period:

Table 160 tagICMVBPPSETTING Enumeration

Constant	Description
ICM_MVBP_OFF	Match/Validate Before Post is on.
ICM_MVBP_ON	Match/Validate Before Post is off.
Unmatch All Intercompany Transactions	Restrict.

Match Option Constants

The following constants represent intercompany matching options:

Table 161 tagICMMATCHOPTIONSENUM Enumeration

Constant	Description
ICT_MATCH_REFERENCE	Match by Reference ID.
ICT_MATCH_TRANSACTIONID	Match by Transaction ID.
ICT_SUPPRESS_INTER	<i>For internal use.</i>
ICT_SUPPRESS_INTRA	<i>For internal use.</i>

Posting Status Constants

The following constants represent posting statuses of intercompany transactions:

Table 162 tagICMPOSTSTATUS Enumeration

Constant	Description
ICM_POSTED	The transaction is posted.
ICM_UNPOSTED	The transaction is not posted.

Matching Status Constants

The following constants represent matching statuses of intercompany transactions:

Table 163 tagICMMATCHSTATUS Enumeration

Constant	Description
ICM_MATCH	The transaction is matched.
ICM_MISMATCH	The transaction is mismatched.
ICM_UNMATCH	The transaction is unmatched.

Processing Action Constants

The following constants represent types of intercompany processing actions:

Table 164 tagICMTRANSPROCESSACTION Enumeration

Constant	Description
ICM_AUTOMATCHTRANS	Auto-match the specified transactions.
ICM_DELETETRANS	Delete the specified transactions.
ICM_DELETETRANS_ALL	Delete all transactions for the specified scenario, year, and period.

Constant	Description
ICM_EDITTRANS	Edit the specified transaction.
ICM_MANUALMATCHTRANS	Manually match the specified transactions.
ICM_NEWTRANS	Create a new transaction.
ICM_POSTTRANS	Post the specified transactions.
ICM_POSTTRANS_ALL	Post all transactions for the specified scenario, year, and period.
ICM_SETREASONCODE	Set reason code.
ICM_UNMATCHTRANS	Unmatch the specified transactions.
ICM_UNMATCHTRANS_ALL	Unmatch all transactions for the specified scenario, year, and period.
ICM_UNPOSTTRANS	Unpost the specified transactions.
ICM_UNPOSTTRANS_ALL	Unpost all transactions for the specified scenario, year, and period.

Transaction Load Mode Constants

The following constants represent modes for loading intercompany transactions:

Table 165 tagICMLOADFILEENUM Enumeration

Constant	Description
ICT_LOAD_MERGE	Load the transactions, merging data for existing transactions.
ICT_LOAD_REPLACE	Load the transactions, replacing data for existing transactions.
ICT_SCAN_MERGE	Scan the transactions, merging data for existing transactions.
ICT_SCAN_REPLACE	Scan the transactions, replacing data for existing transactions.

Filtering and Sorting Options

The following constants represent filtering and sorting options for intercompany transactions:

Table 166 tagICM_MONITOR_FILTER_SORT_FLAGS Enumeration

Constant	Description
ICM_MONITOR_FILTER_LOCKABLE	Lockable lock status filter.
ICM_MONITOR_FILTER_LOCKED	Locked lock status filter.
ICM_MONITOR_FILTER_NOTSTARTED	Not Started process status filter.
ICM_MONITOR_FILTER_STARTED	Started process status filter.

Constant	Description
ICM_MONITOR_FILTER_UNLCKABL	Unlockable lock status filter.
ICM_MONITOR_NO_SORT	Do not sort.
ICM_MONITOR_PROC_SORT	Sort by process status.
ICM_MONITOR_SORT	Sort by locking status.
ICM_MONITOR_SORT_DESC	Sort in descending order.

Process Status Constants

The following constants represent process statuses.

Table 167 tagICMENTITYPROCESSTATUS Enumeration

Constant	Description
ICM_NOTSTARTED	Not Started status.
ICM_STARTED	Started status.

Event Constants

The following constants represent intercompany transaction-related events.

Table 168 tagICEVENTTYPE Enumeration

Constant	Description
ICM_EVENT_MATCH_RPT_BY_ACCT	Intercompany Transactions - Matching Report By Account.
ICM_EVENT_MATCH_RPT_BY_ID	Intercompany Transactions - Matching Report By Transaction ID.
ICM_EVENT_MONITOR	Monitor Intercompany Transactions.
ICM_EVENT_PROCESS_TRANS	Process Intercompany Transactions.
ICM_EVENT_SYSTEM_MATCH_RPT	Intercompany Transaction Report.

Miscellaneous Constants

The following constants have been categorized as “miscellaneous”:

- [“Showing Public and Private Documents” on page 878](#)
- [“Date and Time Format Constants” on page 878](#)
- [“Invalid ID Constant” on page 879](#)

- “Maximum String Length Constants” on page 879
- “Aggregation Type Constants” on page 880
- “Member ID Range” on page 881
- “Number Defaults Constants” on page 881
- “Share Calculation Ownership Constants” on page 881
- “Share Calculation Types Constants” on page 881
- “Validation Type Constants” on page 882
- “Default Security Class Constant” on page 883
- “Module ID Constants” on page 883

Showing Public and Private Documents

The following constants represent whether to show public or private documents. For example, these are used with `HsvReports.EnumDocumentsEx`.

Table 169 tagENUMSHOWPRIVATEDOCS Enumeration

Constant	Description
ENUMSHOWPRIVATEDOCS_DONTSHOW	Show only public documents.
ENUMSHOWPRIVATEDOCS_SHOW	Show only private documents.
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 170 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.

Invalid ID Constant

The tagHFMINVALIDIDS class contains the INVALID_ID constant. This constant indicates represents invalid IDs, and can be used when working with IDs of items such as dimensions and dimension members.

Maximum String Length Constants

The following constants represent the maximum length of various string-based fields.

Table 171 tagHFMSTRINGLENGTHLIMITS Enumeration

Constant	Description
MAXLEN_ANNOTATION	Maximum length of comments in process units.
MAXLEN_APPDESC	Maximum length of application descriptions.
MAXLEN_APPLABEL	Maximum length of application labels.
MAXLEN_CALC_ATTRIBUTE	Maximum length of calculation attributes.
MAXLEN_CLUSTER_NAME	Maximum length of cluster names.
MAXLEN_DECIMAL_DISPLAY	Maximum number of characters for a number-based string.

Constant	Description
MAXLEN_DECIMAL_PRECISION	Maximum number of characters to the right of the decimal character.
MAXLEN_DESCRIPTION_CELL	Maximum length of cell text descriptions.
MAXLEN_DESCRIPTION_LINEITEM	Maximum length of line item descriptions.
MAXLEN_ELEMENTDESC	Maximum length of metadata descriptions.
MAXLEN_ELEMENTDESC_PER	Maximum length of Period dimension member descriptions.
MAXLEN_ELEMENTLABEL	Maximum length of metadata labels.
MAXLEN_ITEMID_STRING	<i>For internal use.</i>
MAXLEN_SERVERNAME	Maximum length of registered application server names.
MAXLEN_SQLTABLENAME	Maximum length of table names in databases.
MAXLEN_USERDEF_ATTRIB	Maximum length of user-defined attributes of dimension members.
MAXLEN_VALUE_LINEITEM	Maximum length of line item values.
MAXLEN_XBRL_TAGS	Maximum length of XBRL tags in Account dimension members.
MAXLEN_YEAR	Maximum length of years.

Aggregation Type Constants

The following table represents the types of aggregation.

Table 172 tagHSV_AGGREGATION_TYPE Enumeration

Constant	Description
HSV_AGGR_TYPE_ADD	Aggregation by addition.
HSV_AGGR_TYPE_DIVIDE	Aggregation by division.
HSV_AGGR_TYPE_INVALID	Invalid aggregation type.
HSV_AGGR_TYPE_MULTIPLY	Aggregation by multiplication.
HSV_AGGR_TYPE_NONE	No aggregation.
HSV_AGGR_TYPE_PERCENT	Aggregation by percentage.
HSV_AGGR_TYPE_SUBTRACT	Aggregation by subtraction.
HSV_AGGR_TYPE_USEDEFAULT	Represents the default aggregation type.

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 173 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 174 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 175 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 176 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 177 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 178 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 Workspace

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	Workspaces
MODULEID_WORKSPACE_PREFERENCES	Workspace Preferences



Change History

In This Appendix

Changes for Release 11.1.3	887
Changes for Release 11.1.1	889
Changes for Release 9.3.1	891
Changes for Release 9.3.0.1	892
Changes for Release 4.1	897
Changes for Release 4.0	902
Changes for Release 3.5.1	907
Changes for Release 3.5	908
Changes for Release 3.4	910
Changes for Release 3.0.4	914
Changes for Release 3.0	916
Changes for Release 2.2	919
Changes for Release 2.1	921
Changes for Release 2.0	923
Changes for Release 1.2	929
Changes for Release 1.1	931

This Appendix lists changes to the Financial Management object model since release 1.0.

Changes for Release 11.1.3

The following topics list changes since release 11.1.1:

- [HsvData Type Library](#)
- [HsvJournals Type Library](#)
- [HsvMetadata Type Library](#)
- [HsvResourceManager Type Library](#)
- [HsvSecurityAccess Type Library](#)
- [HsvSystemInfo Type Library](#)
- [HsxClient Type Library](#)

HsvData Type Library

The following methods have been added:

- “ExtractDrillableRegions” on page 300
- “ExtractDrillableRegionsByURLNames” on page 301
- “GetAllURLNames” on page 312
- “GetURLByName” on page 349
- “GetURLsForCell” on page 350
- “LoadDrillableRegions” on page 355

HsvJournals Type Library

The method “GetJournal2” on page 409 has been added.

HsvMetadata Type Library

The following methods have been added:

- “EnumMembersWithAttribValue” on page 200.
- “GetFrequencyID” on page 174

HsvResourceManager Type Library

The following methods have been added:

- “GetWindowsDateFormatForLocale” on page 821
- “GetLocaleIdFromLanguageId” on page 819
- “GetFormattedDateTimeForLanguage” on page 815

HsvSecurityAccess Type Library

The following new methods are for internal use:

- “GenerateSecurityReportForBiPub” on page 468
- “SetRulesMode” on page 483
- “GetRulesMode” on page 472

HsvSystemInfo Type Library

The new method “[GetFormattedDateTime](#)” on page 513 has been added.

The new method “[AddTaskToRunningTasksAndUpdatePOV](#)” on page 492 is for internal use.

HsxClient Type Library

The new method “[EnumUsersOnSystemEx2](#)” on page 136 has been added:

HsxServer Type Library

The new method “[GetFileTransfer](#)” on page 154 is for internal use.

Changes for Release 11.1.1

The following topics list changes since release 9.3.1:

[HsvCalculate Type Library](#)

[HsvDataCubes Type Library](#)

[HsvICM Type Library](#)

[HsvMDArrays Type Library](#)

[HsvMetadata Type Library](#)

[HsvResourceManager Type Library](#)

[HsvRulesLoadACV Type Library](#)

[HsvSecurityAccess Type Library](#)

[HsvSession Type Library](#)

[HsvSystemInfo Type Library](#)

HsvCalculate Type Library

The following methods have been added:

- “[CalcEPU](#)” on page 377
- “[GetEPUInfo](#)” on page 384
- “[IsEntityAnEPUOwner](#)” on page 385
- The following new methods are only for internal use:
 - [LoadCalcManagerRules](#)
 - [LoadCalcManagerRules2](#)

HsvDataCubes Type Library

The following methods have been changed:

- “BeginEnumerationOfStoredData” on page 688
- “EndEnumerationOfStoredData” on page 688
- “GetOneCellFromStoredItem” on page 690
- “GetPOVFromStoredItem” on page 692

HsvICM Type Library

The following methods have been added:

- “OpenICPeriod2” on page 740
- “UpdatePeriodSettings2” on page 741

HsvMDArrays Type Library

The new method `CreateDataIndexListEx` is only for internal use.

HsvMetadata Type Library

The following changes have been added to the `IHsvTreeInfo` interface:

- “EnumIDsOfChildren” on page 196
- “EnumSortedIDsOfChildren” on page 202
- “EnumPhasedSubmissionStartYears” on page 260
- “GetPhasedSubmissionStartYear” on page 260
- “IsPhasedSubmissionEnabled” on page 266

HsvResourceManager Type Library

The new method `GetCurrentHSSRegistrationVersion` is only for internal use.

HsvRulesLoadACV Type Library

The new method `GetCalcRulesType` is only for internal use.

HsvSecurityAccess Type Library

The new method `GetCalcRulesType` is only for internal use.

HsvSession Type Library

The following new methods are only for internal use:

- [“LockMetadataLoadWithSystemChangeCheck”](#) on page 160
- [“UnlockMetadataLoad”](#) on page 160

HsvSystemInfo Type Library

The following new methods are for internal use:

- [GetLastModifiedDateForArtifact](#)
- [GetCalcRulesType](#)
- [GetWorkingDirectory](#)

Changes for Release 9.3.1

The following topics list changes since release 9.3.0.1:

- [“HsvSession Type Library”](#) on page 891
- [“HsvData Type Library”](#) on page 891
- [“HsvSecurityAccess Type Library”](#) on page 891
- [“HsvcDataLoad Type Library”](#) on page 892

HsvSession Type Library

[GetLicenseExpirationStatus](#) has been deprecated.

HsvData Type Library

The following methods have been added:

- [GetUnassignedGroups](#)
- [SetMinMaxPeriod](#)
- [GetStatusUsingPhaseID](#)

Data load options for phased submissions, decimal character, and thousands separator character have been added. See [Table 61, “Data Load Options,”](#) on page 354.

HsvSecurityAccess Type Library

The following methods have been added to the HsvSecurityAccess object:

- [RenameSecurityClass](#)

- [GetUserInfoFromUniqueID2](#)
- [EnumUsers3](#)
- [EnumUsersWithFilter2](#)
- [EnumUsersInSecurityClass3](#)

The following HsvSecurityAccess methods have been deprecated:

- [GetUserInfoFromUniqueID](#) - use [GetUserInfoFromUniqueID2](#).
- [EnumUsers2](#) - use [EnumUsers3](#).
- [EnumUsersWithFilter](#) - use [EnumUsersWithFilter2](#).

HsvcDataLoad Type Library

Data load options for phased submissions, decimal character, and thousands separator character have been added. See “[Data Load Options](#)” on page 778.

Changes for Release 9.3.0.1

The following topics list changes since release 4.1:

- “[Classic Applications](#)” on page 892
- “[HsxClient Type Library](#)” on page 893
- “[HsxClientUI Type Library](#)” on page 893
- “[HsvMetadata Type Library](#)” on page 893
- “[HsvData Type Library](#)” on page 894
- “[HsvProcessFlow Type Library](#)” on page 895
- “[HsvSystemInfo Type Library](#)” on page 895
- “[HsvSecurityAccess Type Library](#)” on page 894
- “[HsvMDArrays Type Library](#)” on page 896
- “[HsvICM Type Library](#)” on page 896
- “[HsvMetadataLoadACV Type Library](#)” on page 896
- “[HsvcDataLoad Type Library](#)” on page 897
- “[HsvSecurityLoadACV Type Library](#)” on page 897

Classic Applications

Some methods now are supported for only Classic applications. You can determine whether an application is a Classic application with the HsvSecurityAccess method [IsClassicHFMAApplication](#).

HsxClient Type Library

The following methods now apply only to Classic applications. For information on similar functionality for Performance Management Architect applications, see the *Hyperion Enterprise Performance Management Architect Web Services Developer's Guide*:

- [CreateApplicationCAS](#)
- [DeleteApplication](#)
- [RegisterApplicationCAS](#)

The following methods have been added:

- [IsValidApplication](#)
- The following new methods are only for internal use:
 - [EnumUserAppPreferences](#)
 - [UpdateUserAppPreferences](#)
 - [CreateApplicationCASWithAccessCode](#)
 - [DeleteApplicationWithAccessCode](#)
 - [RegisterApplicationCASWithAccessCode](#)

HsxClientUI Type Library

[DeleteApplication](#) now applies only to Classic applications.

HsvMetadata Type Library

The following changes have been made to the HsvMetadata type library:

- The following HsvMetadata object methods now apply only to Classic applications. For information on similar functionality for Performance Management Architect applications, see the *Hyperion Enterprise Performance Management Architect Web Services Developer's Guide*:
 - [Load](#)
 - [EnumLoadOptions](#)
- The following methods have been added to the HsvMetadata object:
 - [GetByIndexValidationAccount](#)
 - [GetFdmAppName](#)
 - [GetSupportSubmissionPhaseForAccountFlag](#)
 - [GetSupportSubmissionPhaseForCustom1Flag](#)
 - [GetSupportSubmissionPhaseForCustom2Flag](#)
 - [GetSupportSubmissionPhaseForCustom3Flag](#)
 - [GetSupportSubmissionPhaseForCustom4Flag](#)

- [GetSupportSubmissionPhaseForICPFlag](#)
- [GetUseSubmissionPhaseFlag](#)
- [LoadWithAccessCode](#) has been added to the HsvMetadata object, but is only for internal use.
- [GetSubmissionGroup](#) has been added to the HsvAccounts object.
- [GetSubmissionGroup](#) has been added to the HsvCustom object.
- [GetSubmissionGroup](#) has been added to the HsvICPs object.

HsvData Type Library

The following methods have been added to the HsvData object:

- [GetCalcStatusEx](#)
- [GetPhaseSubmissionGridForGivenScenarioPeriod](#)
- [SetPhaseSubmissionGridForGivenScenarioPeriod](#)
- [DMELoad](#)

HsvSecurityAccess Type Library

The following changes have been made to the HsvSecurityAccess object:

- The following HsvSecurityAccess object methods now apply only to Classic applications. For information on similar functionality for Performance Management Architect applications, see the *Hyperion Enterprise Performance Management Architect Web Services Developer's Guide*:
 - [AddSecurityClass](#)
 - [DeleteSecurityClass](#)
 - [InsertDefaultSecurityClass](#)
 - [SetSecurityClassLabel](#)
- The following methods have been added:
 - [EnumRolesForPrincipal](#)
 - [EnumRolesForUser](#)
 - [EnumSecurityClassRightsForPrincipal](#)
 - [EnumUsersWithFilter](#)
 - [GetAllSecurityClassRightsForConnectedUser](#)
 - [IsClassicHFMAApplication](#)
 - The internal use methods [AddSecurityClassWithAccessCode](#), [DeleteSecurityClassWithAccessCode](#), [InsertDefaultSecurityClassWithAccessCode](#), and [SetSecurityClassLabelWithAccessCode](#).

The following methods have been added to the IHsvDataSecurity interface:

- [GetProcessUnitAccessRightsAndStateEx](#)
- [GetProcessUnitAccessRightsEx](#)

HsvProcessFlow Type Library

The following methods have been added to the HsvProcessFlow object:

- [GetGroupPhaseFromCell](#)
- [GetPhasedSubmissionHistory](#)
- [GetPhasedSubmissionState](#)
- [GetPhasedSubmissionStateUsingPhaseID](#)
- [PhasedSubmissionApprove](#)
- [PhasedSubmissionApprove2](#)
- [PhasedSubmissionApproveEx](#)
- [PhasedSubmissionGetHistory2](#)
- [PhasedSubmissionGetHistory2UsingPhaseID](#)
- [PhasedSubmissionProcessManagementChangeStateForMultipleEntities2](#)
- [PhasedSubmissionPromote](#)
- [PhasedSubmissionPromote2](#)
- [PhasedSubmissionPublish](#)
- [PhasedSubmissionPublish2](#)
- [PhasedSubmissionPublishEx](#)
- [PhasedSubmissionReject](#)
- [PhasedSubmissionReject2](#)
- [PhasedSubmissionSignOff](#)
- [PhasedSubmissionSignOff2](#)
- [PhasedSubmissionStart](#)
- [PhasedSubmissionStart2](#)
- [PhasedSubmissionStartEx](#)
- [PhasedSubmissionSubmit](#)
- [PhasedSubmissionSubmit2](#)
- [ProcessManagementChangeStateForMultipleEntitiesEx](#)

HsvSystemInfo Type Library

[OutputSystemInfo](#) has been added.

HsvMDArrays Type Library

The following methods have been added to the HsvMDDataBuffer object:

- [EraseRecordFromPMBuffer](#)
- [GetPMErrorRecordCount](#)
- [GetPMRecordCount](#)
- [GetRecordFromPMBuffer](#)
- [GetRecordFromPMErrorBuffer](#)
- [InsertRecordIntoPMBuffer](#)
- [InsertRecordIntoPMErrorBuffer](#)

The following methods have been added to the HsvMDDataBufferLite object:

- [EraseRecordFromPMBuffer](#)
- [GetPMErrorRecordCount](#)
- [GetPMRecordCount](#)
- [GetRecordFromPMBuffer](#)
- [GetRecordFromPMErrorBuffer](#)
- [InsertRecordIntoPMBuffer](#)
- [InsertRecordIntoPMErrorBuffer](#)

The following methods have been added to the HsvICTransactionsData object:

- [GetPartnerQueryDimensionMemberIDs](#)
- [SetPartnerQueryDimensionMemberIDs](#)

HsvICM Type Library

[SavePeriodsSettings](#) has been added to the IHsvAdminICM interface.

HsvMetadataLoadACV Type Library

- The following HsvMetadataLoadACV methods apply only to Classic applications. For information on similar functionality for Performance Management Architect applications, see the *Hyperion Enterprise Performance Management Architect Web Services Developer's Guide*:
 - Load method
 - LoadOptions property

Note: LoadWithAccessCode has been added, but that is only for internal use.

HsvcDataLoad Type Library

The `DMELoad` method has been added. However, this method is only for internal use.

HsvSecurityLoadACV Type Library

The HsvSecurityLoadACV load option `HSV_SECURITYLOAD_OPT_SECURITY_CLASSES` supports loading security classes only for Classic applications. For information on similar functionality for Oracle Hyperion EPM Architect, Fusion Edition applications, see the *Hyperion Enterprise Performance Management Architect Web Services Developer's Guide*

Note: `LoadWithAccessCode` has been added, but that is only for internal use.

Changes for Release 4.1

The following topics list changes for release 4.1:

- “HsxClient Type Library” on page 897
- “HsxServer Type Library” on page 898
- “HsvSession Type Library” on page 898
- “HsvMetadata Type Library” on page 898
- “HsvData Type Library” on page 898
- “HsvCalculate Type Library” on page 898
- “HsvSecurityAccess Type Library” on page 899
- “HsvSystemInfo Type Library” on page 900
- “HsvICM Type Library” on page 900
- “HsvMDArrays Type Library” on page 901
- “HsvResourceManager Type Library” on page 902

HsxClient Type Library

The following changes have been made to the HsxClient object:

- The following methods have been added:
 - `CreateApplicationCAS`
 - `EnumProvisioningProjects`
 - `EnumUsersOnSystemEx`
 - `RegisterApplicationCAS`
- `CreateApplication` has been deprecated - use `CreateApplicationCAS`.

HsxServer Type Library

The following changes have been made to the HsxServer object:

- The `GetClusterInfo` method has been removed.
- The `GetClustersAndServers` method has been added.

HsvSession Type Library

The `GetLicenseExpirationStatus` method has been added to the HsvSession object.

HsvMetadata Type Library

The following changes have been made to the HsvMetadata type library:

- The `GetCalcAttribute` method has been added to the HsvAccounts object.
- The `EnumCurrencies2` method has been added to the HsvCurrencies object.
- The following methods have been added to the HsvMetadata object:
 - `GetAttributeValue`
 - `TranslateAttributeValueForDisplay`
- The `SupportsEmailAlerting` method has been added to the HsvScenarios object.
- The following methods have been added to the IHsvTreeInfo interface:
 - `FindByDesc`
 - `FindMatchingMembersFromHierarchyByDesc`
 - `FindMatchingMembersFromHierarchyWildcard`
 - `GetDefaultMemberID`

HsvData Type Library

The following methods have been added to the HsvData object:

- `GetCalcStatusStatistics`
- `GetMaxCellTextSize`
- `IsValidCellText`

HsvCalculate Type Library

The `FindOverlappingConsolidation` method has been added to the HsvCalculate object.

HsvSecurityAccess Type Library

The following changes have been made to the HsvSecurityAccess object:

- Several methods have been deprecated. The following table lists these methods and the methods that replace them. Note that if you call a deprecated method, it will return an error:

Deprecated Methods	New methods
AddApplicationAdministrator	AddApplicationAdministrator2
AddOrRemoveApplicationAdministrators	AddOrRemoveApplicationAdministrators2
AddOrRemoveRolesFromUser	AddOrRemoveRolesFromUser2
AddOrRemoveUsersFromRole	AddOrRemoveUsersFromRole2
AddUser	AddUser2
AddUserToRole	AddUserToRole2
EnumApplicationAdministrators	EnumApplicationAdministrators2
EnumUsers	EnumUsers2 and EnumUsersOrGroups
EnumUsersInRole	EnumUsersInRole2 and EnumUsersInRole3 Tip: The difference between these two methods is that EnumUsersInRole3 optionally returns information regarding users who belong to groups assigned to the specified role.
EnumUsersInSecurityClass	EnumUsersInSecurityClass2
GetApplicationAdministratorAccessForAllUsers	GetApplicationAdministratorAccessForAllUsers2
GetConnectedUser	GetConnectedUser2
GetRoleAccessForAllUsers	GetRoleAccessForAllUsers2
GetSecurityClassAccessForAllUsers	GetSecurityClassAccessForAllUsers2
GetUserAccessForAllRoles	GetUserAccessForAllRoles2
GetUserAccessForAllSecurityClasses	GetUserAccessForAllSecurityClasses2
GetUserID	GetUserSID
GetUserName	GetUserName2
RemoveApplicationAdministrator	RemoveApplicationAdministrator2
RemoveUser	RemoveUser2
RemoveUserFromRole	RemoveUserFromRole2
SetManySecurityClassRightsForUser	SetManySecurityClassRightsForUser2

Deprecated Methods	New methods
SetSecurityClassRightsForManyUsers	SetSecurityClassRightsForManyUsers2
SetSecurityClassRightsForUser	SetSecurityClassRightsForUser2

- In addition to the methods listed above, the following methods have been added to the HsvSecurityAccess object:
 - [EnumUserClassAccess](#)
 - [EnumUsersInGroup](#)
 - [GetIdentityTypes](#)
 - [GetUserInfoFromUniqueID](#)
 - [SetRolesForUser](#)
 - [SetUserClassAccess](#)
- The following methods are no longer supported:
 - AddUserEx
 - GetOwner
 - GetUserIDFromSID
 - IsValidWindowsUser
 - TakeOwnership
- Users are now identified with security identifiers (SIDs) instead of numeric user IDs. All methods that had taken or returned user IDs are deprecated.

HsvSystemInfo Type Library

The following methods have been added to the HsvSystemInfo object:

- [EnumRunningTasksEx](#)
- [EnumRunningTasksPOV](#)
- [GetModuleName](#)
- [SetCurrentModuleEx](#)
- [AddRefToHsxServer](#)
- [ReleaseHsxServer](#)
- [UpdateRunningTaskPOV](#)

HsvICM Type Library

The following changes have been made to the HsvICM type library:

- The following changes have been made to the HsvICM object:

- The functionality of the [GetICTransactions](#) method's second argument has changed; see the description for details.

Note: The change to this argument's functionality occurred in Release 4.0.1.

- The following methods have been added:
 - [GetEntitiesContacts](#)
 - [GetMonitorICSummary](#)
 - [GetMonitorICTransactions](#)
 - [GetMonitorICDetails](#)
 - [GetUnMatchTransactions](#)
 - [IsOneSideOfTransactionGroupWriteable](#)
 - [SetReasonCodeToICTransactions](#)
- The following changes have been made to the HsvAdminICM object:
 - The following methods have been added:
 - [CheckSecurityForICEExtract](#)
 - [GetICEEntitiesLockStatus](#)
 - The HsvAdminICM method [GetLockStatusICEntities](#) has been deprecated - use [GetICEEntitiesLockStatus](#).

HsvMDArrays Type Library

The following changes have been made to the HsvMDArrays type library:

- The following changes have been made to the HsvICTransactionsData object:
 - The following methods have been added:
 - [GetAccessRights](#)
 - [GetPartnerAsEntityList](#)
 - [GetTransGroupType](#)
 - [IsEntityInPartnerAsEntityList](#)
 - [SetAccessRights](#)
 - [SetPartnerAsEntityList](#)
 - [SetAccessRight](#) has been removed. To specify a user's access rights to a transaction, use [SetAccessRights](#).
- The following methods have been added to the HsvMDDataBuffer object. However, all of these methods are only for internal use:
 - [GetSortedNature](#)
 - [SetSortedNature](#)
 - [Sort](#)

- The following methods have been added to the `HsvMDDataBufferLite` object. However, all of these methods are only for internal use:
 - [GetSortedNature](#)
 - [SetSortedNature](#)
 - [Sort](#)

HsvResourceManager Type Library

The [GetLanguageCountryCodeFromLanguageId](#) method has been added to the `HsvResourceManager` object.

Changes for Release 4.0

The following topics list changes for release 4.0:

- [“HsvSession Type Library” on page 902](#)
- [“HsvMetadata Type Library” on page 903](#)
- [“HsvData Type Library” on page 903](#)
- [“HsvCalculate Type Library” on page 904](#)
- [“HsvJournals Type Library” on page 904](#)
- [“HsvSystemInfo Type Library” on page 904](#)
- [“HsvProcessFlow Type Library” on page 905](#)
- [“HsvReports Type Library” on page 905](#)
- [“HsvStarSchemaACM Type Library” on page 905](#)
- [“HsvICM Type Library” on page 906](#)
- [“HsvMDArrays Type Library” on page 906](#)
- [“HsvRulesLoadACV Type Library” on page 906](#)
- [“HsvcDataLoad Type Library” on page 907](#)
- [“HsvMetadataLoadACV Type Library” on page 907](#)
- [“HsvResourceManager Type Library” on page 907](#)
- [“HsxClient Type Library” on page 907](#)

HsvSession Type Library

The following items have been added to the `HsvSession` type library:

- [ICM](#) property
- [IsBusy](#) method
- [IsRunningTasks](#) method

HsvMetadata Type Library

- The `HsvCurrencies` object has been added. This object represents an application's currencies. For more information, see [“HsvCurrencies Object Methods” on page 270](#).
- An option for extracting system accounts has been added, and the index numbers for the Values and ICPs options have changed to 18 and 19. Extract options are described in [Table 57 on page 165](#).
- The following items have been added to the `HsvMetadata` object:
 - `GetApplicationCurrency` method
 - `Currencies` property
- The `IsSecurityAsPartnerEnabled` method has been added to the `HsvEntities` object.
- The following methods have been added to the `HsvValues` object:
 - `GetCurrencyIDFromValueID`
 - `GetValueIDFromCurrencyID`
- The following changes have been made to the `IHsvTreeInfo` interface:
 - The `lEntity` argument has been added to `EnumMembers2`.
 - The following methods have been added:
 - `EnumSortedIDsOfChildren`
 - `EnumSortedMembers`

HsvData Type Library

The following changes have been made to the `HsvData` object:

- The `pvbDataExistedPriorToClear` argument has been added to `ClearInputData`.
- The `vbEnableDetailedLogging` argument has been added to `CopyInputDataForMultipleEntities`.
- The following methods have been added:
 - `AttachDocumentToCell`
 - `DetachDocumentFromCell`
 - `GetAttachedDocumentsToCell`
 - `GetCountOfAttachedDocumentsToCell`
 - `ExtractDataAuditItems`
 - `GetStatusEx`
 - `SetCells2`
 - `SetFileForLoad`
 - `StartLoad`

HsvCalculate Type Library

The following methods have been added to the HsvCalculate object:

- [GetDefaultExchangeRate](#)
- [GetConsolidationProgress](#)
- [SetVBScriptRules2](#)
- [StopConsolidation](#)

HsvJournals Type Library

The following methods have been added to the HsvJournals type library:

- [GenerateRecurring](#) has been added to the IHsvJournalsEx interface.
- [GetReportData2](#) has been added to the IHsvJournalsReport interface.

HsvSystemInfo Type Library

The following changes have been made to the HsvSystemInfo object:

- The *pvarbstrModuleNames* argument has been added to [EnumAuditTasks](#).
- The *pvarbstrModuleNames* argument has been added to [EnumAuditTasks2](#).
- The following methods have been added:
 - [AddTaskToRunningTasks](#)
 - [ClearRunningTask](#)
 - [EnumActivityServers](#)
 - [EnumRunningTasks](#)
 - [ExtractTaskAudit](#)
 - [GetActivityCodeDesc](#)
 - [GetFormattedResourceString](#)
 - [GetRunningTaskLogFilePathName](#)
 - [GetRunningTaskProgress](#)
 - [GetRunningTasksCount](#)
 - [GetRunningTaskStatus](#)
 - [IsScheduledTaskReadyToRun](#)
 - [KeepRunningTaskStillAlive](#)
 - [StopRunningTask](#)
 - [UpdateRunningTaskLogFilePathName](#)
 - [UpdateRunningTaskProgress](#)

- [UpdateRunningTaskProgressDetails](#)
- [UpdateRunningTaskStatus](#)

HsvProcessFlow Type Library

The following methods have been added to the HsvProcessFlow object:

- [Approve2](#)
- [GetHistory2](#)
- [ProcessManagementChangeStateForMultipleEntities2](#)
- [Promote2](#)
- [Publish2](#)
- [Reject2](#)
- [SignOff2](#)
- [Start2](#)
- [Submit2](#)

Note: These new methods enable you to attach documents to process units and to get names and paths of attached documents.

HsvReports Type Library

The [CheckReportSecurityClass2](#) method has been added to the HsvReports object.

HsvStarSchemaACM Type Library

The following changes have been made to the HsvStarSchemaACM type library:

- The IHsvStarSchemaTemplates interface has been added. For more information, see “[IHsvStarSchemaTemplates Interface](#)” on page 706.
- The following changes have been made to the HsvStarSchemaACM object:
 - The following changes have been made to the [CreateStarSchema](#) method:
 - The data types for the arguments that specify the dimension members have been changed from Long to Variant. In addition, you can now specify dimension members in ways other than with member IDs; for example, you can use member lists and member labels.
 - The *eaExtractType* and *vbExcludeDynamicAccts* arguments have been added.
 - The *vbRunAsynchronous*, *varalssSelctionMethods*, and *plNumDataRecords* arguments have been removed.
 - The data type of the [GetAsynchronousTaskStatus](#) method’s *plCurrentTask* argument has been changed.

- The [GetExtractLogData](#) method has been added.
- For the SS_PUSH_OPTIONS enumeration, the `SSCLEAR_ALL_DATA` and `SSCLEAR_DATA_SYTE` constants have been removed, and the `SSCREATE` and `SSUPDATE` constants have been added. For more information, see the description of `CreateStarSchema`'s `ssPushType` argument.
- The EA_EXTRACT_TYPE_FLAGS enumeration has been added. For more information, see the description of the [CreateStarSchema](#) method's `eaExtractType` argument.
- The EA_TASK_STATUS_FLAGS enumeration has been added. For more information, see the description of the [GetAsynchronousTaskStatus](#) method's `plCurrentTask` argument.
- The SS_SELECTION_METHOD enumeration has been removed.
- The HFMPOVS type has been removed.

HsvICM Type Library

The HsvICM type library has been added. This library exposes Financial Management's intercompany transaction features. For more information, see [Chapter 19, "HsvICM Type Library."](#)

HsvMDArrays Type Library

The following changes have been made to the HsvMDArrays type library:

- The following methods have been added to the HsvMDDataBuffer object:
 - [InsertDataAtBeginning](#)
 - [InsertDescriptionAtBeginning](#)
 - [InsertLineItemsAtBeginning](#)
- The HsvICTransactionsData object has been added. This object contains methods that supplement the intercompany transaction features exposed by the HsvICM object. For more information, see "[HsvICTransactionsData Object Methods](#)" on page 674.

Note: The HsvMetadataSecurityBuffer object has also been added; however, this object is only for internal use.

HsvRulesLoadACV Type Library

The following methods have been added to the HsvRulesLoadACV object:

- [LoadCalcRules2](#)
- [ScriptableLoadCalcRules2](#)

HsvcDataLoad Type Library

The following methods have been added to the HsvcDataLoad object:

- [Load2](#)
- [SetFileForLoad](#)
- [StartLoad](#)

HsvMetadataLoadACV Type Library

An option for extracting system accounts has been added. Extract options are described in “[Metadata Extract Options](#)” on page 765.

HsvResourceManager Type Library

[GetCurrentVersionInUserDisplayFormat](#) has been added to the HsvResourceManager object.

HsxClient Type Library

The following methods have been added to the HsxClient object. However, all of these methods are only for internal use:

- [AuthenticateCSSToken](#)
- [AuthenticateSecurityAgentCredentials](#)
- [AuthenticateUserCredentials](#)

Changes for Release 3.5.1

The following topics list changes for release 3.5.1:

- “[HsvMetadata Type Library](#)” on page 907
- “[HsvData Type Library](#)” on page 908
- “[HsvPOVSelection Type Library](#)” on page 908
- “[HsvResourceManager Type Library](#)” on page 908
- “[Documentation Errors](#)” on page 908

HsvMetadata Type Library

The [GetDefaultHierarchyPosition](#) method has been added to the IHsvTreeInfo interface.

HsvData Type Library

The [GetDataForAllMetadataCombinations](#) method has been added to the HsvData object.

HsvPOVSelection Type Library

The HsvPOVSelection type library has been added. This library exposes an insertable control that provided a user interface for selecting dimension members. For more information, see [Chapter 21, “HsvPOVSelection Type Library.”](#)

HsvResourceManager Type Library

The following methods have been added to the HsvResourceManager object:

- [GetCurrentVersion](#)
- [GetFormattedDateTime](#)
- [GetFormattedErrorWithLineFeed](#)
- [GetFormattedResourceString](#)
- [GetHelpDirectoryForLanguageID](#)
- [GetHFMLanguageIdFromUserLanguages](#)
- [GetUserDisplayDateTimeFormats](#)
- [GetUserLanguageFromHFMLanguageId](#)
- [GetWindowsDateFormat](#)

Documentation Errors

The following documentation errors have been corrected:

- In [“Document File Type Constants” on page 869](#), `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 868](#), `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:

- [“HsxClient Type Library” on page 909](#)

- “HsvMetadata Type Library” on page 909
- “HsvData Type Library” on page 909
- “HsvSystemInfo Type Library” on page 910
- “HsvStarSchemaACM Type Library” on page 910
- “HsvReports Type Library” on page 910
- “HsvMDArrays Type Library” on page 910
- “HsvMetadataLoadACV Type Library” on page 910

HsxClient Type Library

The following methods have been added to the HsxClient object:

- `DoesUserHaveCreateApplicationRights`
- `DoesUserHaveSystemAdminRights`
- `GetHFMEErrorLogRecordSet`
- `GetSSOTokenUsingWebSecurityAgentCredentials`
- `GetWebSecurityAgentSettings`

HsvMetadata Type Library

The following methods have been added to the HsvMetadata object:

- `GetApplicationSettingsTimeStamp`
- `GetConsolidationMethodsTimeStamp`
- `GetCurrencyTimeStamp`

The following methods have been added to the IHsvTreeInfo interface:

- `GetTreeTimeStamp`
- `GetItemIDQL`
- `GetQualifiedLabel`

HsvData Type Library

The following methods have been added to the HsvData object:

- `ClearInvalidData`
- `GetCellHistory2`
- `EnumDataAuditItems2`

HsvSystemInfo Type Library

The following methods have been added to the HsvSystemInfo object:

- [ClearAuditTasks2](#)
- [EnumAuditTasks2](#)

HsvStarSchemaACM Type Library

The HsvStarSchemaACM type library has been added to this document. HsvStarSchemaACM exposes Financial Management's Extended Analytics features. For details, see [Chapter 18](#), "HsvStarSchemaACM Type Library."

HsvReports Type Library

The *lShowPrivateDocs* argument has been added to [EnumDocumentsEx](#).

HsvMDArrays Type Library

The following methods have been added to the HsvMDDataBuffer object:

- [GetCheckLineItemDetailsForCaseInsensitiveDuplicates](#)
- [SetCheckLineItemDetailsForCaseInsensitiveDuplicates](#)

The following methods have been added to the HsvMDDataBufferLite object:

- [GetCheckLineItemDetailsForCaseInsensitiveDuplicates](#)
- [SetCheckLineItemDetailsForCaseInsensitiveDuplicates](#)

HsvMetadataLoadACV Type Library

Load options for loading system accounts and validating metadata integrity have been added.

Changes for Release 3.4

The following topics list changes for release 3.4:

- "HsxClient Type Library" on page 911
- "HsxServer Type Library" on page 911
- "HsvSession Type Library" on page 911
- "HsvMetadata Type Library" on page 911
- "HsvData Type Library" on page 912
- "HsvJournals Type Library" on page 912

- [“HsvSystemInfo Type Library” on page 913](#)
- [“HsvProcessFlow Type Library” on page 913](#)
- [“HsvReports Type Library” on page 913](#)
- [“HsvJournalLoadACV Type Library” on page 913](#)
- [“HsvMDArrays Type Library” on page 914](#)

HsxClient Type Library

The following methods have been added to the HsxClient object:

- [DeleteSystemErrors](#)
- [DisableNewConnections](#)
- [EnableNewConnections](#)
- [EnumProhibitConnections](#)
- [EnumUsersOnSystem](#)
- [KillUsers](#)
- [WarnUsersForShutDown](#)

HsxServer Type Library

The [DeleteXMLErrorsFromDatabase](#) method has been removed from the HsxServer object.

The following members have been added to the HsxServer object:

- [CSSEnabled](#) property
- [DeleteSystemErrors](#) method

HsvSession Type Library

The [HasUserStatusChanged](#) method has been added to the HsvSession object.

HsvMetadata Type Library

The following methods have been added to the HsvScenarios object:

- [GetMissingDataZeroViewForAdjValues](#)
- [GetMissingDataZeroViewForNonAdjValues](#)
- [IsConsolidateYTD](#)

The following methods have been added to the IHsvTreeInfo interface:

- [EnumDefaultAncestors](#)
- [EnumDefaultAncestorsLabels](#)

- [EnumMembers2](#)
- [GetAttributeValue](#)
- [GetDefaultParent](#)
- [GetDefaultParentLabel](#)
- [GetItemGeneration](#)
- [GetItemLevel](#)

HsvData Type Library

The following methods have been added to the HsvData object:

- [ClearDataAuditItems](#)
- [EnumDataAuditItems](#)
- [FormatNumberToText2](#)
- [GetBaseDataForAccount](#)
- [GetCellHistory](#)

HsvJournals Type Library

The following changes have been made to IHsvJournalsEx interface methods:

- The *plSecurityClass* argument has been added to [GetTextTemplate](#).
- The *lSecurityClass* argument has been added to [SaveTextTemplate](#).
- The *plSecurityClass* argument has been added to [GetTemplate](#).
- The *lSecurityClass* argument has been added to [SaveTemplate](#).

The following methods have been added to the IHsvJournalsEx interface:

- [AddJournalGroup](#)
- [EnumJournalGroups](#)
- [EnumJournalGroupsForScenarioYear](#)
- [EnumJournalIDsForExtractFilter](#)
- [GetEntityJournals](#)
- [GetJournalLabelsForIDs](#)
- [GetTemplateLabelsForIDs](#)
- [RemoveAllJournalGroups](#)
- [RemoveJournalGroup](#)
- [ValidateLineItems](#)

HsvSystemInfo Type Library

The following methods have been added to the HsvSystemInfo object:

- [AddTaskToAudit](#)
- [CheckAccess](#)
- [ClearAuditTasks](#)
- [DisableNewConnections](#)
- [EnableNewConnections](#)
- [EnumActivityUsers](#)
- [EnumAuditTasks](#)
- [EnumProhibitConnections](#)
- [GetActivityUserID](#)
- [GetCurrentActivity](#)
- [GetKillUsersStatus](#)
- [GetKillUserStatus](#)
- [KillUsers](#)
- [SetCurrentActivity](#)
- [SetCurrentModule](#)
- [WarnUsersForShutDown](#)

HsvProcessFlow Type Library

The [ProcessManagementChangeStateForMultipleEntities](#) method has been added to the HsvProcessFlow object.

HsvReports Type Library

The following methods have been added to the HsvReports object:

- [CheckSecurityRole](#)
- [EnumDocumentsEx](#)
- [GetDocumentEx](#)
- [SaveDocumentEx](#)

HsvJournalLoadACV Type Library

The [ExtractOptionsEx](#) property and [ExtractEx](#) method have been added to the HsvJournalLoadACV object.

Note: `ExtractOptionsEx` and `ExtractEx` enable you to apply filtering criteria when extracting journals.

HsvMDArrays Type Library

The `HsvCubeData` object has been added. However, in this release the `HsvCubeData` object is only for internal use.

Changes for Release 3.0.4

The following topics list changes for release 3.0.4:

- [“HsxClient Type Library” on page 914](#)
- [“HsxServer Type Library” on page 914](#)
- [“HsvMetadata Type Library” on page 915](#)
- [“HsvData Type Library” on page 915](#)
- [“HsvSystemInfo Type Library” on page 915](#)
- [“HsvReports Type Library” on page 915](#)
- [“HsvRulesLoadACV Type Library” on page 916](#)

HsxClient Type Library

The following changes have been made to the `HsxClient` type library:

- The following methods have been added:
 - `AuthenticateUserOnClusterSSO`
 - `GetLogonInfoSSO`
 - `SetLogonInfoSSO`
- The following methods have been deprecated:
 - `GetLogonInfo`
 - `SetLogonInfo`

HsxServer Type Library

The following changes have been made to the `HsxServer` type library:

- The following methods have been added:
 - `GetHFMErrorsLogRecordSet`

HsvMetadata Type Library

The following changes have been made to the HsvMetadata type library:

- The following methods have been added to the HsvMetadata object:
 - `GetCurrencyValueIDForEntityValueCombination`
 - `GetCurrencyValueIDsForEntityValueCombinations`
- The following methods have been added to the HsvEntities object:
 - `IsOrgByPeriodFilteringOn`
- The following methods have been added to the IHsvTreeInfo interface:
 - `FindMatchingMembersFromHierarchy`
- The following methods have been added to the HsvICPs object:
 - `GetSecurityClassID`

HsvData Type Library

The following changes have been made to the HsvData type library:

- `EnumEntitiesWithDataForScenarioYear` has been added to the HsvData type object.
- The following sparse data-related methods have been added to the HsvData object:
 - `DoesSparseDataExist`
 - `FilterMembersThatHaveData`
 - `FilterMembersThatHaveData2`
 - `GetMembersThatHaveData`
- The following suppression-related methods have been added to the HsvData object:
 - `GetCellsWithRowSuppression2`
 - `GetTextCellsWithRowSuppression2`
- The following methods have been deprecated:
 - `CopyInputData`

HsvSystemInfo Type Library

The following methods have been added to the HsvSystemInfo object:

- `GetExtractFileEncoding`
- `SetExtractFileEncoding`

HsvReports Type Library

The following methods have been added to the HsvReports object:

- DeleteDocuments
- EnumDocuments
- GetDocument
- SaveDocument

HsvRulesLoadACV Type Library

SetSessionAndResource has been added to the HsvRulesLoadACV object.

Changes for Release 3.0

The following topics list changes for release 3.0:

- [“HsxClient Type Library” on page 916](#)
- [“HsxServer Type Library” on page 917](#)
- [“HsvMetadata Type Library” on page 917](#)
- [“HsvData Type Library” on page 918](#)
- [“HsvSecurityAccess Type Library” on page 918](#)
- [“HsvSystemInfo Type Library” on page 918](#)
- [“HsvMDArrays Type Library” on page 919](#)
- [“HsvMetadataLoadACV Type Library” on page 919](#)
- [“HsvRulesLoadACV Type Library” on page 919](#)
- [“Error Numbers” on page 919](#)

HsxClient Type Library

The following changes have been made to the HsxClient type library:

- The following methods have been added:
 - AuthenticateUserOnCluster
 - CreateObjectOnCluster
 - EnumRegisteredClusterNames
 - GetClusterInfo
 - GetServerOnCluster
 - RegisterCluster
 - ScriptableEnumRegisteredClusterNames
 - UnregisterAllClusters
 - UnregisterCluster

- The following methods have been deprecated:
 - CreateObjectOnServer
 - EnumRegisteredServerNames
 - GetServer
 - RegisterServer
 - UnregisterServer

HsxServer Type Library

The following methods have been added to the HsxServer type library:

- DeleteXMLErrorsFromDatabase
- GetClusterInfo
- GetXMLErrorFromDatabase
- GetXMLErrorsListFromDatabase

HsvMetadata Type Library

The following changes have been made to the HsvMetadata type library:

- The following methods have been added to the HsvAccounts object:
 - GetICPTopMember
 - GetIsICP
 - IsICPRestricted
- The following methods have been added to the IHsvTreeInfo interface:
 - GetAllPathsToMember
 - TranslateAttributeValueForDisplay
- The following methods have been added to the HsvCustom object:
 - GetSecurityClassID
 - GetUserDefined1
 - GetUserDefined2
 - GetUserDefined3
- The GetSecurityAsPartnerID method has been added to the HsvEntities object.
- The following methods have been added to the HsvScenarios object:
 - GetMaximumReviewLevel
 - UsesLineItems
- Options for loading system members, Value dimension members, and Intercompany Partner dimension members have been added.

- Options for extracting system members, Value dimension members, and Intercompany Partner dimension members have been added. For details, see [Table 57 on page 165](#).

HsvData Type Library

The following changes have been made to the HsvData type library:

- The `GetCellDescription` method has been added to the HsvData object.
- The `varlaParents` argument has been added to the `CopyInputDataForMultipleEntities` method.
- The following line item-related methods have been added to the HsvData object:
 - `GetCellLineItems`
 - `SetCellLineItems`
 - `GetTextCellLineItems`
 - `SetTextCellLineItems`
 - `InsertLineItemDetails`
 - `DeleteLineItemDetails`
 - `SetCellsLineItems`
 - `SetTextCellsLineItems`
- The following line item-related methods have been deprecated:
 - `GetLineItems` – superseded by `GetCellLineItems`.
 - `SetLineItems` – superseded by `SetCellLineItems`.
 - `GetTextLineItems` – superseded by `GetTextCellLineItems`.
 - `SetTextLineItems` – superseded by `SetTextCellLineItems`.
- The `ClearAllLineItems` method has been removed from the HsvData object.

HsvSecurityAccess Type Library

The following methods have been added to the HsvSecurityAccess object.

- `GetUserNameFromSID`
- `InsertDefaultSecurityClass`

HsvSystemInfo Type Library

The following methods have been added to the HsvSystemInfo object:

- `ExtractDMLScript`
- `GetUserName`
- `LoadDMLScript`

HsvMDArrays Type Library

- The *plNumElementsInDataUnit* argument has been added to the `SetData` method of the `HsvMDDataBuffer` object.
- The *plNumElementsInDataUnit* argument has been added to the `SetData` method of the `HsvMDDataBufferLite` object.

HsvMetadataLoadACV Type Library

The following metadata load options have been added:

- `HSV_METALOAD_OPT_LOAD_SYSTEM_MEMBERS`
- `HSV_METALOAD_OPT_VALUE`
- `HSV_METALOAD_OPT_ICP`

The following metadata extract options have been added:

- `HSV_METAEXTRACT_OPT_EXTRACT_SYSTEM_MEMBERS`
- `HSV_METAEXTRACT_OPT_VALUE`
- `HSV_METAEXTRACT_OPT_ICP`

HsvRulesLoadACV Type Library

The following methods have been added to the `HsvRulesLoadACV` object:

- `ExtractDMLScript`
- `LoadDMLScript`
- `ScriptableLoadDMLScript`

Error Numbers

Error numbers are no longer itemized in an appendix. Financial Management now offers the `HsvResourceManager` type library, which you can use to handle errors. For details, see [Chapter 22, “Error Handling and the HsvResourceManager Type Library.”](#)

Changes for Release 2.2

The following topics list changes for release 2.2:

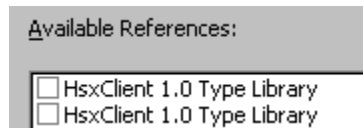
- [“HsxClient, HsxServer, and HsxClientUI References” on page 920](#)
- [“HsvData Type Library” on page 920](#)
- [“HsvCalculate Type Library” on page 920](#)
- [“HsvMDArrays Type Library” on page 921](#)

HsxClient, HsxServer, and HsxClientUI References

HsxClient.dll, HsxServer.exe, and HsxClientUI.dll are no longer installed in the /Common Files\Hyperion Shared\Bin directory. If you have projects that reference these components, you must update the references to the components installed in the new directories, which are as follows:

- HsxClient.dll and HsxClientUI.dll are now in the /Hyperion Financial Management\Client directory.
- HsxServer.exe is now in the /Hyperion Financial Management\Server directory.

Note that the old components might not get uninstalled. This means that the References dialog box in Visual Basic may show two identically named components, as shown in the following example:



Since the component names are identical, use the components' directory locations to select the correct component.

Tip: To prevent the old components from appearing, unregister HsxClient.dll, HsxServer.exe, and HsxClientUI.dll in the /Common Files\Hyperion Shared\Bin directory.

HsvData Type Library

The following methods are new:

- GetCellsWithRowSuppression
- GetTextCellsWithRowSuppression

The following methods now work with the HsvMDDataBufferLite object. In previous releases the methods worked with the HsvMDDataBuffer object, now they work with both HsvMDDataBuffer and HsvMDDataBufferLite:

- AddDataToMDDataBuffer
- UpdateDataUsingMDDataBuffer

HsvCalculate Type Library

The following methods have been added:

- Allocate2
- ChartLogic2
- Consolidate2

- CustomLogic
- Translate2

HsvMDArrays Type Library

The following items were added:

- HsvMDDataBufferLite object.
- IHsvMDDataBufferLite interface.

Changes for Release 2.1

The following topics list changes for release 2.1:

- [“HsvData Type Library” on page 921](#)
- [“HsvJournals Type Library” on page 921](#)
- [“HsvReports Type Library” on page 922](#)
- [“HsvMDArrays Type Library” on page 922](#)
- [“HsvDataCubes Type Library” on page 922](#)
- [“HsvRulesLoadACV Type Library” on page 923](#)
- [“HsvSecurityAccess Type Library” on page 923](#)
- [“HsxClient Type Library” on page 923](#)

HsvData Type Library

The following changes have been made to HsvData:

- The following methods are new:
 - SetCalcStatusLockedForMultipleProcessUnits
 - SetCalcStatusUnlockedForMultipleProcessUnits
- The *varbIncludeDerivedData* argument has been added to `AddDataToMDDataBuffer`.
- The *IParent* and *IValue* arguments have been removed from `ClearAllLineItems`.
- The *sNumDecimals* and *sScale* arguments have been added to `GetTextCells`.
- The *sScale* argument has been added to `SetTextCell`.
- The *sScale* argument has been added to `SetTextCells`.
- The *sAlternateNumDecimals* argument has been added to `FormatNumberToText`.

HsvJournals Type Library

`ValidateValue` has been added to the `IHsvJournalsEx` interface.

HsvReports Type Library

`CheckReportSecurityClass` has been added to the HsvReports object.

HsvMDArrays Type Library

The following changes have been made to the HsvMDDataBuffer object:

- The *lView* argument has been removed from `GetDescription`.
- The *plView* argument has been removed from `GetDescriptionAtIndex`.
- The *lView* argument has been removed from `GetLineItems`.
- The *plView* argument has been removed from `GetLineItemsAtIndex`.
- The *lView* argument has been removed from `InsertDescriptionAtEnd`.
- The *lView* argument has been removed from `InsertLineItemsAtEnd`.
- The *lView* argument has been removed from `SetDescription`.
- The *lView* argument has been removed from `SetLineItems`.

HsvDataCubes Type Library

The description of subcube items has changed, due to changes in some of the type library's methods. The updated description is in [“About Subcube Items” on page 687](#).

The following changes have been made to the HsvCurrencyCube object:

- The *lValue* and *plHandle* arguments have been added to `BeginEnumerationOfStoredData`.
- The *lHandle* argument has been added to `EndEnumerationOfStoredData`.
- The following changes were made to `GetOneCellFromStoredItem`:
 - The *lHandle* and *pvbDimensionMembersAreValid* arguments have been added.
 - The *Value* dimension argument returns a *Value* dimension member ID; in previous releases, the argument took a *Value* member ID. Note that the argument has been renamed from *lValue* to *plValue*.
- The *lHandle*, *plValue*, and *pvbDimensionMembersAreValid* arguments have been added to `GetPOVFromStoredItem`.

The following changes have been made to the HsvNodeCube object:

- The *lValue* and *plHandle* arguments have been added to `BeginEnumerationOfStoredData`.
- The *lHandle* argument has been added to `EndEnumerationOfStoredData`.
- The following changes were made to `GetOneCellFromStoredItem`:
 - The *lHandle* and *pvbDimensionMembersAreValid* arguments have been added.

- The Value dimension argument returns a Value dimension member ID; in previous releases, the argument took a Value member ID. Note that the argument has been renamed from *IValue* to *pIValue*.
- The *IHandle*, *pIValue*, and *pvbDimensionMembersAreValid* arguments have been added to `GetPOVFromStoredItem`.

HsvRulesLoadACV Type Library

The following changes have been made to `SetSession`:

- The *ILanguageID* argument has been added.
- `SetSession` now returns a Boolean to indicate success or failure.

HsvSecurityAccess Type Library

The following constants have been added. However, both constants are only for internal use—if you attempt to use them, an error will occur:

- `HFM_ACCESS_RIGHTS_UNSPECIFIED`
- `HFM_TASK_NONE`

HsxClient Type Library

The `AuthenticateUserOnServer` method has been added. Note that this is currently for internal use only.

Changes for Release 2.0

The following topics list changes for release 2.0:

- [“Security: HsxSecurity and HsvSecurityAccess Type Libraries” on page 924](#)
- [“HsxClient Type Library” on page 924](#)
- [“HsxClientUI Type Library” on page 925](#)
- [“HsxServer Type Library” on page 925](#)
- [“HsvSession Type Library” on page 925](#)
- [“HFMConstants Type Library” on page 925](#)
- [“HsvMetadata Type Library” on page 925](#)
- [“HsvData Type Library” on page 926](#)
- [“HsvCalculate Type Library” on page 926](#)
- [“HsvJournals Type Library” on page 927](#)
- [“HsvProcessFlow Type Library” on page 927](#)

- [“HsvReports Type Library” on page 927](#)
- [“HsvSystemInfo Type Library” on page 928](#)
- [“HsvMDArrays Type Library” on page 928](#)
- [“HsvRulesLoadACV Type Library” on page 928](#)
- [“HsvcDataLoad Type Library” on page 929](#)
- [“HsvJournalLoadACV Type Library” on page 929](#)

Security: HsxSecurity and HsvSecurityAccess Type Libraries

The HsxSecurity type library is no longer supported, and has been replaced by the HsvSecurityAccess type library. The HsxSecurity object’s methods have been moved to the HsvSecurityAccess object. In addition, the following list identifies HsvSecurityAccess items that were not in the HsxSecurity type library:

- The following HsvSecurityAccess object methods are new:
 - `GetTaskAccessForConnectedUserFromList`
 - `GetUserIDFromSID`
 - `IsConnectedUserAllowedToPerformTask`
- The HsvSecurityAccess type library contains the IHsvDataSecurity interface. This interface contains the following methods, which have been removed from the HsvProcessFlow type library:
 - `GetCellLevelAccessRights`
 - `GetProcessUnitAccessRights`
 - `GetProcessUnitAccessRightsAndState`
 - `RefreshAccessRightsCache`
- The access rights constants have been renamed.
- Roles are now represented by enumerated constants. In addition, new roles have been added.
- Constants for tasks have been added.

Note: For HsxSecurity methods with names that contained the word “Administrator,” the corresponding HsvSecurityAccess method names replace “Administrator” with “ApplicationAdministrator.” For example, HsxSecurity contained a method named `IsAdministrator`; in HsvSecurityAccess, this method is named `IsApplicationAdministrator`.

HsxClient Type Library

The following methods are now considered “internal use only”:

- `ScriptableEnumRegisteredServerNames`
- `ScriptableGetLogonInfo`

- `ScriptableOpenApplication`

HsxClientUI Type Library

The `GetServer` method has been added. However, note that this method is only for internal use.

HsxServer Type Library

The following methods are now considered “internal use only”:

- `ScriptableEnumDataSources`
- `ScriptableEnumRegisteredDSNs`

HsvSession Type Library

The `Security` property no longer returns an `HsvSecurity` object reference. Instead, this property returns object references for the `HsvSecurityAccess` object and the `IHsvDataSecurity` interface.

HFMConstants Type Library

The `HFMConstants` type library has been added. This library contains constants that represent numerous Oracle Hyperion Financial Management, Fusion Edition items, and is documented in [Appendix A, “Constants: The HFMConstants Type Library.”](#)

HsvMetadata Type Library

The following list describes changes to the `HsvMetadata` type library:

- The following methods have been added to the `HsvMetadata` object:
 - `GetCellLevelAccountType`
 - `IsOrgByPeriodApplication`
- The valid return values for `HsvAccounts.GetAccountType` are now represented by the constants listed in [“Account Type Constants” on page 826](#).
- The `GetSecurityClassID` method has been added to the `HsvAccounts` object.
- The `GetNumPeriodsInFrequency` method has been added to the `HsvPeriods` object.
- The `GetSecurityClassID` method has been added to the `HsvScenarios` object.
- The following methods have been added to the `IHsvTreeInfo` interface:
 - `GetNumDescendants`
 - `GetNumMembers`
 - `GetNumParents`

- IsMemberABaseOf
- IsMemberADescendantOf
- The GetYearRange method has been added to the HsvYears object.

HsvData Type Library

The following changes have been made to HsvData:

- Enum HSV_DATA_UPDATE_MODE has been added; for details, see [“Update Mode Constants” on page 276](#).
- The Duplicates load option uses the constants in Enum HSV_DATA_UPDATE_MODE instead of the constants that were previously documented.
- The following changes have been made to ClearInputData :
 - The *varlaAccountSubset* argument has been replaced by the *varlaAccountSubsetIncludeList*, *vbUseAccountSubsetIncludeList*, *varlaAccountSubsetExcludeList*, and *vbUseAccountSubsetExcludeList* arguments.
 - The *vbClearNodeLevelValueID* argument’s value is ignored.
- The following changes have been made to CopyInputData :
 - The *vbClearInputValueID*, *vbClearNoneValueID*, and *vbClearNodeLevelValueID* arguments have been renamed to *vbCopyEntityCurrencyValueID*, *vbCopyNoneValueID*, and *vbCopyNodeLevelValueID*, respectively.
 - The *vbCopyNodeLevelValueID* argument’s value is ignored.
 - The *vbCopyDerived* and *IEnumUpdateMode* arguments have been added.
- The CopyInputDataForMultipleEntities method has been added.
- UpdateDataUsingMDDataBuffer has been changed: the *bClearExistingSubCubeData* argument has been removed, and the *IEnumUpdateMode* and *bAccumulateWithinBuffer* arguments have been added.
- AddDataToMDCube, deprecated in release 1.2, has been removed in release 2.0.
- UpdateDataUsingMDCube, deprecated in release 1.2, has been removed in release 2.0.
- The load options have been changed. Two new load options with index numbers 5 and 6 have been added, and the index of the Mode option has been changed from 5 to 7. See [Table 61 on page 354](#) for details.
- Cell statuses are now represented by constants in the HFMConstants type library; for an overview, see [“About Cell Statuses” on page 275](#).
- The *sNumDecimals* and *sScale* arguments have been added to GetTextCell.
- The GetTransactionData method has been added.

HsvCalculate Type Library

The following methods have been removed from the HsvCalculate object:

- `GetRules`
- `SetRules`

The following methods have been added to the `HsvCalculate` object:

- `GetVBScriptRules`
- `SetVBScriptRules`
- `GetCOMDLLRules` (this method is for future use)
- `SetCOMDLLRules` (this method is for future use)

HsvJournals Type Library

In the `IHsvJournalsEx` interface, the `SaveJournal` and `SaveTextJournal` methods throw error number 8004041A (hexadecimal) if you attempt to save a journal as a system-generated autoreversing journal type.

HsvProcessFlow Type Library

The following list describes changes to the `HsvProcessFlow` type library:

- Some methods have been removed from the `HsvProcessFlow` type library. In addition to the methods that have been moved to the `IHsvDataSecurity` interface (as described in [“Security: HsxSecurity and HsvSecurityAccess Type Libraries” on page 924](#)), the following `HsvProcessFlow` methods have been removed:
 - `GetConnectedUser` (You can replace this with `HsvSecurityAccess.GetConnectedUser`. For details, see [“GetConnectedUser” on page 469](#).)
 - `GetRoleAccessRights` (You can replace this with `HsvSecurityAccess.IsConnectedUserInRole`. For details, see [“IsConnectedUserInRole” on page 479](#).)
 - `IsAdministrator` (You can replace this with `HsvSecurityAccess.IsApplicationAdministrator`. For details, see [“IsApplicationAdministrator” on page 478](#).)
- The `HsvProcessFlow` type library no longer contains access rights constants, as the `HsvSecurityAccess` type library now contains all methods that work with access rights.

HsvReports Type Library

To support report descriptions, the following changes have been made to the `HsvReports` object:

- The *pvarabstrDescriptions* argument has been added to `EnumReports`.
- The *pbstrDescription* argument has been added to `GetReport`.
- The *bstrDescription* argument has been added to `SetReport`.

HsvSystemInfo Type Library

Several methods were added and removed from the HsvSystemInfo type library. The following methods were removed:

- GetCalcRules
- GetMemberListRules
- SetCalcRules
- SetMemberListRules

The following methods were added to HsvSystemInfo:

- GetCOMDLLCalcRules
- GetResourceLanguageUserParameters
- GetResourceString
- GetVBScriptCalcRules
- GetVBScriptMemberListRules
- SetCOMDLLCalcRules
- SetResourceLanguageForCurrentSession
- SetResourceLanguageUserParameters
- SetVBScriptCalcRules
- SetVBScriptMemberListRules

HsvMDArrays Type Library

The HsvTransactionData object has been added to the HsvMDArrays type library. Use this object to return the transaction data generated by statutory consolidations.

HsvRulesLoadACV Type Library

New arguments have been added to the following methods, enabling you to validate without loading and to check whether validation messages of various severity levels have been included in the log file:

- LoadCalcRules
- LoadMemberListRules

The following methods have been added to HsvRulesLoadACV. However, they are only for internal use:

- ScriptableLoadCalcRules
- ScriptableLoadMemberListRules

HsvcDataLoad Type Library

The following list describes changes to the HsvcDataLoad type library:

- The HSV_DATALOAD_OPT_ACCUMULATE_WITHIN_FILE and HSV_DATALOAD_OPT_FILE_CONTAINS_SHARES data load options have been added.
- The HSV_DATALOAD_ACCUMULATE constant has been added as a valid value for the HSV_DATALOAD_OPT_DUPLICATES data load option. This allows loading in accumulate mode.

HsvJournalLoadACV Type Library

The HSV_JOURNALEXTRACT_OPT_CATEGORY extract option has been renamed to HSV_JOURNALEXTRACT_OPT_SCENARIO.

Changes for Release 1.2

The following topics list changes for release 1.2:

- [“HsvProcessFlow Type Library” on page 929](#)
- [“HsvData Type Library” on page 929](#)
- [“HsvMetadata Type Library” on page 930](#)
- [“HsvReports Type Library” on page 931](#)
- [“HsvMetadataLoadACV Type Library” on page 931](#)

HsvProcessFlow Type Library

The following list describes changes to the HsvProcessFlow type library:

- The following methods have been added to the HsvProcessFlow object:
 - ApproveEx
 - PublishEx
 - StartEx
- If the scenario passed to HsvProcessFlow.GetHistory does not support process management, the 0x4D2 error occurs. This error has a *Success* severity level, meaning that it will not be returned in Visual Basic; however, the error will be returned in Visual C++.

HsvData Type Library

The following list describes changes to the HsvData type library:

- The following methods have been added to the HsvData object:
 - AddDataToMDDataBuffer

- EnumExtractOptions
- EnumLoadOptions
- Extract
- Load
- UpdateDataUsingMDDataBuffer
- The following methods have been deprecated in the HsvData object:
 - AddDataToMDCube (replaced by AddDataToMDDataBuffer)
 - UpdateDataUsingMDCube (replaced by UpdateDataUsingMDDataBuffer)
- UpdateLoadDataUsingMDCube has been removed from the HsvData object.
- The HsvMDDataBuffer object has been added to the HsvMDArrays type library.
- The HsvMDCube object in the HsvMDArrays type library has been deprecated, and will be removed in a future release. The HsvMDDataBuffer object replaces the HsvMDCube object.

HsvMetadata Type Library

The following list describes changes to the HsvMetadata type library:

- For the valid values returned by `HsvAccounts.GetAccountType`, **9** is now future use. Previously, **9** had indicated the Text account type.
- The following methods have been added to the HsvMetadata object:
 - EnumConsolidationMethodIDs
 - GetConsolidationMethodDescription
 - GetConsolidationMethodInfo
 - GetDefaultValueOfActiveStatusAccount
 - GetICPEntitiesAggregationWeight
 - GetValidationAccount
- The following methods have been added to the HsvAccounts object:
 - GetUserDefined1
 - GetUserDefined2
 - GetUserDefined3
 - GetXBRLTags
- The following methods have been added to the HsvCustom object:
 - GetAggregationWeight
 - IsSwitchSignEnabledForFlow
 - IsSwitchTypeEnabledForFlow
- The `GetHoldingCompany` method has been added to the HsvEntities object.
- The `RefreshCache` method has been removed from the IHsvTreeInfo interface.

HsvReports Type Library

The `DeleteReports` method has been added to the `HsvReports` object.

HsvMetadataLoadACV Type Library

The following metadata load and extract options have been added to the `HsvMetadataLoadACV` type library:

- `HSV_METALOAD_OPT_CONSOL_METHODS`
- `HSV_METALOAD_OPT_CLEAR_CONSOL_METHODS`
- `HSV_METAEXTRACT_OPT_CONSOL_METHODS`

Changes for Release 1.1

The following topics list changes for release 1.1:

- [“HsvMetadata Type Library” on page 931](#)
- [“HsxSecurity Type Library” on page 932](#)
- [“HsvReports Type Library” on page 932](#)
- [“HsvMDArrays Type Library” on page 932](#)
- [“HsvDataCubes Type Library” on page 932](#)
- [“HsvMetadataLoadACV Type Library” on page 933](#)
- [“HsvSecurityLoadACV Type Library” on page 933](#)

HsvMetadata Type Library

The following list describes changes to the `HsvMetadata` type library:

- The `HsvCategories` object has been renamed. It is now the `HsvScenarios` object.
- The `Categories` property of the `HsvMetadata` object has been renamed. It is now the `Scenarios` property.
- The `blIgnoreDups` argument has been added to the `IHsvTreeInfo` interface’s `EnumAncestors` method.
- The `blIgnoreDups` argument has been added to the `IHsvTreeInfo` interface’s `EnumDescendants` method.
- The following methods have been added to the `HsvAccounts` object:
 - `GetNumDecimalPlaces`
 - `GetPlugAccount`
 - `GetTopMemberOfValidCustom1Hierarchy`
 - `GetTopMemberOfValidCustom2Hierarchy`

- GetTopMemberOfValidCustom3Hierarchy
- GetTopMemberOfValidCustom4Hierarchy
- IsConsolidated
- IsCustom1AggregationEnabled
- IsCustom2AggregationEnabled
- IsCustom3AggregationEnabled
- IsCustom4AggregationEnabled
- IsICP
- The following methods have been added to the HsvEntities object:
 - GetAllowAdjustments
 - GetAllowAdjustmentsFromChildren
 - GetUserDefined1
 - GetUserDefined2
 - GetUserDefined3
 - IsDescendant
 - IsICP

HsxSecurity Type Library

The following methods have been added to the HsxSecurity object:

- AddUserEx
- IsValidWindowsUser

HsvReports Type Library

The HsvReports type library has been added to this document. For details, see [Chapter 15](#), “HsvReports Type Library.”

HsvMDArrays Type Library

The HsvMDArrays type library has been added to this document. For details, see [Chapter 16](#), “HsvMDArrays Type Library.”

HsvDataCubes Type Library

The HsvDataCubes type library has been added to this document. For details, see [Chapter 17](#), “HsvDataCubes Type Library.”

HsvMetadataLoadACV Type Library

Security information is no longer loaded and extracted along with metadata. To reflect this change, the following security-related load and extract options have been removed from the HsvMetadataLoadACV type library:

- HSV_METALOAD_OPT_CLEAR_SECURITY_CLASSES
- HSV_METALOAD_OPT_SECURITY_CLASSES
- HSV_METAEXTRACT_OPT_SECURITY_CLASSES

HsvSecurityLoadACV Type Library

The HsvSecurityLoadACV type library has been added; use this to load and extract security information. For details, see [“HsvSecurityLoadACV Type Library” on page 752](#).

Index

Symbols

.LST file

- getting contents of, [519](#)
- loading and validating, [772](#)

.PER files, creating applications and, [125](#)

.RLE file

- getting contents of
 - GetVBScriptCalcRules, [518](#)
 - GetVBScriptRules, [385](#)
- loading and validating
 - LoadCalcRules, [770](#)
 - SetVBScriptCalcRules, [525](#)
 - SetVBScriptRules, [386](#)

A

access rights constants, HsvSecurityAccess, [451](#)

accounts

- AccountType attribute, returning, [232](#)
- Custom1TopMember attribute, returning, [237](#)
- Custom2TopMember attribute, returning, [237](#)
- Custom3TopMember attribute, returning, [238](#)
- Custom4TopMember attribute, returning, [238](#)
- EnableCustom1Aggr attribute, returning, [241](#)
- EnableCustom2Aggr attribute, returning, [241](#)
- EnableCustom3Aggr attribute, returning, [242](#)
- EnableCustom4Aggr attribute, returning, [242](#)
- IsCalculated attribute, returning, [240](#)
- IsConsolidated attribute, returning, [240](#)
- IsICP attribute, returning, [243](#)
- NumDecimalPlaces attribute, returning, [234](#)
- PlugAcct attribute, returning, [235](#)
- testing whether Custom members valid for, [179](#)
- UserDefined1 attribute, returning, [239](#)
- UserDefined2 attribute, returning, [239](#)
- UserDefined3 attribute, returning, [239](#)
- UsesLineItems attribute, returning, [243](#)
- validation account, [178](#)

validation account, returning by index, [169](#)

XBRLTags attribute, returning, [240](#)

Accounts (property), [185](#)

AccountType attribute, returning, [232](#)

ACMs, instantiating, [159](#)

actions, process management, constants for, [853](#)

active status account, returning default value of, [174](#)

activity user ID, [510](#)

AddAccountCustomCombination, [675](#)

AddApplicationAdministrator, [455](#)

AddApplicationAdministrator2, [455](#)

AddDataToMDDDataBuffer, [277](#)

AddICTransactionData, [675](#)

AddJournalGroup, [399](#)

AddOrRemoveApplicationAdministrators, [455](#)

AddOrRemoveApplicationAdministrators2, [455](#)

AddOrRemoveRolesFromUser, [456](#)

AddOrRemoveRolesFromUser2, [456](#)

AddOrRemoveUsersFromRole, [457](#)

AddOrRemoveUsersFromRole2, [457](#)

AddQueryDimensionMemberID, [675](#)

AddQueryField, [675](#)

AddQueryFieldItem, [675](#)

AddQueryOrderField, [675](#)

AddRefToHsxServer, [491](#)

AddSecurityClass, [458](#)

AddSecurityClassWithAccessCode, [458](#)

AddTaskToAudit, [491](#)

AddTaskToRunningTasks, [492](#)

AddTaskToRunningTasksAndUpdatePOV, [492](#)

AddUser, [458](#)

AddUser2, [458](#)

AddUserEx, [459](#)

AddUserToRole, [459](#)

AddUserToRole2, [459](#)

Administrator group, testing whether a user belongs to, [129](#)

AllMVTransPosted, [711](#)

- Allocate, [376](#)
 - Allocate2, [376](#)
 - AllowAdjFromChildren attribute, returning, [248](#)
 - AllowAdjs attribute, returning, [247](#)
 - AllowRulesLoadForEPMAApp, [459](#)
 - application creator rights
 - creating applications and, [124](#)
 - testing whether a user has, [128](#)
 - application currency, returning, [168](#)
 - application profiles, creating applications and, [125](#)
 - application servers
 - activity has occurred on, returning names of, [496](#)
 - applications on, getting, [152](#)
 - data link file, getting, [154](#)
 - names of, getting, [132](#)
 - registering
 - HsxClient object, [142](#)
 - HsxClientUI object, [150](#)
 - system file, getting, [154](#)
 - unregistering
 - HsxClient object, [145](#)
 - HsxClientUI object, [150](#)
 - application settings
 - DefaultCurrency, [168](#)
 - GetByIndexValidationAccount, [169](#)
 - GetValidationAccount, [178](#)
 - OrgByPeriodApplication, [180](#)
 - returning when last updated, [168](#)
 - values, returning
 - as raw value, [167](#)
 - as string, [185](#)
 - applications
 - Classic, testing whether, [478](#)
 - creating, [124](#)
 - deleting, [126](#)
 - getting user information, [138](#)
 - single sign on, [138](#)
 - local storage folder, getting, [137](#)
 - local storage folder, setting, [143](#)
 - logging onto
 - example, [119](#)
 - using HsxClient, [143](#), [144](#)
 - using HsxClientUI, [148](#)
 - on application servers, getting, [152](#)
 - opening
 - using HsxClient, [140](#)
 - using HsxClientUI, [149](#)
 - Approve, [532](#)
 - Approve2, [533](#)
 - ApproveEx, [534](#)
 - ApproveJournals, [399](#)
 - AttachDocumentToCell, [278](#)
 - attribute, metadata, getting value of, [211](#)
 - audit
 - data audit
 - delete audit history, [280](#)
 - enumerate cells, [291](#)
 - one cell, [319](#)
 - task audit
 - adding task to audit history, [491](#)
 - delete audit history, [492](#)
 - getting audit history, [497](#)
 - AuthenticateCSSToken, [124](#)
 - AuthenticateSecurityAgentCredentials, [124](#)
 - AuthenticateUserCredentials, [124](#)
 - AuthenticateUserOnCluster, [124](#)
 - AuthenticateUserOnClusterSSO, [124](#)
 - AuthenticateUserOnServer, [124](#)
- ## B
- base frequency, [256](#)
 - BeginDataEnum
 - HsvICTransactionsData object, [676](#)
 - HsvTransactionData object, [668](#)
 - BeginEnumeration
 - HsvMDDDataBuffer object, [611](#)
 - HsvMDDDataBufferLite object, [640](#)
 - BeginEnumerationOfStoredData
 - HsvCurrencyCube object, [688](#)
 - HsvNodeCube object, [693](#)
 - BeginQueryEnum, [669](#)
 - BeginQueryFieldEnum, [676](#)
 - BeginQueryOrderEnum, [676](#)
- ## C
- CalcAttribute attribute, returning, [233](#)
 - CalcEPU, [377](#)
 - Calculate, [157](#)
 - calculation statuses
 - cells, [314](#)
 - cells, constants for, [842](#)
 - subcube periods, constants for, [843](#)
 - subcubes, [313](#)

- Categories. *See* Scenarios
- cell data
 - examples for getting and setting, [121](#)
 - setting a cell to Null, [359](#)
- cell statuses
 - constants for, [841](#)
 - getting with GetStatus, [335](#)
 - overview, [275](#)
- CELLSTATUS_NOREADACCESS, [843](#)
- ChartLogic, [377](#)
- ChartLogic2, [378](#)
- CheckAccess, [492](#)
- CheckAllItems, [796](#)
- CheckAllItemsUsingSelectedList, [796](#)
- CheckBoxChanged event, [807](#)
- CheckReportSecurity, [712](#)
- CheckReportSecurityClass, [595](#)
- CheckReportSecurityClass2, [596](#)
- CheckSecurityForICExtract, [732](#)
- CheckSecurityRole, [596](#)
- Classic application, testing whether, [478](#)
- ClearAllData, [279](#)
- ClearAllDescriptionsInSubCube, [279](#)
- ClearAuditTasks, [492](#)
- ClearAuditTasks2, [493](#)
- ClearDataAuditItems, [280](#)
- ClearInputData, [281](#)
- ClearInvalidData, [283](#)
- ClearRunningTask, [494](#)
- CloseICPeriod, [732](#)
- ClosePeriod, [395](#)
- COM data type mappings, [40](#)
- connected user, [454](#)
- Consolidate, [379](#)
- Consolidate2, [380](#)
- ConsolidateYTD attribute, Scenario members, [265](#)
- consolidation methods
 - returning description of, [170](#)
 - returning IDs of an application's, [162](#)
 - returning information about, [171](#)
 - returning when last updated, [172](#)
- consolidations, statutory, returning transaction data for, [667](#)
- constants
 - access rights, [451](#)
 - CELLSTATUS_NOREADACCESS, [843](#)
 - data extract options, [781](#)
 - data load options, [778](#)
 - file format constants, metadata loads and extractions, [767](#)
 - journal extract options
 - filtered extracts, [791](#)
 - unfiltered extracts, [789](#)
 - journal load options, [789](#)
 - MEMBER_LIST_ALL_HIERARCHY, [838](#)
 - MEMBERNOTUSED, [835](#)
 - security extract options, [756](#)
 - security load options, [755](#)
- CopyInputData, [283](#)
- CopyInputDataForMultipleEntities, [284](#)
- Count, [744](#)
- CreateApplication, [124](#)
- CreateApplicationCAS, [124](#)
- CreateApplicationCASWithAccessCode, [126](#)
- CreateDataIndexList
 - HsvMDDDataBuffer object, [611](#)
 - HsvMDDDataBufferLite object, [641](#)
- CreateDataIndexListEx, [613](#)
- CreateICTransaction, [712](#)
- CreateObject, [159](#)
- CreateObjectOnCluster, [126](#)
- CreateObjectOnServer, [126](#)
- CreateStarSchema, [700](#)
- Creator group
 - creating applications and, [124](#)
 - user, testing whether a member of, [128](#)
- CSSEnabled, [156](#)
- Currencies, [186](#)
- currencies
 - description, getting, [272](#)
 - exchange rate, returning, [383](#)
 - HsvCurrencies object, [270](#)
 - IDs of, getting, [272](#)
 - labels and IDs of, getting
 - all currencies, [270](#)
 - filtering for DisplayInICT attribute, [271](#)
 - labels of, getting, [273](#)
 - returning when last updated, [173](#)
 - scale, getting, [274](#)
 - translation operator, getting, [273](#)
 - TranslationOperator attribute, [273](#), [274](#)
- currency IDs, [270](#)
- currency subcubes
 - accessing with HsvCurrencyCube object, [688](#)

described, [43](#)
 currency, application, returning, [168](#)
 CurrentValue, [745](#)
 Custom1, [186](#)
 Custom1TopMember attribute, returning, [237](#)
 Custom2, [186](#)
 Custom2TopMember attribute, returning, [237](#)
 Custom3, [186](#)
 Custom3TopMember attribute, returning, [238](#)
 Custom4, [186](#)
 Custom4TopMember attribute, returning, [238](#)
 CustomLogic, [381](#)

D

data

- examples for getting and setting, [121](#)
- extract options, [781](#)
- extracting
 - example, [777](#)
 - overview, [777](#)
- load options, [778](#)
- loading
 - example, [776](#)
 - overview, [776](#)
- objects and interfaces for
 - application data, [69](#)
 - arrays of cells, [93](#)
 - subcubes, access to, [102](#)

data audit

- delete audit history, [280](#)
- enumerate cells, [291](#)
- one cell, [319](#)

data link file, getting path of, [154](#)

Data property, [157](#)

data type mappings, [40](#)

Data View option buttons, [783](#)

decimal character

- getting, [514](#)
- setting, [523](#)

Default security class, constant for, [883](#)

DefaultCurrency application setting, returning, [168](#)

DefaultParent attribute, generation levels and, [218](#)

DefaultValue, [746](#)

DefaultValueForActive attribute, [174](#)

DefCurrency attribute, returning, [248](#)

delegated user management, enumerating users, [467](#)

Delete Application dialog box, displaying, [148](#)

Delete Invalid records feature, [283](#)

DeleteApplication

- HsxClient object, [126](#)
- HsxClientUI object, [148](#)

DeleteApplicationWithAccessCode, [127](#)

DeleteDocument, [597](#)

DeleteICReasonCode, [714](#)

DeleteJournals, [400](#)

DeleteLineItemDetails, [285](#)

DeleteReport, [597](#)

DeleteReports, [598](#)

DeleteSecurityClass, [459](#)

DeleteSecurityClassWithAccessCode, [460](#)

DeleteStarSchema, [702](#)

DeleteSystemErrors, [127](#), [152](#)

DeleteTemplate, [706](#)

DeleteTemplates, [401](#)

DeleteUserParameter, [495](#)

descriptions

- returning the default language for, [514](#)
- setting the default language for, [522](#)

DetachDocumentFromCell, [286](#)

DetermineWindowsLoggedInUser, [127](#)

Dimension, [186](#)

dimension objects, [55](#)

dimensions

- constants that represent, [834](#)
- getting labels of all members, [189](#)
- getting labels of children and parents, [190](#)
- getting member IDs of all members, [188](#)
- getting member IDs of children and parents, [189](#)
- hierarchy, returning position of member in, [212](#)

DimensionWasSelected event, [808](#)

DisableNewConnections

- HsvSystemInfo object, [495](#)
- HsxClient object, [128](#)

DisplayInICT attribute, [271](#)

DMELoad

- HsvcDataLoad object, [783](#)
- HsvData object, [287](#)

DoCellDescriptionsExist, [288](#)

documents

- reports, [603](#)

DoesCellDescriptionExist, [288](#)

DoesCellSupportICTTransactionDetail, [714](#)

DoesDataExist, [289](#)

DoesSparseDataExist, [290](#)

- DoesUserHaveCreateApplicationRights, [128](#)
- DoesUserHaveSystemAdminRights, [129](#)
- domain
 - getting, [138](#)
 - setting, [143](#), [144](#)
 - single sign on, getting, [138](#)
- dynamic member lists, getting members of, [199](#)

- E**
- EA_EXTRACT_TYPE_FLAGS, Enum, [701](#)
- EA_TASK_STATUS_FLAGS, Enum, [704](#)
- EnableCustom1Aggr attribute, returning, [241](#)
- EnableCustom2Aggr attribute, returning, [241](#)
- EnableCustom3Aggr attribute, returning, [242](#)
- EnableCustom4Aggr attribute, returning, [242](#)
- EnableDimension, [797](#)
- EnableNewConnections
 - HsvSystemInfo object, [496](#)
 - HsxClient object, [129](#)
- EndDataEnum
 - HsvICTransactionsData object, [676](#)
 - HsvTransactionData object, [669](#)
- EndEnumeration
 - HsvMDDataBuffer object, [613](#)
 - HsvMDDataBufferLite object, [643](#)
- EndEnumerationOfStoredData
 - HsvCurrencyCube object, [688](#)
 - HsvNodeCube object, [694](#)
- EndQueryEnum, [669](#)
- EndQueryFieldEnum, [677](#)
- EndQueryOrderEnum, [677](#)
- entities
 - AllowAdjFromChildren attribute, returning, [248](#)
 - AllowAdjs attribute, returning, [247](#)
 - DefCurrency attribute, returning, [248](#)
 - HoldingCompany attribute, returning, [249](#)
 - IsICP attribute, returning, [253](#)
 - IsOrgByPeriodFilteringOn attribute, returning, [253](#)
 - SecurityClass attribute, returning, [250](#)
 - UserDefined1 attribute, returning, [251](#)
 - UserDefined2 attribute, returning, [251](#)
 - UserDefined3 attribute, returning, [251](#)
- Entities (property), [187](#)
- EnumActivityServers, [496](#)
- EnumActivityUsers, [496](#)
- EnumAllMemberIDs, [188](#)
- EnumAllMemberLabels, [189](#)
- EnumAllParentAndChildIDs, [189](#)
- EnumAllParentAndChildLabels, [190](#)
- EnumAncestors, [191](#)
- EnumApplicationAdministrators, [460](#)
- EnumApplicationAdministrators2, [460](#)
- EnumApplicationStarSchemas, [703](#)
- EnumAuditTasks, [497](#)
- EnumAuditTasks2, [499](#)
- EnumBaseMemberIDs, [192](#)
- EnumConsolidationMethodIDs, [162](#)
- EnumCurrencies, [270](#)
- EnumCurrencies2, [271](#)
- EnumDataAuditItems, [291](#)
- EnumDataAuditItems2, [293](#)
- EnumDataSources, [152](#)
- EnumDefaultAncestors, [193](#)
- EnumDefaultAncestorsLabels, [194](#)
- EnumDescendants, [194](#)
- EnumDocuments, [598](#)
- EnumDocumentsEx, [600](#)
- EnumEntitiesWithDataForScenarioYear, [293](#)
- EnumExtractOptions
 - HsvData object, [294](#)
 - HsvMetadata object, [163](#)
- EnumIDsOfChildren, [195](#)
- EnumIDsOfIChildren, [196](#)
- EnumJournalGroups, [402](#)
- EnumJournalGroupsForScenarioYear, [402](#)
- EnumJournalIDsForExtractFilter, [403](#)
- EnumLanguages, [164](#)
- EnumLoadOptions
 - HsvData object, [295](#)
 - HsvMetadata object, [164](#)
- EnumMemberLists, [197](#)
- EnumMembers, [197](#)
- EnumMembers2, [199](#)
- EnumMembersWithAttribValue, [200](#)
- EnumParents, [201](#)
- EnumPhasedSubmissionStartYears, [260](#)
- EnumProhibitConnections
 - HsvSystemInfo object, [501](#)
 - HsxClient object, [130](#)
- EnumProvisioningProjects, [131](#)
- EnumRegisteredClusterNames, [132](#)
- EnumRegisteredDSNs
 - HsvStarSchemaACM object, [703](#)

- HsxServer object, [152](#)
 - EnumRegisteredServerNames, [132](#)
 - EnumReports, [602](#)
 - EnumRoles, [460](#)
 - EnumRolesForPrincipal, [461](#)
 - EnumRolesForUser, [461](#)
 - EnumRunningTasks, [502](#)
 - EnumRunningTasksEx, [505](#)
 - EnumRunningTasksPOV, [507](#)
 - EnumSecurityClasses, [462](#)
 - EnumSecurityClassRightsForPrincipal, [463](#)
 - EnumSortedIDsOfChildren, [201](#)
 - EnumSortedIDsOfChildren, [202](#)
 - EnumSortedMembers, [203](#)
 - EnumTemplates, [706](#)
 - EnumUserAppPreferences, [132](#)
 - EnumUserClassAccess, [464](#)
 - EnumUsers, [464](#)
 - EnumUsers2, [464](#)
 - EnumUsers3, [464](#)
 - EnumUsersInGroup, [465](#)
 - EnumUsersInRole, [465](#)
 - EnumUsersInRole2, [465](#)
 - EnumUsersInRole3, [466](#)
 - EnumUsersInSecurityClass, [466](#)
 - EnumUsersInSecurityClass2, [466](#)
 - EnumUsersInSecurityClass3, [467](#)
 - EnumUsersOnSystem, [132](#)
 - EnumUsersOnSystemEx, [134](#)
 - EnumUsersOnSystemEx2, [136](#)
 - EnumUsersOrGroups, [467](#)
 - EnumUsersWithFilter, [467](#)
 - EnumUsersWithFilter2, [467](#)
 - EraseRecordFromPMBuffer
 - HsvMDDataBuffer object, [613](#)
 - HsvMDDataBufferLite object, [643](#)
 - Err object (Visual Basic), [45](#)
 - error handling
 - example, [122](#)
 - overview, [45](#)
 - error numbers, utility for, [822](#)
 - examples, getting started
 - cell data, [121](#)
 - error handling, [122](#)
 - loading and extracting, [122](#)
 - logging on, [119](#)
 - metadata, [120](#)
 - Software Development Toolkit, [119](#)
 - user number formatting preferences, [121](#)
 - exporting data. *See* extracting data
 - exporting journals. *See* extracting journals
 - exporting member lists. *See* extracting member lists
 - exporting metadata. *See* extracting metadata
 - exporting rules. *See* extracting rules
 - Extended Analytics, [699](#)
 - external authentication
 - enabled for cluster, indicating whether, [156](#)
 - getting user information, [138](#)
 - Extract
 - HsvData object, [295](#)
 - HsvMetadata object, [165](#)
 - load- and extract-related objects, [749](#)
 - ExtractCalcRules, [769](#)
 - ExtractDataAuditItems, [299](#)
 - ExtractDrillableRegions, [300](#)
 - ExtractDrillableRegionsByURLNames, [301](#)
 - ExtractEx, [794](#)
 - extracting data
 - example, [777](#)
 - overview, [777](#)
 - extracting journals
 - example
 - filtered extract, [788](#)
 - unfiltered extract, [787](#)
 - overview, [786](#)
 - extracting member lists, [769](#)
 - extracting metadata
 - example, [761](#)
 - file format constants, [767](#)
 - overview, [760](#)
 - extracting rules, [769](#)
 - extracting security
 - example, [755](#)
 - overview, [754](#)
 - ExtractMemberListRules, [769](#)
 - ExtractOptions, [748](#)
 - ExtractOptionsEx, [793](#)
 - ExtractTaskAudit, [509](#)
- ## F
- FDM Application Name setting, [174](#)
 - FilterMembersThatHaveData, [301](#)
 - FilterMembersThatHaveData2, [302](#)
 - Find, [204](#)

FindByDesc, 205
 FindMatchingMembersFromHierarchy, 207
 FindMatchingMembersFromHierarchyByDesc, 208
 FindMatchingMembersFromHierarchyWildcard, 209
 FindOverlappingConsolidation, 381
 FormatNumberToText, 303
 FormatNumberToText2, 305
 FormatStatusToText, 307
 FormatStoredNumberToText, 308
 FormatTextToNumber, 309
 FormatTextToStoredNumber, 310

G

GenerateRecurring, 404
 GenerateSecurityReportForBiPub, 468
 generation level, member, getting, 218
 get_Dimension, 186
 GetAccessRights, 677
 GetAccountCustomCombination, 677
 GetAccountCustomCombinationTotal, 677
 GetAccountType, 232
 GetActivityCodeDesc, 510
 GetActivityUserID, 510
 GetAggregationWeight, 244
 GetAllDescriptionsInSubCube, 311
 GetAllowAdjustments, 247
 GetAllowAdjustmentsFromChildren, 248
 GetAllPathsToMember, 210
 GetAllSecurityClassRightsForConnectedUser, 468
 GetAllURLNames, 312
 GetApplicationAdministratorAccessForAllUsers, 469
 GetApplicationAdministratorAccessForAllUsers2, 469
 GetApplicationAttribute, 167
 GetApplicationCurrency, 168
 GetApplicationDirectory, 510
 GetApplicationFolder, 137
 GetApplicationName, 511
 GetApplicationSettingsTimeStamp, 168
 GetAsynchronousTaskStatus, 703
 GetAttachedDocumentsToCell, 312
 GetAttributeValue, 211
 GetAvailableLanguages, 813
 GetBaseDataForAccount, 313
 GetBaseFrequency, 256
 GetByIndexValidationAccount, 169

GetCalcAttribute, 233
 GetCalcRulesType, 512, 770
 GetCalcStatus, 313
 GetCalcStatusEx, 314
 GetCalcStatusStatistics, 315
 GetCell
 HsvCurrencyCube object, 689
 HsvData object, 316
 HsvNodeCube object, 694
 GetCellDescription, 317
 GetCellDescriptions, 318
 GetCellHistory, 319
 GetCellHistory2, 321
 GetCellJournalEntries, 323
 GetCellLevelAccessRights, 485
 GetCellLevelAccountType, 170
 GetCellLineItems, 325
 GetCells, 326
 GetCellsWithRowSuppression, 327
 GetCellsWithRowSuppression2, 330
 GetCheckedItems, 798
 GetCheckLineItemDetailsForCaseInsensitiveDuplica
 tes
 HsvMDDDataBuffer object, 613
 HsvMDDDataBufferLite object, 643
 GetClusterInfo, 138
 GetClustersAndServers, 153
 GetColumnFilter, 715
 GetCOMDLLCalcRules, 512
 GetCOMDLLRules, 382
 GetConnectedUser, 469
 GetConnectedUser2, 469
 GetConsolidationMethodDescription, 170
 GetConsolidationMethodInfo, 171
 GetConsolidationMethodsTimeStamp, 172
 GetConsolidationProgress, 382
 GetCountOfAttachedDocumentsToCell, 331
 GetCubeIndexFromPOV
 HsvMDDDataBuffer object, 614
 HsvMDDDataBufferLite object, 643
 GetCubePOVFromIndex
 HsvMDDDataBuffer object, 614
 HsvMDDDataBufferLite object, 644
 GetCurrencyCube, 332
 GetCurrencyDescription, 272
 GetCurrencyID, 272
 GetCurrencyIDFromValueID, 268

- GetCurrencyInfo, [715](#)
- GetCurrencyLabel, [273](#)
- GetCurrencyTimeStamp, [173](#)
- GetCurrencyTranslationOperator, [273](#)
- GetCurrencyValueIDForEntityValueCombination, [173](#)
- GetCurrencyValueIDsForEntityValueCombinations, [173](#)
- GetCurrentActivity, [512](#)
- GetCurrentHSSRegistrationVersion, [814](#)
- GetCurrentVersion, [814](#)
- GetCurrentVersionInUserDisplayFormat, [814](#)
- GetData
 - HsvMDDataBuffer object, [615](#)
 - HsvMDDataBufferLite object, [645](#)
- GetDataAtIndex
 - HsvMDDataBuffer object, [616](#)
 - HsvMDDataBufferLite object, [646](#)
- GetDataForAllMetadataCombinations, [333](#)
- GetDataSource, [154](#)
- GetDefaultExchangeRate, [383](#)
- GetDefaultFrequency, [260](#)
- GetDefaultHierarchyPosition, [212](#)
- GetDefaultItemIDHierarchy, [212](#)
- GetDefaultMemberID, [214](#)
- GetDefaultParent, [214](#)
- GetDefaultParentLabel, [214](#)
- GetDefaultValueID, [248](#)
- GetDefaultValueOfActiveStatusAccount, [174](#)
- GetDefaultView, [260](#)
- GetDescription
 - HsvMDDataBuffer object, [618](#)
 - HsvMDDataBufferLite object, [647](#)
 - IHsvTreeInfo interface, [215](#)
- GetDescriptionAtIndex
 - HsvMDDataBuffer object, [619](#)
 - HsvMDDataBufferLite object, [648](#)
- GetDisplayInfo, [216](#)
- GetDisplayInfoForSeveralItems, [217](#)
- GetDocument, [603](#)
- GetDocumentEx, [603](#)
- GetEntitiesContacts, [716](#)
- GetEntityCurrencyID, [716](#)
- GetEntityJournals, [405](#)
- GetEntityPartnerOption, [677](#)
- GetEPUInfo, [384](#)
- GetErrorStatus, [678](#)
- GetExtractFileEncoding, [512](#)
- GetExtractLogData, [704](#)
- GetFdmAppName, [174](#)
- GetFileTransfer, [154](#)
- GetFilterOptions, [678](#)
- GetFixedDimensionMembers, [678](#)
 - HsvCurrencyCube object, [690](#)
 - HsvNodeCube object, [695](#)
 - HsvTransactionData object, [669](#)
- GetFormattedDateTime, [513](#), [815](#)
- GetFormattedDateTimeForLanguage, [815](#)
- GetFormattedError, [816](#)
- GetFormattedErrorWithLineFeed, [817](#)
- GetFormattedResourceString, [513](#), [818](#)
- GetFrequency, [257](#)
- GetFrequencyID, [174](#)
- GetGroupPhaseFromCell, [536](#)
- GetHelpDirectoryForLanguageID, [818](#)
- GetHFMErrorsLogRecordSet
 - HsxClient object, [138](#)
 - HsxServer object, [154](#)
- GetHFMLanguageIdFromUserLanguages, [818](#)
- GetHistory, [537](#)
- GetHoldingCompany, [249](#)
- GetICEntitiesLockStatus, [733](#)
- GetICPEntitiesAggregationWeight, [175](#)
- GetICPTopMember, [233](#)
- GetICReasonCodeID, [717](#)
- GetICReasonCodeLabel, [717](#)
- GetICReasonCodes, [717](#)
- GetICTransactionCell, [679](#)
- GetICTransactionData, [679](#)
- GetICTransactions, [718](#)
- GetIdentityTypes, [470](#)
- GetIsICP, [233](#)
- GetItem, [673](#)
- GetItemGeneration, [218](#)
- GetItemID
 - HsvJournals object, [396](#)
 - IHsvTreeInfo interface, [219](#)
- GetItemIDQL, [220](#)
- GetItemLevel, [220](#)
- GetJournal, [406](#)
- GetJournal2, [409](#)
- GetJournalDisplayData, [411](#)
- GetJournalLabelsForIDs, [412](#)
- GetJournalQueryDefinitionIDs, [413](#)

- GetJournalTemplateItemID, [396](#)
- GetKillUsersStatus, [513](#)
- GetKillUserStatus, [513](#)
- GetLabel, [221](#)
- GetLanguageCountryCodeFromLanguageId, [819](#)
- GetLanguageUserParameters, [514](#)
- GetLastModifiedDateForArtifact, [514](#)
- GetLicenseExpirationStatus, [159](#)
- GetLineItems
 - HsvData object, [333](#)
 - HsvMDDataBuffer object, [620](#)
 - HsvMDDataBufferLite object, [649](#)
- GetLineItemsAtIndex
 - HsvMDDataBuffer object, [621](#)
 - HsvMDDataBufferLite object, [650](#)
- GetListInfo, [798](#)
- GetLocaleIdFromLanguageId, [819](#)
- GetLockStatusICEntities, [733](#)
- GetLockStatusICEntity, [733](#)
- GetLogonInfo, [138](#)
- GetLogonInfoSSO, [138](#)
- GetMatchCode, [681](#)
- GetMaxCellTextSize, [333](#)
- GetMaximumReviewLevel, [261](#)
- GetMember, [799](#)
- GetMemberID, custom function, [219](#)
- GetMemberListID, [221](#)
- GetMemberListName, [222](#)
- GetMembersThatHaveData, [333](#)
- GetMissingDataZeroViewForAdjValues, [262](#)
- GetMissingDataZeroViewForNonAdjValues, [262](#)
- GetModuleName, [514](#)
- GetMonitorICDetails, [718](#)
- GetMonitorICSummary, [719](#)
- GetMonitorICTransactions, [720](#)
- GetNodeCube, [334](#)
- GetNumBaseMembers, [222](#)
- GetNumberFormattingUserParameters, [514](#)
- GetNumCellsForData
 - HsvMDDataBuffer object, [622](#)
 - HsvMDDataBufferLite object, [652](#)
- GetNumCellsForDescriptions
 - HsvMDDataBuffer object, [623](#)
 - HsvMDDataBufferLite object, [652](#)
- GetNumCellsForLineItems
 - HsvMDDataBuffer object, [623](#)
 - HsvMDDataBufferLite object, [653](#)
- GetNumCheckedItems, [800](#)
- GetNumChildren, [223](#)
- GetNumDecimalPlaces, [234](#)
- GetNumDescendants, [223](#)
- GetNumItems, [674](#)
- GetNumMembers, [224](#)
- GetNumParents, [225](#)
- GetNumPeriodsInCubeForData
 - HsvMDDataBuffer object, [624](#)
 - HsvMDDataBufferLite object, [653](#)
- GetNumPeriodsInCubeForDescriptions
 - HsvMDDataBuffer object, [624](#)
 - HsvMDDataBufferLite object, [654](#)
- GetNumPeriodsInCubeForLineItems
 - HsvMDDataBuffer object, [625](#)
 - HsvMDDataBufferLite object, [654](#)
- GetNumPeriodsInFrequency, [257](#)
- GetNumRoles, [470](#)
- GetNumTransactionsCached, [681](#)
- GetOBPInfo, [801](#)
- GetOneCellFromStoredItem
 - HsvCurrencyCube object, [690](#)
 - HsvNodeCube object, [695](#)
- GetOwner, [470](#)
- GetPagingOption, [682](#)
- GetPartnerAsEntityList, [682](#)
- GetPartnerQueryDimensionMemberIDs, [682](#)
- GetPeriodFromFrequency, [258](#)
- GetPeriodIndexFromPOVForData
 - HsvMDDataBuffer object, [625](#)
 - HsvMDDataBufferLite object, [654](#)
- GetPeriodIndexFromPOVForDescriptions
 - HsvMDDataBuffer object, [626](#)
 - HsvMDDataBufferLite object, [656](#)
- GetPeriodIndexFromPOVForLineItems
 - HsvMDDataBuffer object, [627](#)
 - HsvMDDataBufferLite object, [657](#)
- GetPeriodPOVFromIndexForData
 - HsvMDDataBuffer object, [628](#)
 - HsvMDDataBufferLite object, [658](#)
- GetPeriodPOVFromIndexForDescriptions
 - HsvMDDataBuffer object, [629](#)
 - HsvMDDataBufferLite object, [658](#)
- GetPeriodPOVFromIndexForLineItems
 - HsvMDDataBuffer object, [629](#)
 - HsvMDDataBufferLite object, [659](#)
- GetPeriodsInFrequency, [259](#)

GetPeriodStatusList, [397](#)
 GetPersistedSettings, [705](#)
 GetPhasedSubmissionHistory, [540](#)
 GetPhasedSubmissionStartYear, [260](#)
 GetPhasedSubmissionState, [541](#)
 GetPhasedSubmissionStateUsingPhaseID, [542](#)
 GetPhaseSubmissionGridForGivenScenarioPeriod, [335](#)
 GetPlugAccount, [235](#)
 GetPMErrorRecordCount
 HsvMDDDataBuffer object, [630](#)
 HsvMDDDataBufferLite object, [659](#)
 GetPMRecordCount
 HsvMDDDataBuffer object, [630](#)
 HsvMDDDataBufferLite object, [659](#)
 GetPOVFromStoredItem
 HsvCurrencyCube object, [692](#)
 HsvNodeCube object, [697](#)
 GetProcessUnitAccessRights, [486](#)
 GetProcessUnitAccessRightsAndState, [487](#)
 GetProcessUnitAccessRightsAndStateEx, [487](#)
 GetProcessUnitAccessRightsEx, [489](#)
 GetQualifiedLabel, [225](#)
 GetQueryDimensionMemberIDs, [682](#)
 GetQueryFieldInformation, [682](#)
 GetQueryFieldItem, [682](#)
 GetQueryItem, [669](#)
 GetQueryOrderField, [682](#)
 GetRecordFromPMBuffer
 HsvMDDDataBuffer object, [630](#)
 HsvMDDDataBufferLite object, [659](#)
 GetRecordFromPMErrorBuffer
 HsvMDDDataBuffer object, [630](#)
 HsvMDDDataBufferLite object, [659](#)
 GetReport, [604](#)
 GetReportData, [447](#)
 GetReportData2, [449](#)
 GetResourceLanguageUserParameters, [515](#)
 GetResourceString
 HsvResourceManager object, [820](#)
 HsvSystemInfo object, [515](#)
 GetResourceStringFromHR, [820](#)
 GetRoleAccessForAllUsers, [470](#)
 GetRoleAccessForAllUsers2, [471](#)
 GetRoleID, [471](#)
 GetRoleLabel, [471](#)
 GetRowFilter, [721](#)
 GetRulesMode, [472](#)
 GetRunningTaskLogFilePathName, [515](#)
 GetRunningTaskProgress, [516](#)
 GetRunningTasksCount, [516](#)
 GetRunningTaskStatus, [516](#)
 GetScale, [274](#)
 GetSecurityAsPartnerID, [249](#)
 GetSecurityClassAccessForAllUsers, [472](#)
 GetSecurityClassAccessForAllUsers2, [472](#)
 GetSecurityClassID, [244](#)
 HsvAccounts object, [236](#)
 HsvEntities object, [250](#)
 HsvScenarios object, [263](#)
 HsvSecurityAccess object, [472](#)
 GetSecurityClassLabel, [473](#)
 GetSecurityClassRightsForConnectedUser, [473](#)
 GetServer
 HsxClient object, [139](#)
 HsxClientUI object, [148](#)
 GetServerName, [517](#)
 GetServerOnCluster, [139](#)
 GetSettingsICPeriod, [734](#)
 GetSettingsICPeriods, [735](#)
 GetSortedNature
 HsvMDDDataBuffer object, [630](#)
 HsvMDDDataBufferLite object, [659](#)
 GetSSOTokenUsingWebSecurityAgentCredentials, [139](#)
 GetState, [543](#)
 GetStatus, [335](#)
 GetStatusEx, [337](#)
 GetStatusUsingPhaseID, [338](#)
 GetSubmissionGroup
 HsvAccounts object, [236](#)
 HsvCustom object, [245](#)
 HsvICPs object, [255](#)
 GetSupportSubmissionPhaseForAccountFlag, [175](#)
 GetSupportSubmissionPhaseForCustom1Flag, [175](#)
 GetSupportSubmissionPhaseForCustom2Flag, [176](#)
 GetSupportSubmissionPhaseForCustom3Flag, [176](#)
 GetSupportSubmissionPhaseForCustom4Flag, [176](#)
 GetSupportSubmissionPhaseForICPFlag, [177](#)
 GetSystemDataLinkFile, [154](#)
 GetSystemFolder, [154](#)
 GetTaskAccessForConnectedUserFromList, [474](#)
 GetTemplate, [415](#), [707](#)
 GetTemplateDisplayData, [417](#)

[GetTemplateLabelsForIDs](#), 418
[GetTemplateQueryDefinitionIDs](#), 418
[GetTextCell](#), 339
[GetTextCellLineItems](#), 341
[GetTextCells](#), 342
[GetTextCellsWithRowSuppression](#), 344
[GetTextCellsWithRowSuppression2](#), 346
[GetTextJournal](#), 420
[GetTextLineItems](#), 348
[GetTextTemplate](#), 422
[GetTopMemberOfValidCustom1Hierarchy](#), 237
[GetTopMemberOfValidCustom2Hierarchy](#), 237
[GetTopMemberOfValidCustom3Hierarchy](#), 238
[GetTopMemberOfValidCustom4Hierarchy](#), 238
[GetTotalTransactions](#), 682
[GetTransactionData](#)
 HsvData object, 348
 HsvTransactionData object, 669
[GetTransCurrencyID](#), 721
[GetTransGroupType](#), 682
[GetTreeCapabilities](#), 225
[GetTreeName](#), 226
[GetTreeTimeStamp](#), 226
[GetUnassignedGroups](#), 349
[GetUnMatchTransactions](#), 721
[GetURLByName](#), 349
[GetURLsForCell](#), 350
[GetUserAccessForAllRoles](#), 474
[GetUserAccessForAllRoles2](#), 475
[GetUserAccessForAllSecurityClasses](#), 475
[GetUserAccessForAllSecurityClasses2](#), 475
[GetUserDefined1](#), 245
 HsvAccounts object, 239
 HsvEntities object, 251
 HsvScenarios object, 264
[GetUserDefined2](#), 245
 HsvAccounts object, 239
 HsvEntities object, 251
 HsvScenarios object, 265
[GetUserDefined3](#), 246
 HsvAccounts object, 239
 HsvEntities object, 251
 HsvScenarios object, 265
[GetUserDisplayDataTimeFormats](#), 820
[GetUserID](#), 475
[GetUserIDFromSID](#), 476
[GetUserInfoFromUniqueID](#), 476

[GetUserInfoFromUniqueID2](#), 476
[GetUserLanguageFromHFMLanguageId](#), 820
[GetUserName](#)
 HsvSecurityAccess object, 477
 HsvSystemInfo object, 517
[GetUserName2](#), 477
[GetUserNameFromSID](#), 477
[GetUserParameter](#), 518
[GetUserSID](#), 477
[GetUseSubmissionPhaseFlag](#), 177
[GetValidationAccount](#), 178
[GetValueIDFromCurrencyID](#), 269
[GetVariance](#), 425
[GetVBScriptCalcRules](#), 518
[GetVBScriptMemberListRules](#), 519
[GetVBScriptRules](#), 385
[GetWebSecurityAgentSettings](#), 139
[GetWindowsDateFormat](#), 821
[GetWindowsDateFormatForLocale](#), 821
[GetWorkingDirectory](#), 520
[GetXBRLLTags](#), 240
[GetXMLErrorFromDatabase](#), 155
[GetXMLErrorsListFromDatabase](#), 155
[GetYearRange](#), 269

H

[HasChildren](#), 227
[HasSystemChanged](#), 159
[HasUserStatusChanged](#), 160
[HFM_ACCESS_RIGHTS_ALL](#), 451
[HFM_ACCESS_RIGHTS_NONE](#), 451
[HFM_ACCESS_RIGHTS_READANDPROMOTE](#), 451
[HFM_ACCESS_RIGHTS_READONLY](#), 451
[HFM_ACCESS_RIGHTS_VIEW](#), 452
[HFM_USER_GROUP_ENUM_FLAGS](#), 453
[HoldingCompany](#) attribute, returning, 249
[HRESULTS](#), utility for, 822
[HSV_DATA_UPDATE_MODE](#), 276
[HSV_DATAEXTRACT_OPTION](#) constants, 781
[HSV_DATALOAD_OPTION](#) constants, 778
[HSV_JOURNALEXTRACT_EX_OPTION](#) constants, 791
[HSV_JOURNALEXTRACT_OPTION](#) constants, 789
[HSV_JOURNALLOAD_OPTION](#) constants, 789
[HSV_METALOADEX_FILE_FORMAT](#) constants, 767

- HSV_SECURITYEXTRACT_OPTION constants, [756](#)
- HSV_SECURITYLOAD_OPTION constants, [755](#)
- HsvAccounts object
 - [GetAccountType](#), [232](#)
 - [GetCalcAttribute](#), [233](#)
 - [GetICPTopMember](#), [233](#)
 - [GetIsICP](#), [233](#)
 - [GetNumDecimalPlaces](#), [234](#)
 - [GetPlugAccount](#), [235](#)
 - [GetSecurityClassID](#), [236](#)
 - [GetSubmissionGroup](#), [236](#)
 - [GetTopMemberOfValidCustom1Hierarchy](#), [237](#)
 - [GetTopMemberOfValidCustom2Hierarchy](#), [237](#)
 - [GetTopMemberOfValidCustom3Hierarchy](#), [238](#)
 - [GetTopMemberOfValidCustom4Hierarchy](#), [238](#)
 - [GetUserDefined1](#), [239](#)
 - [GetUserDefined2](#), [239](#)
 - [GetUserDefined3](#), [239](#)
 - [GetXBRLETags](#), [240](#)
 - [IsCalculated](#), [240](#)
 - [IsConsolidated](#), [240](#)
 - [IsCustom1AggregationEnabled](#), [241](#)
 - [IsCustom2AggregationEnabled](#), [241](#)
 - [IsCustom3AggregationEnabled](#), [242](#)
 - [IsCustom4AggregationEnabled](#), [242](#)
 - [IsICP](#), [243](#)
 - [IsICPRestricted](#), [243](#)
 - methods, summarized, [59](#)
 - object references, assigning, [232](#)
 - overview, [232](#)
 - [UsesLineItems](#), [243](#)
- HsvAdminICM object
 - [CheckSecurityForICExtract](#), [732](#)
 - [GetICEntitiesLockStatus](#), [733](#)
 - [OpenICPeriod2](#), [740](#)
 - [UpdatePeriodSettings2](#), [741](#)
- HsvCalculate object
 - [Allocate](#), [376](#)
 - [Allocate2](#), [376](#)
 - [CalcEPU](#), [377](#)
 - [ChartLogic](#), [377](#)
 - [ChartLogic2](#), [378](#)
 - [Consolidate](#), [379](#)
 - [Consolidate2](#), [380](#)
 - [CustomLogic](#), [381](#)
 - [FindOverlappingConsolidation](#), [381](#)
 - [GetCOMDLLRules](#), [382](#)
 - [GetConsolidationProgress](#), [382](#)
 - [GetDefaultExchangeRate](#), [383](#)
 - [GetEPUInfo](#), [384](#)
 - [GetVBScriptRules](#), [385](#)
 - [IsEntityAnEPUOwner](#), [385](#)
 - [LoadCalcManagerRules](#), [386](#)
 - methods, summarized, [74](#)
 - object references, assigning, [376](#)
 - overview, [376](#)
 - [SetCOMDLLRules](#), [386](#)
 - [SetVBScriptRules](#), [386](#)
 - [SetVBScriptRules2](#), [388](#)
 - [StopConsolidation](#), [390](#)
 - [Translate](#), [390](#)
 - [Translate2](#), [391](#)
- HsvCalculate type library
 - HsvCalculate object, [376](#)
 - overview, [375](#)
- HsvCategories object. *See* HsvScenarios object
- HsvcDataLoad object
 - [DMELoad](#), [783](#)
 - [Extract](#), [749](#)
 - extract options, [781](#)
 - [ExtractOptions](#), [748](#)
 - [Load](#), [750](#)
 - load options, [778](#)
 - [Load2](#), [783](#)
 - [LoadOptions](#), [748](#)
 - methods, summarized, [114](#)
 - object references, assigning, [783](#)
 - overview, [783](#)
 - properties, summarized, [113](#)
 - [SetFileForLoad](#), [784](#)
 - [SetSession](#), [751](#)
 - [StartLoad](#), [784](#)
- HsvcDataLoad type library
 - [HsvcDataLoad](#), [783](#)
 - IHsvLoadExtractOption interface, [775](#)
 - IHsvLoadExtractOptions interface, [775](#)
 - overview, [775](#)
- HsvCubeData object (deprecated), [609](#)
- HsvCurrencies object, [270](#)
 - [EnumCurrencies](#), [270](#)
 - [EnumCurrencies2](#), [271](#)
 - [GetCurrencyDescription](#), [272](#)
 - [GetCurrencyID](#), [272](#)

- GetCurrencyLabel, 273
- GetCurrencyTranslationOperator, 273
- GetScale, 274
- HsvCurrencyCube object
 - BeginEnumerationOfStoredData, 688
 - EndEnumerationOfStoredData, 688
 - GetCell, 689
 - GetFixedDimensionMembers, 690
 - GetOneCellFromStoredItem, 690
 - GetPOVFromStoredItem, 692
 - methods, summarized, 103
 - overview, 688
- HsvCustom object
 - GetAggregationWeight, 244
 - GetSecurityClassID, 244
 - GetSubmissionGroup, 245
 - GetUserDefined1, 245
 - GetUserDefined2, 245
 - GetUserDefined3, 246
 - IsCalculated, 246
 - IsSwitchSignEnabledForFlow, 247
 - IsSwitchTypeEnabledForFlow, 247
 - methods, summarized, 61
 - object references, assigning, 244
 - overview, 244
- HsvData object
 - AddDataToMDDDataBuffer, 277
 - AttachDocumentToCell, 278
 - ClearAllData, 279
 - ClearAllDescriptionsInSubCube, 279
 - ClearDataAuditItems, 280
 - ClearInputData, 281
 - ClearInvalidData, 283
 - CopyInputData, 283
 - CopyInputDataForMultipleEntities, 284
 - DeleteLineItemDetails, 285
 - DetachDocumentFromCell, 286
 - DMEload, 287
 - DoCellDescriptionsExist, 288
 - DoesCellDescriptionExist, 288
 - DoesDataExist, 289
 - DoesSparseDataExist, 290
 - EnumDataAuditItems, 291
 - EnumDataAuditItems2, 293
 - EnumEntitiesWithDataForScenarioYear, 293
 - EnumExtractOptions, 294
 - EnumLoadOptions, 295
 - Extract, 295
 - ExtractDataAuditItems, 299
 - ExtractDrillableRegions, 300
 - ExtractDrillableRegionsByURLNames, 301
 - FilterMembersThatHaveData, 301
 - FilterMembersThatHaveData2, 302
 - FormatNumberToText, 303
 - FormatNumberToText2, 305
 - FormatStatusToText, 307
 - FormatStoredNumberToText, 308
 - FormatTextToNumber, 309
 - FormatTextToStoredNumber, 310
 - GetAllDescriptionsInSubCube, 311
 - GetAllURLNames, 312
 - GetAttachedDocumentsToCell, 312
 - GetBaseDataForAccount, 313
 - GetCalcStatus, 313
 - GetCalcStatusEx, 314
 - GetCalcStatusStatistics, 315
 - GetCell, 316
 - GetCellDescription, 317
 - GetCellDescriptions, 318
 - GetCellHistory, 319
 - GetCellHistory2, 321
 - GetCellJournalEntries, 323
 - GetCellLineItems, 325
 - GetCells, 326
 - GetCellsWithRowSuppression, 327
 - GetCellsWithRowSuppression2, 330
 - GetCountOfAttachedDocumentsToCell, 331
 - GetCurrencyCube, 332
 - GetDataForAllMetadataCombinations, 333
 - GetLineItems, 333
 - GetMaxCellTextSize, 333
 - GetMembersThatHaveData, 333
 - GetNodeCube, 334
 - GetPhaseSubmissionGridForGivenScenarioPeriod, 335
 - GetStatus, 335
 - GetStatusEx, 337
 - GetStatusUsingPhaseID, 338
 - GetTextCell, 339
 - GetTextCellLineItems, 341
 - GetTextCells, 342
 - GetTextCellsWithRowSuppression, 344
 - GetTextCellsWithRowSuppression2, 346
 - GetTextLineItems, 348

- GetTransactionData, [348](#)
- GetUnassignedGroups, [349](#)
- GetURLByName, [349](#)
- GetURLsForCell, [350](#)
- InsertLineItemDetails, [352](#)
- IsValidCellText, [353](#)
- Load, [353](#)
- LoadDrillableRegions, [355](#)
- methods, summarized, [69](#)
- object references, assigning, [69](#)
- overview, [276](#)
- SetCalcStatusLocked, [357](#)
- SetCalcStatusLockedForMultipleProcessUnits, [358](#)
- SetCalcStatusUnlocked, [358](#)
- SetCalcStatusUnlockedForMultipleProcessUnits, [359](#)
- SetCell, [359](#)
- SetCellDescriptions, [361](#)
- SetCellLineItems, [362](#)
- SetCells, [363](#)
- SetCells2, [365](#)
- SetCellsLineItems, [367](#)
- SetFileForLoad, [368](#)
- SetLineItems, [368](#)
- SetPhaseSubmissionGridForGivenScenarioPeriod, [368](#)
- SetTextCell, [369](#)
- SetTextCellLineItems, [370](#)
- SetTextCells, [371](#)
- SetTextCellsLineItems, [373](#)
- SetTextLineItems, [374](#)
- StartLoad, [374](#)
- UpdateDataUsingMDDDataBuffer, [374](#)
- HsvData type library
 - HsvData object, [276](#)
 - HsvDataPipe object, [275](#)
 - IHsvDataValidation interface, [275](#)
 - overview, [275](#)
- HsvDataCubes type library
 - HsvCurrencyCube, [688](#)
 - HsvNodeCube, [693](#)
 - overview, [687](#)
- HsvDataPipe object, [275](#)
- HsvEntities object
 - GetAllowAdjustments, [247](#)
 - GetAllowAdjustmentsFromChildren, [248](#)
 - GetDefaultValueID, [248](#)
 - GetHoldingCompany, [249](#)
 - GetSecurityAsPartnerID, [249](#)
 - GetSecurityClassID, [250](#)
 - GetUserDefined1, [251](#)
 - GetUserDefined2, [251](#)
 - GetUserDefined3, [251](#)
 - IsChild, [252](#)
 - IsDescendant, [252](#)
 - IsICP, [253](#)
 - IsOrgByPeriodFilteringOn, [253](#)
 - IsSecurityAsPartnerEnabled, [254](#)
 - methods, summarized, [62](#)
 - object references, assigning, [247](#), [254](#)
 - overview, [247](#)
- HsvICM interface
 - object references, assigning, [711](#)
- HsvICM object, [711](#)
 - AllMVTransPosted, [711](#)
 - CheckReportSecurity, [712](#)
 - CreateICTransaction, [712](#)
 - DeleteICReasonCode, [714](#)
 - DoesCellSupportICTransactionDetail, [714](#)
 - GetColumnFilter, [715](#)
 - GetCurrencyInfo, [715](#)
 - GetEntitiesContacts, [716](#)
 - GetEntityCurrencyID, [716](#)
 - GetICReasonCodeID, [717](#)
 - GetICReasonCodeLabel, [717](#)
 - GetICReasonCodes, [717](#)
 - GetICTransactions, [718](#)
 - GetMonitorICDetails, [718](#)
 - GetMonitorICSummary, [719](#)
 - GetMonitorICTransactions, [720](#)
 - GetRowFilter, [721](#)
 - GetTransCurrencyID, [721](#)
 - GetUnMatchTransactions, [721](#)
 - IsOneSideOfTransactionGroupWriteable, [722](#)
 - MatchAutoAccounts, [723](#)
 - MatchAutoIDs, [724](#)
 - NoTransMatchedOrPosted, [725](#)
 - ProcessAllICTrans, [725](#)
 - ProcessICTransactions, [726](#)
 - SaveColumnFilter, [728](#)
 - SaveICReasonCode, [728](#)
 - SaveICTransaction, [728](#)
 - SaveRowFilter, [730](#)
 - SetReasonCodeToICTransactions, [730](#)

- UnMatchICTransactions, [731](#)
- HsvICM type library, [711](#)
 - HsvICM, [711](#)
 - IHsvAdminICM, [732](#)
- HsvICPs object
 - GetSubmissionGroup, [255](#)
- HsvICPs object references, assigning, [187](#)
- HsvICPs property, [187](#)
- HsvICTransactionsData object, [674](#)
 - AddAccountCustomCombination, [675](#)
 - AddICTransactionData, [675](#)
 - AddQueryDimensionMemberID, [675](#)
 - AddQueryField, [675](#)
 - AddQueryFieldItem, [675](#)
 - AddQueryOrderField, [675](#)
 - BeginDataEnum, [676](#)
 - BeginQueryFieldEnum, [676](#)
 - BeginQueryOrderEnum, [676](#)
 - EndDataEnum, [676](#)
 - EndQueryFieldEnum, [677](#)
 - EndQueryOrderEnum, [677](#)
 - GetAccessRights, [677](#)
 - GetAccountCustomCombination, [677](#)
 - GetAccountCustomCombinationTotal, [677](#)
 - GetEntityPartnerOption, [677](#)
 - GetErrorStatus, [678](#)
 - GetFilterOptions, [678](#)
 - GetFixedDimensionMembers, [678](#)
 - GetICTransactionCell, [679](#)
 - GetICTransactionData, [679](#)
 - GetMatchCode, [681](#)
 - GetNumTransactionsCached, [681](#)
 - GetPagingOption, [682](#)
 - GetPartnerAsEntityList, [682](#)
 - GetPartnerQueryDimensionMemberIDs, [682](#)
 - GetQueryDimensionMemberIDs, [682](#)
 - GetQueryFieldInformation, [682](#)
 - GetQueryFieldItem, [682](#)
 - GetQueryOrderField, [682](#)
 - GetTotalTransactions, [682](#)
 - GetTransGroupType, [682](#)
 - Initialize, [682](#)
 - InitializeSequenceIdMap, [683](#)
 - IsEntityInPartnerAsEntityList, [684](#)
 - IsICTransactionValid, [684](#)
 - object references, assigning, [674](#)
 - RemoveQueryOrder, [684](#)
 - SetAccessRights, [684](#)
 - SetEntityPartnerOption, [684](#)
 - SetErrorStatus, [685](#)
 - SetFilterOptions, [685](#)
 - SetICTransactionData, [685](#)
 - SetPagingOption, [685](#)
 - SetPartnerAsEntityList, [685](#)
 - SetPartnerQueryDimensionMemberIDs, [685](#)
 - SetQueryDimensionMemberIDs, [685](#)
 - SetTotalTransactions, [685](#)
 - SortByCell, [685](#)
 - UnInitialize, [685](#)
- HsvJournalLoadACV object
 - Extract, [749](#)
 - extract options
 - filtered extracts, [791](#)
 - unfiltered extracts, [789](#)
 - ExtractEx, [794](#)
 - ExtractOptions, [748](#)
 - ExtractOptionsEx, [793](#)
 - Load, [750](#)
 - load options, [789](#)
 - LoadOptions, [748](#)
 - methods, [793](#)
 - methods, summarized, [115](#)
 - object references, assigning, [793](#)
 - properties, [793](#)
 - properties, summarized, [115](#)
 - SetSession, [751](#)
- HsvJournalLoadACV type library
 - HsvJournalLoadACV, [793](#)
 - IHsvLoadExtractOption interface, [785](#)
 - IHsvLoadExtractOptions interface, [785](#)
 - overview, [784](#)
- HsvJournals object
 - ClosePeriod, [395](#)
 - GetItemID, [396](#)
 - GetJournalTemplateItemID, [396](#)
 - GetPeriodStatusList, [397](#)
 - IsPeriodOpen, [398](#)
 - methods, summarized, [76](#)
 - object references, assigning, [395](#)
 - OpenPeriod, [398](#)
 - overview, [395](#)
- HsvJournals type library
 - HsvJournals object, [395](#)
 - IHsvJournalsEx interface, [399](#)

- IHsvJournalsReport interface, [444](#)
- overview, [393](#)
- HsvJournalsEx object
 - GetJournal2, [409](#)
- HsvMDArrays type library
 - HsvCubeData (deprecated), [609](#)
 - HsvICTransactionsData, [674](#)
 - HsvMDCube (deprecated), [609](#)
 - HsvMDDataBuffer, [610](#)
 - HsvMDDataBufferLite, [639](#)
 - HsvMDIndexList, [673](#)
 - HsvMetadataSecurityBuffer (deprecated), [609](#)
 - HsvTransactionData, [667](#)
 - IHsvMDDataBufferLite, [667](#)
 - overview, [609](#)
- HsvMDCube object (deprecated), [609](#)
- HsvMDDataBuffer object
 - BeginEnumeration, [611](#)
 - CreateDataIndexList, [611](#)
 - CreateDataIndexListEx, [613](#)
 - EndEnumeration, [613](#)
 - EraseRecordFromPMBuffer, [613](#)
 - GetCheckLineItemDetailsForCaseInsensitiveDuplicates, [613](#)
 - GetCubeIndexFromPOV, [614](#)
 - GetCubePOVFromIndex, [614](#)
 - GetData, [615](#)
 - GetDataAtIndex, [616](#)
 - GetDescription, [618](#)
 - GetDescriptionAtIndex, [619](#)
 - GetLineItems, [620](#)
 - GetLineItemsAtIndex, [621](#)
 - GetNumCellsForData, [622](#)
 - GetNumCellsForDescriptions, [623](#)
 - GetNumCellsForLineItems, [623](#)
 - GetNumPeriodsInCubeForData, [624](#)
 - GetNumPeriodsInCubeForDescriptions, [624](#)
 - GetNumPeriodsInCubeForLineItems, [625](#)
 - GetPeriodIndexFromPOV, [626](#)
 - GetPeriodIndexFromPOVForData, [625](#)
 - GetPeriodIndexFromPOVForLineItems, [627](#)
 - GetPeriodPOVFromIndex, [629](#)
 - GetPeriodPOVFromIndexForData, [628](#)
 - GetPeriodPOVFromIndexForLineItems, [629](#)
 - GetPMErrorRecordCount, [630](#)
 - GetPMRecordCount, [630](#)
 - GetRecordFromPMBuffer, [630](#)
 - GetRecordFromPMErrorBuffer, [630](#)
 - GetSortedNature, [630](#)
 - indexes for methods, obtaining, [610](#)
 - InsertDataAtBeginning, [630](#)
 - InsertDataAtEnd, [631](#)
 - InsertDescriptionAtBeginning, [632](#)
 - InsertDescriptionAtEnd, [633](#)
 - InsertLineItemsAtBeginning, [633](#)
 - InsertLineItemsAtEnd, [634](#)
 - InsertRecordIntoPMBuffer, [635](#)
 - InsertRecordIntoPMErrorBuffer, [635](#)
 - methods, summarized, [94](#)
 - object references, assigning, [93](#)
 - overview, [610](#)
 - RemoveAll, [635](#)
 - SetCheckLineItemDetailsForCaseInsensitiveDuplicates, [635](#)
 - SetData, [636](#)
 - SetDescription, [637](#)
 - SetGrowByAmount, [638](#)
 - SetLineItems, [638](#)
 - SetSortedNature, [639](#)
 - Sort, [639](#)
- HsvMDDataBufferLite object
 - BeginEnumeration, [640](#)
 - CreateDataIndexList, [641](#)
 - EndEnumeration, [643](#)
 - EraseRecordFromPMBuffer, [643](#)
 - GetCheckLineItemDetailsForCaseInsensitiveDuplicates, [643](#)
 - GetCubeIndexFromPOV, [643](#)
 - GetCubePOVFromIndex, [644](#)
 - GetData, [645](#)
 - GetDataAtIndex, [646](#)
 - GetDescription, [647](#)
 - GetDescriptionAtIndex, [648](#)
 - GetLineItems, [649](#)
 - GetLineItemsAtIndex, [650](#)
 - GetNumCellsForData, [652](#)
 - GetNumCellsForDescriptions, [652](#)
 - GetNumCellsForLineItems, [653](#)
 - GetNumPeriodsInCubeForData, [653](#)
 - GetNumPeriodsInCubeForDescriptions, [654](#)
 - GetNumPeriodsInCubeForLineItems, [654](#)
 - GetPeriodIndexFromPOVForData, [654](#)
 - GetPeriodIndexFromPOVForDescriptions, [656](#)
 - GetPeriodIndexFromPOVForLineItems, [657](#)

- GetPeriodPOVFromIndexForData, 658
- GetPeriodPOVFromIndexForDescriptions, 658
- GetPeriodPOVFromIndexForLineItems, 659
- GetPMErrorRecordCount, 659
- GetPMRecordCount, 659
- GetRecordFromPMBuffer, 659
- GetRecordFromPMErrorBuffer, 659
- GetSortedNature, 659
- indexes for methods, obtaining, 610
- InsertDataAtEnd, 660
- InsertDescriptionAtEnd, 661
- InsertLineItemsAtEnd, 661
- InsertRecordIntoPMBuffer, 662
- InsertRecordIntoPMErrorBuffer, 662
- methods, summarized, 96
- object references, assigning, 96
- overview, 639
- RemoveAll, 662
- SetCheckLineItemDetailsForCaseInsensitiveDuplications, 663
- SetData, 663
- SetDescription, 664
- SetGrowByAmount, 665
- SetLineItems, 665
- SetSortedNature, 666
- Sort, 666
- HsvMDIndexList object, 673
 - GetItem, 673
 - GetNumItems, 674
- HsvMetadata object
 - Accounts, 185
 - Currencies, 186
 - Custom1, 186
 - Custom2, 186
 - Custom3, 186
 - Custom4, 186
 - Dimension, 186
 - Entities, 187
 - EnumConsolidationMethodIDs, 162
 - EnumExtractOptions, 163
 - EnumLanguages, 164
 - EnumLoadOptions, 164
 - Extract, 165
 - GetApplicationAttribute, 167
 - GetApplicationCurrency, 168
 - GetApplicationSettingsTimeStamp, 168
 - GetByIndexValidationAccount, 169
 - GetCellLevelAccountType, 170
 - GetConsolidationMethodDescription, 170
 - GetConsolidationMethodInfo, 171
 - GetConsolidationMethodsTimeStamp, 172
 - GetCurrencyTimeStamp, 173
 - GetCurrencyValueIDForEntityValueCombination, 173
 - GetCurrencyValueIDsForEntityValueCombinations, 173
 - GetDefaultValueOfActiveStatusAccount, 174
 - GetFdmAppName, 174
 - GetFrequencyID, 174
 - GetICPEntitiesAggregationWeight, 175
 - GetSupportSubmissionPhaseForAccountFlag, 175
 - GetSupportSubmissionPhaseForCustom1Flag, 175
 - GetSupportSubmissionPhaseForCustom2Flag, 176
 - GetSupportSubmissionPhaseForCustom3Flag, 176
 - GetSupportSubmissionPhaseForCustom4Flag, 176
 - GetSupportSubmissionPhaseForICPFlag, 177
 - GetUseSubmissionPhaseFlag, 177
 - GetValidationAccount, 178
 - ICPs, 187
 - IsCustomMemberValidForAccount, 179
 - IsOrgByPeriodApplication, 180
 - Load, 180
 - LoadWithAccessCode, 185
 - methods, summarized, 57
 - object references, assigning, 162
 - overview, 162
 - Periods, 187
 - properties, summarized, 56
 - Scenarios, 187
 - TranslateApplicationAttributeForDisplay, 185
 - Values, 188
 - Views, 188
 - Years, 188
- HsvMetadata type library
 - HsvAccounts object, 232
 - HsvCurrencies, 270
 - HsvCustom object, 244
 - HsvEntities object, 247
 - HsvMetadata object
 - methods, 162

- properties, 185
- HsvPeriods object, 256
- HsvScenarios object, 259
- HsvValues, 268
- HsvYears, 269
- IHsvTreeInfo interface, 188
- object hierarchy, 56
- overview, 161
- HsvMetadataLoadACV object
 - Extract, 749
 - ExtractOptions, 748
 - Load, 750
 - LoadOptions, 748
 - methods, 768
 - methods, summarized, 112
 - object references, assigning, 768
 - properties, 768
 - properties, summarized, 112
 - SetSession, 751
- HsvMetadataLoadACV type library
 - HsvMetadataLoadACV methods, 768
 - HsvMetadataLoadACV properties, 768
 - IHsvLoadExtractOption interface, 759
 - IHsvLoadExtractOptions interface, 759
 - overview, 758
- HsvMetadataSecurityBuffer object (deprecated), 609
- HsvNodeCube object
 - BeginEnumerationOfStoredData, 693
 - EndEnumerationOfStoredData, 694
 - GetCell, 694
 - GetFixedDimensionMembers, 695
 - GetOneCellFromStoredItem, 695
 - GetPOVFromStoredItem, 697
 - methods, summarized, 103
 - overview, 693
- HsvPeriods object
 - GetBaseFrequency, 256
 - GetFrequency, 257
 - GetNumPeriodsInFrequency, 257
 - GetPeriodFromFrequency, 258
 - GetPeriodsInFrequency, 259
 - methods, summarized, 63
 - object references, assigning, 256
 - overview, 256
- HsvPOVSelection object, 796
 - CheckAllItems, 796
 - CheckAllItemsUsingSelectedList, 796
 - CheckBoxChanged event, 807
 - DimensionWasSelected event, 808
 - EnableDimension, 797
 - events, 807
 - GetCheckedItems, 798
 - GetListInfo, 798
 - GetMember, 799
 - GetNumCheckedItems, 800
 - GetOBPInfo, 801
 - Initialize, 801
 - InitializeEx, 802
 - InitializeHTTP, 802
 - InitializeWithConnection, 802
 - multi-select mode, 806
 - MultiCheckBoxChanged event, 809
 - SelectDimension, 802
 - SelectionChanged event, 809
 - SetCheckedItems, 802
 - SetDimUnk, 804
 - SetListInfo, 804
 - SetMember, 805
 - SetOBPInfo, 805
 - SetResourceAndLanguage, 806
 - ShowCheckBoxes, 806
 - single-select mode, 806
- HsvPOVSelection type library, 795
 - HsvPOVSelection, 796
- HsvProcessFlow object
 - Approve, 532
 - Approve2, 533
 - ApproveEx, 534
 - GetGroupPhaseFromCell, 536
 - GetHistory, 537
 - GetPhasedSubmissionHistory, 540
 - GetPhasedSubmissionState, 541
 - GetPhasedSubmissionStateUsingPhaseID, 542
 - GetState, 543
 - methods, summarized, 88
 - object references, assigning, 531
 - overview, 531
 - PhasedSubmissionApprove, 544
 - PhasedSubmissionApprove2, 545
 - PhasedSubmissionApproveEx, 547
 - PhasedSubmissionGetHistory2, 548
 - PhasedSubmissionGetHistory2UsingPhaseID, 549
 - PhasedSubmissionProcessManagementChangeStateForMultipleEntities2, 551

- PhasedSubmissionPromote, [552](#)
- PhasedSubmissionPromote2, [554](#)
- PhasedSubmissionPublish, [555](#)
- PhasedSubmissionPublish2, [556](#)
- PhasedSubmissionPublishEx, [558](#)
- PhasedSubmissionReject, [559](#)
- PhasedSubmissionReject2, [560](#)
- PhasedSubmissionSignOff, [561](#)
- PhasedSubmissionSignOff2, [562](#)
- PhasedSubmissionStart, [564](#)
- PhasedSubmissionStart2, [565](#)
- PhasedSubmissionStartEx, [567](#)
- PhasedSubmissionSubmit, [569](#)
- PhasedSubmissionSubmit2, [570](#)
- ProcessManagementChangeStateForMultipleEntities, [571](#)
- ProcessManagementChangeStateForMultipleEntities2, [572](#)
- ProcessManagementChangeStateForMultipleEntitiesEx, [574](#)
- Promote, [575](#)
- Promote2, [576](#)
- Publish, [578](#)
- Publish2, [579](#)
- PublishEx, [581](#)
- Reject, [582](#)
- SignOff, [584](#)
- SignOff2, [585](#)
- Start, [586](#)
- Start2, [588](#)
- StartEx, [589](#)
- submission phases and process units, [531](#)
- Submit, [590](#)
- Submit2, [591](#)
- HsvProcessFlow type library
 - HsvProcessFlow object, [531](#)
 - overview, [531](#)
- HsvReports object
 - CheckReportSecurityClass, [595](#)
 - CheckReportSecurityClass2, [596](#)
 - CheckSecurityRole, [596](#)
 - DeleteDocument, [597](#)
 - DeleteReport, [597](#)
 - DeleteReports, [598](#)
 - EnumDocuments, [598](#)
 - EnumDocumentsEx, [600](#)
 - EnumReports, [602](#)
 - GetDocument, [603](#)
 - GetDocumentEx, [603](#)
 - GetReport, [604](#)
 - methods, summarized, [92](#)
 - object references, assigning, [92](#)
 - overview, [595](#)
 - Reports property and, [158](#)
 - SaveDocument, [605](#)
 - SaveDocumentEx, [606](#)
 - SetReport, [607](#)
- HsvReports type library
 - HsvReports, [595](#)
 - overview, [595](#)
- HsvResourceManager object, [812](#)
 - GetAvailableLanguages, [813](#)
 - GetCurrentHSSRegistrationVersion, [814](#)
 - GetCurrentVersion, [814](#)
 - GetCurrentVersionInUserDisplayFormat, [814](#)
 - GetFormattedDateTime, [815](#)
 - GetFormattedDateTimeForLanguage, [815](#)
 - GetFormattedError, [816](#)
 - GetFormattedErrorWithLineFeed, [817](#)
 - GetFormattedResourceString, [818](#)
 - GetHelpDirectoryForLanguageID, [818](#)
 - GetHFMLanguageIdFromUserLanguages, [818](#)
 - GetLanguageCountryCodeFromLanguageId, [819](#)
 - GetLocaleIdFromLanguageId, [819](#)
 - GetResourceString, [820](#)
 - GetResourceStringFromHR, [820](#)
 - GetUserDisplayDataTimeFormats, [820](#)
 - GetUserLanguageFromHFMLanguageId, [820](#)
 - GetWindowsDateFormat, [821](#)
 - GetWindowsDateFormatForLocale, [821](#)
 - Initialize, [821](#)
- HsvResourceManager type library, [811](#)
 - HsvResourceManager, [812](#)
- HsvRulesLoadACV object
 - ExtractCalcRules, [769](#)
 - ExtractMemberListRules, [769](#)
 - GetCalcRulesType, [770](#)
 - LoadCalcRules, [770](#)
 - LoadCalcRules2, [771](#)
 - LoadMemberListRules, [772](#)
 - methods, summarized, [112](#)
 - object references, assigning, [769](#)
 - overview, [769](#)
 - ScriptableLoadCalcRules, [773](#)

- ScriptableLoadCalcRules2, [773](#)
- ScriptableLoadMemberListRules, [773](#)
- SetSession, [773](#)
- SetSessionAndResource, [774](#)
- HsvRulesLoadACV type library
 - HsvRulesLoadACV, [769](#)
 - overview, [768](#)
- HsvScenarios object
 - EnumPhasedSubmissionStartYears, [260](#)
 - GetDefaultFrequency, [260](#)
 - GetDefaultView, [260](#)
 - GetMaximumReviewLevel, [261](#)
 - GetMissingDataZeroViewForAdjValues, [262](#)
 - GetMissingDataZeroViewForNonAdjValues, [262](#)
 - GetPhasedSubmissionStartYear, [260](#)
 - GetSecurityClassID, [263](#)
 - GetUserDefined1, [264](#)
 - GetUserDefined2, [265](#)
 - GetUserDefined3, [265](#)
 - IsConsolidateYTD, [265](#)
 - IsPhasedSubmissionEnabled, [266](#)
 - methods, summarized, [63](#), [64](#)
 - object references, assigning, [259](#)
 - overview, [259](#)
 - SupportsAccountAllocations, [266](#)
 - SupportsEmailAlerting, [266](#)
 - SupportsProcessFlow, [267](#)
 - SupportsTargetSetting, [267](#)
 - SupportsTimePeriodAllocations, [267](#)
 - UsesLineItems, [268](#)
- HsvSecurityAccess object
 - AddApplicationAdministrator, [455](#)
 - AddApplicationAdministrator2, [455](#)
 - AddOrRemoveApplicationAdministrators, [455](#)
 - AddOrRemoveApplicationAdministrators2, [455](#)
 - AddOrRemoveRolesFromUser, [456](#)
 - AddOrRemoveRolesFromUser2, [456](#)
 - AddOrRemoveUsersFromRole, [457](#)
 - AddOrRemoveUsersFromRole2, [457](#)
 - AddSecurityClass, [458](#)
 - AddSecurityClassWithAccessCode, [458](#)
 - AddUser, [458](#)
 - AddUser2, [458](#)
 - AddUserEx, [459](#)
 - AddUserToRole, [459](#)
 - AddUserToRole2, [459](#)
 - AllowRulesLoadForEPMAApp, [459](#)
 - DeleteSecurityClass, [459](#)
 - DeleteSecurityClassWithAccessCode, [460](#)
 - EnumApplicationAdministrators, [460](#)
 - EnumApplicationAdministrators2, [460](#)
 - EnumRoles, [460](#)
 - EnumRolesForPrincipal, [461](#)
 - EnumRolesForUser, [461](#)
 - EnumSecurityClasses, [462](#)
 - EnumSecurityClassRightsForPrincipal, [463](#)
 - EnumUserClassAccess, [464](#)
 - EnumUsers, [464](#)
 - EnumUsers2, [464](#)
 - EnumUsers3, [464](#)
 - EnumUsersInGroup, [465](#)
 - EnumUsersInRole, [465](#)
 - EnumUsersInRole2, [465](#)
 - EnumUsersInRole3, [466](#)
 - EnumUsersInSecurityClass, [466](#)
 - EnumUsersInSecurityClass2, [466](#)
 - EnumUsersInSecurityClass3, [467](#)
 - EnumUsersOrGroups, [467](#)
 - EnumUsersWithFilter, [467](#)
 - EnumUsersWithFilter2, [467](#)
 - GenerateSecurityReportForBiPub, [468](#)
 - GetAllSecurityClassRightsForConnectedUser, [468](#)
 - GetApplicationAdministratorAccessForAllUsers, [469](#)
 - GetApplicationAdministratorAccessForAllUsers2, [469](#)
 - GetConnectedUser, [469](#)
 - GetConnectedUser2, [469](#)
 - GetIdentityTypes, [470](#)
 - GetNumRoles, [470](#)
 - GetOwner, [470](#)
 - GetRoleAccessForAllUsers, [470](#)
 - GetRoleAccessForAllUsers2, [471](#)
 - GetRoleID, [471](#)
 - GetRoleLabel, [471](#)
 - GetRulesMode, [472](#)
 - GetSecurityClassAccessForAllUsers, [472](#)
 - GetSecurityClassAccessForAllUsers2, [472](#)
 - GetSecurityClassID, [472](#)
 - GetSecurityClassLabel, [473](#)
 - GetSecurityClassRightsForConnectedUser, [473](#)
 - GetTaskAccessForConnectedUserFromList, [474](#)
 - GetUserAccessForAllRoles, [474](#)
 - GetUserAccessForAllRoles2, [475](#)

- GetUserAccessForAllSecurityClasses, [475](#)
- GetUserAccessForAllSecurityClasses2, [475](#)
- GetUserID, [475](#)
- GetUserIDFromSID, [476](#)
- GetUserInfoFromUniqueID, [476](#)
- GetUserInfoFromUniqueID2, [476](#)
- GetUserName, [477](#)
- GetUserName2, [477](#)
- GetUserNameFromSID, [477](#)
- GetUserSID, [477](#)
- InsertDefaultSecurityClass, [478](#)
- InsertDefaultSecurityClassWithAccessCode, [478](#)
- IsApplicationAdministrator, [478](#)
- IsClassicHFMAApplication, [478](#)
- IsConnectedUserAllowedToPerformTask, [478](#)
- IsConnectedUserInRole, [479](#)
- IsValidWindowsUser, [479](#)
- methods, summarized, [79](#)
- object references, assigning, [454](#)
- overview, [454](#)
- RemoveApplicationAdministrator, [480](#)
- RemoveApplicationAdministrator2, [480](#)
- RemoveUser, [480](#)
- RemoveUser2, [480](#)
- RemoveUserFromRole, [481](#)
- RemoveUserFromRole2, [481](#)
- RenameSecurityClass, [481](#)
- SetManySecurityClassRightsForUser, [481](#)
- SetManySecurityClassRightsForUser2, [482](#)
- SetRolesForUser, [482](#)
- SetRulesMode, [483](#)
- SetSecurityClassLabel, [483](#)
- SetSecurityClassLabelWithAccessCode, [483](#)
- SetSecurityClassRightsForManyUsers, [483](#)
- SetSecurityClassRightsForManyUsers2, [483](#)
- SetSecurityClassRightsForUser, [484](#)
- SetSecurityClassRightsForUser2, [484](#)
- SetUserClassAccess, [484](#)
- TakeOwnership, [485](#)
- HsvSecurityAccess type library
 - access rights constants, [451](#)
 - connected user, [454](#)
 - HsvSecurityAccess, [454](#)
 - IHsvDataSecurity, [485](#)
 - overview, [451](#)
 - role IDs, [452](#)
 - security identifiers (SIDs), [454](#)
 - task IDs, [452](#)
- HsvSecurityLoadACV object
 - Extract, [749](#)
 - extract options, [756](#)
 - ExtractOptions, [748](#)
 - Load, [758](#)
 - load options, [755](#)
 - LoadOptions, [748](#)
 - methods, summarized, [111](#)
 - object references, assigning, [757](#)
 - overview, [757](#)
 - properties, summarized, [111](#)
 - SetSession, [751](#)
- HsvSecurityLoadACV type library
 - HsvSecurityLoadACV, [757](#)
 - IHsvLoadExtractOption interface, [753](#)
 - IHsvLoadExtractOptions interface, [753](#)
 - overview, [752](#)
- HsvSession object
 - Calculate, [157](#)
 - CreateObject, [159](#)
 - Data property, [157](#)
 - GetLicenseExpirationStatus, [159](#)
 - HasSystemChanged, [159](#)
 - HasUserStatusChanged, [160](#)
 - ICM, [158](#)
 - IsBusy, [160](#)
 - IsRunningTasks, [160](#)
 - Journals, [158](#)
 - LockMetadataLoadWithSystemChangeCheck, [160](#)
 - Metadata, [158](#)
 - methods, [159](#)
 - object references, assigning, [157](#)
 - ProcessFlow, [158](#)
 - properties, [157](#)
 - properties and methods, summarized, [54](#)
 - Reports, [158](#)
 - Security, [158](#)
 - SystemInfo, [158](#)
 - UnlockMetadataLoad, [160](#)
- HsvSession type library
 - HsvSession object
 - methods, [159](#)
 - properties, [157](#)
 - overview, [157](#)
- HsvStarSchemaACM object, [699](#)
 - CreateStarSchema, [700](#)

- DeleteStarSchema, 702
- EnumApplicationStarSchemas, 703
- EnumRegisteredDSNs, 703
- GetAsynchronousTaskStatus, 703
- GetExtractLogData, 704
- GetPersistedSettings, 705
- object references, assigning, 700
- QuitAsynchronousTask, 705
- SetDefaultTablePrefix, 705
- SetPersistedSettings, 705
- SetSession, 705
- TestSQLConnection, 706
- HsvStarSchemaACM type library, 699
 - HsvStarSchemaACM, 699
 - IHsvStarSchemaTemplates, 706
- HsvSystemInfo object
 - AddRefToHsxServer, 491
 - AddTaskToAudit, 491
 - AddTaskToRunningTasks, 492
 - AddTaskToRunningTasksAndUpdatePOV, 492
 - CheckAccess, 492
 - ClearAuditTasks, 492
 - ClearAuditTasks2, 493
 - ClearRunningTask, 494
 - DeleteUserParameter, 495
 - DisableNewConnections, 495
 - EnableNewConnections, 496
 - EnumActivityServers, 496
 - EnumActivityUsers, 496
 - EnumAuditTasks, 497
 - EnumAuditTasks2, 499
 - EnumProhibitConnections, 501
 - EnumRunningTasks, 502
 - EnumRunningTasksEx, 505
 - EnumRunningTasksPOV, 507
 - ExtractTaskAudit, 509
 - GetActivityCodeDesc, 510
 - GetActivityUserID, 510
 - GetApplicationDirectory, 510
 - GetApplicationName, 511
 - GetCalcRulesType, 512
 - GetCOMDLLCalcRules, 512
 - GetCurrentActivity, 512
 - GetExtractFileEncoding, 512
 - GetFormattedDateTime, 513
 - GetFormattedResourceString, 513
 - GetKillUsersStatus, 513
 - GetKillUserStatus, 513
 - GetLanguageUserParameters, 514
 - GetLastModifiedDateForArtifact, 514
 - GetModuleName, 514
 - GetNumberFormattingUserParameters, 514
 - GetResourceLanguageUserParameters, 515
 - GetResourceString, 515
 - GetRunningTaskLogFilePathName, 515
 - GetRunningTaskProgress, 516
 - GetRunningTasksCount, 516
 - GetRunningTaskStatus, 516
 - GetServerName, 517
 - GetUserName, 517
 - GetUserParameter, 518
 - GetVBScriptCalcRules, 518
 - GetVBScriptMemberListRules, 519
 - GetWorkingDirectory, 520
 - IsScheduledTaskReadyToRun, 520
 - KeepRunningTaskStillAlive, 520
 - KillUsers, 520
 - methods, summarized, 85
 - object references, assigning, 491
 - OutputSystemInfo, 521
 - overview, 491
 - ReleaseHsxServer, 521
 - SetApplicationDirectory, 521
 - SetCOMDLLCalcRules, 521
 - SetCurrentActivity, 521
 - SetCurrentModule, 521
 - SetCurrentModuleEx, 522
 - SetExtractFileEncoding, 522
 - SetLanguageUserParameters, 522
 - SetNumberFormattingUserParameters, 523
 - SetResourceLanguageForCurrentSession, 523
 - SetResourceLanguageUserParameters, 524
 - SetUserParameter, 524
 - SetVBScriptCalcRules, 525
 - SetVBScriptMemberListRules, 527
 - StopRunningTask, 529
 - UpdateRunningTaskLogFilePathName, 529
 - UpdateRunningTaskPOV, 529
 - UpdateRunningTaskProgress, 529
 - UpdateRunningTaskProgressDetails, 529
 - UpdateRunningTaskStatus, 529
 - WarnUsersForShutDown, 529
- HsvSystemInfo type library
 - HsvSystemInfo object, 491

- overview, [491](#)
- HsvTransactionData object
 - BeginDataEnum, [668](#)
 - BeginQueryEnum, [669](#)
 - EndDataEnum, [669](#)
 - EndQueryEnum, [669](#)
 - GetFixedDimensionMembers, [669](#)
 - GetQueryItem, [669](#)
 - GetTransactionData, [669](#)
 - Initialize, [672](#)
 - object references, assigning, [667](#)
 - overview, [667](#)
 - SetAccessRight, [672](#)
 - SetQueryItem, [672](#)
 - SetTransactionData, [673](#)
- HsvTreeInfo object
 - EnumMembersWithAttribValues, [200](#)
 - EnumSortedIDsOfChildren, [202](#)
- HsvValues object
 - GetCurrencyIDFromValueID, [268](#)
 - GetValueIDFromCurrencyID, [269](#)
 - methods, [268](#)
- HsvValues object, object references, assigning, [188](#)
- HsvViews object, object references, assigning, [188](#)
- HsvYears object
 - GetYearRange, [269](#)
 - object references, assigning, [269](#)
 - overview, [269](#)
- HsxClient object
 - AuthenticateCSSToken, [124](#)
 - AuthenticateSecurityAgentCredentials, [124](#)
 - AuthenticateUserCredentials, [124](#)
 - AuthenticateUserOnCluster, [124](#)
 - AuthenticateUserOnClusterSSO, [124](#)
 - AuthenticateUserOnServer, [124](#)
 - CreateApplicationCAS, [124](#)
 - CreateApplicationCASWithAccessCode, [126](#)
 - CreateObjectOnCluster, [126](#)
 - CreateObjectOnServer, [126](#)
 - DeleteApplication, [126](#)
 - DeleteApplicationWithAccessCode, [127](#)
 - DeleteSystemErrors, [127](#)
 - DetermineWindowsLoggedOnUser, [127](#)
 - DisableNewConnections, [128](#)
 - DoesUserHaveCreateApplicationRights, [128](#)
 - DoesUserHaveSystemAdminRights, [129](#)
 - EnableNewConnections, [129](#)
 - EnumProhibitConnections, [130](#)
 - EnumProvisioningProjects, [131](#)
 - EnumRegisteredClusterNames, [132](#)
 - EnumRegisteredServerNames, [132](#)
 - EnumUserAppPreferences, [132](#)
 - EnumUsersOnSystem, [132](#)
 - EnumUsersOnSystemEx, [134](#)
 - GetApplicationFolder, [137](#)
 - GetClusterInfo, [138](#)
 - GetHFMErrLogRecordSet, [138](#)
 - GetLogonInfo, [138](#)
 - GetLogonInfoSSO, [138](#)
 - GetServer, [139](#)
 - GetServerOnCluster, [139](#)
 - GetSSOTokenUsingWebSecurityAgentCredentials, [139](#)
 - GetWebSecurityAgentSettings, [139](#)
 - IsValidApplication, [139](#)
 - KillUsers, [140](#)
 - methods, summarized, [50](#)
 - object references, assigning, [123](#)
 - OpenApplication, [140](#)
 - overview, [123](#)
 - RegisterApplicationCAS, [141](#)
 - RegisterApplicationCASWithAccessCode, [142](#)
 - RegisterCluster, [142](#)
 - RegisterServer, [142](#)
 - ScriptableEnumRegisteredClusterNames, [142](#)
 - ScriptableEnumRegisteredServerNames, [143](#)
 - ScriptableGetLogonInfo, [143](#)
 - ScriptableOpenApplication, [143](#)
 - SetApplicationFolder, [143](#)
 - SetLogonInfo, [143](#)
 - UnregisterAllClusters, [144](#)
 - UnregisterCluster, [144](#)
 - UnregisterServer, [145](#)
 - UpdateUserAppPreferences, [145](#)
 - WarnUsersForShutDown, [145](#)
- HsxClient type library
 - duplicate references in References dialog box, [920](#)
 - HsxClient object, [123](#)
 - overview, [123](#)
- HsxClientUI object
 - DeleteApplication, [148](#)
 - GetServer, [148](#)
 - Initialize, [148](#)
 - Logon, [148](#)

- methods, summarized, [52](#)
 - object references, assigning, [147](#)
 - OpenApplication, [149](#)
 - overview, [147](#)
 - RegisterServer, [150](#)
 - HsxClientUI type library
 - duplicate references in References dialog box, [920](#)
 - HsxClientUI object, [147](#)
 - overview, [147](#)
 - HsxServer object
 - CSSEnabled, [156](#)
 - DeleteSystemErrors, [152](#)
 - EnumDataSources, [152](#)
 - EnumRegisteredDSNs, [152](#)
 - GetClustersAndServers, [153](#)
 - GetDataSource, [154](#)
 - GetHFMErrorsLogRecordSet, [154](#)
 - GetSystemDataLinkFile, [154](#)
 - GetSystemFolder, [154](#)
 - GetXMLErrorFromDatabase, [155](#)
 - GetXMLErrorsListFromDatabase, [155](#)
 - methods, summarized, [53](#)
 - object references, assigning, [151](#)
 - overview, [151](#)
 - ScriptableEnumDataSources, [156](#)
 - ScriptableEnumRegisteredDSNs, [156](#)
 - HsxServer type library
 - duplicate references in References dialog box, [920](#)
 - HsxServer object, [151](#)
 - overview, [151](#)
- I**
- ICM, [158](#)
 - ICPs, [187](#)
 - identity types, [453](#)
 - returning for users, [470](#)
 - IDs, member, [161](#)
 - IHsvAdminICM interface, [732](#)
 - CloseICPeriod, [732](#)
 - GetLockStatusICEntities, [733](#)
 - GetLockStatusICEntity, [733](#)
 - GetSettingsICPeriod, [734](#)
 - GetSettingsICPeriods, [735](#)
 - LoadICTransactions, [737](#)
 - LockICEntity, [739](#)
 - object references, assigning, [732](#)
 - OpenICPeriod, [739](#)
 - SavePeriodsSettings, [740](#)
 - UnlockICEntity, [740](#)
 - UpdatePeriodSettings, [741](#)
 - IHsvDataSecurity interface
 - GetCellLevelAccessRights, [485](#)
 - GetProcessUnitAccessRights, [486](#)
 - GetProcessUnitAccessRightsAndState, [487](#)
 - GetProcessUnitAccessRightsAndStateEx, [487](#)
 - GetProcessUnitAccessRightsEx, [489](#)
 - methods, summarized, [84](#)
 - object references, assigning, [485](#)
 - overview, [485](#)
 - RefreshAccessRightsCache, [490](#)
 - IHsvDataValidation interface, [275](#)
 - IHsvJournalsEx interface
 - AddJournalGroup, [399](#)
 - ApproveJournals, [399](#)
 - DeleteJournals, [400](#)
 - DeleteTemplates, [401](#)
 - EnumJournalGroups, [402](#)
 - EnumJournalGroupsForScenarioYear, [402](#)
 - EnumJournalIDsForExtractFilter, [403](#)
 - GenerateRecurring, [404](#)
 - GetEntityJournals, [405](#)
 - GetJournal, [406](#)
 - GetJournalDisplayData, [411](#)
 - GetJournalLabelsForIDs, [412](#)
 - GetJournalQueryDefinitionIDs, [413](#)
 - GetTemplate, [415](#)
 - GetTemplateDisplayData, [417](#)
 - GetTemplateLabelsForIDs, [418](#)
 - GetTemplateQueryDefinitionIDs, [418](#)
 - GetTextJournal, [420](#)
 - GetTextTemplate, [422](#)
 - GetVariance, [425](#)
 - methods, summarized, [77](#)
 - object references, assigning, [399](#)
 - overview, [399](#)
 - PostJournals, [426](#)
 - RejectJournals, [427](#)
 - RemoveAllJournalGroups, [428](#)
 - RemoveJournalGroup, [428](#)
 - SaveJournal, [428](#)
 - SaveTemplate, [431](#)
 - SaveTextJournal, [434](#)
 - SaveTextTemplate, [437](#)
 - SubmitJournals, [439](#)

- UnpostJournals, [440](#)
- UnsubmitJournals, [441](#)
- ValidateLineItems, [442](#)
- ValidateValue, [443](#)
- IHsvJournalsReport interface
 - GetReportData, [447](#)
 - GetReportData2, [449](#)
 - methods, summarized, [79](#)
 - object references, assigning, [444](#)
 - overview, [444](#)
- IHsvLoadExtractOption interface
 - CurrentValue, [745](#)
 - DefaultValue, [746](#)
 - HsvcDataLoad type library, [775](#)
 - HsvJournalLoadACV type library, [785](#)
 - HsvMetadataLoadACV type library, [759](#)
 - HsvSecurityLoadACV type library, [753](#)
 - MaxValue, [746](#)
 - MinValue, [746](#)
 - Name, [747](#)
 - object references, assigning, [745](#)
 - OptionID, [747](#)
 - overview, [745](#)
 - properties, summarized, [110](#)
 - ValidationList, [747](#)
- IHsvLoadExtractOptions interface
 - Count, [744](#)
 - HsvcDataLoad type library, [775](#)
 - HsvJournalLoadACV type library, [785](#)
 - HsvMetadataLoadACV type library, [759](#)
 - HsvSecurityLoadACV type library, [753](#)
 - Item, [744](#)
 - object references, assigning, [744](#)
 - overview, [744](#)
 - properties, summarized, [109](#)
- IHsvMDDataBufferLite interface, [667](#)
 - SetMinCubesInCache, [667](#)
- IHsvStarSchemaTemplates interface, [706](#)
 - DeleteTemplate, [706](#)
 - EnumTemplates, [706](#)
 - GetTemplate, [707](#)
 - object references, assigning, [706](#)
 - SetTemplate, [707](#)
- IHsvTreeInfo interface
 - EnumAllMemberIDs, [188](#)
 - EnumAllMemberLabels, [189](#)
 - EnumAllParentAndChildIDs, [189](#)
 - EnumAllParentAndChildLabels, [190](#)
 - EnumAncestors, [191](#)
 - EnumBaseMemberIDs, [192](#)
 - EnumDefaultAncestors, [193](#)
 - EnumDefaultAncestorsLabels, [194](#)
 - EnumDescendants, [194](#)
 - EnumIDsOfChildren, [195](#)
 - EnumIDsOfIChildren, [196](#)
 - EnumMemberLists, [197](#)
 - EnumMembers, [197](#)
 - EnumMembers2, [199](#)
 - EnumParents, [201](#)
 - EnumSortedIDsOfChildren, [201](#)
 - EnumSortedMembers, [203](#)
 - Find, [204](#)
 - FindByDesc, [205](#)
 - FindMatchingMembersFromHierarchy, [207](#)
 - FindMatchingMembersFromHierarchyByDesc, [208](#)
 - FindMatchingMembersFromHierarchyWildCard, [209](#)
 - GetAllPathsToMember, [210](#)
 - GetAttributeValue, [211](#)
 - GetDefaultHierarchyPosition, [212](#)
 - GetDefaultItemIDHierarchy, [212](#)
 - GetDefaultParent, [214](#)
 - GetDefaultParentLabel, [214](#)
 - GetDescription, [215](#)
 - GetDisplayInfo, [216](#)
 - GetDisplayInfoForSeveralItems, [217](#)
 - GetItemGeneration, [218](#)
 - GetItemID, [219](#)
 - GetItemIDQL, [220](#)
 - GetItemLevel, [220](#)
 - GetLabel, [221](#)
 - GetMemberListID, [221](#)
 - GetMemberListName, [222](#)
 - GetNumBaseMembers, [222](#)
 - GetNumChildren, [223](#)
 - GetNumDescendants, [223](#)
 - GetNumMembers, [224](#)
 - GetNumParents, [225](#)
 - GetQualifiedLabel, [225](#)
 - GetTreeCapabilities, [225](#)
 - GetTreeName, [226](#)
 - GetTreeTimeStamp, [226](#)
 - HasChildren, [227](#)

- IsMemberABaseOf, 228
- IsMemberAChildOf, 228
- IsMemberADescendantOf, 229
- methods, summarized, 65
- object references, assigning, 188
- overview, 188
- SortMembersBasedOnList, 230
- TranslateAttributeValueForDisplay, 230
- IHsvTreeInfo object
 - GetDefaultMemberID, 214
- IHsxClient object
 - EnumUsersOnSystemEx2, 136
- IHsxServer object
 - GetFileTransfer, 154
- indexes, obtaining for HsvMDDDataBuffer and HsvMDDDataBufferLite objects, 610
- Initialize
 - HsvICTransactionsData object, 682
 - HsvPOVSelection object, 801
 - HsvResourceManager object, 821
 - HsvTransactionData object, 672
 - HsxClientUI object, 148
- InitializeEx, 802
- InitializeHTTP, 802
- InitializeSequenceIdMap, 683
- InitializeWithConnection, 802
- InsertDataAtBeginning, 630
- InsertDataAtEnd
 - HsvMDDDataBuffer object, 631
 - HsvMDDDataBufferLite object, 660
- InsertDefaultSecurityClass, 478
- InsertDefaultSecurityClassWithAccessCode, 478
- InsertDescriptionAtBeginning, 632
- InsertDescriptionAtEnd
 - HsvMDDDataBuffer object, 633
 - HsvMDDDataBufferLite object, 661
- InsertLineItemDetails, 352
- InsertLineItemsAtBeginning, 633
- InsertLineItemsAtEnd
 - HsvMDDDataBuffer object, 634
 - HsvMDDDataBufferLite object, 661
- InsertRecordIntoPMBuffer
 - HsvMDDDataBuffer object, 635
 - HsvMDDDataBufferLite object, 662
- InsertRecordIntoPMErrorBuffer
 - HsvMDDDataBuffer object, 635
 - HsvMDDDataBufferLite object, 662
- intercompany matching
 - by account, 723
 - by ID, 724
- Intercompany Partner dimension
 - children of member, sorting, 201
 - members, sorting, 203
 - object for, 254
- intercompany transactions
 - managing, 732
 - processing, 711
 - sequence IDs, 680
 - testing if accounts used in, 243
 - testing if entities support, 253
- invalid records, deleting, 283
- IsApplicationAdministrator, 478
- IsBusy, 160
- IsCalculated
 - HsvAccounts object, 240
 - HsvCustom object, 246
- IsCalculated attribute, returning, 240
- IsChild, 252
- IsClassicHFMAApplication, 478
- IsConnectedUserAllowedToPerformTask, 478
- IsConnectedUserInRole, 479
- IsConsolidated, 240
- IsConsolidated attribute, returning, 240
- IsConsolidateYTD, 265
- IsCustom1AggregationEnabled, 241
- IsCustom2AggregationEnabled, 241
- IsCustom3AggregationEnabled, 242
- IsCustom4AggregationEnabled, 242
- IsCustomMemberValidForAccount, 179
- IsDescendant, 252
- IsEntityAnEPUOwner, 385
- IsEntityInPartnerAsEntityList, 684
- IsICP
 - HsvAccounts object, 243
 - HsvEntities object, 253
- IsICPRestricted, 243
- IsICTransactionValid, 684
- IsMemberABaseOf, 228
- IsMemberAChildOf, 228
- IsMemberADescendantOf, 229
- IsOneSideOfTransactionGroupWriteable, 722
- IsOrgByPeriodApplication, 180
- IsOrgByPeriodFilteringOn
 - HsvEntities object, 253

- IsPeriodOpen, [398](#)
 - IsPhasedSubmissionEnabled, [266](#)
 - IsRunningTasks, [160](#)
 - IsScheduledTaskReadyToRun, [520](#)
 - IsSecurityAsPartnerEnabled, [254](#)
 - IsSwitchSignEnabledForFlow, [247](#)
 - IsSwitchTypeEnabledForFlow, [247](#)
 - IsValidApplication, [139](#)
 - IsValidCellText, [353](#)
 - IsValidWindowsUser, [479](#)
 - Item, [744](#)
 - items, subcubes, [687](#)
- J**
- journals
 - extract options
 - filtered extracts, [791](#)
 - unfiltered extracts, [789](#)
 - extracting
 - example, filtered extract, [788](#)
 - example, unfiltered extract, [787](#)
 - overview, [786](#)
 - load options, [789](#)
 - loading
 - example, [785](#)
 - overview, [785](#)
 - objects and interfaces for, [75](#)
 - Journals property, [158](#)
- K**
- KeepRunningTaskStillAlive, [520](#)
 - KillUsers
 - HsvSystemInfo object, [520](#)
 - HsxClient object, [140](#)
- L**
- language
 - getting the ID of a language for description, [164](#)
 - returning an application's languages for description, [164](#)
 - returning the default language for descriptions, [514](#)
 - returning the default language for resource strings, [515](#)
 - setting the default language for descriptions, [522](#)
 - setting the default language for resource strings, [524](#)
 - setting the default language for resource strings for a session, [523](#)
 - language code, obtaining, [818](#)
 - language IDs, obtaining for supported languages, [813](#)
 - level, member, getting, [220](#)
 - line items, examples for getting and setting, [121](#)
 - Load
 - HsvcDataLoad, HsvJournalLoadACV, and HsvMetadataLoadACV objects, [750](#)
 - HsvData object, [353](#)
 - HsvMetadata object, [180](#)
 - HsvSecurityLoadACV object, [758](#)
 - Load2, [783](#)
 - LoadCalcManagerRules, [386](#)
 - LoadCalcRules, [770](#)
 - LoadCalcRules2, [771](#)
 - LoadDrillableRegions, [355](#)
 - LoadICTransactions, [737](#)
 - loading data, [776](#)
 - loading journals, [785](#)
 - loading member lists, [772](#)
 - loading metadata
 - file format constants, [767](#)
 - overview, [759](#)
 - loading rules, [770](#)
 - loading security, [753](#)
 - LoadMemberListRules, [772](#)
 - LoadOptions, [748](#)
 - LoadWithAccessCode
 - HsvMetadata object, [185](#)
 - local storage folder
 - getting, [137](#)
 - setting, [143](#)
 - LockICEntity, [739](#)
 - LockMetadataLoadWithSystemChangeCheck, [160](#)
 - logging on
 - example, [119](#)
 - HsxClient object, [143](#), [144](#)
 - HsxClientUI object, [148](#)
 - Logon, [148](#)
 - Logon dialog box, displaying, [148](#)
 - LST file
 - getting contents of, [519](#)
 - loading and validating, [772](#)

M

MatchAutoAccounts, [723](#)
 MatchAutoIDs, [724](#)
 matching, intercompany
 by account, [723](#)
 by ID, [724](#)
 MaximumReviewLevel attribute, [261](#)
 MaxValue, [746](#)
 member IDs, [161](#)
 member lists
 dynamic, getting members of, [199](#)
 extracting, [769](#)
 getting members of, [197](#)
 loading, [772](#)
 MEMBER_LIST_ALL_HIERARCHY, [838](#)
 MEMBERNOTUSED, [835](#)
 Metadata, [158](#)
 metadata
 attribute, getting value of, [211](#)
 examples, [120](#)
 extract options
 file format constants, [767](#)
 overview, [765](#)
 extracting
 example, [761](#)
 overview, [760](#)
 load options
 file format constants, [767](#)
 overview, [761](#)
 loading
 example, [760](#)
 overview, [759](#)
 objects and interfaces for, [55](#)
 testing to see if reloaded, [159](#)
 metadata status
 constants for, [843](#)
 overview, [275](#)
 methods, [159](#)
 MinValue, [746](#)
 multi-select mode, [806](#)
 MultiCheckBoxChanged event, [809](#)

N

Name, [747](#)
 node subcubes
 accessing with HsvNodeCube object, [693](#)
 described, [43](#)

NoTransMatchedOrPosted, [725](#)
 Number property, Err object (Visual Basic), [45](#)
 NumDecimalPlaces attribute, returning, [234](#)

O

Open Application dialog box, displaying, [149](#)
 OpenApplication
 HsxClient object, [140](#)
 HsxClientUI object, [149](#)
 OpenICPeriod, [739](#)
 OpenICPeriod2, [740](#)
 opening application, example, [119](#)
 OpenPeriod, [398](#)
 OptionID, [747](#)
 Organization by Period, [180](#)
 organization by period
 testing if entities support, [253](#)
 OrgByPeriodApplication application setting, [180](#)
 OutputSystemInfo, [521](#)

P

parent
 getting ID of a member's default, [214](#)
 getting label of a member's default, [214](#)
 password, setting, [143](#), [144](#)
 Periods property, [187](#)
 periods, object for, [256](#)
 phased submissions
 enabled, testing whether, [177](#)
 phased submissions, considerations for, [531](#)
 PhasedSubmissionApprove, [544](#)
 PhasedSubmissionApprove2, [545](#)
 PhasedSubmissionApproveEx, [547](#)
 PhasedSubmissionGetHistory2, [548](#)
 PhasedSubmissionGetHistory2UsingPhaseID, [549](#)
 PhasedSubmissionProcessManagementChangeState
 ForMultipleEntities2, [551](#)
 PhasedSubmissionPromote, [552](#)
 PhasedSubmissionPromote2, [554](#)
 PhasedSubmissionPublish, [555](#)
 PhasedSubmissionPublish2, [556](#)
 PhasedSubmissionPublishEx, [558](#)
 PhasedSubmissionReject, [559](#)
 PhasedSubmissionReject2, [560](#)
 PhasedSubmissionSignOff, [561](#)
 PhasedSubmissionSignOff2, [562](#)

PhasedSubmissionStart, 564
 PhasedSubmissionStart2, 565
 PhasedSubmissionStartEx, 567
 PhasedSubmissionSubmit, 569
 PhasedSubmissionSubmit2, 570
 PlugAcct attribute, returning, 235
 Point of View
 control for user interface, 795
 IHsvTreeInfo, methods for specifying members, 188
 Point of View dialog box, 795
 PostJournals, 426
 preferences, user
 decimal character
 getting, 514
 setting, 523
 language
 getting, 514
 setting, 522
 thousands separator
 getting, 514
 setting, 523
 prerequisites, 39
 process management
 constants for actions, 853
 constants for review levels, 853
 submission phases and process units, 531
 Process Management, type library for, 531
 ProcessAllICTrans, 725
 ProcessFlow, 158
 ProcessICTransactions, 726
 ProcessManagementChangeStateForMultipleEntities, 571
 ProcessManagementChangeStateForMultipleEntities2, 572
 ProcessManagementChangeStateForMultipleEntitiesEx, 574
 projects, provisioning, returning names of, 131
 Promote, 575
 Promote2, 576
 provisioning projects, returning names of, 131
 Publish, 578
 Publish2, 579
 PublishEx, 581

Q

QuitAsynchronousTask, 705

R

records, invalid, deleting, 283
 referencing type libraries, 47
 RefreshAccessRightsCache, 490
 Register Cluster dialog box, displaying, 150
 RegisterApplicationCAS, 141
 RegisterApplicationCASWithAccessCode, 142
 RegisterCluster, 142
 registering application servers
 HsxClient object, 142
 HsxClientUI object, 150
 RegisterServer
 HsxClient object, 142
 HsxClientUI object, 150
 Reject, 582
 RejectJournals, 427
 ReleaseHsxServer, 521
 RemoveAll
 HsvMDDDataBuffer object, 635
 HsvMDDDataBufferLite object, 662
 RemoveAllJournalGroups, 428
 RemoveApplicationAdministrator, 480
 RemoveApplicationAdministrator2, 480
 RemoveJournalGroup, 428
 RemoveQueryOrder, 684
 RemoveUser, 480
 RemoveUser2, 480
 RemoveUserFromRole, 481
 RemoveUserFromRole2, 481
 RenameSecurityClass, 481
 reports
 journals, 447
 reports, 604
 Reports property, 158
 Resource Manager, 812
 resource strings
 returning the default language for, 515
 setting the default language, 524
 setting the default language for a session, 523
 review levels, constants for, 853
 RLE file
 getting contents of
 GetVBScriptCalcRules, 518
 GetVBScriptRules, 385
 loading and validating
 LoadCalcRules, 770
 SetVBScriptCalcRules, 525

- SetVBScriptRules, [386](#)
 - role IDs, [452](#)
 - rules
 - extracting, [769](#)
 - loading, [770](#)
- S**
- SaveColumnFilter, [728](#)
 - SaveDocument, [605](#)
 - SaveDocumentEx, [606](#)
 - SaveICReasonCode, [728](#)
 - SaveICTransaction, [728](#)
 - SaveJournal, [428](#)
 - SavePeriodsSettings, [740](#)
 - SaveRowFilter, [730](#)
 - SaveTemplate, [431](#)
 - SaveTextJournal, [434](#)
 - SaveTextTemplate, [437](#)
 - Scenarios, [187](#)
 - ScriptableEnumDataSources, [156](#)
 - ScriptableEnumRegisteredClusterNames, [142](#)
 - ScriptableEnumRegisteredDSNs, [156](#)
 - ScriptableEnumRegisteredServerNames, [143](#)
 - ScriptableGetLogonInfo, [143](#)
 - ScriptableLoadCalcRules, [773](#)
 - ScriptableLoadCalcRules2, [773](#)
 - ScriptableLoadMemberListRules, [773](#)
 - ScriptableOpenApplication, [143](#)
 - SDK, contains examples, [119](#)
 - security
 - connected user, [454](#)
 - extract options, [756](#)
 - extracting
 - example, [755](#)
 - overview, [754](#)
 - load options, [755](#)
 - loading
 - example, [754](#)
 - overview, [753](#)
 - role IDs, [452](#)
 - security identifiers (SIDs), [454](#)
 - task IDs, [452](#)
 - type library for, [451](#)
 - security class, Default, constant for, [883](#)
 - security identifier
 - returning username for, [477](#)
 - user, getting for, [477](#)
 - security identifiers, [454](#)
 - Security property, [158](#)
 - SecurityAsPartner attribute, [249](#)
 - SecurityClass attribute, returning, [250](#)
 - Select Member List dialog box, [795](#)
 - SelectDimension, [802](#)
 - SelectionChanged event, [809](#)
 - sequence IDs, intercompanytransactions, [680](#)
 - Server-Side Application components, instantiating, [159](#)
 - servers, application
 - applications on, getting, [152](#)
 - data link file, getting, [154](#)
 - names of, getting, [132](#)
 - registering
 - HsxClient object, [142](#)
 - HsxClientUI object, [150](#)
 - system file, getting, [154](#)
 - unregistering
 - HsxClient object, [145](#)
 - HsxClientUI object, [150](#)
 - SetAccessRight, [672](#)
 - SetAccessRights, [684](#)
 - SetApplicationDirectory, [521](#)
 - SetApplicationFolder, [143](#)
 - SetCalcStatusLocked, [357](#)
 - SetCalcStatusLockedForMultipleProcessUnits, [358](#)
 - SetCalcStatusUnlocked, [358](#)
 - SetCalcStatusUnlockedForMultipleProcessUnits, [359](#)
 - SetCell, [359](#)
 - SetCellDescriptions, [361](#)
 - SetCellLineItems, [362](#)
 - SetCells, [363](#)
 - SetCells2, [365](#)
 - SetCellsLineItems, [367](#)
 - SetCheckedItems, [802](#)
 - SetCheckLineItemDetailsForCaseInsensitiveDuplicat
 - es
 - HsvMDDataBuffer object, [635](#)
 - HsvMDDataBufferLite object, [663](#)
 - SetCOMDLLCalcRules, [521](#)
 - SetCOMDLLRules, [386](#)
 - SetCurrentActivity, [521](#)
 - SetCurrentModule, [521](#)
 - SetCurrentModuleEx, [522](#)
 - SetData
 - HsvMDDataBuffer object, [636](#)

- HsvMDDataBufferLite object, 663
- SetDefaultTablePrefix, 705
- SetDescription
 - HsvMDDataBuffer object, 637
 - HsvMDDataBufferLite object, 664
- SetDimUnk, 804
- SetEntityPartnerOption, 684
- SetErrorStatus, 685
- SetExtractFileEncoding, 522
- SetFileForLoad
 - HsvcDataLoad object, 784
 - HsvData object, 368
- SetFilterOptions, 685
- SetGrowByAmount
 - HsvMDDataBuffer object, 638
 - HsvMDDataBufferLite object, 665
- SetICTransactionData, 685
- SetLanguageUserParameters, 522
- SetLineItems
 - HsvData object, 368
 - HsvMDDataBuffer object, 638
 - HsvMDDataBufferLite object, 665
- SetListInfo, 804
- SetLogonInfo, 143
- SetManySecurityClassRightsForUser, 481
- SetManySecurityClassRightsForUser2, 482
- SetMember, 805
- SetMinCubesInCache, 667
- SetNumberFormattingUserParameters, 523
- SetOBPInfo, 805
- SetPagingOption, 685
- SetPartnerAsEntityList, 685
- SetPartnerQueryDimensionMemberIDs, 685
- SetPersistedSettings, 705
- SetPhaseSubmissionGridForGivenScenarioPeriod, 368
- SetQueryDimensionMemberIDs, 685
- SetQueryItem, 672
- SetReasonCodeToICTransactions, 730
- SetReport, 607
- SetResourceAndLanguage, 806
- SetResourceLanguageForCurrentSession, 523
- SetResourceLanguageUserParameters, 524
- SetRolesForUser, 482
- SetRulesMode, 483
- SetSecurityClassLabel, 483
- SetSecurityClassLabelWithAccessCode, 483
- SetSecurityClassRightsForManyUsers, 483
- SetSecurityClassRightsForManyUsers2, 483
- SetSecurityClassRightsForUser, 484
- SetSecurityClassRightsForUser2, 484
- SetSession, 705
 - HsvcDataLoad, HsvJournalLoadACV, HsvMetadataLoadACV, and HsvSecurityLoadACV objects, 751
 - HsvRulesLoadACV object, 773
- SetSessionAndResource
 - HsvRulesLoadACV object, 774
- SetSortedNature
 - HsvMDDataBuffer object, 639
 - HsvMDDataBufferLite object, 666
- SetTemplate, 707
- SetTextCell, 369
- SetTextCellLineItems, 370
- SetTextCells, 371
- SetTextCellsLineItems, 373
- SetTextLineItems, 374
- SetTotalTransactions, 685
- SetTransactionData, 673
- SetUserClassAccess, 484
- SetUserParameter, 524
- SetVBScriptCalcRules, 525
- SetVBScriptMemberListRules, 527
- SetVBScriptRules, 386
- SetVBScriptRules2, 388
- Shared Services
 - returning provisioning project names, 131
- ShowCheckBoxes, 806
- SID, returning username for, 477
- SIDs, 454
- SignOff, 584
- SignOff2, 585
- single-select mode, 806
- Software Development Toolkit, contains examples, 119
- Sort
 - HsvMDDataBuffer object, 639
 - HsvMDDataBufferLite object, 666
- SortByCell, 685
- SortMembersBasedOnList, 230
- SS_PUSH_OPTIONS, Enum, 700
- star schema, 699
 - creating or updating, 700
 - deleting, 702

Start, [586](#)
 Start2, [588](#)
 StartEx, [589](#)
 StartLoad
 HsvcDataLoad object, [784](#)
 HsvData object, [374](#)
 statuses, cells
 getting with GetStatus, [335](#)
 overview, [275](#)
 statutory consolidations, returning transaction data for, [667](#)
 StopConsolidation, [390](#)
 StopRunningTask, [529](#)
 storage folder, local
 getting, [137](#)
 setting, [143](#)
 subcube
 accessing with HsvDataCubes type library, [687](#)
 calculation status, getting with GetCalcStatus, [313](#)
 data synchronization and, [44](#)
 items, [687](#)
 overview, [43](#)
 Submission Group property
 accounts, returning for, [236](#)
 Custom dimension members, returning for, [245](#)
 Intercompany Partner members, returning for, [255](#)
 submission phases, considerations for, [531](#)
 Submit, [590](#)
 Submit2, [591](#)
 SubmitJournals, [439](#)
 SupportsAccountAllocations, [266](#)
 SupportsEmailAlerting, [266](#)
 SupportsProcessFlow, [267](#)
 SupportsProcessManagement attribute, testing for, [267](#)
 SupportsTargetSetting, [267](#)
 SupportsTimePeriodAllocations, [267](#)
 SwitchSignForFlow attribute, returning, [247](#)
 SwitchTypeForFlow attribute, returning, [247](#)
 system changes, testing for with HasSystemChanged, [159](#)
 system data link file, getting path of, [154](#)
 system file, getting path of, [154](#)
 system folder, [510](#)
 SystemInfo, [158](#)

T
 TakeOwnership, [485](#)
 task audit
 adding task to audit history, [491](#)
 delete audit history, [492](#)
 getting audit history, [497](#)
 task IDs, [452](#)
 TestSQLConnection, [706](#)
 thousands separator
 getting, [514](#)
 setting, [523](#)
 token, external authentication, getting, [138](#)
 transaction data, statutory consolidations, [667](#)
 transaction status
 constants for, [846](#)
 overview, [275](#)
 Translate, [390](#)
 Translate2, [391](#)
 TranslateApplicationAttributeForDisplay, [185](#)
 TranslateAttributeValueForDisplay, [230](#)
 triplets, Value dimension, [43](#)
 type libraries
 HsvCalculate, [375](#)
 HsvcDataLoad, [775](#)
 HsvData, [275](#)
 HsvDataCubes, [687](#)
 HsvICM, [711](#)
 HsvJournalLoadACV, [784](#)
 HsvJournals, [393](#)
 HsvMDArrays, [609](#)
 HsvMetadata, [161](#)
 HsvMetadataLoadACV, [758](#)
 HsvPOVSelection, [795](#)
 HsvProcessFlow, [531](#)
 HsvReports, [595](#)
 HsvResourceManager, [811](#)
 HsvRulesLoadACV, [768](#)
 HsvSecurityAccess, [451](#)
 HsvSecurityLoadACV, [752](#)
 HsvSession, [157](#)
 HsvStarSchemaACM, [699](#)
 HsvSystemInfo, [491](#)
 HsxClient, [123](#)
 HsxClientUI, [147](#)
 HsxServer, [151](#)
 referencing in projects, [47](#)
 type mappings, [40](#)

U

UnInitialize, 685
 UnlockICEntity, 740
 UnlockMetadataLoad, 160
 UnMatchICTransactions, 731
 UnpostJournals, 440
 UnregisterAllClusters, 144
 UnregisterCluster, 144
 UnregisterServer, 145
 UnsubmitJournals, 441
 update mode, 276
 UpdateDataUsingMDDDataBuffer, 374
 UpdatePeriodSettings, 741
 UpdatePeriodSettings2, 741
 UpdateRunningTaskLogFilePathName, 529
 UpdateRunningTaskPOV, 529
 UpdateRunningTaskProgress, 529
 UpdateRunningTaskProgressDetails, 529
 UpdateRunningTaskStatus, 529
 UpdateUserAppPreferences, 145
 user preferences

- decimal character
 - getting, 514
 - setting, 523
- examples, number formatting, getting and setting, 121
- language
 - getting, 514
 - setting, 522
- thousands separator
 - getting, 514
 - setting, 523

 user, connected, 454
 UserDefined1 attribute, returning

- accounts, 239
- entities, 251
- scenarios, 264

 UserDefined2 attribute, returning

- accounts, 239
- entities, 251
- scenarios, 265

 UserDefined3 attribute, returning

- accounts, 239
- entities, 251
- scenarios, 265

 username

- getting, 138

setting, 143, 144
 single sign on

- getting, 138

 users, enumerating, 467
 UsesLineItems

- HsvAccounts object, 243
- HsvScenarios object, 268

V

ValidateLineItems, 442
 ValidateValue, 443
 Validation Account n properties, returning, 169
 Validation Account property, returning, 178
 ValidationList, 747
 Value dimension triplets, 43
 Values, 188
 Views, 188

W

WarnUsersForShutDown

- HsvSystemInfo object, 529
- HsxClient object, 145

X

XBRLTags attribute, returning, 240

Y

Years, 188

Z

ZeroViewForAdj attribute, getting, 262
 ZeroViewForNonadj attribute, getting, 262

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z