# Documentation Accessibility

47

## Chapter 1. Overview

49

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

## Chapter 2. About the Type Libraries and Objects

57

- Type Library References
- Objects, Properties, and Methods
  - HsxClient Type Library Overview
  - HsxClientUI Type Library Overview
  - HsxServer Type Library Overview
  - HsvSession Type Library Overview
  - HsvMetadata Type Library Overview
  - HsvData Type Library Overview
  - HsvCalculate Type Library Overview
  - HsvJournals Type Library Overview
  - HsvSecurityAccess Type Library Overview
  - HsvSystemInfo Type Library Overview
  - HsvProcessFlow Type Library Overview
  - HsvReports Type Library Overview
  - HsvMDArrays Type Library Overview
  - HsvDataCubes Type Library Overview
  - HsvDQI Type Library Overview
  - HsvStarSchemaACM Type Library Overview
  - HsvICM Type Library
  - Type Libraries for Loading and Extracting Information
  - HsvPOVSelection Type Library
  - HsvResourceManager Type Library Overview
  - HfmSliceCOM Type Library
<table>
<thead>
<tr>
<th>Method</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetNumMembers</td>
<td>153</td>
</tr>
<tr>
<td>GetPovAtIndex</td>
<td>153</td>
</tr>
<tr>
<td>InitializeFromPovString</td>
<td>154</td>
</tr>
<tr>
<td>SetFixedMember</td>
<td>154</td>
</tr>
<tr>
<td>SetMemberArrayForDim</td>
<td>155</td>
</tr>
<tr>
<td>SetMemberAt</td>
<td>155</td>
</tr>
</tbody>
</table>

**Chapter 5. HsxClient Type Library**

HsxClient Object Methods

- AuthenticateCSSToken                       | 157  |
- AuthenticateSecurityAgentCredentials       | 158  |
- AuthenticateUserCredentials                | 158  |
- AuthenticateUserOnClient                   | 158  |
- AuthenticateUserOnCluster                  | 158  |
- AuthenticateUserOnClusterSSO               | 159  |
- AuthenticateUserOnServer                   | 159  |
- CreateApplication                          | 159  |
- CreateApplicationCAS                       | 159  |
- CreateApplicationCASWithAccessCode         | 161  |
- CreateObjectOnCluster                      | 161  |
- CreateObjectOnServer                       | 161  |
- DeleteApplication                          | 161  |
- DeleteApplicationWithAccessCode            | 162  |
- DeleteSystemErrors                         | 162  |
- DetermineWindowsLoggedOnUser               | 162  |
- DisableNewConnections                      | 162  |
- DoesUserHaveCreateApplicationRights        | 163  |
- DoesUserHaveSystemAdminRights              | 163  |
- EnableNewConnections                       | 164  |
- EnumProhibitConnections                    | 165  |
- EnumProvisioningProjects                   | 166  |
- EnumRegisteredClusterNames                 | 166  |
- EnumRegisteredClustersOrServers            | 167  |
- EnumRegisteredServerNames                  | 167  |
- EnumUserAppPreferences                      | 167  |
- EnumsUsersOnSystem                         | 167  |
- EnumsUsersOnSystemEx                       | 169  |
- EnumsUsersOnSystemEx2                      | 170  |
- GetApplicationFolder                       | 172  |
Chapter 7. HsxServer Type Library

HsxServer Object Methods
- DeleteSystemErrors ................................................ 186
- DeleteSystemErrorsEx .............................................. 186
- EnumDataSources ................................................ 186
- EnumRegisteredDSNs .............................................. 187
- GetClustersAndServers ............................................. 187
- GetDataSource ................................................... 188
- GetFileTransfer ................................................... 188
- GetHFMErrorLogRecordSet ......................................... 188
- GetSystemDataLinkFile ............................................. 188
- GetSystemFolder ................................................. 188
- GetXMLErrorFromDatabase ......................................... 189
- GetXMLErrorsListFromDatabase ...................................... 189
- ScriptableEnumDataSources ......................................... 190
- ScriptableEnumRegisteredDSNs ...................................... 190

HsxServer Object Properties
- CSSEnabled ..................................................... 190

Chapter 8. HsvSession Type Library

HsvSession Object Properties
- Calculate ....................................................... 191
- Data ........................................................... 191
- ICM ........................................................... 192
- Journals ........................................................ 192
- Metadata ....................................................... 192
- ProcessFlow ..................................................... 192
- Reports ........................................................ 192
- Security ........................................................ 192
- SystemInfo ...................................................... 192

HsvSession Object Methods
- CreateObject .................................................... 193
- GetECID ....................................................... 193
- get_HsvDQI ..................................................... 194
- GetLicenseExpirationStatus ........................................ 194
- HasSystemChanged ............................................... 194
- HasUserStatusChanged ............................................. 194
- IsBusy ......................................................... 194
IsRunningTasks .................................................. 195
LockMetadataLoadWithSystemChangeCheck ............................. 195
UnlockMetadataLoad .............................................. 195

Chapter 9. HsvMetadata Type Library ................................................... 197
About Member IDs ................................................... 197
HsvMetadata Object Methods ........................................... 198
EnumCellTextLabels ............................................... 198
EnumConsolidationMethodIDs ....................................... 199
EnumCustomDimsForApp .......................................... 199
EnumCustomDimsForAppEx ........................................ 199
EnumDimProperties ............................................... 200
EnumExtractOptions .............................................. 200
EnumLanguages .................................................. 201
EnumLoadOptions ................................................ 202
EnumMemberProperties ............................................ 203
Extract ......................................................... 203
ExtractModuleConfigurations ........................................ 206
GetApplicationAttribute ............................................ 206
GetApplicationAttributeExtDim ...................................... 206
GetApplicationCurrency ............................................ 207
GetApplicationSettingsTimeStamp .................................... 207
GetAppProfileInfo ................................................ 208
GetByIndexValidationAccount ....................................... 208
GetCellLevelAccountType ........................................... 209
GetCellLevelAccountTypeExtDim ..................................... 209
GetCellTextIdFromLabel ............................................ 209
GetCellTextIdsFromLabels .......................................... 210
GetCellTextLabelDescription ......................................... 210
GetCellTextLabelFromId ............................................ 211
GetConsolidationMethodDescription .................................. 211
GetConsolidationMethodInfo ........................................ 212
GetConsolidationMethodsTimeStamp .................................. 213
GetCurrencyTimeStamp ............................................ 214
GetCurrencyValueIDForEntityValueCombination ......................... 214
GetCurrencyValueIDsForEntityValueCombinations ........................ 215
GetCustomX .................................................... 215
GetDefaultValueOfActiveStatusAccount ................................. 216
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetDimension</td>
<td>216</td>
</tr>
<tr>
<td>GetDimensionIdFromName</td>
<td>217</td>
</tr>
<tr>
<td>GetFdmAppName</td>
<td>217</td>
</tr>
<tr>
<td>GetFrequencyID</td>
<td>217</td>
</tr>
<tr>
<td>GetICPEntitiesAggregationWeight</td>
<td>218</td>
</tr>
<tr>
<td>GetICPEntitiesAggregationWeightEx</td>
<td>218</td>
</tr>
<tr>
<td>GetMaxCustomDimNumber</td>
<td>218</td>
</tr>
<tr>
<td>GetMaxDimId</td>
<td>218</td>
</tr>
<tr>
<td>GetNumDimensions</td>
<td>219</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForAccountFlag</td>
<td>219</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustom1Flag</td>
<td>219</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustom2Flag</td>
<td>219</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustom3Flag</td>
<td>219</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustom4Flag</td>
<td>220</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustomXFlag</td>
<td>220</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForICPFlag</td>
<td>220</td>
</tr>
<tr>
<td>GetUseSubmissionPhaseFlag</td>
<td>221</td>
</tr>
<tr>
<td>GetValidationAccount</td>
<td>222</td>
</tr>
<tr>
<td>GetValidationAccountMembers</td>
<td>223</td>
</tr>
<tr>
<td>IsCustomDimValidForApp</td>
<td>223</td>
</tr>
<tr>
<td>IsCustomMemberValidForAccount</td>
<td>224</td>
</tr>
<tr>
<td>IsOrgByPeriodApplication</td>
<td>225</td>
</tr>
<tr>
<td>Load</td>
<td>225</td>
</tr>
<tr>
<td>LoadModuleConfigurations</td>
<td>229</td>
</tr>
<tr>
<td>LoadWithAccessCode</td>
<td>229</td>
</tr>
<tr>
<td>TranslateApplicationAttributeForDisplay</td>
<td>229</td>
</tr>
<tr>
<td>TranslateApplicationAttributeForDisplayExtDim</td>
<td>229</td>
</tr>
<tr>
<td>HsvMetadata Object Properties</td>
<td>230</td>
</tr>
<tr>
<td>Accounts</td>
<td>230</td>
</tr>
<tr>
<td>Currencies</td>
<td>230</td>
</tr>
<tr>
<td>Custom1</td>
<td>231</td>
</tr>
<tr>
<td>Custom2</td>
<td>231</td>
</tr>
<tr>
<td>Custom3</td>
<td>231</td>
</tr>
<tr>
<td>Custom4</td>
<td>231</td>
</tr>
<tr>
<td>Dimension</td>
<td>231</td>
</tr>
<tr>
<td>Entities</td>
<td>232</td>
</tr>
<tr>
<td>ICPs</td>
<td>232</td>
</tr>
<tr>
<td>Periods</td>
<td>232</td>
</tr>
<tr>
<td>Scenarios</td>
<td>232</td>
</tr>
<tr>
<td>IHsvTreeInfo Interface Methods</td>
<td></td>
</tr>
<tr>
<td>---------------------------------------------------------------------</td>
<td>---</td>
</tr>
<tr>
<td>EnumAllMemberIDs</td>
<td>233</td>
</tr>
<tr>
<td>EnumAllMemberLabels</td>
<td>233</td>
</tr>
<tr>
<td>EnumAllParentAndChildIDs</td>
<td>234</td>
</tr>
<tr>
<td>EnumAllParentAndChildLabels</td>
<td>235</td>
</tr>
<tr>
<td>EnumAncestors</td>
<td>236</td>
</tr>
<tr>
<td>EnumBaseMemberIDs</td>
<td>237</td>
</tr>
<tr>
<td>EnumDefaultAncestors</td>
<td>237</td>
</tr>
<tr>
<td>EnumDefaultAncestorsLabels</td>
<td>238</td>
</tr>
<tr>
<td>EnumDescendants</td>
<td>238</td>
</tr>
<tr>
<td>EnumIDsOfChildren</td>
<td>240</td>
</tr>
<tr>
<td>EnumIDsOfIChildren</td>
<td>240</td>
</tr>
<tr>
<td>EnumMemberLists</td>
<td>242</td>
</tr>
<tr>
<td>EnumMembers</td>
<td>242</td>
</tr>
<tr>
<td>EnumMembers2</td>
<td>243</td>
</tr>
<tr>
<td>EnumMembersWithAttribValue</td>
<td>245</td>
</tr>
<tr>
<td>EnumParents</td>
<td>245</td>
</tr>
<tr>
<td>EnumSortedIDsOfChildren</td>
<td>246</td>
</tr>
<tr>
<td>EnumSortedIDsOfIChildren</td>
<td>247</td>
</tr>
<tr>
<td>EnumSortedMembers</td>
<td>247</td>
</tr>
<tr>
<td>Find</td>
<td>248</td>
</tr>
<tr>
<td>FindByDesc</td>
<td>250</td>
</tr>
<tr>
<td>FindMatchingMembersFromHierarchy</td>
<td>251</td>
</tr>
<tr>
<td>FindMatchingMembersFromHierarchyByDesc</td>
<td>252</td>
</tr>
<tr>
<td>FindMatchingMembersFromHierarchyWildCard</td>
<td>253</td>
</tr>
<tr>
<td>GetAllPathsToMember</td>
<td>254</td>
</tr>
<tr>
<td>GetAttributeValue</td>
<td>255</td>
</tr>
<tr>
<td>GetDefaultHierarchyPosition</td>
<td>256</td>
</tr>
<tr>
<td>GetDefaultItemIDHierarchy</td>
<td>256</td>
</tr>
<tr>
<td>GetDefaultMemberID</td>
<td>257</td>
</tr>
<tr>
<td>GetDefaultParent</td>
<td>258</td>
</tr>
<tr>
<td>GetDefaultParentLabel</td>
<td>258</td>
</tr>
<tr>
<td>GetDescription</td>
<td>259</td>
</tr>
<tr>
<td>GetDisplayInfo</td>
<td>260</td>
</tr>
<tr>
<td>GetDisplayInfoForSeveralItems</td>
<td>261</td>
</tr>
<tr>
<td>GetItemGeneration</td>
<td>262</td>
</tr>
<tr>
<td>Method Name</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>GetItemID</td>
<td>263</td>
</tr>
<tr>
<td>GetItemIDQL</td>
<td>264</td>
</tr>
<tr>
<td>GetItemLevel</td>
<td>264</td>
</tr>
<tr>
<td>GetLabel</td>
<td>264</td>
</tr>
<tr>
<td>GetMemberListID</td>
<td>265</td>
</tr>
<tr>
<td>GetMemberListName</td>
<td>265</td>
</tr>
<tr>
<td>GetNumBaseMembers</td>
<td>266</td>
</tr>
<tr>
<td>GetNumChildren</td>
<td>266</td>
</tr>
<tr>
<td>GetNumDescendants</td>
<td>267</td>
</tr>
<tr>
<td>GetNumMembers</td>
<td>268</td>
</tr>
<tr>
<td>GetNumParents</td>
<td>268</td>
</tr>
<tr>
<td>GetQualifiedLabel</td>
<td>269</td>
</tr>
<tr>
<td>GetTreeCapabilities</td>
<td>269</td>
</tr>
<tr>
<td>GetTreeName</td>
<td>269</td>
</tr>
<tr>
<td>GetTreeTimeStamp</td>
<td>270</td>
</tr>
<tr>
<td>HasChildren</td>
<td>270</td>
</tr>
<tr>
<td>IsMemberABaseOf</td>
<td>271</td>
</tr>
<tr>
<td>IsMemberAChildOf</td>
<td>272</td>
</tr>
<tr>
<td>IsMemberADescendantOf</td>
<td>272</td>
</tr>
<tr>
<td>SortMembersBasedOnList</td>
<td>273</td>
</tr>
<tr>
<td>TranslateAttributeValueForDisplay</td>
<td>274</td>
</tr>
</tbody>
</table>

**HsvAccounts Object Methods**

<table>
<thead>
<tr>
<th>Method Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAccountType</td>
<td>275</td>
</tr>
<tr>
<td>GetCalcAttribute</td>
<td>276</td>
</tr>
<tr>
<td>GetICPTopMember</td>
<td>276</td>
</tr>
<tr>
<td>GetIsICP</td>
<td>277</td>
</tr>
<tr>
<td>GetNumDecimalPlaces</td>
<td>278</td>
</tr>
<tr>
<td>GetPlugAccount</td>
<td>279</td>
</tr>
<tr>
<td>GetSecurityClassID</td>
<td>279</td>
</tr>
<tr>
<td>GetSubmissionGroup</td>
<td>280</td>
</tr>
<tr>
<td>GetTopMemberOfValidCustom1Hierarchy</td>
<td>280</td>
</tr>
<tr>
<td>GetTopMemberOfValidCustom2Hierarchy</td>
<td>281</td>
</tr>
<tr>
<td>GetTopMemberOfValidCustom3Hierarchy</td>
<td>281</td>
</tr>
<tr>
<td>GetTopMemberOfValidCustom4Hierarchy</td>
<td>282</td>
</tr>
<tr>
<td>GetUserDefined1</td>
<td>282</td>
</tr>
<tr>
<td>GetUserDefined2</td>
<td>283</td>
</tr>
<tr>
<td>GetUserDefined3</td>
<td>283</td>
</tr>
<tr>
<td>GetXBRLTags</td>
<td>283</td>
</tr>
<tr>
<td>IsCalculated</td>
<td>283</td>
</tr>
</tbody>
</table>
Chapter 10. HsvData Type Library

About Cell Statuses
Update Mode Constants ............................................... 320
HsvData Object Methods .............................................. 320
   AddDataToMDDataBuffer .......................................... 321
   AddDataToMDDataBufferExtDim ..................................... 321
   AttachDocumentToCell ............................................ 321
   AttachDocumentToCellExtDim ....................................... 321
   ClearAllData ..................................................... 322
   ClearAllDescriptionsInSubCube .................................... 322
   ClearDataAuditItems .............................................. 323
   ClearDataAuditItemsEx ............................................ 324
   ClearInputData .................................................. 324
   ClearInvalidData ................................................. 326
   CopyInputData .................................................. 327
   CopyInputDataForMultipleEntities .................................... 327
   DeleteLineItemDetails .............................................. 328
   DeleteLineItemDetailsExtDim ........................................ 328
   DetachDocumentFromCell .......................................... 329
   DetachDocumentFromCellExtDim .................................... 329
   DMELoad ...................................................... 329
   DoCellDescriptionsExist ............................................ 329
   DoesCellDescriptionExist ........................................... 330
   DoesCellDescriptionExistExtDim ..................................... 330
   DoesDataExist ................................................... 330
   DoesSparseDataExist ............................................... 331
   DoesSparseDataExistExtDim ........................................... 331
   EnumDataAuditItems .............................................. 332
   EnumDataAuditItemsExtDim ........................................ 332
   EnumDataAuditItems2ExtDim ....................................... 333
   EnumEntitiesWithDataForScenarioYear ................................. 333
   EnumExtractOptions .............................................. 333
   EnumLoadOptions ................................................ 334
   Extract ......................................................... 335
   ExtractDataAuditItems ............................................. 339
   ExtractDataAuditItemsExtDim ....................................... 339
   ExtractDrillableRegions ........................................... 340
   ExtractDrillableRegionsByUrlNames .................................. 340
   FilterMembersThatHaveData ........................................ 341
   FilterMembersThatHaveDataExtDim .................................... 341
   FilterMembersThatHaveData2 ....................................... 342
FilterMembersThatHaveData2ExtDim .................................. 342
FormatNumberToText ............................................. 342
FormatNumberToText2 ............................................ 344
FormatStatusToText ............................................. 346
FormatStoredNumberToText ........................................ 347
FormatTextToNumber ............................................. 348
FormatTextToStoredNumber ........................................ 349
GetAllDescriptionsInSubCube ........................................ 350
GetAllDescriptionsInSubCubeExtDim .................................. 350
GetAllURLNames ................................................. 350
GetAttachedDocumentsToCell ....................................... 351
GetAttachedDocumentsToCellExtDim .................................. 351
GetBaseDataForAccount ............................................ 351
GetBaseDataForAccountExtDim ...................................... 352
GetCalcStatus .................................................... 352
GetCalcStatusEx .................................................. 352
GetCalcStatusExExtDim ............................................ 353
GetCalcStatusStatistics ............................................. 353
GetCell ........................................................ 354
GetCellExtDim ................................................... 354
GetCellDescription ................................................ 355
GetCellDescriptionExtDim .......................................... 355
GetCellDescriptions ............................................... 355
GetCellDescriptionsExtDim ......................................... 355
GetCellHistory ................................................... 356
GetCellHistoryExtDim ............................................. 356
GetCellHistory2 .................................................. 357
GetCellHistory2ExtDim ............................................ 357
GetCellJournalEntries .............................................. 358
GetCellJournalEntriesExtDim ........................................ 358
GetCellLineItems ................................................. 358
GetCellLineItemsExtDim ........................................... 358
GetCells ........................................................ 359
GetCellsExtDim ................................................... 359
GetCellsWithRowSuppression ...................................... 359
GetCellsWithRowSuppressionExtDim .................................. 360
GetCellsWithRowSuppression2 ..................................... 361
GetCellsWithRowSuppression2ExtDim .................................. 361
GetCountOfAttachedDocumentsToCell ................................. 361
Chapter 11. HsvCalculate Type Library

HsvCalculate Object Methods

Allocate
Allocate2
CalcEPU
ChartLogic
Chapter 12. HsvJournals Type Library ........................................... 409

Journal Column Return Values ........................................... 409
Template Column Return Values ........................................... 410
HsvJournals Object Methods ........................................... 411
  ClosePeriod ........................................................ 411
  GetItemID ........................................................ 412
  GetJournalTemplateItemID ........................................... 412
  GetPeriodStatusList ................................................. 413
  IsPeriodOpen ..................................................... 414
  OpenPeriod ...................................................... 414

IHsvJournalsEx Interface Methods ........................................ 415
  AddJournalGroup ................................................. 415
  ApproveJournals .................................................. 415
  DeleteJournals ................................................... 416
  DeleteTemplates .................................................. 417
EnumJournalGroups ............................................... 418
EnumJournalGroupsEx ............................................. 418
EnumJournalGroupsForScenarioYear .................................. 419
EnumJournalIDsForExtractFilter ...................................... 419
GenerateRecurring ................................................ 420
GetEntityJournals ................................................. 421
GetEntityJournalsExtDim ........................................... 421
GetEntityJournals2 ................................................ 422
GetEntityJournals2ExtDim .......................................... 422
GetJournal ...................................................... 423
GetJournalExtDim ................................................ 423
GetJournal2 ..................................................... 425
GetJournal2ExtDim ............................................... 425
GetJournalDisplayData ............................................. 426
GetJournalLabelsForIDs ............................................ 428
GetJournalQueryDefinitionIDs ....................................... 428
GetTemplate .................................................... 430
GetTemplateExtDim ............................................... 430
GetTemplateDisplayData ........................................... 431
GetTemplateLabelsForIDs ........................................... 432
GetTemplateQueryDefinitionIDs ...................................... 433
GetTextJournal ................................................... 434
GetTextJournalExtDim ............................................. 434
GetTextTemplate .................................................. 436
GetTextTemplateExtDim ........................................... 436
GetVariance ..................................................... 437
PostJournals ................................................... 439
RejectJournals ................................................... 440
RemoveAllJournalGroups ........................................... 441
RemoveJournalGroup ................................................ 441
SaveJournal ..................................................... 441
SaveJournalExtDim ................................................ 441
SaveTemplate .................................................... 443
SaveTemplateExtDim ................................................ 443
SaveTextJournal .................................................. 445
SaveTextJournalExtDim ............................................ 445
SaveTextTemplate .................................................. 448
SaveTextTemplateExtDim ........................................... 448
SubmitJournals ................................................... 450
Chapter 13. HsvSecurityAccess Type Library

Access Rights Constants .................................................. 461
Role ID Constants .......................................................... 462
Task Constants .......................................................... 462
E-mail Alerting Constants ................................................ 462
User Groups - User Type Flag Constants ................................ 462
Identity Type Constants ................................................... 463
Search Filters .......................................................... 463
Security Identifiers .......................................................... 464
The Connected User ...................................................... 464
HsvSecurityAccess Object Methods ....................................... 464
AddApplicationAdministrator ........................................ 465
AddApplicationAdministrator2 ......................................... 465
AddOrRemoveApplicationAdministrators .............................. 465
AddOrRemoveApplicationAdministrators2 ............................ 465
AddOrRemoveRolesFromUser ........................................... 466
AddOrRemoveRolesFromUser2 ........................................... 466
AddOrRemoveUsersFromRole ........................................... 467
AddOrRemoveUsersFromRole2 .......................................... 467
AddSecurityClass ....................................................... 468
AddSecurityClassWithAccessCode ...................................... 468
AddUser .......................................................... 468
AddUser2 .......................................................... 468
AddUserEx .......................................................... 468
AddUserToRole ......................................................... 469
AddUserToRole2 ......................................................... 469
AllowRulesLoadForEPMAApp ........................................... 469
<table>
<thead>
<tr>
<th>Function Name</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSecurityClassAccessForAllUsers</td>
<td>482</td>
</tr>
<tr>
<td>GetSecurityClassAccessForAllUsers2</td>
<td>482</td>
</tr>
<tr>
<td>GetSecurityClassID</td>
<td>482</td>
</tr>
<tr>
<td>GetSecurityClassLabel</td>
<td>483</td>
</tr>
<tr>
<td>GetSecurityClassRightsForConnectedUser</td>
<td>483</td>
</tr>
<tr>
<td>GetTaskAccessForConnectedUserFromList</td>
<td>483</td>
</tr>
<tr>
<td>GetUserAccessForAllRoles</td>
<td>484</td>
</tr>
<tr>
<td>GetUserAccessForAllRoles2</td>
<td>484</td>
</tr>
<tr>
<td>GetUserAccessForAllSecurityClasses</td>
<td>485</td>
</tr>
<tr>
<td>GetUserAccessForAllSecurityClasses2</td>
<td>485</td>
</tr>
<tr>
<td>GetUserID</td>
<td>485</td>
</tr>
<tr>
<td>GetUserIDFromSID</td>
<td>485</td>
</tr>
<tr>
<td>GetUserInfoFromUniqueID</td>
<td>485</td>
</tr>
<tr>
<td>GetUserInfoFromUniqueID2</td>
<td>486</td>
</tr>
<tr>
<td>GetUserName</td>
<td>486</td>
</tr>
<tr>
<td>GetUserName2</td>
<td>487</td>
</tr>
<tr>
<td>GetUserNameFromSID</td>
<td>487</td>
</tr>
<tr>
<td>GetUserSID</td>
<td>487</td>
</tr>
<tr>
<td>InsertDefaultSecurityClass</td>
<td>487</td>
</tr>
<tr>
<td>InsertDefaultSecurityClassWithAccessCode</td>
<td>488</td>
</tr>
<tr>
<td>IsApplicationAdministrator</td>
<td>488</td>
</tr>
<tr>
<td>IsClassicHFMApplication</td>
<td>488</td>
</tr>
<tr>
<td>IsConnectedUserAllowedToPerformTask</td>
<td>488</td>
</tr>
<tr>
<td>IsConnectedUserInRole</td>
<td>489</td>
</tr>
<tr>
<td>IsValidWindowsUser</td>
<td>489</td>
</tr>
<tr>
<td>RemoveApplicationAdministrator</td>
<td>489</td>
</tr>
<tr>
<td>RemoveApplicationAdministrator2</td>
<td>489</td>
</tr>
<tr>
<td>RemoveUser</td>
<td>490</td>
</tr>
<tr>
<td>RemoveUser2</td>
<td>490</td>
</tr>
<tr>
<td>RemoveUserFromRole</td>
<td>490</td>
</tr>
<tr>
<td>RemoveUserFromRole2</td>
<td>490</td>
</tr>
<tr>
<td>RenameSecurityClass</td>
<td>491</td>
</tr>
<tr>
<td>SetManySecurityClassRightsForUser</td>
<td>491</td>
</tr>
<tr>
<td>SetManySecurityClassRightsForUser2</td>
<td>491</td>
</tr>
<tr>
<td>SetRolesForUser</td>
<td>492</td>
</tr>
<tr>
<td>SetRulesMode</td>
<td>492</td>
</tr>
<tr>
<td>SetSecurityClassLabel</td>
<td>492</td>
</tr>
<tr>
<td>SetSecurityClassLabelWithAccessCode</td>
<td>493</td>
</tr>
<tr>
<td>SetSecurityClassRightsForManyUsers</td>
<td>493</td>
</tr>
</tbody>
</table>
Chapter 14. HsvSystemInfo Type Library

Contents
KillUsers .......................................................... 533
OutputSystemInfo ................................................... 533
ReleaseHsxServer .................................................... 534
SaveLastModifiedDateAndModifiedBy .................................. 534
SetApplicationDirectory ............................................... 534
SetCOMDLLCalcRules ................................................ 534
SetCurrentActivity ................................................... 534
SetCurrentModule ................................................... 534
SetCurrentModuleEx ................................................. 535
SetExtractFileEncoding ................................................ 535
SetLanguageUserParameters ............................................ 535
SetNumberFormattingUserParameters ..................................... 536
SetResourceLanguageForCurrentSession .................................. 536
SetResourceLanguageUserParameters ..................................... 537
SetUserParameter .................................................... 537
SetVBScriptCalcRules ................................................. 538
SetVBScriptMemberListRules ........................................... 539
StopRunningTask .................................................... 541
TraceLog .......................................................... 541
UpdateRunningTaskLogFilePathName ..................................... 542
UpdateRunningTaskPOV .............................................. 542
UpdateRunningTaskProgress .......................................... 542
UpdateRunningTaskProgressDetails ..................................... 542
UpdateRunningTaskStatus ............................................. 542
WarnUsersForShutDown .............................................. 542

Chapter 15. HsvProcessFlow Type Library ................................................ 543
Process Units versus Submission Phases .................................. 543
Approve ........................................................... 544
Approve2 .......................................................... 545
ApproveEx ......................................................... 546
GetGroupPhaseFromCell .............................................. 547
GetGroupPhaseFromCellExtDim ......................................... 548
GetHistory ......................................................... 548
GetHistory2 ......................................................... 550
GetPhasedSubmissionHistory ........................................... 551
GetPhasedSubmissionHistoryExtDim ..................................... 551
GetPhasedSubmissionState ............................................. 552
GetPhasedSubmissionStateExtDim ..................................... 552
GetPhasedSubmissionStateUsingPhaseID ................................... 552
GetPhasedSubmissionStateUsingPhaseIDExtDim .................................. 553
GetState ........................................................................ 553
PhasedSubmissionApprove ............................................... 554
PhasedSubmissionApproveExtDim ........................................... 554
PhasedSubmissionApprove2 .............................................. 555
PhasedSubmissionApprove2ExtDim ....................................... 555
PhasedSubmissionApproveEx ............................................. 556
PhasedSubmissionApproveExExtDim ...................................... 556
PhasedSubmissionGetHistory2 .......................................... 557
PhasedSubmissionGetHistory2ExtDim .................................... 557
PhasedSubmissionGetHistory2UsingPhaseID ................................ 558
PhasedSubmissionGetHistory2UsingPhaseIDExtDim ...................... 558
PhasedSubmissionProcessManagementChangeStateForMultipleEntities2 .... 560
PhasedSubmissionProcessManagementChangeStateForMultipleEntities2ExtDim 560
PhasedSubmissionPromote ............................................. 561
PhasedSubmissionPromoteExtDim ........................................ 561
PhasedSubmissionPromote2 ............................................. 562
PhasedSubmissionPromote2ExtDim ...................................... 562
PhasedSubmissionPublish .............................................. 563
PhasedSubmissionPublishExtDim ........................................ 563
PhasedSubmissionPublish2 ............................................. 563
PhasedSubmissionPublish2ExtDim ....................................... 564
PhasedSubmissionPublishEx ............................................ 565
PhasedSubmissionPublishExExtDim ...................................... 565
PhasedSubmissionReject ............................................... 566
PhasedSubmissionRejectExtDim ........................................ 566
PhasedSubmissionReject2 ............................................... 566
PhasedSubmissionReject2ExtDim ....................................... 566
PhasedSubmissionSignOff .............................................. 567
PhasedSubmissionSignOffExtDim ....................................... 567
PhasedSubmissionSignOff2 ............................................. 568
PhasedSubmissionSignOff2ExtDim ...................................... 568
PhasedSubmissionStart ................................................ 569
PhasedSubmissionStartExtDim .......................................... 569
PhasedSubmissionStart2 ............................................... 570
PhasedSubmissionStart2ExtDim ......................................... 570
PhasedSubmissionStartEx .............................................. 571
PhasedSubmissionStartExExtDim ....................................... 571
Chapter 17. HsvMDArrays Type Library .................................................. 611
Types of Indexes ............................................................. 612
HsvMDDataBuffer Object Methods ............................................. 612
   BeginEnumeration .................................................. 613
   CreateDataIndexList .............................................. 613
   CreateDataIndexListExtDim ....................................... 613
   CreateDataIndexListEx .......................................... 614
   EndEnumeration ................................................... 614
   EraseRecordFromPMBuffer ........................................ 614
   GetCheckLineItemDetailsForCaseInsensitiveDuplicates ......... 615
   GetCubeIndexFromPOV ............................................ 615
   GetCubePOVFromIndex ............................................ 615
   GetData .......................................................... 616
   GetDataExtDim .................................................... 616
   GetDataAtIndex .................................................. 617
   GetDataAtIndexExtDim ........................................... 617
   GetDescription ................................................... 618
   GetDescriptionExtDim .......................................... 618
   GetDescriptionAtIndex .......................................... 618
   GetDescriptionAtIndexExtDim ................................... 618
   GetLineItems .................................................... 619
   GetLineItemsExtDim ............................................. 619
   GetLineItemsAtIndex ............................................. 620
   GetLineItemsAtIndexExtDim .................................... 620
   GetNumCellsForData ............................................. 620
   GetNumCellsForDescriptions .................................... 621
   GetNumCellsForLineItems ...................................... 621
   GetNumPeriodsInCubeForData ................................... 622
   GetNumPeriodsInCubeForDescriptions ......................... 622
   GetNumPeriodsInCubeForLineItems ............................ 623
   GetPeriodIndexFromPOVForData ................................ 623
   GetPeriodIndexFromPOVForDescriptions ...................... 624
   GetPeriodIndexFromPOVForLineItems ........................ 625
   GetPeriodPOVFromIndexForData ................................ 626
GetPeriodPOVFromIndexForDescriptions ......................... 627
GetPeriodPOVFromIndexForLineItems ............................. 627
GetPMErrorRecordCount ......................................... 627
GetPMRecordCount ............................................. 628
GetRecordFromPMBuffer ......................................... 628
GetRecordFromPMErrorBuffer ..................................... 628
GetSortedNature .................................................. 628
InsertDataAtBeginning .......................................... 628
InsertDataAtBeginningExtDim .................................... 628
InsertDataAtEnd ............................................... 628
InsertDataAtEndExtDim .......................................... 628
InsertDescriptionAtBeginning .................................... 629
InsertDescriptionAtBeginningExtDim ............................ 629
InsertDescriptionAtEnd ........................................ 629
InsertDescriptionAtEndExtDim .................................. 629
InsertLineItemsAtBeginning ...................................... 630
InsertLineItemsAtBeginningExtDim ............................... 630
InsertLineItemsAtEnd .......................................... 630
InsertLineItemsAtEndExtDim .................................... 630
InsertLineItemsAtBeginningExtDim .............................. 630
InsertRecordIntoPMBuffer ..................................... 631
InsertRecordIntoPMErrorBuffer ................................ 631
RemoveAll .................................................... 631
SetCheckLineItemDetailsForCaseInsensitiveDuplicates ............ 632
SetData .......................................................... 632
SetDataExtDim .................................................. 632
SetDescription .................................................. 633
SetDescriptionExtDim .......................................... 633
SetGrowByAmount ................................................ 633
SetLineItems ..................................................... 633
SetLineItemsExtDim ............................................ 633
SetSortedNature .................................................. 634
Sort ............................................................ 634
HsvMDDDataBufferLite Object Methods ............................... 634
BeginEnumeration ............................................... 635
CreateDataIndexList ............................................ 635
CreateDataIndexListExtDim ..................................... 636
EraseRecordFromPMBuffer ....................................... 636
EndEnumeration .................................................. 637
GetCheckLineItemDetailsForCaseInsensitiveDuplicates ............. 637
SetData ........................................................ 654
SetDataExtDim ................................................... 654
SetDescription ................................................... 655
SetDescriptionExtDim ............................................. 655
SetGrowByAmount ................................................ 655
SetLineItems ..................................................... 655
SetLineItemsExtDim ............................................... 656
SetSortedNature .................................................. 656
Sort ........................................................... 656
IHsvMDDataBufferLite Interface ......................................... 656
SetMinCubesInCache .............................................. 657
HsvTransactionData Object Methods ............................... 657
BeginDataEnum .................................................. 658
BeginQueryEnum ................................................. 658
EndDataEnum ................................................... 658
EndQueryEnum .................................................. 659
GetFixedDimensionMembers ........................................ 659
GetQueryItem ................................................... 659
GetQueryItemExtDim .............................................. 659
GetTransactionData ............................................... 659
GetTransactionDataExtDim ......................................... 659
Initialize ........................................................ 660
SetAccessRight ................................................... 660
SetQueryItem .................................................... 661
SetQueryItemExtDim .............................................. 661
SetTransactionData ................................................ 661
SetTransactionDataExtDim .......................................... 661
HsvMDIndexList Object Methods ........................................ 661
GetItem ........................................................ 662
GetItemExtDim .................................................. 662
GetNumItems ................................................... 662
HsvICTransactionsData Object Methods ............................. 663
AddAccountCustomCombination ..................................... 663
AddAccountCustomCombinationExtDim ............................... 663
AddICTransactionData ............................................. 664
AddICTransactionDataExtDim ......................................... 664
AddQueryDimensionMemberID ...................................... 664
AddQueryField ................................................... 664
AddQueryFieldItem ............................................... 664
AddQueryOrderField ..............................................  664
BeginDataEnum ..................................................  664
BeginQueryFieldEnum ..........................................  665
BeginQueryOrderEnum ..........................................  665
EndDataEnum ...................................................  665
EndQueryFieldEnum ..............................................  665
EndQueryOrderEnum ..............................................  665
GetAccessRights ..................................................  665
GetAccountCustomCombination ....................................  666
GetAccountCustomCombinationExtDim ..........................  666
GetAccountCustomCombinationTotal ..........................  666
GetEntityPartnerOption ...........................................  666
GetErrorStatus ...................................................  666
GetFilterOptions ..................................................  667
GetFixedDimensionMembers .......................................  667
GetICTransactionCell .............................................  667
GetICTransactionCellExtDim ......................................  667
GetICTransactionData .............................................  668
GetICTransactionDataExtDim ......................................  668
GetMatchCode ...................................................  669
GetNumTransactionsCached .......................................  670
GetPagingOption ..................................................  670
GetPartnerAsEntityList ..........................................  670
GetPartnerQueryDimensionMemberIDs ..........................  670
GetQueryDimensionMemberIDs ....................................  670
GetQueryFieldInformation .......................................  670
GetQueryFieldItem ................................................  670
GetQueryOrderField ...............................................  670
GetTotalTransactions ............................................  670
GetTransGroupType ...............................................  670
Initialize ........................................................  671
InitializeSequenceIdMap .........................................  672
IsEntityInPartnerAsEntityList ..................................  672
IsICTransactionValid .............................................  672
IsICTransactionValidExtDim ......................................  672
RemoveQueryOrder ...............................................  672
SetAccessRights ..................................................  672
SetEntityPartnerOption ..........................................  673
SetErrorStatus ...................................................  673
Chapter 18. HsvDataCubes Type Library ............................................. 675

About Subcube Items .......................................................... 675

HsvCurrencyCube Object Methods ........................................ 676

BeginEnumerationOfStoredData .......................................... 676
EndEnumerationOfStoredData ............................................ 676
GetCell ................................................................. 677
GetCellExtDim .......................................................... 677
GetFixedDimensionMembers ............................................. 677
GetOneCellFromStoredItem .............................................. 678
GetOneCellFromStoredItemExtDim .................................... 678
GetPOVFromStoredItem .................................................. 679
GetPOVFromStoredItemExtDim ........................................ 679

HsvNodeCube Object Methods ............................................. 680

BeginEnumerationOfStoredData .......................................... 680
EndEnumerationOfStoredData ............................................ 680
GetCell ................................................................. 681
GetCellExtDim .......................................................... 681
GetFixedDimensionMembers ............................................. 681
GetOneCellFromStoredItem .............................................. 682
GetOneCellFromStoredItemExtDim .................................... 682
GetPOVFromStoredItem .................................................. 683
GetPOVFromStoredItemExtDim ........................................ 683

Chapter 19. HsvDQI Type Library ............................................. 685

HsvDQI Object Methods .................................................. 685

HsvDQI Object Methods .................................................. 685

Chapter 20. HsvStarSchemaACM Type Library .................................. 725

HsvStarSchemaACM Object Methods ..................................... 725
CreateStarSchema ................................................. 726
CreateStarSchemaExtDim ........................................... 726
CreateStarSchemaAndReturnTaskID ................................... 727
CreateStarSchemaFromHTTP ........................................ 728
DeleteStarSchema ................................................. 728
EnumApplicationStarSchemas ......................................... 728
EnumRegisteredDSNs .............................................. 728
GetAsynchronousTaskStatus ........................................ 729
GetExtractLogData ................................................ 730
GetPersistedSettings ............................................... 731
GetPersistedSettingsExtDim ......................................... 731
QuitAsynchronousTask ............................................. 731
SetDefaultTablePrefix .............................................. 731
SetPersistedSettings ................................................ 731
SetPersistedSettingsExtDim ......................................... 731
SetSession .................................................................. 731
TestSQLConnection ............................................... 732
IHsvStarSchemaTemplates Interface ...................................... 732
DeleteTemplate .................................................. 732
EnumTemplates .................................................. 732
GetTemplate .................................................... 733
SetTemplate ..................................................... 733

Chapter 21. HsvICM Type Library ...................................................... 737
HsvICM Object Methods .............................................. 737
AllMVTransPosted ................................................ 737
CheckReportSecurity .............................................. 738
CreateICTransaction ............................................... 738
CreateICTransactionExtDim ......................................... 738
DeleteICReasonCode .............................................. 739
DoesCellSupportICTransactionDetail .................................. 740
DoesCellSupportICTransactionDetailExtDim ............................. 740
GetColumnFilter ................................................. 740
GetCurrencyInfo ................................................. 740
GetEntitiesContacts ............................................... 741
GetEntityCurrencyID .............................................. 741
GetICReasonCodeID ................................................ 741
GetICReasonCodeLabel ............................................ 742
GetICReasonCodes ................................................ 742
GetICTransactions ................................................  743
GetMonitorICDetails ..............................................  743
GetMonitorICSummary ............................................  744
GetMonitorICTransactions ..........................................  745
GetRowFilter ....................................................  746
GetTransCurrencyID ..............................................  746
GetUnMatchTransactions ...........................................  746
IsOneSideOfTransactionGroupWriteable ..........................  747
MatchAutoAccounts ...............................................  748
MatchAutoIDs ....................................................  749
NoTransMatchedOrPosted ..........................................  749
ProcessAllICTrans ................................................  750
ProcessICTransactions .............................................  751
ProcessICTransactionsExtDim ......................................  751
SaveColumnFilter .................................................  752
SaveICReasonCode ................................................  752
SaveICTransaction ................................................  752
SaveICTransactionExtDim ..........................................  753
SaveRowFilter ....................................................  754
SetReasonCodeToICTransactions ...................................  754
UnMatchICTransactions ............................................  755
IHsvAdminICM Interface Methods .......................................  755
CheckSecurityForICExtract ........................................  756
CloseICPeriod ...................................................  756
GetICEntitiesLockStatus .........................................  757
GetLockStatusICEntities .........................................  757
GetLockStatusICEntity .............................................  757
GetSettingsICPeriod ...............................................  758
GetSettingsICPeriods ..............................................  759
LoadICTransactions ...............................................  761
LoadICTransactionsExtDim .........................................  761
LockICEntity ....................................................  761
OpenICPeriod ...................................................  762
OpenICPeriod2 ..................................................  762
SavePeriodsSettings ................................................  763
UnlockICEntity ..................................................  763
UpdatePeriodSettings ..............................................  763
UpdatePeriodSettings2 .............................................  764
Chapter 22. Type Libraries for Loading and Extracting Information ............................... 765
Load and Extract Option Interfaces ................................................. 765
IHsvLoadExtractOptions Interface Properties .................................. 766
IHsvLoadExtractOption Interface Properties .................................. 767
Common Properties and Methods ............................................. 769
Common Load and Extract Properties ..................................... 770
Common Load and Extract Methods ....................................... 771
HsvSecurityLoadACV Type Library ............................................ 774
IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for
HsvSecurityLoadACV .................................................. 774
Loading Security Information ............................................ 775
Extracting Security Information ......................................... 776
Security Load Options .................................................. 777
Security Extract Options ................................................. 778
HsvSecurityLoadACV Object Properties .................................. 779
HsvSecurityLoadACV Object Methods .................................... 779
HsvMetadataLoadACV Type Library ........................................... 780
IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for
HsvMetadataLoadACV .................................................. 781
Loading Metadata .................................................... 781
Extracting Metadata ................................................... 782
Metadata Load Options .................................................. 783
Metadata Extract Options ................................................ 786
HsvMetadataLoadACV Object Properties .................................. 789
HsvMetadataLoadACV Object Methods .................................... 789
HsvRulesLoadACV Type Library ............................................. 790
HsvRulesLoadACV Object Methods ....................................... 790
HsvcDataLoad Type Library ................................................ 796
IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for
HsvcDataLoad .......................................................... 797
Loading Data ............................................................. 797
Extracting Data .......................................................... 798
Data Load Options ........................................................ 800
Data Extract Options .................................................... 802
HsvcDataLoad Object Properties ......................................... 804
HsvcDataLoad Object Methods ............................................. 805
HsvJournalLoadACV Type Library ........................................... 806
IHsvLoadExtractOptions and IHsvLoadExtractOption Interfaces for
HsvJournalLoadACV .................................................... 806
Loading Journals ......................................................... 807
Chapter 23. HsvPOVSelection Type Library ................................................ 817
HsvPOVSelection Object Methods ................................................ 818

CheckAllItems ................................................... 818
CheckAllItemsUsingSelectedList ........................................ 818
EnableDimension ................................................. 819
GetCheckedItems ................................................. 820
GetListInfo ...................................................... 820
GetMember ..................................................... 821
GetNumCheckedItems ............................................. 822
GetOBPInfo ..................................................... 822
Initialize ........................................................ 823
InitializeEx ...................................................... 824
InitializeHTTP ................................................... 824
InitializeWithConnection ........................................... 824
SelectDimension .................................................. 824
SetCheckedItems ................................................. 824
SetDimUnk ..................................................... 825
SetListInfo ...................................................... 826
SetMember ...................................................... 826
SetOBPInfo ..................................................... 827
SetResourceAndLanguage ........................................... 828
ShowCheckBoxes ................................................. 828

Chapter 24. Error Handling and the HsvResourceManager Type Library ............. 833
Error Handling with Financial Management ........................................ 833
HsvResourceManager Object Methods .......................................... 834
Handling Errors with the HsvResourceManager Object ...................... 834
System Message Detail Strings .............................................. 835
Appendix A. Constants: The HFMConstants Type Library ................................. 845

Metadata-Related Constants ................................................................. 845

Account Dimension Constants ......................................................... 846
Custom Dimension Constants .......................................................... 850
Consolidation Method Constants ....................................................... 852
Currency Attribute Constants ........................................................... 853
Entity Dimension Constants .............................................................. 854
View Dimension Constants ................................................................. 855
Supported Language Constants .......................................................... 856
Dimension-Related Constants ............................................................ 857
Intercompany Partner Constants ........................................................ 858
Period Dimension Constants ............................................................... 860
Scenario Attribute Constants ............................................................. 861
Member List Constants ...................................................................... 862
Value Dimension Constants ............................................................... 862
IHsvTreeInfo Interface Constants ........................................................ 864
Application Setting Attribute ID Constants ........................................... 864
Cell Status Constants ......................................................... 866
Cell Calculation Status Constants ........................................... 866
Cell Status Constants ......................................................... 867
Cell Metadata Status Constants ............................................. 867
Subcube Period Calculation Status Constants .......................... 868
Transaction Dimension Constants ......................................... 869
Cell Transaction Type Constants .......................................... 871
Calculation Status Types .................................................... 871
Additional Status Information Constant ................................. 872
Journal-Related Constants ................................................... 872
Journal Action Constants ..................................................... 872
Balance Type Constants ..................................................... 873
Journal Column Display Constants ......................................... 873
Debit/Credit Constants ....................................................... 874
Period Status Constants ...................................................... 874
Journal Report Display Option Constants ............................... 874
Journal Report Sort Option Constants .................................... 875
Journal Report Total Flag Constants ....................................... 875
Journal Status Constants ..................................................... 875
Journal Type Constants ....................................................... 876
Journal Web Session Parameter Constants .............................. 876
Template Column Display Constants ...................................... 876
Template Type Constants ..................................................... 877
Journal and Template Type Constants ..................................... 877
Process Management Constants ............................................. 877
Process Management Action Constants ................................... 878
Process Management Review Level Constants .......................... 878
Report Column Constants .................................................... 879
Consolidation Type Constants .............................................. 880
Tier Constants ............................................................... 880
Security Constants ........................................................... 881
Access Rights Constants ...................................................... 881
Role Constants ............................................................... 881
Task Constants ............................................................... 884
Web Constants ............................................................... 888
Data Explorer Task Constants .............................................. 888
Data Explorer Process Management Constants .......................... 889
Data Grid Definition Constants ............................................ 890
Data Grid Member Expansion Mode Constants ........................... 890

Contents xxxix
Data Grid Dimension Expansion Mode Constants ........................................ 891
Data Grid Transaction Information Constants ........................................ 891
Data Information Display Constants .................................................. 891
Data Display Page Constant ............................................................. 892
Member Display Constants ............................................................. 892
Metadata Information Constants ..................................................... 892
Document Type Constants ................................................................... 893
Document File Type Constants ........................................................... 894
Extracted File Encoding Constants ..................................................... 894
User Activity Constants ...................................................................... 895
Task Status Constants ........................................................................ 897
Log Severity Constants ...................................................................... 898
Intercompany Transaction Constants .................................................. 898
  Lock Status Constants ..................................................................... 899
  Period Status Constants .................................................................. 899
  Match/Verify Before Post Constants ............................................... 899
  Match Option Constants .................................................................. 900
  Posting Status Constants ............................................................... 900
  Matching Status Constants ............................................................. 900
  Processing Action Constants .......................................................... 901
Transaction Load Mode Constants ....................................................... 901
Filtering and Sorting Options ............................................................. 901
Process Status Constants ................................................................. 902
Event Constants ................................................................................. 902
Miscellaneous Constants ..................................................................... 903
  Showing Public and Private Documents .......................................... 903
  Date and Time Format Constants ................................................... 903
  Invalid ID Constant ........................................................................ 904
  Maximum String Length Constants .................................................. 904
  Aggregation Type Constants ............................................................ 905
  Member ID Range .......................................................................... 906
  Number Defaults Constants ............................................................ 906
  Share Calculation Ownership Constants ......................................... 906
  Share Calculation Types Constants .................................................. 907
  Validation Type Constants .............................................................. 907
  Default Security Class Constant ..................................................... 908
  Module ID Constants ........................................................................ 908
Appendix B. HTTP Listener for Extended Analytics ............................................. 913
  Communicating with the HTTP Listener ........................................... 913
  Format of Input Parameters ..................................................... 914
  Application Tag ........................................................................ 915
  Input Parameter Headers ......................................................... 915
  ExtractType ........................................................................... 915
  PushOption .................................................................................. 915
  Element Selections for all Dimension ......................................... 916
  Input Parameter File Example ................................................... 916
  Response Message ................................................................. 917

Appendix C. Change History ........................................................................ 919
  Changes for Release 11.1.2.3 ...................................................... 919
    HsvCalculate Type Library ........................................................ 920
    HsvDataType Library ............................................................... 920
    HsvDQI Type Library ................................................................ 920
    HsvJournals Type Library ......................................................... 920
    HsvMetadata Type Library ....................................................... 920
    HsvSecurityAccess Type Library ............................................. 920
    HsvSession Type Library ........................................................ 920
    HsvSystemInfo Type Library .................................................... 921
  Changes for Release 11.1.2.2.300 ................................................ 921
  Changes for Release 11.1.2.2 ....................................................... 921
  Changes for Release 11.1.2.1 ....................................................... 922
    HsvJournals Type Library ........................................................ 922
    Hsv SystemInfo Type Library .................................................... 922
  Changes for Release 11.1.2 ......................................................... 922
    HsvData Type Library ............................................................. 922
    HsvJournals Type Library ........................................................ 923
    HsvReports Type Library .......................................................... 923
    HsvStarSchemaACM Type Library .......................................... 923
    HsvSystemInfo Type Library .................................................... 923
    HsxClient Type Library ........................................................... 923
  Changes for Release 11.1.3 ......................................................... 923
    HsvData Type Library ............................................................. 923
    HsvJournals Type Library ........................................................ 924
    HsvMetadata Type Library ....................................................... 924
    HsvResourceManager Type Library ....................................... 924
    HsvSecurityAccess Type Library ............................................. 924
Documentation Accessibility

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

Access to Oracle Support

Oracle customers have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.
You can use the objects described in this book to programmatically extend Oracle Hyperion Financial Management. 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 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:

<table>
<thead>
<tr>
<th>Visual Basic 6 Types</th>
<th>COM Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>VARIANT_BOOL</td>
</tr>
<tr>
<td>Byte</td>
<td>Byte</td>
</tr>
<tr>
<td>String</td>
<td>BSTR</td>
</tr>
<tr>
<td>Double</td>
<td>Double</td>
</tr>
<tr>
<td>Single</td>
<td>Float</td>
</tr>
<tr>
<td>Integer</td>
<td>short</td>
</tr>
<tr>
<td>Long</td>
<td>long</td>
</tr>
<tr>
<td>Object</td>
<td>IDispatch</td>
</tr>
<tr>
<td>Object</td>
<td>IUnknown</td>
</tr>
<tr>
<td>Variant (array)</td>
<td>Variant *</td>
</tr>
<tr>
<td>Variant</td>
<td>Variant</td>
</tr>
</tbody>
</table>

Features Exposed by the Objects

The objects covered in this book expose many Financial Management features, including these 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
- Running consolidations, translations, and calculations
- Opening and closing periods
- Processing journals
- Defining application security
- Process Management
- Loading and extracting member lists, rules, data, and journals
Application Server Tier Type Libraries

After an HsvSession object is instantiated, the client can initiate tasks such as consolidations and data entry. 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

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

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>
<thead>
<tr>
<th>Type Library</th>
<th>Corresponding Features</th>
</tr>
</thead>
<tbody>
<tr>
<td>HsvSecurityLoadACV</td>
<td>Loading and extracting security information.</td>
</tr>
<tr>
<td>HsvMetadataLoadACV</td>
<td>Extracting metadata.</td>
</tr>
<tr>
<td>HsvRulesLoadACV</td>
<td>Loading and extracting member lists and rules.</td>
</tr>
<tr>
<td>HsvcDataLoad</td>
<td>Loading and extracting data.</td>
</tr>
<tr>
<td>HsvJournalLoadACV</td>
<td>Loading and extracting journals and journal templates.</td>
</tr>
</tbody>
</table>

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 are displayed.

Note: See Chapter 23, “HsvPOVSelection Type Library.”
Additional Architectural Considerations

These 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 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 data synchronization across application servers is automatically applied by Financial Management; you do not need to write any code to do this. See “Data Across Multiple Application Servers: Subcube Caching” on page 54.

About Subcubes

Several Financial Management methods work with subcubes. A subcube consists of all the cells that share the same members of these 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 is 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]

The following list provides examples of subcube types:

- **Currency subcube**: A currency subcube stores the cells for the year, Actual scenario, Connecticut entity, and USD currency. The USD currency is represented by the [USD], [USD Adjs], and [USD Total] Value dimension triplet.

- **Node subcube**: A node subcube stores the cells for the year, 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, the system 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 this process to ensure that data is synchronized:

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 was updated by another application server since the current application server last cached the subcube.

3. If the subcube was 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 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. 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 indicates this with the ByVal keyword; otherwise the argument is passed by reference.

Error Handling

The methods in this document return HRESULT error numbers for 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 completes, Err.Number returns zero.

Financial Management enables you to obtain simple descriptions of errors that you can display to users and technical error information that can help debug issues. If Financial Management is localized, error messages are also localized. see Chapter 24, “Error Handling and the HsvResourceManager Type Library.”
About the Type Libraries and Objects

In This Chapter

Type Library References ...................................................................................57
Objects, Properties, and Methods........................................................................59

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

The Financial Management 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 need 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 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>
<thead>
<tr>
<th>Type Library</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HsxClient</td>
<td>Provides client tier functionality such as logging on and opening applications.</td>
</tr>
<tr>
<td>HsxClientUI</td>
<td>Provides dialog boxes for some of the client tier functionality.</td>
</tr>
<tr>
<td>HsxServer</td>
<td>Gets application server tier information such as the server’s Financial Management applications and data sources.</td>
</tr>
<tr>
<td>HsvSession</td>
<td>Provides access to Financial Management applications. Use HsvSession’s properties to access subordinate objects such as HsvData and HsvMetadata.</td>
</tr>
<tr>
<td>HsvMetadata</td>
<td>Gets information about an application’s metadata. HsvMetadata contains many objects, enabling you to get information about dimension members.</td>
</tr>
<tr>
<td>HsvData</td>
<td>Gets and sets data for an application.</td>
</tr>
<tr>
<td>HsvCalculate</td>
<td>Executes consolidations, calculations, and translations.</td>
</tr>
<tr>
<td>HsvJournals</td>
<td>Processes journals, opens and closes periods, and generates reports.</td>
</tr>
<tr>
<td>HsvSecurityAccess</td>
<td>Defines an application’s security.</td>
</tr>
<tr>
<td>HsvSystemInfo</td>
<td>Gets and sets various types of system information such as application directories and server names.</td>
</tr>
<tr>
<td>HsvProcessFlow</td>
<td>Exposes the Process Management features, enabling you to take actions for and get histories of process units.</td>
</tr>
<tr>
<td>HsvReports</td>
<td>Gets report information, enumerates an application’s reports, and saves and deletes reports.</td>
</tr>
<tr>
<td>HsvICM</td>
<td>Exposes the intercompany transaction features.</td>
</tr>
<tr>
<td>HsvMDArrays</td>
<td>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.</td>
</tr>
<tr>
<td>HsvDataCubes</td>
<td>Provides access to cell data and information at the subcube level.</td>
</tr>
<tr>
<td>HsvStarSchemaACM</td>
<td>Exposes the Extract Data Extended Analytics features.</td>
</tr>
<tr>
<td>HsvSecurityLoadACV</td>
<td>Loads and extracts security settings, using load and extract files on client computers.</td>
</tr>
<tr>
<td>HsvMetadataLoadACV</td>
<td>Loads metadata using files on client computers.</td>
</tr>
<tr>
<td>HsvRulesLoadACV</td>
<td>Loads and extracts rules and member lists, using load and extract files on client computers.</td>
</tr>
<tr>
<td>HsvcDataLoad</td>
<td>Loads and extracts data, using load and extract files on client computers.</td>
</tr>
<tr>
<td>HsvJournalLoadACV</td>
<td>Loads and extracts journals and journal templates, using load and extract files on client computers.</td>
</tr>
<tr>
<td>HsvPOVSelection</td>
<td>Exposes an insertable control that provides a user interface for selecting dimension members.</td>
</tr>
<tr>
<td>HsvResourceManager</td>
<td>Provides an interface to the Resource Manager.</td>
</tr>
</tbody>
</table>

**Note:** The Resource Manager exposes error message strings for Financial Management. If the interface is localized, error messages are also localized.

| HfmSliceCOM       | Provides objects needed to support configurable dimensionality.                                                                                                                                             |
Objects, Properties, and Methods

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

- “HsxClient Type Library Overview” on page 59
- “HsxClientUI Type Library Overview” on page 62
- “HsxServer Type Library Overview” on page 63
- “HsvSession Type Library Overview” on page 64
- “HsvMetadata Type Library Overview” on page 65
- “HsvData Type Library Overview” on page 78
- “HsvCalculate Type Library Overview” on page 85
- “HsvJournals Type Library Overview” on page 86
- “HsvSecurityAccess Type Library Overview” on page 90
- “HsvSystemInfo Type Library Overview” on page 96
- “HsvProcessFlow Type Library Overview” on page 99
- “HsvReports Type Library Overview” on page 103
- “HsvMDArrays Type Library Overview” on page 104
- “HsvDataCubes Type Library Overview” on page 115
- “HsvStarSchemaACM Type Library Overview” on page 120
- “HsvICM Type Library” on page 121
- “Type Libraries for Loading and Extracting Information” on page 124
- “HsvPOVSelection Type Library” on page 131
- “HsvResourceManager Type Library Overview” on page 133

HsxClient Type Library Overview

The HsxClient type library contains 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:

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

Use the SetLogonInfoSSO and OpenApplication methods to log on and to connect to applications. 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. See “HsxClientUI Type Library Overview” on page 62.
The following table lists the HsxClient object’s methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AuthenticateCSSToken</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>AuthenticateSecurityAgentCredentials</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>AuthenticateUserCredentials</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>AuthenticateUseronClient</td>
<td>Allows authentication operation to be done on the Web server.</td>
</tr>
<tr>
<td>AuthenticateUserOnCluster</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>AuthenticateUserOnClusterSSO</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>AuthenticateUserOnServer</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>CreateApplication</td>
<td><em>Deprecated</em></td>
</tr>
<tr>
<td>CreateApplicationCAS</td>
<td>Creates a Classic application.</td>
</tr>
<tr>
<td>CreateApplicationCASWithAccessCode</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>CreateObjectOnCluster</td>
<td>Instantiates an object on a cluster or application server.</td>
</tr>
<tr>
<td>CreateObjectOnServer</td>
<td><em>Deprecated</em> - use CreateObjectOnCluster.*</td>
</tr>
<tr>
<td>DeleteApplication</td>
<td>Deletes a Classic application.</td>
</tr>
<tr>
<td>DeleteApplicationWithAccessCode</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>DeleteSystemErrors</td>
<td>Deletes records of system errors for a cluster.</td>
</tr>
<tr>
<td>DetermineWindowsLoggedOnUser</td>
<td>Returns the domain name and the username of the user who is logged onto Windows.</td>
</tr>
<tr>
<td>DisableNewConnections</td>
<td>Disables new Financial Management connections to a cluster for the specified application and application server criteria.</td>
</tr>
<tr>
<td>DoesUserHaveCreateApplicationRights</td>
<td>Indicates whether the connected user is a member of the Creator group for an application server cluster.</td>
</tr>
<tr>
<td>DoesUserHaveSystemAdminRights</td>
<td>Indicates whether the connected user is a member of the Administrator group for an application server cluster.</td>
</tr>
<tr>
<td>EnableNewConnections</td>
<td>Enables new Financial Management connections to a cluster for the specified application and application server criteria.</td>
</tr>
<tr>
<td>EnumProhibitConnections</td>
<td>Returns information on the applications, application servers, and users for which connections are disabled on a cluster.</td>
</tr>
<tr>
<td>EnumProvisioningProjects</td>
<td>Returns the names of the provisioning projects for Oracle Hyperion Shared Services associated with a cluster.</td>
</tr>
<tr>
<td>EnumRegisteredClusterNames</td>
<td>Returns an array containing the names of the clusters or application servers registered for the client.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnumRegisteredClustersOrServers</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EnumRegisteredServerNames</td>
<td>Deprecated - use EnumRegisteredClusterNames.</td>
</tr>
<tr>
<td>EnumUserAppPreferences</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EnumUsersOnSystem</td>
<td>Returns the usernames of and other information applicable to the users logged on to a cluster.</td>
</tr>
<tr>
<td>EnumUsersOnSystemEx</td>
<td>Returns the usernames of and other information applicable to users logged on to a cluster; you can specify the language in which the names of the user’s active modules are returned.</td>
</tr>
<tr>
<td>EnumUsersOnSystemEx2</td>
<td>Returns the usernames and other information applicable to all users logged on to or logged out from a cluster.</td>
</tr>
<tr>
<td>GetApplicationFolder</td>
<td>Returns the path of an application’s local storage folder.</td>
</tr>
<tr>
<td>GetClusterInfo</td>
<td>Returns the name of the cluster to which the specified application server is assigned.</td>
</tr>
<tr>
<td>GetClustersAndServers</td>
<td>Enumerates clusters and application servers.</td>
</tr>
<tr>
<td>GetHFMErrorLogRecordSet</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetLogonInfo</td>
<td>Deprecated - use GetLogonInfoSSO.</td>
</tr>
<tr>
<td>GetLogonInfoSSO</td>
<td>Gets the domain, username, and external authentication token for the connected user.</td>
</tr>
<tr>
<td>GetModifyApplicationInfo</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetModifyApplicationStatus</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetServerOnCluster</td>
<td>Returns an object reference to the HsxServer object that represents the specified cluster or server.</td>
</tr>
<tr>
<td>GetSSOTokenUsingWebSecurityAgentCredentials</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetWebSecurityAgentSettings</td>
<td>For internal use.</td>
</tr>
<tr>
<td>IsValidApplication</td>
<td>Indicates whether the specified application exists on the specified cluster.</td>
</tr>
<tr>
<td>KillUsers</td>
<td>Logs off users from a cluster.</td>
</tr>
<tr>
<td>ModifyApplication</td>
<td>For internal use.</td>
</tr>
<tr>
<td>OpenApplication</td>
<td>Opens an application. OpenApplication also sets object references to the HsxServer and HsvSession objects.</td>
</tr>
<tr>
<td>RegisterApplicationCAS</td>
<td>Registers a Classic application with Shared Services.</td>
</tr>
<tr>
<td>RegisterApplicationCASWithAccessCode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>RegisterCluster</td>
<td>Registers a cluster or application server for a client.</td>
</tr>
</tbody>
</table>
### Method Description

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RegisterServer</td>
<td>Deprecated - use RegisterCluster.</td>
</tr>
<tr>
<td>ScriptableEnumRegisteredClusterNames</td>
<td>For internal use.</td>
</tr>
<tr>
<td>ScriptableEnumRegisteredServerNames</td>
<td>For internal use.</td>
</tr>
<tr>
<td>ScriptableGetLogonInfoSSO</td>
<td>For internal use.</td>
</tr>
<tr>
<td>ScriptableOpenApplication</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetApplicationFolder</td>
<td>Sets an application’s local storage folder.</td>
</tr>
<tr>
<td>SetLogonInfo</td>
<td>Deprecated - use SetLogonInfoSSO.</td>
</tr>
<tr>
<td>SetLogonInfoSSO</td>
<td>Sets logon information such as the username, password, domain or sets an SSO token.</td>
</tr>
<tr>
<td>UnregisterAllClusters</td>
<td>Unregisters all the clusters or application servers that are registered for the client.</td>
</tr>
<tr>
<td>UnregisterCluster</td>
<td>Unregisters the specified cluster or application server.</td>
</tr>
<tr>
<td>UnregisterServer</td>
<td>Deprecated - use UnregisterCluster.</td>
</tr>
<tr>
<td>WarnUsersForShutDown</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>


### HsxClientUI Type Library Overview

The HsxClientUI type library contains 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:

```vba
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. See “HsxClient Type Library Overview” on page 59.

The following table lists the HsxClientUI object’s methods.

#### Table 5  HsxClientUI Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteApplication</td>
<td>Displays the Delete Application dialog box.</td>
</tr>
</tbody>
</table>
Method | Description
---|---
GetServer | For internal use.
Initialize | Provides the HsxClientUI object with access to the client layer. You must call `Initialize` before calling any of the other HsxClientUI methods.
Logon | Displays the Logon dialog box.
OpenApplication | Displays the Open Application dialog box.
RegisterServer | Displays the Register Server dialog box.

See “HsxClientUI Object Methods” on page 181.

**HsxServer Type Library Overview**

The HsxServer type library contains 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 64 describes the HsxServer object’s one property.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteSystemErrors</td>
<td>For internal use.</td>
</tr>
<tr>
<td>DeleteSystemErrorsEx</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EnumDataSources</td>
<td>Returns the names and descriptions of the applications on an application server.</td>
</tr>
<tr>
<td>EnumRegisteredDSNs</td>
<td>Returns an array of the registered Data Source Names on the application server.</td>
</tr>
<tr>
<td>GetClustersAndServers</td>
<td>Returns the names of the clusters and servers associated with an HsxServer object reference.</td>
</tr>
<tr>
<td>GetDataSource</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetFileTransfer</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetHFMErrLogRecordSet</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetSystemDataLinkFile</td>
<td>Returns the name and path of the data link file for the application server.</td>
</tr>
<tr>
<td>GetSystemFolder</td>
<td>Returns the name and path of the system file for the application server.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetXMLErrorFromDatabase</td>
<td>Returns the message for a system error, using the system error’s reference number.</td>
</tr>
<tr>
<td>GetXMLErrorsListFromDatabase</td>
<td>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.</td>
</tr>
<tr>
<td>ScriptableEnumDataSources</td>
<td>For internal use.</td>
</tr>
<tr>
<td>ScriptableEnumRegisteredDSNs</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

See “HsxServer Object Methods” on page 185.

**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 the HsvSession object. The HsvSession object provides properties for accessing subordinate objects, such as the objects used for accessing metadata, data, journals, and security. 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. See “OpenApplication” on page 176.

The following table lists the properties of the HsvSession object, and Table 8 lists the object’s methods.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Calculate</td>
<td>Returns an HsvCalculate object reference.</td>
</tr>
<tr>
<td>Data</td>
<td>Returns an HsvData object reference.</td>
</tr>
<tr>
<td>ICM</td>
<td>Returns an object reference to the HsvICM object or the IHsvAdminICM interface.</td>
</tr>
<tr>
<td>Metadata</td>
<td>Returns an HsvMetadata object reference.</td>
</tr>
<tr>
<td>Reports</td>
<td>Returns an HsvReports object reference.</td>
</tr>
<tr>
<td>SystemInfo</td>
<td>Returns an HsvSystemInfo object reference.</td>
</tr>
</tbody>
</table>
The HsvSession object also contains the following methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateObject</td>
<td>Instantiates an object on the application server on which an application was opened. Use CreateObject to instantiate server-side objects that you develop.</td>
</tr>
<tr>
<td>GetEcid</td>
<td>Returns the Execution Context ID (ECID) for ODL logging.</td>
</tr>
<tr>
<td>get_HsvDQI</td>
<td>Returns the DQI object reference from a valid session that can be used to create, define, retrieve data and status in form of Grids from an application.</td>
</tr>
<tr>
<td>GetLicenseExpirationStatus</td>
<td>Deprecated. For internal use.</td>
</tr>
<tr>
<td>HasSystemChanged</td>
<td>Indicates whether an application's system information changed in a way that might require a consolidation, calculation, or translation to be run. For example, HasSystemChanged returns TRUE if a metadata file was loaded after the last consolidation.</td>
</tr>
<tr>
<td>HasUserStatusChanged</td>
<td>Indicates whether the connected user was logged off by an administrator.</td>
</tr>
<tr>
<td>IsBusy</td>
<td>For internal use.</td>
</tr>
<tr>
<td>IsRunningTasks</td>
<td>For internal use.</td>
</tr>
<tr>
<td>LockMetadataLoadWithSystemChangeCheck</td>
<td>For internal use.</td>
</tr>
<tr>
<td>UnlockMetadataLoad</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

**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

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

**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; see “Metadata” on page 192.

The following table lists the HsvMetadata object’s properties, and Table 11 lists the object’s methods.

### Table 10  HsvMetadata Object Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accounts</td>
<td>Sets object references to the HsvAccounts object, and also sets references to the IHsvTreeInfo interface for use with the Account dimension.</td>
</tr>
<tr>
<td>Currencies</td>
<td>Sets object references to the HsvCurrencies object.</td>
</tr>
</tbody>
</table>
### Property

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


The following table lists the HsvMetadata object’s methods:

### Table 11  HsvMetadata Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnumCellTextLabels</td>
<td>Enumerates the cell text labels for an application.</td>
</tr>
<tr>
<td>EnumConsolidationMethodIDs</td>
<td>Returns IDs that identify an application’s consolidation methods.</td>
</tr>
<tr>
<td>EnumCustomDimsForApp</td>
<td>Enumerates the valid custom dimensions for an application.</td>
</tr>
<tr>
<td>EnumCustomDimsForAppEx</td>
<td>Enumerates the valid custom dimensions and dimension size for an application.</td>
</tr>
<tr>
<td>EnumDimProperties</td>
<td>Extracts property IDs, names and types for a dimension.</td>
</tr>
<tr>
<td>EnumExtractOptions</td>
<td>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.</td>
</tr>
<tr>
<td>EnumLanguages</td>
<td>Returns the numeric IDs and labels of an application's languages.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnumLoadOptions</td>
<td>Returns a two-dimensional array of the metadata load options that can be passed to Load.</td>
</tr>
<tr>
<td>EnumMemberProperties</td>
<td>Extracts member property IDs and values for a dimension member.</td>
</tr>
<tr>
<td>Extract</td>
<td>Extracts metadata into a text file that is created on the application server.</td>
</tr>
<tr>
<td>ExtractModuleConfigurations</td>
<td>Extracts the definition file for module configurations.</td>
</tr>
<tr>
<td>GetApplicationAttribute</td>
<td>Deprecated - use GetApplicationAttributeExtDim.</td>
</tr>
<tr>
<td>GetApplicationAttributeExtDim</td>
<td>Returns the raw value of an application setting attribute.</td>
</tr>
<tr>
<td>GetApplicationCurrency</td>
<td>Returns the label of the application’s default currency.</td>
</tr>
<tr>
<td>GetApplicationSettingsTimeStamp</td>
<td>Returns a timestamp that indicates when the application settings were last updated.</td>
</tr>
<tr>
<td>GetApplicationProfileInfo</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetByIndexValidationAccount</td>
<td>Returns the member ID of a validation account, using the index of the account.</td>
</tr>
<tr>
<td>GetCellLevelAccountType</td>
<td>Deprecated - use GetCellLevelAccountTypeExtDim.</td>
</tr>
<tr>
<td>GetCellLevelAccountTypeExtDim</td>
<td>Returns the account type of a cell, using the member IDs of the cell’s Account and Custom dimension members.</td>
</tr>
<tr>
<td>GetCellTextLabelFromLabel</td>
<td>Retrieves the ID associated with a cell text label.</td>
</tr>
<tr>
<td>GetCellTextIdsFromLabels</td>
<td>Retrieves the 2-D array of cell text label IDs from 2-D array of cell text labels.</td>
</tr>
<tr>
<td>GetCellTextLabelDescription</td>
<td>Retrieves the description for a specified cell text ID and language.</td>
</tr>
<tr>
<td>GetCellTextLabelFromId</td>
<td>Retrieves the cell text label for a specified cell text ID.</td>
</tr>
<tr>
<td>GetConsolidationMethodDescription</td>
<td>Returns a consolidation method’s description for the specified language.</td>
</tr>
<tr>
<td>GetConsolidationMethodInfo</td>
<td>Returns the attributes of a consolidation method, using the method’s ID.</td>
</tr>
<tr>
<td>GetConsolidationMethodsTimeStamp</td>
<td>Returns a timestamp that indicates when the application’s consolidation methods were last updated.</td>
</tr>
<tr>
<td>GetCurrencyTimeStamp</td>
<td>Returns a timestamp that indicates when the application’s currencies were last updated.</td>
</tr>
<tr>
<td>GetCurrencyValueIDForEntityValueCombination</td>
<td>Returns the currency value dimension ID for the specified parent, entity, and value triplet.</td>
</tr>
<tr>
<td>GetCurrencyValueIDsForEntityValueCombinations</td>
<td>Returns the currency value dimension IDs for the specified parent, entity, and value triplet.</td>
</tr>
<tr>
<td>GetCustomX</td>
<td>Retrieves the custom dimension object for the specified custom dimension ID.</td>
</tr>
<tr>
<td>GetDefaultScaleAndNumDecimal</td>
<td>Returns the default value for the scale setting and number of decimal places.</td>
</tr>
<tr>
<td>GetDefaultValueOfActiveStatusAccount</td>
<td>Returns the default value for an application’s active status account.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetDimension</td>
<td>Returns an object reference to the specified dimension.</td>
</tr>
<tr>
<td>GetDimensionIdFromName</td>
<td>Converts a dimension name to its associated ID.</td>
</tr>
<tr>
<td>GetFdmAppName</td>
<td>Returns the value of the FDM Application Name setting for the application.</td>
</tr>
<tr>
<td>GetFrequencyID</td>
<td>The frequency (YTD, MTD, QTD, and so forth) for which to return an internal numeric ID.</td>
</tr>
<tr>
<td>GetICPEntitiesAggregationWeight</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetICPEntitiesAggregationWeightEx</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetMaxCustomDimNumber</td>
<td>Returns the maximum custom dimension number for the application, in the range 1-n.</td>
</tr>
<tr>
<td>GetMaxDimId</td>
<td>Retrieves the maximum valid dimension ID for an application.</td>
</tr>
<tr>
<td>GetNumDimensions</td>
<td>Retrieves the total number of dimensions in the application.</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForAccountFlag</td>
<td>Indicates whether phased submissions are enabled for the Account dimension.</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustom1Flag</td>
<td>Deprecated - use GetSupportSubmissionPhaseForCustomXFlag.</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustom2Flag</td>
<td>Deprecated - use GetSupportSubmissionPhaseForCustomXFlag.</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustom3Flag</td>
<td>Deprecated - use GetSupportSubmissionPhaseForCustomXFlag.</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustom4Flag</td>
<td>Deprecated - use GetSupportSubmissionPhaseForCustomXFlag.</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForCustomXFlag</td>
<td>Indicates whether phased submissions are enabled for the Custom dimension.</td>
</tr>
<tr>
<td>GetSupportSubmissionPhaseForICPFlag</td>
<td>Indicates whether phased submissions are enabled for the Intercompany Partner dimension.</td>
</tr>
<tr>
<td>GetUseSubmissionPhaseFlag</td>
<td>Indicates whether phased submissions are enabled for the application.</td>
</tr>
<tr>
<td>GetValidationAccount</td>
<td>Returns the member ID of an application’s Validation Account setting.</td>
</tr>
<tr>
<td>GetValidationAccountMembers</td>
<td>Returns all of the validation Account members for an application.</td>
</tr>
<tr>
<td>IsCustomDimValidForApp</td>
<td>Indicates whether a Custom dimension is valid for an application.</td>
</tr>
<tr>
<td>IsCustomMemberValidForAccount</td>
<td>Indicates whether a Custom dimension member is valid for an account.</td>
</tr>
<tr>
<td>IsOrgByPeriodApplication</td>
<td>Indicates whether the OrgByPeriodApplication application setting is on or off.</td>
</tr>
<tr>
<td>Load</td>
<td>Loads metadata into a Classic application, using a load file on the application server.</td>
</tr>
<tr>
<td>LoadModuleConfigurations</td>
<td>Loads the definition file for the module configurations.</td>
</tr>
<tr>
<td>LoadWithAccessCode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>TranslateApplicationAttributeForDisplay</td>
<td>Deprecated - use TranslateApplicationAttributeForDisplayExtDim.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>TranslateApplicationAttributeForDisplayExtDim</td>
<td>Returns a user-readable string that represents the value of an application setting attribute.</td>
</tr>
</tbody>
</table>

See “HsvMetadata Object Methods” on page 198.

**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; see “Accounts” on page 230.

The following table lists the HsvAccounts object’s methods.

### Table 12 HsvAccounts Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAccountType</td>
<td>Indicates an account’s type.</td>
</tr>
<tr>
<td>GetCalcAttribute</td>
<td>Returns the value of an Account dimension member’s CalcAttribute attribute.</td>
</tr>
<tr>
<td>GetICPTopMember</td>
<td>Returns the member ID of the Intercompany Partner dimension member assigned as an Account dimension member’s ICPTopMember attribute.</td>
</tr>
<tr>
<td>GetIsICP</td>
<td>Indicates the value assigned to the IsICP attribute of an Account dimension member.</td>
</tr>
<tr>
<td>GetNumDecimalPlaces</td>
<td>Returns the maximum number of digits to the right of the decimal point that an account supports.</td>
</tr>
<tr>
<td>GetPlugAccount</td>
<td>Returns the member ID of an account’s plug account.</td>
</tr>
<tr>
<td>GetSecurityClassID</td>
<td>Returns the ID of the security class assigned to an account, using the account’s member ID.</td>
</tr>
<tr>
<td>GetSubmissionGroup</td>
<td>Returns the value of an account’s Submission Group property.</td>
</tr>
<tr>
<td>GetTopMemberOfValidCustom1Hierarchy</td>
<td>Returns the member ID of the top member of an account’s Custom 1 dimension hierarchy.</td>
</tr>
<tr>
<td>GetTopMemberOfValidCustom2Hierarchy</td>
<td>Returns the member ID of the top member of an account’s Custom 2 dimension hierarchy.</td>
</tr>
<tr>
<td>GetTopMemberOfValidCustom3Hierarchy</td>
<td>Returns the member ID of the top member of an account’s Custom 3 dimension hierarchy.</td>
</tr>
<tr>
<td>GetTopMemberOfValidCustom4Hierarchy</td>
<td>Returns the member ID of the top member of an account’s Custom 4 dimension hierarchy.</td>
</tr>
<tr>
<td>GetUserDefined1</td>
<td>Returns an account’s UserDefined1 attribute, using the account’s member ID.</td>
</tr>
<tr>
<td>GetUserDefined2</td>
<td>Returns an account’s UserDefined2 attribute, using the account’s member ID.</td>
</tr>
<tr>
<td>GetUserDefined3</td>
<td>Returns an account’s UserDefined3 attribute, using the account’s member ID.</td>
</tr>
<tr>
<td>GetXBRLTags</td>
<td>Returns an account’s XBRLTags attribute, using the account’s member ID.</td>
</tr>
<tr>
<td>IsCalculated</td>
<td>Indicates whether an account’s data is calculated by Financial Management or is manually entered.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsConsolidated</td>
<td>Returns a Boolean that indicates whether an account’s data is consolidated to parent entities.</td>
</tr>
<tr>
<td>IsCustom1AggregationEnabled</td>
<td>Returns the value to which an account’s EnableCustom1Aggr attribute is set.</td>
</tr>
<tr>
<td>IsCustom2AggregationEnabled</td>
<td>Returns the value to which an account’s EnableCustom2Aggr attribute is set.</td>
</tr>
<tr>
<td>IsCustom3AggregationEnabled</td>
<td>Returns the value to which an account’s EnableCustom3Aggr attribute is set.</td>
</tr>
<tr>
<td>IsCustom4AggregationEnabled</td>
<td>Returns the value to which an account’s EnableCustom4Aggr attribute is set.</td>
</tr>
<tr>
<td>IsICP</td>
<td>Returns a Boolean that indicates whether the account is used in intercompany transactions.</td>
</tr>
<tr>
<td>IsICPRestricted</td>
<td>Indicates whether an Account dimension member is restricted from having Intercompany Partner transactions with itself.</td>
</tr>
<tr>
<td>UsesLineItems</td>
<td>Indicates whether an account allows line items.</td>
</tr>
</tbody>
</table>

See “HsvAccounts Object Methods” on page 275.

**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; see “HsvCustom Object Methods” on page 287.

The following table lists the methods of the HsvCustom object.

<table>
<thead>
<tr>
<th>Table 13 HsvCustom Object Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>GetAggregationWeight</td>
</tr>
<tr>
<td>GetSecurityClassID</td>
</tr>
<tr>
<td>GetSubmissionGroup</td>
</tr>
<tr>
<td>GetUserDefined1</td>
</tr>
<tr>
<td>GetUserDefined2</td>
</tr>
<tr>
<td>GetUserDefined3</td>
</tr>
<tr>
<td>IsCalculated</td>
</tr>
<tr>
<td>IsSwitchSignEnabledForFlow</td>
</tr>
<tr>
<td>IsSwitchTypeEnabledForFlow</td>
</tr>
</tbody>
</table>
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; see “Entities” on page 232.

The following table lists the methods of the HsvEntities object.

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

See “HsvEntities Object Methods” on page 291.

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; see “ICPs” on page 232.

The following table lists the methods of the HsvICPs object.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetSecurityClassID</td>
<td>Returns the ID of the security class assigned to an Intercompany Partner.</td>
</tr>
<tr>
<td>GetSubmissionGroup</td>
<td>Returns the value of an Intercompany Partner dimension member’s Submission Group property.</td>
</tr>
</tbody>
</table>
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; see “Periods” on page 232.

The following table lists the methods of the HsvPeriods object.

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

See “HsvPeriods Object Methods” on page 299.

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; see “Scenarios” on page 232.

The following table lists the methods of the HsvScenarios object.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetPhasedSubmissionStartYear</td>
<td>Returns the Phased Submission start year for the scenario.</td>
</tr>
<tr>
<td>EnumPhasedSubmissionStartYears</td>
<td>Returns and array of Scenario IDs and their corresponding Phased Submission start years.</td>
</tr>
<tr>
<td>GetDefaultFrequency</td>
<td>Returns the number that identifies the default frequency of a Scenario dimension member.</td>
</tr>
<tr>
<td>GetDefaultView</td>
<td>Returns the number that identifies the default view of a Scenario dimension member.</td>
</tr>
<tr>
<td>GetMaximumReviewLevel</td>
<td>Returns the value assigned to the MaximumReviewLevel attribute of a Scenario dimension member.</td>
</tr>
<tr>
<td>GetMissingDataZeroViewForAdjValues</td>
<td>Retums the member ID of the View dimension member assigned as the ZeroViewForAdj attribute of a specified Scenario dimension member.</td>
</tr>
<tr>
<td>GetMissingDataZeroViewForNonAdjValues</td>
<td>Retums the member ID of the View dimension member assigned as the ZeroViewForNonadj attribute of a specified Scenario dimension member.</td>
</tr>
<tr>
<td>GetSecurityClassID</td>
<td>Returns the ID of the security class assigned to a scenario, using the scenario's member ID.</td>
</tr>
</tbody>
</table>
### 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. See “HsvValues Object Methods” on page 311.

#### Table 18  HsvValues Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCurrencyIDFromValueID</td>
<td>Returns a currency ID for a Value dimension member ID.</td>
</tr>
<tr>
<td>GetValueIDFromCurrencyID</td>
<td>Returns the member ID of a Value dimension member that corresponds to a currency.</td>
</tr>
</tbody>
</table>

### 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; see “Views” on page 233.

### HsvYears Object Overview

The HsvYears object provides the `GetYearRange` method, which returns an application’s valid range of years. See “HsvYears Object Methods” on page 312.
Assign HsvYears object references with the `Years` property; see “Years” on page 233.

**IHsvTreeInfo Interface Overview**

The IHsvTreeInfo interface is implemented in all of the dimension objects listed in Table 9 on page 66. 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 66. See “IHsvTreeInfo Interface Methods” on page 233.

The following table summarizes the IHsvTreeInfo interface’s methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnumAllMemberIDs</td>
<td>Returns numeric IDs that identify all the members of a dimension.</td>
</tr>
<tr>
<td>EnumAllMemberLabels</td>
<td>Returns the labels of all the members of a dimension.</td>
</tr>
<tr>
<td>EnumAllParentAndChildIDs</td>
<td>Returns arrays of IDs that represent the parent-child relationships of a dimension’s members.</td>
</tr>
<tr>
<td>EnumAllParentAndChildLabels</td>
<td>Returns arrays that contain member labels and that represent the parent-child relationships of a dimension’s members.</td>
</tr>
<tr>
<td>EnumAncestors</td>
<td>Returns an array containing the member IDs of a member’s ancestors.</td>
</tr>
<tr>
<td>EnumBaseMemberIDs</td>
<td>Returns an array containing the member IDs of a parent’s base-level members. You can also use EnumBaseMemberIDs to get all of a dimension’s base-level members.</td>
</tr>
<tr>
<td>EnumDefaultAncestors</td>
<td>Returns the member IDs of a member’s default ancestors.</td>
</tr>
<tr>
<td>EnumDefaultAncestorsLabels</td>
<td>Returns the labels of a member’s default ancestors.</td>
</tr>
<tr>
<td>EnumDescendants</td>
<td>Returns an array containing the member IDs of a member’s descendants.</td>
</tr>
<tr>
<td>EnumIDsOfChildren</td>
<td>Returns either the top members of a dimension hierarchy or the child members of a parent member.</td>
</tr>
<tr>
<td>EnumIDsOfChildren</td>
<td>Returns children and the parent item of the requested children.</td>
</tr>
<tr>
<td>EnumMemberLists</td>
<td>Returns the names of the member lists for a dimension.</td>
</tr>
<tr>
<td>EnumMembers</td>
<td>Returns an array containing the member IDs of the dimension members in either the default dimension hierarchy or in a member list.</td>
</tr>
<tr>
<td>EnumMembers2</td>
<td>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 Scenario, Year, Period, and Entity dimension members.</td>
</tr>
<tr>
<td>EnumMembersWithAttribValue</td>
<td>Enumerates the attribute IDs and corresponding labels for an attribute.</td>
</tr>
<tr>
<td>EnumParents</td>
<td>Returns an array containing the member IDs of a member’s parent members.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnumSortedIDsOfChildren</td>
<td>Returns an array containing the member IDs an Intercompany Partner dimension member's children, with the members sorted according to the specified sorting criteria.</td>
</tr>
<tr>
<td>EnumSortedIDsofChildren</td>
<td>Returns Sorted children and the parent item of the specified children.</td>
</tr>
<tr>
<td>EnumSortedMembers</td>
<td>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.</td>
</tr>
<tr>
<td>Find</td>
<td>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.</td>
</tr>
<tr>
<td>FindByDesc</td>
<td>Returns the member IDs of those members of a member list with descriptions that match a search string in a language.</td>
</tr>
<tr>
<td>FindMatchingMembersFromHierarchy</td>
<td>Returns the member IDs of the members with labels that match a search string.</td>
</tr>
<tr>
<td>FindMatchingMembersFromHierarchyByDesc</td>
<td>Returns the member IDs of the members with descriptions in a language that match a search string.</td>
</tr>
<tr>
<td>FindMatchingMembersFromHierarchyWildCard</td>
<td>Returns the member IDs of the members with labels or descriptions that match a search string; the search string can include wildcard characters.</td>
</tr>
</tbody>
</table>
### Method Description

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetMemberListID</td>
<td>Returns the numeric ID of a member list.</td>
</tr>
<tr>
<td>GetMemberListName</td>
<td>Returns the name of a member list for which you have the ID number.</td>
</tr>
<tr>
<td>GetNumBaseMembers</td>
<td>Returns a count of the base-level members beneath a dimension member.</td>
</tr>
<tr>
<td>GetNumChildren</td>
<td>Returns the number of children that are one level beneath a dimension member in a dimension hierarchy.</td>
</tr>
<tr>
<td>GetNumDescendants</td>
<td>Returns the number of descendants beneath a member in a dimension's hierarchy, using the member’s ID.</td>
</tr>
<tr>
<td>GetNumMembers</td>
<td>Returns the number of members in a dimension.</td>
</tr>
<tr>
<td>GetNumParents</td>
<td>Returns the number of parents for a dimension member, using the member’s ID.</td>
</tr>
<tr>
<td>GetQualifiedLabel</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetTreeCapabilities</td>
<td>Returns various properties of a dimension.</td>
</tr>
<tr>
<td>GetTreeName</td>
<td>Returns the name of the dimension to which the IHsvTreeInfo interface is set.</td>
</tr>
<tr>
<td>GetTreeTimeStamp</td>
<td>Returns a timestamp that indicates when the dimension was last updated.</td>
</tr>
<tr>
<td>HasChildren</td>
<td>Indicates whether a dimension member has child members.</td>
</tr>
<tr>
<td>IsMemberABaseOf</td>
<td>Indicates whether a member is in a base-level position beneath another member in a dimension hierarchy.</td>
</tr>
<tr>
<td>IsMemberAChildOf</td>
<td>Indicates whether one member is a child of another member.</td>
</tr>
<tr>
<td>IsMemberADescendantOf</td>
<td>Indicates whether one member is a descendant beneath another member in a dimension hierarchy.</td>
</tr>
<tr>
<td>SortMembersBasedOnList</td>
<td>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.</td>
</tr>
<tr>
<td>TranslateAttributeValueForDisplay</td>
<td>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.</td>
</tr>
</tbody>
</table>

See “IHsvTreeInfo Interface Methods” on page 233.

### 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. See “HsvCurrencies Object Methods” on page 313.
### HsvCurrencies Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EnumCurrencies</td>
<td>Returns arrays containing the currency IDs and labels of the application’s</td>
</tr>
<tr>
<td></td>
<td>currencies.</td>
</tr>
<tr>
<td>EnumCurrencies2</td>
<td>Returns arrays containing the currency IDs and labels of the application’s</td>
</tr>
<tr>
<td></td>
<td>currencies, with the option to return only those currencies for which the</td>
</tr>
<tr>
<td></td>
<td>DisplayInICT attribute is enabled.</td>
</tr>
<tr>
<td>GetCurrencyDescription</td>
<td>Returns a currency’s description in a language.</td>
</tr>
<tr>
<td>GetCurrencyID</td>
<td>Returns the currency ID of a currency.</td>
</tr>
<tr>
<td>GetCurrencyLabel</td>
<td>Returns a currency label, using a currency ID.</td>
</tr>
<tr>
<td>GetCurrencyTranslationOperator</td>
<td>Returns the conversion operator for a currency.</td>
</tr>
<tr>
<td>GetScale</td>
<td>Returns the scale of a currency.</td>
</tr>
</tbody>
</table>

### 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:

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

The following table lists the methods of the HsvData object.

### HsvData Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddDataToMDDataBuffer</td>
<td>Deprecated - use AddDataToMDDataBufferExtDim.</td>
</tr>
<tr>
<td>AddDataToMDDataBufferExtDim</td>
<td>Adds an application’s cell to an HsvMDDataBuffer or HsvMDDataBufferLite object. Any data, description, or line items for the cell are added.</td>
</tr>
<tr>
<td>AttachDocumentToCell</td>
<td>Deprecated - use AttachDocumentToCellExtDim.</td>
</tr>
<tr>
<td>AttachDocumentToCellExtDim</td>
<td>Attaches a previously loaded document to a cell.</td>
</tr>
<tr>
<td>ClearAllData</td>
<td>Deletes all data from an application.</td>
</tr>
<tr>
<td>ClearAllDescriptionsInSubCube</td>
<td>Removes the cell text from all the cells in a subcube.</td>
</tr>
<tr>
<td>ClearDataAuditItems</td>
<td>Deletes the audit history for all data changes that occurred before a date and time.</td>
</tr>
<tr>
<td>ClearDataAuditItemsEx</td>
<td>Deletes the audit history for all data changes that occurred before a date and time.</td>
</tr>
<tr>
<td>ClearInputData</td>
<td>Deletes data from some or all of a subcube’s cells.</td>
</tr>
<tr>
<td>ClearInvalidData</td>
<td>Scans for or deletes invalid records.</td>
</tr>
<tr>
<td>CopyInputData</td>
<td>Deprecated - use CopyInputDataForMultiple Entities.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CopyInputDataForMultipleEntities</td>
<td>Copies data for one or more entities from one set of cells to another set of cells.</td>
</tr>
<tr>
<td>DeleteLineItemDetails</td>
<td>Deprecated - use DeleteLineItemDetailsExtDim.</td>
</tr>
<tr>
<td>DeleteLineItemDetailsExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>DetachDocumentFromCell</td>
<td>Deprecated - use DetachDocumentFromCellExtDim.</td>
</tr>
<tr>
<td>DetachDocumentFromCellExtDim</td>
<td>Detaches a document from a cell.</td>
</tr>
<tr>
<td>DoCellDescriptionsExist</td>
<td>Indicates whether one or more of the cells for a combination of Scenario, Year, and Entity dimension members contain cell text descriptions.</td>
</tr>
<tr>
<td>DoesCellDescriptionExist</td>
<td>Deprecated - use DoesCellDescriptionExistExtDim.</td>
</tr>
<tr>
<td>DoesCellDescriptionExistExtDim</td>
<td>Indicates whether a cell contains cell text.</td>
</tr>
<tr>
<td>DoesDataExist</td>
<td>Indicates whether any data exists in one or more cells.</td>
</tr>
<tr>
<td>DoesSparseDataExist</td>
<td>Deprecated - use DoesSparseDataExistExtDim.</td>
</tr>
<tr>
<td>DoesSparseDataExistExtDim</td>
<td>Indicates whether sparse data exists for the specified cell.</td>
</tr>
<tr>
<td>EnumDataAuditItems</td>
<td>Deprecated - use EnumDataAuditItemsExtDim.</td>
</tr>
<tr>
<td>EnumDataAuditItemsExtDim</td>
<td>Returns data audit information from a range of audit records that meet the specified filtering criteria.</td>
</tr>
<tr>
<td>EnumDataAuditItems2</td>
<td>Deprecated - use EnumDataAuditItems2ExtDim.</td>
</tr>
<tr>
<td>EnumDataAuditItems2ExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EnumEntitiesWithDataForScenarioYear</td>
<td>Returns a variant array of Entity ID’s that have data for a scenario and year.</td>
</tr>
<tr>
<td>EnumExtractOptions</td>
<td>Returns a two-dimensional array of the data extract options that can be passed to HsvData.Extract. The array includes the options' names and default values.</td>
</tr>
<tr>
<td>EnumLoadOptions</td>
<td>Returns a two-dimensional array of the data load options that can be passed to HsvData.Load. The array includes the options' names and default values.</td>
</tr>
<tr>
<td>Extract</td>
<td>Extracts data into a text file that is created on the application server.</td>
</tr>
<tr>
<td>ExtractDataAuditItems</td>
<td>Deprecated - use ExtractDataAuditItemsExtDim.</td>
</tr>
<tr>
<td>ExtractDataAuditItemsExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>ExtractDrillableRegions</td>
<td>Retrieves all ERPI URL definitions.</td>
</tr>
<tr>
<td>ExtractDrillableRegionsByURLNames</td>
<td>Retrieves only the URLs that appear in the input array varabstrURLNames.</td>
</tr>
<tr>
<td>FilterMembersThatHaveData</td>
<td>Deprecated - use FilterMembersThatHaveDataExtDim.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>FilterMembersThatHaveDataExtDim</td>
<td>Returns a list of members for the specified dimension for which data exists for the specified POVs.</td>
</tr>
<tr>
<td>FilterMembersThatHaveData2</td>
<td>Deprecated - use FilterMembersThatHaveData2ExtDim.</td>
</tr>
<tr>
<td>FilterMembersThatHaveData2ExtDim</td>
<td>For a specified dimension, returns the filtered of member IDs that have data for any of the specified POVs.</td>
</tr>
<tr>
<td>FormatNumberToText</td>
<td>Returns a number as a String. FormatNumberToText scales the number and puts a fixed number of digits to the right of the decimal point.</td>
</tr>
<tr>
<td>FormatNumberToText2</td>
<td>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.</td>
</tr>
<tr>
<td>FormatStatusToText</td>
<td>Returns a string description of the numeric calculation status passed as the argument.</td>
</tr>
<tr>
<td>FormatStoredNumberToText</td>
<td>Converts a number passed as a Double to a String, formatting the String with the decimal and thousands delimiters that you specify.</td>
</tr>
<tr>
<td>FormatTextToNumber</td>
<td>Returns a number stored in a String as a Double. FormatTextToNumber also scales the returned number.</td>
</tr>
<tr>
<td>FormatTextToStoredNumber</td>
<td>Converts a number passed as a String to a Double.</td>
</tr>
<tr>
<td>GetAllDescriptionsInSubCube</td>
<td>Deprecated - use GetAllDescriptionsInSubCubeExtDim.</td>
</tr>
<tr>
<td>GetAllDescriptionsInSubCubeExtDim</td>
<td>Returns an array containing the cell text descriptions of one or more cells that have the same Scenario, Year, and Value dimension members.</td>
</tr>
<tr>
<td>GetAllURLNames</td>
<td>Retrieves the full list of URL names currently in Financial Management.</td>
</tr>
<tr>
<td>GetAttachedDocumentsToCell</td>
<td>Deprecated - use GetAttachedDocumentsToCellExtDim.</td>
</tr>
<tr>
<td>GetAttachedDocumentsToCellExtDim</td>
<td>Returns the names and paths of the documents attached to a cell.</td>
</tr>
<tr>
<td>GetBaseDataForAccount</td>
<td>Deprecated - use GetBaseDataForAccountExtDim.</td>
</tr>
<tr>
<td>GetBaseDataForAccountExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetCalcStatus</td>
<td>Returns the calculation status of a subcube.</td>
</tr>
<tr>
<td>GetCalcStatusStatistics</td>
<td>Returns arrays of flags that indicate which calculation statuses apply to the specified entities and periods of a subcube.</td>
</tr>
<tr>
<td>GetCell</td>
<td>Deprecated - use GetCellExtDim.</td>
</tr>
<tr>
<td>GetCellExtDim</td>
<td>Returns the data in a cell, as well as the cell’s status. The data is returned as a Double.</td>
</tr>
<tr>
<td>GetCellDescription</td>
<td>Deprecated - use GetCellDescriptionExtDim.</td>
</tr>
<tr>
<td>GetCellDescriptionExtDim</td>
<td>Returns the cell text description of a cell.</td>
</tr>
<tr>
<td>GetCellDescriptions</td>
<td>Deprecated - use GetCellDescriptionsExtDim.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetCellDescriptionsExtDim</td>
<td>Returns the cell text descriptions of one or more cells.</td>
</tr>
<tr>
<td>GetCellHistory</td>
<td>Deprecated - use GetCellHistoryExtDim.</td>
</tr>
<tr>
<td>GetCellHistoryExtDim</td>
<td>Returns the audit history of a cell’s data changes, with cell values returned as Doubles.</td>
</tr>
<tr>
<td>GetCellHistory2</td>
<td>Deprecated - use GetCellHistory2ExtDim.</td>
</tr>
<tr>
<td>GetCellHistory2ExtDim</td>
<td>Returns the audit history of a cell’s data changes, with cell values returned as formatted strings.</td>
</tr>
<tr>
<td>GetCellJournalEntriesExtDim</td>
<td>Returns arrays containing the data in and IDs of the journal entries for a cell.</td>
</tr>
<tr>
<td>GetCellLineItems</td>
<td>Deprecated - use GetCellLineItemsExtDim.</td>
</tr>
<tr>
<td>GetCellLineItemsExtDim</td>
<td>Returns arrays of the data and descriptions for the specified cell’s line items. Data is returned in a Double array.</td>
</tr>
<tr>
<td>GetCells</td>
<td>Deprecated - use GetCellsExtDim.</td>
</tr>
<tr>
<td>GetCellsExtDim</td>
<td>Returns the data and statuses of cells. The cells’ data is returned as a Double array.</td>
</tr>
<tr>
<td>GetCellTextAndAttachment</td>
<td>Returns cell text and attachments for a cell.</td>
</tr>
<tr>
<td>GetCellTextLabelsAndFieldsForPOV</td>
<td>Returns cell text label IDs, labels, cell text, and attachment indicators for a POV.</td>
</tr>
<tr>
<td>GetCellTextLabelsAndFieldsForSlice</td>
<td>Returns cell text label IDs, labels, cell text, and attachment indicators for a multiple POVs.</td>
</tr>
<tr>
<td>GetCellsWithRowSuppression</td>
<td>Deprecated - use GetCellsWithRowSuppressionExtDim.</td>
</tr>
<tr>
<td>GetCellsWithRowSuppressionExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>GetCellsWithRowSuppression2</td>
<td>Deprecated - use GetCellsWithRowSuppression2ExtDim.</td>
</tr>
<tr>
<td>GetCellsWithRowSuppression2ExtDim</td>
<td>Returns the data and statuses of cells, optionally excluding rows of cells that match specified criteria.</td>
</tr>
<tr>
<td>GetCountOfAttachedDocumentsToCell</td>
<td>Deprecated - use GetCountOfAttachedDocumentsToCellExtDim.</td>
</tr>
<tr>
<td>GetCountOfAttachedDocumentsToCellExtDim</td>
<td>Returns a count of the documents attached to a cell.</td>
</tr>
<tr>
<td>GetCurrencyCube</td>
<td>Returns an object reference to the HsvCurrencyCube object. The object reference provides access to the subcube identified by the member IDs passed to GetCurrencyCube.</td>
</tr>
<tr>
<td>GetDataForAllMetadataCombinations</td>
<td>Deprecated - use</td>
</tr>
<tr>
<td>GetDataForAllMetadataCombinations</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetDataGridExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetDynamicAccountRulesCache</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetLineItems</td>
<td>Deprecated - use GetLineItemsExtDim.</td>
</tr>
<tr>
<td>GetLineItemsExtDim</td>
<td>Deprecated - superseded by GetCellLineItems.</td>
</tr>
<tr>
<td>GetMaxCellTextSize</td>
<td>Returns the maximum number of characters that can be inserted as cell text.</td>
</tr>
<tr>
<td>GetMembersThatHaveData</td>
<td>Deprecated - use GetMembersThatHaveDataExtDim.</td>
</tr>
<tr>
<td>GetMembersThatHaveDataExtDim</td>
<td>Returns members that have data.</td>
</tr>
<tr>
<td>GetMultiServerMaxSyncDelayForChanges</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetNodeCube</td>
<td>Returns an object reference to the HsvNodeCube object. The object reference provides access to the subcube identified by the member IDs passed to GetNodeCube.</td>
</tr>
<tr>
<td>GetPhaseSubmissionGridForGivenScenarioPeriod</td>
<td>Returns an array representing the submission groups assigned to the specified combinations of Scenario dimension member, Period dimension members, and submission phases.</td>
</tr>
<tr>
<td>GetPhaseSubmissionGridForGivenScenarioPeriodExt</td>
<td>Returns an array representing the submission groups assigned to specific scenarios and periods.</td>
</tr>
<tr>
<td>GetPhasedSubmissionGroupAssignmentRecords</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetProcessManagementDat</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetStatus</td>
<td>Deprecated - use GetStatusExtDim.</td>
</tr>
<tr>
<td>GetStatusExtDim</td>
<td>Returns the transaction status, metadata status, and calculation status of a cell.</td>
</tr>
<tr>
<td>GetStatusEx</td>
<td>Deprecated - use GetStatusExExtDim.</td>
</tr>
<tr>
<td>GetStatusExExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>GetStatusUsingPhaseID</td>
<td>Deprecated - use GetStatusUsingPhaseIDExtDim.</td>
</tr>
<tr>
<td>GetStatusUsingPhaseIDExtDim</td>
<td>Returns the transaction status, metadata status, and calculation status of either a cell or a submission phase.</td>
</tr>
<tr>
<td>GetTextCell</td>
<td>Deprecated - use GetTextCellExtDim.</td>
</tr>
<tr>
<td>GetTextCellExtDim</td>
<td>Returns the data in a cell, as well as the cell’s status. The data is returned as a String.</td>
</tr>
<tr>
<td>GetTextCellLineItems</td>
<td>Deprecated - use GetTextCellLineItemsExtDim.</td>
</tr>
<tr>
<td>GetTextCellLineItemsExtDim</td>
<td>Returns arrays of the data and descriptions for the specified cell’s line items. Data is returned in a String array.</td>
</tr>
<tr>
<td>GetTextCells</td>
<td>Deprecated - use GetTextCellsExtDim.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetTextCellsExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>GetTextCellsWithRowSuppression</td>
<td>Deprecated - use GetTextCellsWithRowSuppressionExtDim.</td>
</tr>
<tr>
<td>GetTextCellsWithRowSuppressionExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>GetTextCellsWithRowSuppression2</td>
<td>Deprecated - use GetTextCellsWithRowSuppression2ExtDim.</td>
</tr>
<tr>
<td>GetTextCellsWithRowSuppression2ExtDim</td>
<td>Returns the data and status for rows that are not suppressed.</td>
</tr>
<tr>
<td>GetTextLineItems</td>
<td>Deprecated - use GetTextLineItemsExtDim.</td>
</tr>
<tr>
<td>GetTextLineItemsExtDim</td>
<td>Deprecated – superseded by GetTextCellLineItemsExtDim.</td>
</tr>
<tr>
<td>GetTransactionData</td>
<td>Populates an HsvTransactionData object with an array of transaction data.</td>
</tr>
<tr>
<td>GetUnassignedGroups</td>
<td>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.</td>
</tr>
<tr>
<td>GetUnassignedGroupsEx</td>
<td>Returns the names of submission groups assigned to submission phases but not to members.</td>
</tr>
<tr>
<td>GetURLByName</td>
<td>Returns an XML block representing the specified URL.</td>
</tr>
<tr>
<td>GetURLsForCell</td>
<td>Retrieves all URLs whose regions cover a specified POV.</td>
</tr>
<tr>
<td>InsertLineItemDetails</td>
<td>Deprecated - use InsertLineItemDetailsExtDim.</td>
</tr>
<tr>
<td>InsertLineItemDetailsExtDim</td>
<td>Inserts line item descriptions in the cells that intersect the specified dimension members.</td>
</tr>
<tr>
<td>IsValidCellText</td>
<td>Indicates whether a string exceeds the application’s maximum cell text size.</td>
</tr>
<tr>
<td>lIndexRecordProcessed</td>
<td>For internal use.</td>
</tr>
<tr>
<td>Load</td>
<td>Loads data into an application, using a load file on the application server.</td>
</tr>
<tr>
<td>LoadDrillableRegions</td>
<td>Loads the definition file for the drillable regions.</td>
</tr>
<tr>
<td>SetCalcStatusLocked</td>
<td>Locks the cells for a period in a subcube.</td>
</tr>
<tr>
<td>SetCalcStatusLockedForMultipleProcessUnits</td>
<td>Locks the cells for one or more process units.</td>
</tr>
<tr>
<td>SetCalcStatusUnlocked</td>
<td>Unlocks the cells for a period in a subcube.</td>
</tr>
<tr>
<td>SetCalcStatusUnlockedForMultipleProcessUnits</td>
<td>Unlocks the cells for one or more process units.</td>
</tr>
<tr>
<td>SetCell</td>
<td>Deprecated - use</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SetCellExtDim</td>
<td>Sets a cell's data; you can either insert data or set the cell to Null. Cell data is passed as a Double.</td>
</tr>
<tr>
<td>SetCellDescriptions</td>
<td>Deprecated - use SetCellDescriptionsExtDim.</td>
</tr>
<tr>
<td>SetCellDescriptionsExtDim</td>
<td>Inserts cell text into one or more cells.</td>
</tr>
<tr>
<td>SetCellLineItems</td>
<td>Deprecated - use SetCellLineItemsExtDim.</td>
</tr>
<tr>
<td>SetCellLineItemsExtDim</td>
<td>Appends or updates line items for the specified cell. Line item data is passed in a Double array.</td>
</tr>
<tr>
<td>SetCells</td>
<td>Deprecated - use SetCellsExtDim.</td>
</tr>
<tr>
<td>SetCellsExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>SetCells2</td>
<td>Deprecated - use SetCells2ExtDim.</td>
</tr>
<tr>
<td>SetCells2ExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>SetCellsLineItems</td>
<td>Deprecated - use SetCellsLineItemsExtDim.</td>
</tr>
<tr>
<td>SetCellsLineItemsExtDim</td>
<td>Appends or updates line items for the specified cells. Line item data is passed in a Double array.</td>
</tr>
<tr>
<td>SetFileForLoad</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetLineItems</td>
<td>Deprecated - use SetCellLineItemsExtDim.</td>
</tr>
<tr>
<td>SetMinMaxPeriod</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetPhaseSubmissionGridForGivenScenarioPeriod</td>
<td>Assigns submission groups to the specified combinations of Scenario dimension member, Period dimension members, and submission phases.</td>
</tr>
<tr>
<td>SetPhaseSubmissionGridForGivenScenarioPeriodEx</td>
<td>Assigns submission groups to the specified combinations of Scenario dimension member, Period dimension members, and submission phases using member labels.</td>
</tr>
<tr>
<td>SetTextCell</td>
<td>Deprecated - use SetTextCellExtDim.</td>
</tr>
<tr>
<td>SetTextCellExtDim</td>
<td>Inserts data into a cell, passing the data as a String. SetTextCell also enables you to scale the data that is passed.</td>
</tr>
<tr>
<td>SetTextCellLineItems</td>
<td>Deprecated - use SetTextCellLineItemsExtDim.</td>
</tr>
<tr>
<td>SetTextCellLineItemsExtDim</td>
<td>Appends or updates line items for the specified cell. Line item data is passed in a String array.</td>
</tr>
<tr>
<td>SetTextCells</td>
<td>Deprecated - use SetTextCellsExtDim.</td>
</tr>
<tr>
<td>SetTextCellsExtDim</td>
<td>Inserts data into cells. The cells' data is passed in a String array. It also enables you to scale the data that is passed.</td>
</tr>
<tr>
<td>SetTextCellsLineItems</td>
<td>Deprecated - use SetTextCellsLineItemsExtDim.</td>
</tr>
</tbody>
</table>
### Method Description

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetTextCellsLineItemsExtDim</td>
<td>Appends or updates line items for the specified cells. Line item data is passed in a String array.</td>
</tr>
<tr>
<td>SetTextLineItems</td>
<td>Deprecated - use SetTextLineItemsExtDim.</td>
</tr>
<tr>
<td>StartLoad</td>
<td>For internal use.</td>
</tr>
<tr>
<td>StartLoadTask</td>
<td>For internal use.</td>
</tr>
<tr>
<td>UpdateDataUsingMDDataBuffer</td>
<td>Inserts an HsvMDDataBuffer or HsvMDDataBufferLite object’s cells into the corresponding cells of an application.</td>
</tr>
<tr>
<td>ValidateMembersAndConvertToIDs</td>
<td>For internal use.</td>
</tr>
<tr>
<td>varAIGgregationMapAddr</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

See “HsvData Object Methods” on page 320.

### HsvCalculate Type Library Overview

The HsvCalculate type library contains 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:

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

The following table lists the methods of the HsvCalculate object.

### Table 22  HsvCalculate Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Allocate</td>
<td>Allocates an entity’s data for the specified Scenario, Year, Period, and Value dimension members.</td>
</tr>
<tr>
<td>Allocate2</td>
<td>Allocates an entity’s data across a range of periods for the specified Scenario, Year, and Value dimension members.</td>
</tr>
<tr>
<td>CalcEPU</td>
<td>Runs the Equity Pickup for the specified Scenario, Year, and Period.</td>
</tr>
<tr>
<td>ChartLogic</td>
<td>Calculates an entity’s data for the specified Scenario, Year, Period, and Value dimension members.</td>
</tr>
<tr>
<td>ChartLogic2</td>
<td>Calculates an entity’s data across a range of periods for the specified Scenario, Year, and Value dimension members.</td>
</tr>
<tr>
<td>Consolidate</td>
<td>Consolidates an entity’s data for the specified Scenario, Year, and Period dimension members.</td>
</tr>
<tr>
<td>Consolidate2</td>
<td>Consolidates an entity’s data across a range of periods for the specified Scenario and Year dimension members.</td>
</tr>
<tr>
<td>CustomLogic</td>
<td>For internal use.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnumOnDemandRules</td>
<td>Returns the labels for OnDemand rules.</td>
</tr>
<tr>
<td>ExecuteOnDemandRules</td>
<td>Executes OnDemand rules.</td>
</tr>
<tr>
<td>FindOverlappingConsolidation</td>
<td>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.</td>
</tr>
<tr>
<td>GetCOMDLLRules</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetConsolidationProgress</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetDefaultExchangeRate</td>
<td>Deprecated - use GetDefaultExchangeRateExtDim.</td>
</tr>
<tr>
<td>GetDefaultExchangeRateExtDim</td>
<td>Returns the exchange rate between two currencies for the specified Point of View.</td>
</tr>
<tr>
<td>GetEPUInfo</td>
<td>Retrieves the equity pickup information for the specified Scenario, Year, and Period. The information returns include: owner, owned, percentage ownership, and status.</td>
</tr>
<tr>
<td>GetEPUInfoExtDim</td>
<td>Retrieves the equity pickup information for the specified Scenario, Year, and Period. The information returns include: owner, owned, percentage ownership, and status.</td>
</tr>
<tr>
<td>GetVBScriptRules</td>
<td>Returns the rules loaded into an application. The rules are returned as an array of bytes.</td>
</tr>
<tr>
<td>IsEntityAnEPUOwner</td>
<td>Indicates whether the entity is an EPU owner for specified Scenario, Year, and Period.</td>
</tr>
<tr>
<td>LoadCalcManagerRules</td>
<td>For internal use.</td>
</tr>
<tr>
<td>LoadCalcManagerRules2</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetCOMDLLRules</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetVBScriptRules</td>
<td>Loads or scans a rules file.</td>
</tr>
<tr>
<td>SetVBScriptRules2</td>
<td>Loads or scans a rules file, optionally validating whether the rules violate the referential integrity of any intercompany transactions.</td>
</tr>
<tr>
<td>StopConsolidation</td>
<td>For internal use.</td>
</tr>
<tr>
<td>Translate</td>
<td>Translates an entity’s data from one currency to another for the specified Scenario, Year, Period, and Value dimension members.</td>
</tr>
<tr>
<td>Translate2</td>
<td>Translates an entity’s data from one currency to another across a range of periods for the specified Scenario, Year, and Value dimension members.</td>
</tr>
</tbody>
</table>

See “HsvCalculate Object Methods” on page 392.

**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 87.

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 87.

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 90.

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:

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

The following table lists the HsvJournals object’s methods.

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

See “HsvJournals Object Methods” on page 411.

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:

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

The following table lists the methods of the IHsvJournalsEx interface.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddJournalGroup</td>
<td>Creates a journal group.</td>
</tr>
<tr>
<td>ApproveJournals</td>
<td>Approves one or more journals.</td>
</tr>
<tr>
<td>DeleteJournals</td>
<td>Deletes one or more journals.</td>
</tr>
<tr>
<td>DeleteTemplates</td>
<td>Deletes one or more templates.</td>
</tr>
<tr>
<td>EnumJournalGroups</td>
<td>Returns the names and descriptions of an application's journal groups.</td>
</tr>
<tr>
<td>EnumJournalGroupsEx</td>
<td>Returns the names and descriptions of an application's journal groups.</td>
</tr>
<tr>
<td>EnumJournalGroupsForScenarioYear</td>
<td>Returns the names and descriptions of the journal groups assigned to journals for a scenario and year.</td>
</tr>
<tr>
<td>EnumJournalIDsForExtractFilter</td>
<td>Returns the IDs of journals and templates that match the specified filtering criteria.</td>
</tr>
<tr>
<td>GenerateRecurring</td>
<td>Generates a journal from a recurring template.</td>
</tr>
<tr>
<td>GetEntityJournals</td>
<td>Deprecated - use GetEntityJournalsExtDim.</td>
</tr>
<tr>
<td>GetEntityJournalsExtDim</td>
<td>Returns journal entry information for all journals that match the specified Point of View.</td>
</tr>
<tr>
<td>GetEntityJournals2</td>
<td>Deprecated - use GetEntityJournals2ExtDim.</td>
</tr>
<tr>
<td>GetEntityJournals2ExtDim</td>
<td>Returns journal entry information for all journals, including journal IDs, that match the specified Point of View.</td>
</tr>
<tr>
<td>GetJournalExtDim</td>
<td>Returns a variety of information for a journal. Line item amounts are returned in a Double array.</td>
</tr>
<tr>
<td>GetJournal2</td>
<td>Deprecated - use GetJournal2ExtDim.</td>
</tr>
<tr>
<td>GetJournal2ExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>GetJournalDisplayData</td>
<td>Returns various types of information for journals. The types of information returned correspond to the display columns in Journals.</td>
</tr>
<tr>
<td>GetJournalLabelsForIDs</td>
<td>Returns the labels of the journals that contain the specified Scenario and Year dimension members and that correspond to a set of journal IDs.</td>
</tr>
<tr>
<td>GetJournalQueryDefinitionIDs</td>
<td>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.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetTemplate</td>
<td>Deprecated - use GetTemplateExtDim.</td>
</tr>
<tr>
<td>GetTemplateExtDim</td>
<td>Returns a variety of information for a template. Line item amounts are returned in a Double array.</td>
</tr>
<tr>
<td>GetTemplateDisplayData</td>
<td>Returns various types of information for templates. The types of information returned correspond to the display columns in Journals.</td>
</tr>
<tr>
<td>GetTemplateLabelsForIDs</td>
<td>Returns the labels of the journal templates for a set of journal template IDs.</td>
</tr>
<tr>
<td>GetTemplateQueryDefinitionIDs</td>
<td>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.</td>
</tr>
<tr>
<td>GetTextJournal</td>
<td>Deprecated - use GetTextJournalExtDim.</td>
</tr>
<tr>
<td>GetTextJournalExtDim</td>
<td>Returns a variety of information for a journal. Line item amounts are returned in a String array.</td>
</tr>
<tr>
<td>GetTextTemplate</td>
<td>Deprecated - use GetTextTemplateExtDim.</td>
</tr>
<tr>
<td>GetTextTemplateExtDim</td>
<td>Returns a variety of information for a template. Line item amounts are returned in a String array.</td>
</tr>
<tr>
<td>GetVariance</td>
<td>Returns a journal’s total debit and credit amounts and the difference between these amounts.</td>
</tr>
<tr>
<td>PostJournals</td>
<td>Posts one or more journals.</td>
</tr>
<tr>
<td>RejectJournals</td>
<td>Rejects one or more journals.</td>
</tr>
<tr>
<td>RemoveAllJournalGroups</td>
<td>Removes all journal groups from an application.</td>
</tr>
<tr>
<td>RemoveJournalGroup</td>
<td>Removes the specified journal group.</td>
</tr>
<tr>
<td>SaveJournal</td>
<td>Deprecated - use SaveJournalExtDim.</td>
</tr>
<tr>
<td>SaveJournalExtDim</td>
<td>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. <strong>Tip:</strong> To create a journal from a template, pass the values returned by GetTemplateExtDim to SaveJournalExtDim.</td>
</tr>
<tr>
<td>SaveTemplate</td>
<td>Deprecated - use SaveTemplateExtDim.</td>
</tr>
<tr>
<td>SaveTemplateExtDim</td>
<td>Creates a new journal template, or updates an existing journal template. Line item amounts are passed in a Double array.</td>
</tr>
<tr>
<td>SaveTextJournal</td>
<td>Deprecated - use SaveTextJournalExtDim.</td>
</tr>
<tr>
<td>SaveTextJournalExtDim</td>
<td>Creates a new journal, or saves changes to a journal that has a Working or Submitted status. Line item amounts are passed in a String array. <strong>Tip:</strong> To create a journal from a template, pass the values returned by GetTextTemplate to SaveTextJournalExtDim.</td>
</tr>
<tr>
<td>SaveTextTemplate</td>
<td>Deprecated - use SaveTextTemplateExtDim.</td>
</tr>
<tr>
<td>SaveTextTemplateExtDim</td>
<td>Creates a new journal template, or updates an existing journal template. Line item amounts are passed in a String array.</td>
</tr>
<tr>
<td>SubmitJournals</td>
<td>Submits one or more journals.</td>
</tr>
</tbody>
</table>
### IHsvJournalsReport Interface Methods


The following table lists the methods of the IHsvJournalsReport interface.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetReportData</td>
<td>Deprecated - use GetReportDataExtDim.</td>
</tr>
<tr>
<td>GetReportDataExtDim</td>
<td>Returns data for journals; only actual journal amounts are returned.</td>
</tr>
<tr>
<td>GetReportData2</td>
<td>Deprecated - use GetReportDataExtDim.</td>
</tr>
<tr>
<td>GetReportData3</td>
<td>Returns data for journals; scaled and actual journal amounts are returned.</td>
</tr>
</tbody>
</table>

### 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 91.
- 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 95.
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:

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

The following table lists the methods of the HsvSecurityAccess object.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddApplicationAdministrator</td>
<td>Deprecated - use AddApplicationAdministrator2.</td>
</tr>
<tr>
<td>AddApplicationAdministrator2</td>
<td>Assigns the Application Administrator role to a user.</td>
</tr>
<tr>
<td>AddOrRemoveApplicationAdministrators</td>
<td>Deprecated - use AddOrRemoveApplicationAdministrators2.</td>
</tr>
<tr>
<td>AddOrRemoveApplicationAdministrators2</td>
<td>Adds or removes one or more users from the Application Administrator role.</td>
</tr>
<tr>
<td>AddOrRemoveRolesFromUser</td>
<td>Deprecated - use AddOrRemoveRolesFromUser2.</td>
</tr>
<tr>
<td>AddOrRemoveRolesFromUser2</td>
<td>Adds or removes a user from one or more roles.</td>
</tr>
<tr>
<td>AddOrRemoveUsersFromRole</td>
<td>Deprecated - use AddOrRemoveUsersFromRole2.</td>
</tr>
<tr>
<td>AddOrRemoveUsersFromRole2</td>
<td>Assigns users to or removes them from a role.</td>
</tr>
<tr>
<td>AddSecurityClass</td>
<td>Adds a security class to a Classic application.</td>
</tr>
<tr>
<td>AddSecurityClassWithAccessCode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>AddUser2</td>
<td>Deprecated - use AddUser2.</td>
</tr>
<tr>
<td>AddUser</td>
<td>Adds a user or user group to an application.</td>
</tr>
<tr>
<td>AddUserEx</td>
<td>This method is not supported as of Release 4.1.</td>
</tr>
<tr>
<td>AddUserToRole</td>
<td>Deprecated - use AddUserToRole2.</td>
</tr>
<tr>
<td>AddUserToRole2</td>
<td>Assigns a user to a role.</td>
</tr>
<tr>
<td>AllowRulesLoadForEPMAApp</td>
<td>For internal use.</td>
</tr>
<tr>
<td>DeleteSecurityClass</td>
<td>Deletes a security class from a Classic application.</td>
</tr>
<tr>
<td>DeleteSecurityClasses</td>
<td>Deletes a set of security classes from a Classic application.</td>
</tr>
<tr>
<td>DeleteSecurityClassWithAccessCode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EnumApplicationAdministrators</td>
<td>Deprecated - use EnumApplicationAdministrators2.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnumApplicationAdministrators2</td>
<td>Returns arrays containing the security identifiers and usernames of the users assigned to the Application Administrator role.</td>
</tr>
<tr>
<td>EnumRoles</td>
<td>Returns an array containing the names of an application’s roles.</td>
</tr>
<tr>
<td>EnumRolesForPrincipal</td>
<td>Returns the IDs of a user or group’s roles.</td>
</tr>
<tr>
<td>EnumRolesForUser</td>
<td>Returns the localized names of a user’s roles.</td>
</tr>
<tr>
<td>EnumSecurityClasses</td>
<td>Returns arrays containing an application’s security class IDs and names.</td>
</tr>
<tr>
<td>EnumSecurityClassRightsForPrincipal</td>
<td>Returns arrays representing a user’s access and email alerting rights to security classes.</td>
</tr>
<tr>
<td>EnumUserClassAccess</td>
<td>Returns the access and email alerting rights that the specified users have for the specified security classes.</td>
</tr>
<tr>
<td>EnumUsers</td>
<td>Deprecated - use EnumUsers2.</td>
</tr>
<tr>
<td>EnumUsers2</td>
<td>Deprecated - use EnumUsers3.</td>
</tr>
<tr>
<td>EnumUsers3</td>
<td>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.</td>
</tr>
<tr>
<td>EnumUsersInGroup</td>
<td>Returns the security identifiers, usernames, and identity types of the users in a user group.</td>
</tr>
<tr>
<td>EnumUsersInRole</td>
<td>Deprecated - use EnumUsersInRole2.</td>
</tr>
<tr>
<td>EnumUsersInRole2</td>
<td>Returns the security identifiers and usernames of the users and groups assigned to a role.</td>
</tr>
<tr>
<td>EnumUsersInRole3</td>
<td>Returns the security identifiers and usernames of the users and groups assigned to a role, and optionally allows you to return this information for users of any groups assigned to the role.</td>
</tr>
<tr>
<td>EnumUsersInSecurityClass</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EnumUsersInSecurityClass2</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EnumUsersInSecurityClass3</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EnumUsersInSecurityClass4</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EnumUsersOrGroups</td>
<td>Returns the security identifiers, usernames, and identity types of the users in a user group.</td>
</tr>
<tr>
<td>EnumUsersWithFilter</td>
<td>Deprecated - use EnumUsersWithFilter2.</td>
</tr>
<tr>
<td>EnumUsersWithFilter2</td>
<td>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.</td>
</tr>
<tr>
<td>GenerateSecurityReportForBiPub</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

92 About the Type Libraries and Objects
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAllSecurityClassRightsForConnectedUser</td>
<td>Returns arrays indicating the connected user’s rights to all security classes.</td>
</tr>
<tr>
<td>GetApplicationAdministratorAccessForAllUsers</td>
<td><strong>Deprecated</strong> - use GetApplicationAdministratorAccessForAllUsers2.</td>
</tr>
<tr>
<td>GetApplicationAdministratorAccessForAllUsers2</td>
<td>Indicates which of an application’s users are assigned to the Application Administrator role.</td>
</tr>
<tr>
<td>GetConnectedUser</td>
<td><strong>Deprecated</strong> - use GetConnectedUser2.</td>
</tr>
<tr>
<td>GetConnectedUser2</td>
<td>Returns the security identifier and username of the connected user.</td>
</tr>
<tr>
<td>GetIdentityTypes</td>
<td>Returns the identity types of the specified users.</td>
</tr>
<tr>
<td>GetNumRoles</td>
<td>Returns a count of the number of roles in an application.</td>
</tr>
<tr>
<td>GetOwner</td>
<td>This method is not supported as of Release 4.1.</td>
</tr>
<tr>
<td>GetRoleAccessForAllUsers</td>
<td><strong>Deprecated</strong> - use GetRoleAccessForAllUsers2.</td>
</tr>
<tr>
<td>GetRoleAccessForAllUsers2</td>
<td>Indicates whether the application users are assigned to a role.</td>
</tr>
<tr>
<td>GetRoleID</td>
<td>Returns the ID of a role, using the role name.</td>
</tr>
<tr>
<td>GetRoleLabel</td>
<td>Returns the name of a role, using a role ID.</td>
</tr>
<tr>
<td>GetRulesMode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetSecurityClassAccessForAllUsers</td>
<td><strong>Deprecated</strong> - use GetSecurityClassAccessForAllUsers2.</td>
</tr>
<tr>
<td>GetSecurityClassAccessForAllUsers2</td>
<td>Returns the access rights to a security class for all of an application’s users.</td>
</tr>
<tr>
<td>GetSecurityClassID</td>
<td>Returns the ID of a security class, using the security class’s name.</td>
</tr>
<tr>
<td>GetSecurityClassLabel</td>
<td>Returns the name of a security class, using a security class ID.</td>
</tr>
<tr>
<td>GetSecurityClassRightsForConnectedUser</td>
<td>Indicates the access rights that the connected user has to a security class.</td>
</tr>
<tr>
<td>GetTaskAccessForConnectedUserFromList</td>
<td>Indicates whether the connected user is allowed to perform one or more tasks.</td>
</tr>
<tr>
<td>GetUserAccessForAllRoles</td>
<td><strong>Deprecated</strong> - use GetUserAccessForAllRoles2.</td>
</tr>
<tr>
<td>GetUserAccessForAllRoles2</td>
<td>Indicates whether a user is assigned to the application’s roles.</td>
</tr>
<tr>
<td>GetUserAccessForAllSecurityClasses</td>
<td><strong>Deprecated</strong> - use GetUserAccessForAllSecurityClasses2.</td>
</tr>
<tr>
<td>GetUserAccessForAllSecurityClasses2</td>
<td>Returns the level of access rights that a user has for each security class in an application.</td>
</tr>
<tr>
<td>GetUserID</td>
<td><strong>Deprecated</strong> - use GetUserSID.</td>
</tr>
<tr>
<td>GetUserIDFromSID</td>
<td>This method is not supported as of Release 4.1.</td>
</tr>
<tr>
<td>UserInfoFromUniqueID</td>
<td><strong>Deprecated.</strong> Use UserInfoFromUniqueID2.</td>
</tr>
<tr>
<td>UserInfoFromUniqueID2</td>
<td>Returns user information such as the username, first name, and last name, using the user’s external authentication token and security identifier.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetUserName</td>
<td>Deprecated - use GetUserName2.</td>
</tr>
<tr>
<td>GetUserName2</td>
<td>Returns a username, using a security identifier.</td>
</tr>
<tr>
<td>GetUserNameFromSID</td>
<td>Returns the username of a user, using the user’s security identifier (SID).</td>
</tr>
<tr>
<td>GetUserSID</td>
<td>Returns the security identifier for a user.</td>
</tr>
<tr>
<td>InsertDefaultSecurityClass</td>
<td>For internal use.</td>
</tr>
<tr>
<td>InsertDefaultSecurityClassWithAccessCode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>IsApplicationAdministrator</td>
<td>Indicates whether the connected user is assigned to the Application Administrator role.</td>
</tr>
<tr>
<td>IsClassicHFMApplication</td>
<td>Indicates whether an application is a Classic application.</td>
</tr>
<tr>
<td>IsConnectedUserAllowedToPerformTask</td>
<td>Indicates whether the connected user has rights to a task.</td>
</tr>
<tr>
<td>IsConnectedUserInRole</td>
<td>Indicates whether the connected user is assigned to a role.</td>
</tr>
<tr>
<td>IsValidWindowsUser</td>
<td>This method is not supported as of Release 4.1.</td>
</tr>
<tr>
<td>RemoveApplicationAdministrator</td>
<td>Deprecated - use RemoveApplicationAdministrator2.</td>
</tr>
<tr>
<td>RemoveApplicationAdministrator2</td>
<td>Removes a user from the Application Administrator role.</td>
</tr>
<tr>
<td>RemoveUser</td>
<td>Deprecated - use RemoveUser2.</td>
</tr>
<tr>
<td>RemoveUser2</td>
<td>Removes a user from an application.</td>
</tr>
<tr>
<td>RemoveUserFromRole</td>
<td>Deprecated - use RemoveUserFromRole2.</td>
</tr>
<tr>
<td>RemoveUserFromRole2</td>
<td>Removes a user from a role.</td>
</tr>
<tr>
<td>RenameSecurityClass</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetManySecurityClassRightsForUser</td>
<td>Deprecated - use SetManySecurityClassRightsForUser2.</td>
</tr>
<tr>
<td>SetManySecurityClassRightsForUser2</td>
<td>Sets a user’s access and email alerting rights for one or more security classes.</td>
</tr>
<tr>
<td>SetRolesForUser</td>
<td>Specifies one or more roles for a user.</td>
</tr>
<tr>
<td>SetRulesMode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetSecurityClassLabel</td>
<td>Changes the name of a security class in a Classic application.</td>
</tr>
<tr>
<td>SetSecurityClassLabelWithAccessCode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetSecurityClassRightsForManyUsers</td>
<td>Deprecated - use SetSecurityClassRightsForManyUsers2.</td>
</tr>
<tr>
<td>SetSecurityClassRightsForManyUsers2</td>
<td>Sets one or more users’ access and email alerting rights to a security class.</td>
</tr>
<tr>
<td>SetSecurityClassRightsForUser</td>
<td>Deprecated - use SetSecurityClassRightsForUser2.</td>
</tr>
<tr>
<td>SetSecurityClassRightsForUser2</td>
<td>Sets a user’s access and email alerting rights for a security class.</td>
</tr>
</tbody>
</table>
Method | Description
--- | ---
SetUserClassAccess | Sets security class access and email alerting rights for one or more users.
TakeOwnership | This method is not supported as of Release 4.1.


**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:

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

The following table lists the IHsvDataSecurity interface’s methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCellLevelAccessRights</td>
<td>Deprecated - use GetCellLevelAccessRightsExtDim.</td>
</tr>
<tr>
<td>GetCellLevelAccessRightsExtDim</td>
<td>Returns the access rights that the connected user has to a cell.</td>
</tr>
<tr>
<td>GetProcessUnitAccessRights</td>
<td>Returns the access rights that the connected user has to a process unit.</td>
</tr>
<tr>
<td>GetProcessUnitAccessRightsAndState</td>
<td>Returns the access rights that the connected user has for a process unit, as well as the process unit’s current level.</td>
</tr>
<tr>
<td>GetProcessUnitAccessRightsEx</td>
<td>Deprecated - use GetProcessUnitAccessRightsExExtDim.</td>
</tr>
<tr>
<td>GetProcessUnitAccessRightsExExtDim</td>
<td>Returns the access rights that the connected user has to a submission phase, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>GetProcessUnitAccessRightsAndStateEx</td>
<td>Deprecated - use GetProcessUnitAccessRightsAndStateExExtDim.</td>
</tr>
<tr>
<td>GetProcessUnitAccessRightsAndStateExExtDim</td>
<td>Returns the following information for a submission phase, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td></td>
<td>• The connected user’s access rights to the submission phase.</td>
</tr>
<tr>
<td></td>
<td>• The review level of the submission phase.</td>
</tr>
<tr>
<td>RefreshAccessRightsCache</td>
<td>Refreshes the access rights on the application server.</td>
</tr>
</tbody>
</table>

See “IHsvDataSecurity Interface Methods” on page 495.
**HsvSystemInfo Type Library Overview**

The HsvSystemInfo type library contains 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:

```vbnet
Dim cHsvSystemInfo As HsvSystemInfo
Set cHsvSystemInfo = cHsvSession.SystemInfo
```

The following table lists the HsvSystemInfo object’s methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>AddRefToHsxServer</code></td>
<td>For internal use.</td>
</tr>
<tr>
<td><code>AddTaskToAudit</code></td>
<td>Adds a task for the user to the audit log.</td>
</tr>
<tr>
<td><code>AddTaskToAuditWithAttachment</code></td>
<td>Adds a task for the user to the audit log with attachment.</td>
</tr>
<tr>
<td><code>AddTaskToRunningTasks</code></td>
<td>For internal use.</td>
</tr>
<tr>
<td><code>AddTaskToRunningTasksAndUpdatePOV</code></td>
<td>For internal use.</td>
</tr>
<tr>
<td><code>CheckAccess</code></td>
<td>Indicates whether the current user has access to the application.</td>
</tr>
<tr>
<td><code>ClearAuditTasks</code></td>
<td>Deletes the task audit history for all tasks that meet the specified criteria.</td>
</tr>
<tr>
<td><code>ClearAuditTasks2</code></td>
<td>Deletes the audit history for a task that meets the specified criteria.</td>
</tr>
<tr>
<td><code>ClearRunningTask</code></td>
<td>For internal use.</td>
</tr>
<tr>
<td><code>DeleteUserParameter</code></td>
<td>Deletes a user parameter created with <code>SetUserParameter</code>.</td>
</tr>
<tr>
<td><code>DisableNewConnections</code></td>
<td>Disables new Financial Management connections for the specified user and application server criteria.</td>
</tr>
<tr>
<td><code>EnableNewConnections</code></td>
<td>Enables new Financial Management connections for the specified user and application server criteria.</td>
</tr>
<tr>
<td><code>EnumActivityServers</code></td>
<td>Returns the names of the application servers for which there are task audit and data audit records.</td>
</tr>
<tr>
<td><code>EnumActivityUsers</code></td>
<td>Returns the usernames of all users who have performed at least one activity in the application.</td>
</tr>
<tr>
<td><code>EnumActivityUsersForRunningTasks</code></td>
<td>For internal use.</td>
</tr>
<tr>
<td><code>EnumAuditTasks</code></td>
<td>Returns task audit information from a range of audit records that meet the selection criteria. Criteria include date range, application server, and user.</td>
</tr>
<tr>
<td><code>EnumAuditTasksEx</code></td>
<td>Returns task audit information from a range of audit records that meet the selection criteria. Criteria include date range, application server, and user.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>EnumAuditTasks2</td>
<td>Returns task audit information from a range of audit records that meet the selection criteria. Criteria include task, date range, application server, and user.</td>
</tr>
<tr>
<td>EnumProhibitConnections</td>
<td>Returns information on the applications, application servers, and users for which connections are disabled.</td>
</tr>
<tr>
<td>EnumRunningTasks</td>
<td>Returns information about the running tasks that meet the selection criteria.</td>
</tr>
<tr>
<td>EnumRunningTasksEx</td>
<td>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.</td>
</tr>
<tr>
<td>EnumRunningTasksPOV</td>
<td>Returns the dimension members, consolidation types, and other information for the running consolidations that meet the selection criteria.</td>
</tr>
<tr>
<td>ExtractTaskAudit</td>
<td>Extracts to a file the task audit records that meet the specified criteria. Criteria include date range, application server, user, and task.</td>
</tr>
<tr>
<td>ExtractTaskAudit2</td>
<td>Extracts to a file the task audit records that meet the specified criteria. Criteria include date range, application server, user, and task.</td>
</tr>
<tr>
<td>GetActivityCodeDesc</td>
<td>Returns the description of a type of activity.</td>
</tr>
<tr>
<td>GetActivityUserID</td>
<td>Returns the activity user ID for a username.</td>
</tr>
<tr>
<td>GetAllInfoForTask</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetApplicationDirectory</td>
<td>Returns the name of the application folder for an application.</td>
</tr>
<tr>
<td>GetApplicationName</td>
<td>Returns the name of the application to which the client is connected.</td>
</tr>
<tr>
<td>GetCalcRulestype</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetCOMDLLCalcRules</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetCurrentActivity</td>
<td>Returns information about the user’s current activity.</td>
</tr>
<tr>
<td>GetExtractFileEncoding</td>
<td>Returns the type of file encoding for extracted files.</td>
</tr>
<tr>
<td>GetFormattedDateTime</td>
<td>Returns a string representation of the double-byte date/time value using the language ID for the connected user.</td>
</tr>
<tr>
<td>GetFormattedResourceString</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetRunningTaskLogFilePathName</td>
<td>Returns the name and path of the log file for a running task.</td>
</tr>
<tr>
<td>GetRunningTaskProgress</td>
<td>Returns information on the progress of a running task.</td>
</tr>
<tr>
<td>GetRunningTasksCount</td>
<td>Returns the number of currently running tasks for the application.</td>
</tr>
<tr>
<td>GetRunningTaskStatus</td>
<td>Returns the status of a task.</td>
</tr>
<tr>
<td>GetKillUsersStatus</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetKillUserStatus</td>
<td>Indicates whether an administrator has logged off the current user.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetLanguageUserParameters</td>
<td>Gets the language in which member descriptions are displayed for the connected user.</td>
</tr>
<tr>
<td>GetLastModifiedDateForArtifact</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetLastModifiedDateAndModifiedByForArtifact</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetModuleName</td>
<td>Returns the name of the module represented by a module ID.</td>
</tr>
<tr>
<td>GetNumberFormattingUserParameters</td>
<td>Returns the double-byte Integers that identify the user's decimal and thousands separator characters.</td>
</tr>
<tr>
<td>GetResourceLanguageUserParameters</td>
<td>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.</td>
</tr>
<tr>
<td>GetResourceString</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetResourceStringFromHR</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetServerName</td>
<td>Returns the name of the server to which the client is connected.</td>
</tr>
<tr>
<td>GetTaskAuditAttachment</td>
<td>Returns the names of the attachments for a specified task audit.</td>
</tr>
<tr>
<td>GetUserName</td>
<td>Returns the username of the connected user.</td>
</tr>
<tr>
<td>GetUserParameter</td>
<td>Returns the value of a user parameter set with <code>SetUserParameter</code>.</td>
</tr>
<tr>
<td>GetVBScriptCalcRules</td>
<td>Returns the rules loaded into an application. The rules are returned as an array of bytes.</td>
</tr>
<tr>
<td>GetVBScriptCalcRulesEX</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetVBScriptMemberListRules</td>
<td>Returns an application’s member lists as an array of bytes in the <code>.LST</code> file format.</td>
</tr>
<tr>
<td>GetWorkingDirectory</td>
<td>For internal use.</td>
</tr>
<tr>
<td>IsLoggingEnabled</td>
<td>Returns a zero value if diagnostic logging is not enabled, and a positive value if diagnostic logging is enabled.</td>
</tr>
<tr>
<td>IsScheduledTaskReadyToRun</td>
<td>For internal use.</td>
</tr>
<tr>
<td>KeepRunningTaskStillAlive</td>
<td>For internal use.</td>
</tr>
<tr>
<td>KillUsers</td>
<td>Logs off users.</td>
</tr>
<tr>
<td>OutputSystemInfo</td>
<td>For internal use.</td>
</tr>
<tr>
<td>pIHsvDSSEcurityUnk</td>
<td>For internal use.</td>
</tr>
<tr>
<td>pIHsvDSReportsUnk</td>
<td>For internal use.</td>
</tr>
<tr>
<td>ReleaseHsxServer</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SaveLastModifiedDateAndModifiedBy</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetApplicationDirectory</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetCOMDLLCalcRules</td>
<td>For internal use.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SetCurrentActivity</td>
<td>Sets the activity for the current user.</td>
</tr>
<tr>
<td>SetCurrentModule</td>
<td>Sets the current module using a module name.</td>
</tr>
<tr>
<td>SetCurrentModuleEx</td>
<td>Sets the current module using a module ID.</td>
</tr>
<tr>
<td>SetExtractFileEncoding</td>
<td>Sets the file encoding type for extracted files.</td>
</tr>
<tr>
<td>SetLanguageUserParameters</td>
<td>Sets the language in which member descriptions are displayed for the connected user.</td>
</tr>
<tr>
<td>SetNumberFormattingUserParameters</td>
<td>Sets the user’s decimal and thousands separator characters.</td>
</tr>
<tr>
<td>SetResourceLanguageForCurrentSession</td>
<td>Sets the language for a user’s resource strings in the current session. The specified language does not persist beyond the current session.</td>
</tr>
<tr>
<td>SetResourceLanguageUserParameters</td>
<td>Sets the user’s default language for resource strings such as error messages and other strings that are generated on the server.</td>
</tr>
<tr>
<td>SetUserParameter</td>
<td>Sets a parameter for the user. Use <code>SetUserParameter</code> to create and edit custom parameters for items such as personal preferences.</td>
</tr>
<tr>
<td>SetVBScriptCalcRules</td>
<td>Validates and loads a rules file; a flag determines whether <code>SetVBScriptCalcRules</code> loads after validation or validates without loading.</td>
</tr>
<tr>
<td>SetVBScriptMemberListRules</td>
<td>Validates and loads a member lists file; a flag determines whether <code>SetVBScriptMemberListRules</code> loads after validation or validates without loading.</td>
</tr>
<tr>
<td>StopRunningTask</td>
<td>Stops a running task.</td>
</tr>
<tr>
<td>TraceLog</td>
<td>Writes a text value to the diagnostic log if logging is enabled.</td>
</tr>
<tr>
<td>UpdateRunningTaskLogFilePathName</td>
<td>For internal use.</td>
</tr>
<tr>
<td>UpdateRunningTaskPOV</td>
<td>For internal use.</td>
</tr>
<tr>
<td>UpdateRunningTaskProgress</td>
<td>For internal use.</td>
</tr>
<tr>
<td>UpdateRunningTaskProgressDetails</td>
<td>For internal use.</td>
</tr>
<tr>
<td>UpdateRunningTaskStatus</td>
<td>For internal use.</td>
</tr>
<tr>
<td>UpdateUserAppPreferences</td>
<td>For internal use.</td>
</tr>
<tr>
<td>WarnUsersForShutDown</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>


**HsvProcessFlow Type Library Overview**

The HsvProcessFlow type library contains 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:

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

The following table lists the methods of the HsvProcessFlow object.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Approve</td>
<td>Approves a process unit.</td>
</tr>
<tr>
<td>Approve2</td>
<td>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.</td>
</tr>
<tr>
<td>ApproveEx</td>
<td>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.</td>
</tr>
<tr>
<td>GetGroupPhaseFromCell</td>
<td>Deprecated - use GetGroupPhaseFromCellExtDim.</td>
</tr>
<tr>
<td>GetGroupPhaseFromCellExtDim</td>
<td>Returns the IDs of the submission group and submission phase to which a cell is assigned.</td>
</tr>
<tr>
<td>GetHistory</td>
<td>Returns arrays containing the dates and times, users, actions, states, and comments of a process unit.</td>
</tr>
<tr>
<td>GetHistory2</td>
<td>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.</td>
</tr>
<tr>
<td>GetState</td>
<td>Returns the current state of a process unit.</td>
</tr>
<tr>
<td>GetPhasedSubmissionHistory</td>
<td>Deprecated - use GetPhasedSubmissionHistoryExtDim.</td>
</tr>
<tr>
<td>GetPhasedSubmissionHistoryExtDim</td>
<td>Returns arrays containing the history of a submission phase, using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>GetPhasedSubmissionState</td>
<td>Deprecated - use GetPhasedSubmissionStateExtDim.</td>
</tr>
<tr>
<td>GetPhasedSubmissionStateExtDim</td>
<td>Returns the current review level of a cell’s submission phase.</td>
</tr>
<tr>
<td>GetPhasedSubmissionStateUsingPhaseID</td>
<td>Deprecated - use GetPhasedSubmissionStateUsingPhaseIDExtDim.</td>
</tr>
<tr>
<td>GetPhasedSubmissionStateUsingPhaseIDExtDim</td>
<td>Returns the review level of a submission phase, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>PhasedSubmissionApprove</td>
<td>Deprecated - use PhasedSubmissionApproveExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionApproveExtDim</td>
<td>Approves a submission phase, using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionApprove2</td>
<td>Deprecated - use PhasedSubmissionApprove2ExtDim.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PhasedSubmissionApprove2ExtDim</td>
<td>Approves a submission phase, and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>PhasedSubmissionApproveEx</td>
<td>Deprecated - use PhasedSubmissionApproveExExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionApproveExExtDim</td>
<td>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, using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionGetHistory2</td>
<td>Deprecated - use PhasedSubmissionGetHistory2ExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionGetHistory2ExtDim</td>
<td>Returns arrays containing the history of a submission phase using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionGetHistory2UsingPhaseID</td>
<td>Deprecated - use PhasedSubmissionGetHistory2UsingPhaseIDExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionGetHistory2UsingPhaseIDExtDim</td>
<td>Returns arrays containing the history of a submission phase using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>PhasedSubmissionProcessManagementChange</td>
<td>Deprecated - use PhasedSubmissionProcessManagementChangeStateForMultipleEntities2ExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionProcessManagementChange</td>
<td>Applies a process management action and attaches documents to submission phases for multiple Entity dimension members, using the member IDs of cells in the phases.</td>
</tr>
<tr>
<td>PhasedSubmissionProcessManagementChange</td>
<td>StateForMultipleEntities2ExtDim</td>
</tr>
<tr>
<td>PhasedSubmissionPromote</td>
<td>Deprecated - use PhasedSubmissionPromoteExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionPromoteExtDim</td>
<td>Promotes a submission phase to a specified review level, using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionPromote2</td>
<td>Deprecated - use PhasedSubmissionPromote2ExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionPromote2ExtDim</td>
<td>Promotes a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>PhasedSubmissionPublish</td>
<td>Deprecated - use PhasedSubmissionPublishExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionPublishExtDim</td>
<td>Publishes a submission phase using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionPublish2</td>
<td>Deprecated - use PhasedSubmissionPublish2ExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionPublish2ExtDim</td>
<td>Publishes a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>PhasedSubmissionPublishEx</td>
<td>Deprecated - use PhasedSubmissionPublishExExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionPublishExExtDim</td>
<td>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, using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionReject</td>
<td>Deprecated - use PhasedSubmissionRejectExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionRejectExtDim</td>
<td>Rejects a submission phase, using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionReject2</td>
<td>Deprecated - use PhasedSubmissionReject2ExtDim.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PhasedSubmissionReject2ExtDim</td>
<td>Rejects a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>PhasedSubmissionSignOff</td>
<td>Deprecated - use PhasedSubmissionSignOffExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionSignOffExtDim</td>
<td>Signs off on a submission phase using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionSignOff2</td>
<td>Deprecated - use PhasedSubmissionSignOff2ExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionSignOff2ExtDim</td>
<td>Signs off on a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>PhasedSubmissionStart</td>
<td>Deprecated - use PhasedSubmissionStartExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionStartExtDim</td>
<td>Starts a submission phase using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionStart2</td>
<td>Deprecated - use PhasedSubmissionStart2ExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionStart2ExtDim</td>
<td>Starts a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>PhasedSubmissionStartEx</td>
<td>Deprecated - use PhasedSubmissionStartExExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionStartExExtDim</td>
<td>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, using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionSubmit</td>
<td>Deprecated - use PhasedSubmissionSubmitExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionSubmitExtDim</td>
<td>Submits a submission phase using the member IDs of a cell in the phase.</td>
</tr>
<tr>
<td>PhasedSubmissionSubmit2</td>
<td>Deprecated - use PhasedSubmissionSubmit2ExtDim.</td>
</tr>
<tr>
<td>PhasedSubmissionSubmit2ExtDim</td>
<td>Submits a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID.</td>
</tr>
<tr>
<td>ProcessManagementChangeStateForMultipleEntities</td>
<td>Applies a process management action to process units for one or more Entity dimension members.</td>
</tr>
<tr>
<td>ProcessManagementChangeStateForMultipleEntities2</td>
<td>Deprecated - use ProcessManagementChangeStateForMultipleEntities2ExtDim.</td>
</tr>
<tr>
<td>ProcessManagementChangeStateForMultipleEntities2ExtDim</td>
<td>Applies a process management action and attaches documents to process units for one or more Entity dimension members.</td>
</tr>
<tr>
<td>ProcessManagementChangeStateForMultipleEntitiesEx</td>
<td>Deprecated - use ProcessManagementChangeStateForMultipleEntitiesExExtDim.</td>
</tr>
<tr>
<td>ProcessManagementChangeStateForMultipleEntitiesExExtDim</td>
<td>Applies a process management action and attaches documents to submission phases for multiple Entity dimension members, using the member IDs of cells in the phases.</td>
</tr>
<tr>
<td>Promote</td>
<td>Promotes a process unit to a specified review level.</td>
</tr>
<tr>
<td>Promote2</td>
<td>Promotes a process unit to a specified review level, and provides the option of attaching one or more documents.</td>
</tr>
<tr>
<td>Publish</td>
<td>Publishes a process unit.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Publish2</td>
<td>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.</td>
</tr>
<tr>
<td>PublishEx</td>
<td>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.</td>
</tr>
<tr>
<td>Reject</td>
<td>Demotes a process unit from its current state to its previous state.</td>
</tr>
<tr>
<td>Reject2</td>
<td>Demotes a process unit from its current level to its previous level and optionally attaches documents.</td>
</tr>
<tr>
<td>SignOff</td>
<td>Signs off on a process unit.</td>
</tr>
<tr>
<td>SignOff2</td>
<td>Signs off on a process unit and attaches documents.</td>
</tr>
<tr>
<td>Start</td>
<td>Starts a process unit.</td>
</tr>
<tr>
<td>Start2</td>
<td>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.</td>
</tr>
<tr>
<td>StartEx</td>
<td>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.</td>
</tr>
<tr>
<td>Submit</td>
<td>Submits a process unit.</td>
</tr>
<tr>
<td>Submit2</td>
<td>Submits a process unit and provides the option to attach one or more documents.</td>
</tr>
</tbody>
</table>

See Chapter 15, “HsvProcessFlow Type Library.”

**HsvReports Type Library Overview**

The HsvReports type library contains 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:

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

The following table lists the methods of the HsvReports object.

**Table 30  HsvReports Object Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CheckReportSecurityClass</td>
<td>Indicates whether the connected user has either Read or All access rights to a report’s security class.</td>
</tr>
<tr>
<td>CheckReportSecurityClass2</td>
<td>Indicates whether the connected user has a level of access rights to a report’s security class.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CheckSecurityRole</td>
<td>Indicates whether the connected user is assigned to the security role that enables users to save reports of a report type.</td>
</tr>
<tr>
<td>DeleteDocuments</td>
<td>Deletes documents from the application server.</td>
</tr>
<tr>
<td>DeleteReport</td>
<td>Deletes a report from the application server.</td>
</tr>
<tr>
<td>DeleteReports</td>
<td>Deletes one or more reports from the application server.</td>
</tr>
<tr>
<td>DoesDocumentExist</td>
<td>Checks whether a document exists.</td>
</tr>
<tr>
<td>DoesFolderExist</td>
<td>Checks whether a folder exists.</td>
</tr>
<tr>
<td>EnumDocuments</td>
<td>Returns the names, descriptions, timestamps, and security class IDs of documents that meet the search criteria.</td>
</tr>
<tr>
<td>EnumDocumentsEx</td>
<td>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.</td>
</tr>
<tr>
<td>EnumReports</td>
<td>Returns the names, descriptions, and timestamps of reports on the application server.</td>
</tr>
<tr>
<td>GetDocument</td>
<td>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.</td>
</tr>
<tr>
<td>GetDocumentEx</td>
<td>Returns the definition of a document, as well as other properties such as the document’s type, file type, and security class.</td>
</tr>
<tr>
<td>GetReport</td>
<td>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.</td>
</tr>
<tr>
<td>SaveDocument</td>
<td>Saves a document to the application server.</td>
</tr>
<tr>
<td>SaveDocumentEx</td>
<td>Saves a document to the application server and specifies the document’s content type and privacy flag.</td>
</tr>
<tr>
<td>SetReport</td>
<td>Saves a report to the application server.</td>
</tr>
<tr>
<td>ValidateMembersAgainstSlice</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

See Chapter 16, “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 105.
- 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 108.
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 111.

- 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 111.

- The HsvMDIndexList object contains an array of indexes that represent points-of-view in an HsvMDDataBuffer or HsvMDDataBufferLite object. See “HsvMDIndexList Object Overview” on page 112.

- The HsvICTransactionsData object supplements the HsvICM object by providing access to intercompany transactions. See “HsvICTransactionsData Object Overview” on page 113.

**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. 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. You must include the New keyword, otherwise an error occurs.

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

The following table summarizes the HsvMDDataBuffer object’s methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeginEnumeration</td>
<td>Locks an HsvMDDataBuffer object, and returns the number of subcubes contained by the object.</td>
</tr>
<tr>
<td>CreateDataIndexList</td>
<td>Deprecated - use CreateDataIndexListExtDim.</td>
</tr>
<tr>
<td>CreateDataIndexListExtDim</td>
<td>Creates an HsvMDIndexList object based on the items in an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CreateDataIndexListEx</td>
<td>Deprecated - use CreateDataIndexListExtDim.</td>
</tr>
<tr>
<td>EndEnumeration</td>
<td>Unlocks an HsvMDDataBuffer object locked by BeginEnumeration.</td>
</tr>
<tr>
<td>EraseRecordFromPMBuffer</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetCheckLineItemDetailsForCaseInsensitiveDuplicates</td>
<td>Indicates whether case-insensitive duplicate checking is enabled for the HsvMDDataBuffer instance.</td>
</tr>
<tr>
<td>GetCubeIndexFromPOV</td>
<td>Returns the index of a subcube in an HsvMDDataBuffer object, using the IDs of the subcube’s dimension members.</td>
</tr>
<tr>
<td>GetCubePOVFromIndex</td>
<td>Returns the IDs of a subcube’s dimension members, using the subcube index.</td>
</tr>
<tr>
<td>GetData</td>
<td>Deprecated - use GetDataExtDim.</td>
</tr>
<tr>
<td>GetDataExtDim</td>
<td>Returns a cell’s data, using the member IDs that identify the cell.</td>
</tr>
<tr>
<td>GetDataAtIndex</td>
<td>Deprecated - use GetDataAtIndexExtDim.</td>
</tr>
<tr>
<td>GetDataAtIndexExtDim</td>
<td>Returns the data for a cell, using the indexes that identify the cell.</td>
</tr>
<tr>
<td>GetDescription</td>
<td>Deprecated - use GetDescriptionExtDim.</td>
</tr>
<tr>
<td>GetDescriptionExtDim</td>
<td>Returns a cell’s description, using the member IDs that identify the cell.</td>
</tr>
<tr>
<td>GetDescriptionAtIndex</td>
<td>Deprecated - use GetDescriptionAtIndexExtDim.</td>
</tr>
<tr>
<td>GetDescriptionAtIndexExtDim</td>
<td>Returns the description for a cell, using the indexes that identify the cell.</td>
</tr>
<tr>
<td>GetLineItems</td>
<td>Deprecated - use GetLineItemsExtDim.</td>
</tr>
<tr>
<td>GetLineItemsExtDim</td>
<td>Returns the data and descriptions for a cell’s line items, using the IDs that identify the cell’s dimension members.</td>
</tr>
<tr>
<td>GetLineItemsAtIndex</td>
<td>Deprecated - use GetLineItemsAtIndexExtDim.</td>
</tr>
<tr>
<td>GetLineItemsAtIndexExtDim</td>
<td>Returns the data and descriptions for a cell’s line items, using the indexes that identify the cell.</td>
</tr>
<tr>
<td>GetNumCellsForData</td>
<td>Returns the number of cells that are in a period of a subcube and that contain data.</td>
</tr>
<tr>
<td>GetNumCellsForDescriptions</td>
<td>Returns the number of cells that are in a period of a subcube and that contain descriptions.</td>
</tr>
<tr>
<td>GetNumCellsForLineItems</td>
<td>Returns the number of cells that are in a period of a subcube and that contain line items.</td>
</tr>
<tr>
<td>GetNumPeriodsInCubeForData</td>
<td>Returns the number of periods that are in a subcube and that contain cells with data.</td>
</tr>
<tr>
<td>GetNumPeriodsInCubeForDescriptions</td>
<td>Returns the number of periods that are in a subcube and that contain cells with descriptions.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>GetNumPeriodsInCubeForLineItems</td>
<td>Returns the number of periods that are in a subcube and that contain cells with line items.</td>
</tr>
<tr>
<td>GetPeriodIndexFromPOVForData</td>
<td>Returns the index of a period within a subcube, using the subcube’s index and the period’s member ID. An index is returned only for periods with cells that contain data.</td>
</tr>
<tr>
<td>GetPeriodIndexFromPOVForDescriptions</td>
<td>Returns the index of a period within a subcube, using the subcube’s index and the period’s member ID. An index is returned only for periods with cells that contain descriptions.</td>
</tr>
<tr>
<td>GetPeriodIndexFromPOVForLineItems</td>
<td>Returns the index of a period within a subcube, using the subcube’s index and the period’s member ID. An index is returned only for periods with cells that contain line items.</td>
</tr>
<tr>
<td>GetPeriodPOVFromIndexForData</td>
<td>Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain data.</td>
</tr>
<tr>
<td>GetPeriodPOVFromIndexForDescriptions</td>
<td>Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain descriptions.</td>
</tr>
<tr>
<td>GetPeriodPOVFromIndexForLineItems</td>
<td>Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain line items.</td>
</tr>
<tr>
<td>GetPMErrorRecordCount</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetPMRecordCount</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetRecordFromPMBuffer</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetRecordFromPMErrorBuffer</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetSortedNature</td>
<td>For internal use</td>
</tr>
<tr>
<td>InsertDataAtBeginning</td>
<td>Deprecated - use InsertDataAtBeginningExtDim.</td>
</tr>
<tr>
<td>InsertDataAtBeginningExtDim</td>
<td>Inserts data for a cell at the top of an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>InsertDataAtEnd</td>
<td>Deprecated - use InsertDataAtEndExtDim.</td>
</tr>
<tr>
<td>InsertDataAtEndExtDim</td>
<td>Inserts data for a cell at the bottom of an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>InsertDescriptionAtBeginning</td>
<td>Deprecated - use InsertDescriptionAtBeginningExtDim.</td>
</tr>
<tr>
<td>InsertDescriptionAtBeginningExtDim</td>
<td>Inserts a cell description at the top of an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>InsertDescriptionAtEnd</td>
<td>Deprecated - use InsertDescriptionAtEndExtDim.</td>
</tr>
<tr>
<td>InsertDescriptionAtEndExtDim</td>
<td>Inserts a cell description at the bottom of an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>InsertLineItemsAtBeginning</td>
<td>Deprecated - use InsertLineItemsAtBeginningExtDim.</td>
</tr>
<tr>
<td>InsertLineItemsAtBeginningExtDim</td>
<td>Inserts line items for a cell at the top of an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>InsertLineItemsAtEnd</td>
<td>Deprecated - use InsertLineItemsAtEndExtDim.</td>
</tr>
<tr>
<td>InsertLineItemsAtEndExtDim</td>
<td>Inserts line items for a cell at the bottom of an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>InsertLineItemsAtEndExtDim</td>
<td>Inserts line items for a cell at the bottom of an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>InsertRecordIntoPMBuffer</td>
<td>For internal use</td>
</tr>
<tr>
<td>InsertRecordIntoPMErrorBuffer</td>
<td>For internal use</td>
</tr>
<tr>
<td>RemoveAll</td>
<td>Clears all the cells from an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>SetCheckLineItemDetailsForCaseInsensitiveDuplicates</td>
<td>Specifies whether the system should check for case-insensitive duplicate line item descriptions.</td>
</tr>
<tr>
<td>SetData</td>
<td>Deprecated - use SetDataExtDim.</td>
</tr>
<tr>
<td>SetDataExtDim</td>
<td>Inserts a cell's data into an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>SetDescription</td>
<td>Deprecated - use SetDescriptionExtDim.</td>
</tr>
<tr>
<td>SetDescriptionExtDim</td>
<td>Inserts a cell's description into an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>SetGrowByAmount</td>
<td>Changes the amount of memory that the HsvMDDataBuffer object automatically allocates.</td>
</tr>
<tr>
<td>SetLineItems</td>
<td>Deprecated - use SetLineItemsExtDim.</td>
</tr>
<tr>
<td>SetLineItemsExtDim</td>
<td>Inserts a cell's line items into an HsvMDDataBuffer object.</td>
</tr>
<tr>
<td>SetSortedNature</td>
<td>For internal use</td>
</tr>
<tr>
<td>Sort</td>
<td>For internal use</td>
</tr>
</tbody>
</table>

**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. 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 are cached; see “IHsvMDDataBufferLite Interface” on page 111.

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. You must include the `New` keyword, otherwise an error occurs.
Dim cDataBufferLite As HsvMDDataBufferLite
Set cDataBufferLite = New HSVMDARRAYSLib.HsvMDDataBufferLite
The following table summarizes the HsvMDDataBufferLite object’s methods.

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeginEnumeration</td>
<td>Locks an HsvMDDataBufferLite object, and returns the number of subcubes contained by the object.</td>
</tr>
<tr>
<td>CreateDataIndexList</td>
<td>Deprecated - use CreateDataIndexListExtDim.</td>
</tr>
<tr>
<td>CreateDataIndexListExtDim</td>
<td>Creates an HsvMDIndexList object based upon the items in an HsvMDDataBufferLite object.</td>
</tr>
<tr>
<td>EndEnumeration</td>
<td>Unlocks an HsvMDDataBufferLite object locked by BeginEnumeration.</td>
</tr>
<tr>
<td>EraseRecordFromPMBuffer</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetCheckLineItemDetailsForCaseInsensitiveDuplicates</td>
<td>Indicates whether case-insensitive duplicate checking is enabled for the HsvMDDataBufferLite instance.</td>
</tr>
<tr>
<td>GetCubeIndexFromPOV</td>
<td>Returns the index of a subcube in an HsvMDDataBufferLite object, using the IDs of the subcube’s dimension members.</td>
</tr>
<tr>
<td>GetCubePOVFromIndex</td>
<td>Returns the IDs of a subcube’s dimension members, using the subcube’s index.</td>
</tr>
<tr>
<td>GetData</td>
<td>Deprecated - use GetDataExtDim.</td>
</tr>
<tr>
<td>GetDataExtDim</td>
<td>Returns a cell’s data, using the member IDs that identify the cell.</td>
</tr>
<tr>
<td>GetDataAtIndex</td>
<td>Returns the data for a cell, using the indexes that identify the cell.</td>
</tr>
<tr>
<td>GetDescription</td>
<td>Deprecated - use GetDescriptionExtDim.</td>
</tr>
<tr>
<td>GetDescriptionExtDim</td>
<td>Returns a cell’s description, using the member IDs that identify the cell.</td>
</tr>
<tr>
<td>GetDescriptionAtIndex</td>
<td>Returns the description for a cell, using the indexes that identify the cell.</td>
</tr>
<tr>
<td>GetLineItems</td>
<td>Deprecated - use GetLineItemsExtDim.</td>
</tr>
<tr>
<td>GetLineItemsExtDim</td>
<td>Returns the data and descriptions for a cell’s line items, using the IDs that identify the cell’s dimension members.</td>
</tr>
<tr>
<td>GetLineItemsAtIndex</td>
<td>Deprecated - use GetLineItemsAtIndexExtDim.</td>
</tr>
<tr>
<td>GetLineItemsAtIndexExtDim</td>
<td>Returns the data and descriptions for a cell’s line items, using the indexes that identify the cell.</td>
</tr>
<tr>
<td>GetNumCellsForData</td>
<td>Returns the number of cells that are in a period of a subcube and that contain data.</td>
</tr>
<tr>
<td>GetNumCellsForDescriptions</td>
<td>Returns the number of cells that are in a period of a subcube and that contain descriptions.</td>
</tr>
<tr>
<td>GetNumCellsForLineItems</td>
<td>Returns the number of cells that are in a period of a subcube and that contain line items.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetNumPeriodsInCubeForData</td>
<td>Returns the number of periods that are in a subcube and that contain cells with data.</td>
</tr>
<tr>
<td>GetNumPeriodsInCubeForDescriptions</td>
<td>Returns the number of periods that are in a subcube and that contain cells with descriptions.</td>
</tr>
<tr>
<td>GetNumPeriodsInCubeForLineItems</td>
<td>Returns the number of periods that are in a subcube and that contain cells with line items.</td>
</tr>
<tr>
<td>GetPeriodIndexFromPOVForData</td>
<td>Returns the index of a period within a subcube, using the subcube’s index and the period’s member ID. An index is returned only for periods with cells that contain data.</td>
</tr>
<tr>
<td>GetPeriodIndexFromPOVForDescriptions</td>
<td>Returns the index of a period within a subcube, using the subcube’s index and the period’s member ID. An index is returned only for periods with cells that contain descriptions.</td>
</tr>
<tr>
<td>GetPeriodIndexFromPOVForLineItems</td>
<td>Returns the index of a period within a subcube, using the subcube’s index and the period’s member ID. An index is returned only for periods with cells that contain line items.</td>
</tr>
<tr>
<td>GetPeriodPOVFromIndexForData</td>
<td>Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain data.</td>
</tr>
<tr>
<td>GetPeriodPOVFromIndexForDescriptions</td>
<td>Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain descriptions.</td>
</tr>
<tr>
<td>GetPeriodPOVFromIndexForLineItems</td>
<td>Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain line items.</td>
</tr>
<tr>
<td>GetPMErrorRecordCount</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetPMRecordCount</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetRecordFromPMBuffer</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetRecordFromPMErrorBuffer</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetSortedNature</td>
<td>For internal use</td>
</tr>
<tr>
<td>InsertDataAtEnd</td>
<td>Deprecated - use InsertDataAtEndExtDim.</td>
</tr>
<tr>
<td>InsertDataAtEndExtDim</td>
<td>Inserts data for a cell at the bottom of an HsvMDDataBufferLite object.</td>
</tr>
<tr>
<td>InsertDescriptionAtEnd</td>
<td>Deprecated - use InsertDescriptionAtEndExtDim.</td>
</tr>
<tr>
<td>InsertDescriptionAtEndExtDim</td>
<td>Inserts a cell description at the bottom of an HsvMDDataBufferLite object.</td>
</tr>
<tr>
<td>InsertLineItemsAtEnd</td>
<td>Deprecated - use InsertLineItemsAtEndExtDim.</td>
</tr>
<tr>
<td>InsertLineItemsAtEndExtDim</td>
<td>Inserts line items for a cell at the bottom of an HsvMDDataBufferLite object.</td>
</tr>
<tr>
<td>InsertRecordIntoPMBuffer</td>
<td>For internal use</td>
</tr>
<tr>
<td>InsertRecordIntoPMErrorBuffer</td>
<td>For internal use</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>RemoveAll</td>
<td>Clears all the cells from an HsvMDDataBufferLite object.</td>
</tr>
<tr>
<td>SetCheckLineItemDetailsForCaseInsensitiveDuplicates</td>
<td>Specifies whether the system should check for case-insensitive duplicate line item descriptions.</td>
</tr>
<tr>
<td>SetData</td>
<td>Deprecated - use SetDataExtDim.</td>
</tr>
<tr>
<td>SetDataExtDim</td>
<td>Inserts a cell’s data into an HsvMDDataBufferLite object.</td>
</tr>
<tr>
<td>SetDescription</td>
<td>Deprecated - use SetDescriptionExtDim.</td>
</tr>
<tr>
<td>SetDescriptionExtDim</td>
<td>Inserts a cell’s description into an HsvMDDataBufferLite object.</td>
</tr>
<tr>
<td>SetGrowByAmount</td>
<td>Changes the amount of memory that the HsvMDDataBufferLite object automatically allocates.</td>
</tr>
<tr>
<td>SetLineItems</td>
<td>Deprecated - use SetLineItemsExtDim.</td>
</tr>
<tr>
<td>SetLineItemsExtDim</td>
<td>Inserts a cell’s line items into an HsvMDDataBufferLite object.</td>
</tr>
<tr>
<td>SetSortedNature</td>
<td>For internal use</td>
</tr>
<tr>
<td>Sort</td>
<td>For internal use</td>
</tr>
</tbody>
</table>

**IHsvMDDataBufferLite Interface**

The IHsvMDDataBufferLite interface provides the `SetMinCubesInCache` method, which sets the number of minimum subcubes cached to RAM for the HsvMDDataBufferLite object. See “IHsvMDDataBufferLite Interface” on page 656.

**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 is populated.

Create HsvTransactionData object references with the following syntax. You must include the `New` keyword, otherwise an error occurs.

```vba
Dim cHsvTransData As HsvTransactionData
Set cHsvTransData = New HSVMDARRAYSLib.HsvTransactionData
```

The following table summarizes the HsvTransactionData object’s methods.
Table 33  HsvTransactionData Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeginDataEnum</td>
<td>Begins an enumeration of an HsvTransactionData object, and returns a count of the items with which HsvData.GetTransactionData populated the object.</td>
</tr>
<tr>
<td>BeginQueryEnum</td>
<td>For internal use</td>
</tr>
<tr>
<td>EndDataEnum</td>
<td>Ends an HsvTransactionData object enumeration. You should call EndDataEnum after you have finished working with an enumeration that was started with BeginDataEnum.</td>
</tr>
<tr>
<td>EndQueryEnum</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetFixedDimensionMembers</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetQueryItem</td>
<td>Deprecated - use GetQueryItemExtDim.</td>
</tr>
<tr>
<td>GetQueryItemExtDim</td>
<td>For internal use</td>
</tr>
<tr>
<td>GetTransactionData</td>
<td>Deprecated - use GetTransactionDataExtDim.</td>
</tr>
<tr>
<td>GetTransactionDataExtDim</td>
<td>Returns source and destination data for a transaction, using 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.</td>
</tr>
<tr>
<td>Initialize</td>
<td>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.</td>
</tr>
<tr>
<td>SetAccessRight</td>
<td>For internal use</td>
</tr>
<tr>
<td>SetQueryItem</td>
<td>Deprecated - use SetQueryItemExtDim.</td>
</tr>
<tr>
<td>SetQueryItemExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>SetTransactionData</td>
<td>Deprecated - use SetTransactionDataExtDim.</td>
</tr>
<tr>
<td>SetTransactionDataExtDim</td>
<td>For internal use</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetItem</td>
<td>Deprecated - use GetItemExtDim.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetItemExtDim</td>
<td>Returns the member IDs of the dimension members for an index in an HsvMDIndexList object.</td>
</tr>
<tr>
<td>GetNumItems</td>
<td>Returns the number of indexes in an HsvMDIndexList object. Use this to loop with GetItem.</td>
</tr>
</tbody>
</table>

**HsvICTransactionsData Object Overview**

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

See “HsvICTransactionsData Object Methods” on page 663. The following table summarized the HsvICTransactionsData object’s methods.

**Table 35  HsvICTransactionsData Object Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddAccountCustomCombination</td>
<td>Deprecated - use AddAccountCustomCombinationExtDim.</td>
</tr>
<tr>
<td>AddAccountCustomCombinationExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>AddICTransactionData</td>
<td>Deprecated - use AddICTransactionDataExtDim.</td>
</tr>
<tr>
<td>AddICTransactionDataExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>AddQueryDimensionMemberID</td>
<td>For internal use.</td>
</tr>
<tr>
<td>AddQueryField</td>
<td>For internal use.</td>
</tr>
<tr>
<td>AddQueryFieldItem</td>
<td>For internal use.</td>
</tr>
<tr>
<td>AddQueryOrderField</td>
<td>For internal use.</td>
</tr>
<tr>
<td>BeginDataEnum</td>
<td>Begins the enumeration of an HsvICTransactionsData instance’s array of transactions, and returns a count of the transactions that the array contains.</td>
</tr>
<tr>
<td>BeginQueryFieldEnum</td>
<td>For internal use.</td>
</tr>
<tr>
<td>BeginQueryOrderEnum</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EndDataEnum</td>
<td>Cleans up an HsvICTransactionsData instance; call EndDataEnum after you finish working with an enumeration of transactions for which you called BeginDataEnum.</td>
</tr>
<tr>
<td>EndQueryFieldEnum</td>
<td>For internal use.</td>
</tr>
<tr>
<td>EndQueryOrderEnum</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetAccessRights</td>
<td>Gets the user’s read and write access rights to an intercompany transaction.</td>
</tr>
<tr>
<td>GetAccountCustomCombination</td>
<td>Deprecated - use GetAccountCustomCombinationExtDim.</td>
</tr>
<tr>
<td>GetAccountCustomCombinationExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>GetAccountCustomCombinationTotal</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetEntityPartnerOption</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetErrorStatus</td>
<td>Returns the HRESULT associated with a intercompany transaction.</td>
</tr>
<tr>
<td>GetFilterOptions</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetFixedDimensionMembers</td>
<td>Returns the member IDs of the Scenario, Year, and Period dimension members for an HsvICTransactionsData instance's transactions.</td>
</tr>
<tr>
<td>GetICTransactionCellExtDim</td>
<td>Returns the member IDs of an intercompany transaction's Entity, Intercompany Partner, Account, and Custom dimension members.</td>
</tr>
<tr>
<td>GetICTransactionData</td>
<td>Deprecated - use GetICTransactionDateExtDim.</td>
</tr>
<tr>
<td>GetICTransactionDateExtDim</td>
<td>Returns an intercompany transaction's details, including the transaction's amounts, currency, dimension members, and so on.</td>
</tr>
<tr>
<td>GetMatchCode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetNumTransactionsCached</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetPagingOption</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetPartnerAsEntityList</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetPartnerQueryDimensionMemberIDs</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetQueryDimensionMemberIDs</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetQueryDimensionMemberIDs</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetQueryFieldInformation</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetQueryFieldItem</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetQueryOrderField</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetTotalTransactions</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetTransGroupType</td>
<td>For internal use.</td>
</tr>
<tr>
<td>Initialize</td>
<td>Specifies the scenario, year, and period of the transactions that the HsvICTransactionsData instance contains.</td>
</tr>
<tr>
<td>InitializeSequenceIdMap</td>
<td>For internal use.</td>
</tr>
<tr>
<td>IsEntityInPartnerAsEntityList</td>
<td>For internal use.</td>
</tr>
<tr>
<td>IsICTransactionValidExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>RemoveQueryOrder</td>
<td>For internal use.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>SetAccessRights</td>
<td>Sets the user’s read and write access rights to an intercompany transaction.</td>
</tr>
<tr>
<td>SetEntityPartnerOption</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetErrorStatus</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetFilterOptions</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetICTransactionData</td>
<td>Deprecated - use SetICTransactionDataExtDim.</td>
</tr>
<tr>
<td>SetICTransactionDataExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetPagingOption</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetPartnerAsEntityList</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetPartnerQueryDimensionMemberIDs</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetQueryDimensionMemberIDs</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetTotalTransactions</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SortByCell</td>
<td>For internal use.</td>
</tr>
<tr>
<td>UnInitialize</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

### 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 115.

- The HsvNodeCube object provides access to node subcubes. The HsvNodeCube methods are listed in Table 37 on page 116.

### HsvCurrencyCube Object Overview

The HsvCurrencyCube object is used to access currency subcubes, which are described in “About Subcubes” on page 53. 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>
<thead>
<tr>
<th>Table 36 HsvCurrencyCube Object Methods</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
</tr>
<tr>
<td>BeginEnumerationOfStoredData</td>
</tr>
<tr>
<td>EndEnumerationOfStoredData</td>
</tr>
</tbody>
</table>
### Table 37  HsvNodeCube Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetCell</td>
<td><em>Deprecated</em> - use GetCellExtDim.</td>
</tr>
<tr>
<td>GetCellExtDim</td>
<td>Returns the data in and transaction status of a cell in a subcube.</td>
</tr>
<tr>
<td>GetFixedDimensionMembers</td>
<td>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.</td>
</tr>
<tr>
<td>GetOneCellFromStoredItem</td>
<td><em>Deprecated</em> - use GetOneCellFromStoredItemExtDim.</td>
</tr>
<tr>
<td>GetOneCellFromStoredItemExtDim</td>
<td>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.</td>
</tr>
<tr>
<td>GetPOVFromStoredItem</td>
<td><em>Deprecated</em> - use GetPOVFromStoredItemExtDim</td>
</tr>
<tr>
<td>GetPOVFromStoredItemExtDim</td>
<td>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.</td>
</tr>
</tbody>
</table>

See “HsvCurrencyCube Object Methods” on page 676.

**HsvNodeCube Object Overview**

The HsvNodeCube object is used to access node subcubes, which are described in “About Subcubes” on page 53. 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.
HsvDQI Type Library Overview

The HsvDQI type library provides methods for intelligent data retrieval.

HsvDQI Object Overview

The HsvDQI object contains methods for data retrieval. An HsvDQI object should be obtained from the HsvSession API get_HsvDQI() only.

Table 38  HsvDQI Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddEdgeDimension</td>
<td>Adds a dimension to the edge dimension.</td>
</tr>
<tr>
<td>AddEdgeDimensionID</td>
<td>Adds a dimension to the edge definition (input as IDs).</td>
</tr>
<tr>
<td>AddEdgeSource</td>
<td>Adds a source to the edge.</td>
</tr>
<tr>
<td>AddEdgeSources</td>
<td>Adds a set of sources to the edge.</td>
</tr>
<tr>
<td>AddSourceDimension</td>
<td>Adds a dimension to a source object.</td>
</tr>
<tr>
<td>AddSourceDimensionID</td>
<td>Adds a dimension to a source object definition (input as IDs).</td>
</tr>
<tr>
<td>AddToSourceJoinTypesForDimensions</td>
<td>Adds Join types to be used between dimensions in the source to resolve source definition.</td>
</tr>
<tr>
<td>CollapseAll</td>
<td>Collapses all nodes in a specified edge completely for a grid.</td>
</tr>
<tr>
<td>CollapseGrid</td>
<td>Collapses a specific member in a grid.</td>
</tr>
<tr>
<td>CollapseGridDimension</td>
<td>Collapses a dimension in an index in an edge of the grid.</td>
</tr>
<tr>
<td>CollapseGridDimensionID</td>
<td>Collapses a dimension in an index in an edge of the grid (input as IDs).</td>
</tr>
<tr>
<td>CollapseGridID</td>
<td>Collapses a specific member in the edge of the grid (input as IDs).</td>
</tr>
<tr>
<td>CreateEdge</td>
<td>Creates a DQI edge with specified type for a grid.</td>
</tr>
<tr>
<td>CreateGrid</td>
<td>Creates a DQI Grid instance in an application.</td>
</tr>
<tr>
<td>CreatePageControl</td>
<td>Creates a DQI page control object for a grid.</td>
</tr>
<tr>
<td>CreateSource</td>
<td>Creates a DQI source with specified type.</td>
</tr>
<tr>
<td>DefineGrid</td>
<td>Validates and defines the grid definition.</td>
</tr>
<tr>
<td>DeleteGrid</td>
<td>Deletes the grid and related objects from DQI internally.</td>
</tr>
<tr>
<td>ExpandAll</td>
<td>Expands all nodes in specified edge completely for a specified grid.</td>
</tr>
<tr>
<td>ExpandGrid</td>
<td>Expands a specific member in a grid.</td>
</tr>
<tr>
<td>ExpandGridDimension</td>
<td>Expands a dimension in an index in the specified edge of the grid.</td>
</tr>
<tr>
<td>ExpandGridDimensionID</td>
<td>Expands a dimension in an index in the specified edge of the grid (input as IDs).</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ExpandGridID</td>
<td>Expands a specific member in the edge of the grid (input as IDs).</td>
</tr>
</tbody>
</table>
| FreezeEdgeDefinition                        | Freezes the edge definition after which dimensions cannot be added to the edge.
<p>| GetAllDataForGridPageDimensionIndex         | Returns the data for the specified page dimension based on the current state of the grid. |
| GetAllVisibleDimensionDetails               | Returns all the visible headers for the edge of the grid with IDs and labels in output for members. |
| GetGridCompleteDimensionDetails             | Returns all the visible and hidden headers for the edge of the grid with IDs and labels in output for members. |
| GetGridDetails                              | Returns the number of rows, columns, and page dimension entries in the grid. |
| GetGridDimensionDetails                     | Returns the headers for the specified edge of grid.                         |
| GetGridDimensionDetailsIDsAndLabels         | Returns the headers for the requested edge of grid with IDs and labels in output for members. |
| GetIndexesRangeForSourcesOnEdge             | Returns the start and end indexes to figure out the range of indexes for each logical source on the edge. |
| GetPageControlDetails                       | Returns the page directions to be enabled for the page control for a grid.   |
| GetPOFForIndexes                            | Returns the POV for specified combination of x,y indexes in grid.            |
| GetPOVIDsForIndexes                         | Returns the POV for the specified cell x, y in the grid (output as IDs).     |
| GetSourceIDForIndexes                       | Returns the row and column logical source IDs for the specified set of x and y coordinates. |
| JoinSources                                 | Joins two sources with specified Join type to form the output source definition. |
| JoinSourcesInSequence                       | Joins two or more sources in order based on specified join types to form the output source definition. |
| SelectGridPageDimensionIndex                | Selects the active page dimension index to determine the active page dimension. |
| SelectMemberSourceAncestors                 | Selects ancestors of a specified member to a source of type member source for the active dimension or last added dimension. |
| SelectMemberSourceAncestorsID               | Selects ancestors of a specified member to member source for the active dimension (input as IDs). |
| SelectMemberSourceBase                      | Selects base members of a specified member to a source of type member source for the active dimension or last added dimension. |
| SelectMemberSourceBaseID                    | Selects base members of a specified member to member source for the active dimension (input as IDs). |
| SelectMemberSourceChildren                  | Selects a member and its children to a member source for active dimension.   |
| SelectMemberSourceChildrenID                | Selects a member and its children to a member source for active dimension (input as IDs). |
| SelectMemberSourceDescendants               | Selects descendants of a specified member to a source of type member source for the active dimension or last added dimension. |</p>
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SelectMemberSourceHierarchy</td>
<td>Selects a hierarchy of members to a member source for active dimension.</td>
</tr>
<tr>
<td>SelectMemberSourceHierarchyExpanded</td>
<td>Selects a hierarchy of members to member source for the active dimension and sets them to expanded state.</td>
</tr>
<tr>
<td>SelectMemberSourceHierarchyExpandedID</td>
<td>Selects a hierarchy of members to a member source for the active dimension with expanded state (input as IDs).</td>
</tr>
<tr>
<td>SelectMemberSourceHierarchyID</td>
<td>Selects a hierarchy of members to a member source for the active dimension (input as IDs).</td>
</tr>
<tr>
<td>SelectMemberSourceMemberID</td>
<td>Selects a single member to member source for the active dimension (input as IDs).</td>
</tr>
<tr>
<td>SelectMemberSourceMemberIDs</td>
<td>Selects set of members to a source of type member source for active dimension (input as IDs).</td>
</tr>
<tr>
<td>SelectMemberSourceMembers</td>
<td>Selects a set of members to a source of type member source for active dimension.</td>
</tr>
<tr>
<td>SelectMemberSourceParentsID</td>
<td>Selects parents of a specified member to member source for the active dimension (input as IDs).</td>
</tr>
<tr>
<td>SelectMemberSourceParents</td>
<td>Selects parents of a specified member to a source of type for the active dimension or last added dimension.</td>
</tr>
<tr>
<td>SelectMemberSourceUserDefinedMemberList</td>
<td>Selects the members in a specified list into a member source for active dimension.</td>
</tr>
<tr>
<td>SelectMemberSourceUserDefinedMemberList2</td>
<td>Selects the members in a specified list into a member source for active dimension.</td>
</tr>
<tr>
<td>SelectPageSourceMemberIDsForDim</td>
<td>Adds the dimension and set of members (input as IDs) to a page source definition.</td>
</tr>
<tr>
<td>SelectPageSourceMembersForDim</td>
<td>Adds the dimension and set of members to a page source.</td>
</tr>
<tr>
<td>SelectSliceSourceMemberIDForDim</td>
<td>Adds the dimension and member to a slice source definition input as IDs.</td>
</tr>
<tr>
<td>SelectSliceSourceMemberForDim</td>
<td>Adds the dimension and member to a slice source definition.</td>
</tr>
<tr>
<td>SetColsForPageControl</td>
<td>Sets the number of columns for the page control object.</td>
</tr>
<tr>
<td>SetEdgeRepeatedMetaDataSuppression</td>
<td>Sets the suppression of repeating metadata headers in the edge to on or off.</td>
</tr>
<tr>
<td>SetGridEdge</td>
<td>Sets the edge to a grid definition.</td>
</tr>
<tr>
<td>SetGridNavigationType</td>
<td>Sets the navigation type for a grid.</td>
</tr>
<tr>
<td>SetGridOptionDescriptionEnabled</td>
<td>Enables or disables the description in headers returned for a grid.</td>
</tr>
<tr>
<td>SetGridOptionDimensionExpansion</td>
<td>Sets the dimension expansion feature on or off for a grid.</td>
</tr>
<tr>
<td>SetGridOptionDisplayParentForEntityHierarchy</td>
<td>Disables the Parent,Child format in output for Entity members in hierarchy selection in output headers.</td>
</tr>
<tr>
<td>SetGridPageControl</td>
<td>Sets a PageControl to a grid definition.</td>
</tr>
<tr>
<td>SetGridSuppressionType</td>
<td>Sets a specific type of suppression on a row/column edge to on or off.</td>
</tr>
<tr>
<td>SetPageMovement</td>
<td>Sets the direction of page movement.</td>
</tr>
</tbody>
</table>
### Method Description

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>SetRowsForPageControl</td>
<td>Sets the number of rows for the page control object.</td>
</tr>
<tr>
<td>SetSourceNotSuppressible</td>
<td>Disables or overrides the suppression options set at grid or edge level for rows and columns resulting from the edge source.</td>
</tr>
<tr>
<td>SetSourceOptionFillDefaultParentForEntity</td>
<td>Sets the default parent for members during entity member selection in a specific source.</td>
</tr>
<tr>
<td>SetSourceSubType</td>
<td>Sets subtype for a source.</td>
</tr>
</tbody>
</table>

### HsvStarSchemaACM Type Library Overview


### HsvStarSchemaACM Object Overview

The HsvStarSchemaACM object enables you to work with Extended Analytics data 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 725.

The following table lists the HsvStarSchemaACM object’s methods.

#### Table 39  HsvStarSchemaACM Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CreateStarSchema</td>
<td><em>Deprecated</em> - use CreateStarSchemaExtDim.</td>
</tr>
<tr>
<td>CreateStarSchemaExtDim</td>
<td>Creates or updates a star schema by exporting data for all cells that intersect the specified dimension members.</td>
</tr>
<tr>
<td>CreateStarSchemaAndReturnTaskID</td>
<td><em>Deprecated</em> - use CreateStarSchemaExtDim.</td>
</tr>
<tr>
<td>CreateStarSchemaFromHTTP</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>DeleteStarSchema</td>
<td>Deletes a star schema from a database.</td>
</tr>
<tr>
<td>EnumApplicationStarSchemas</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>EnumRegisteredDSNs</td>
<td>Returns an array of the Extended Analytics Data Source Names registered on the application server.</td>
</tr>
<tr>
<td>GetAsynchronousTaskStatus</td>
<td>Returns status information for the asynchronous thread launched by CreateStarSchemaExtDim.</td>
</tr>
<tr>
<td>GetExtractLogData</td>
<td>Returns a string that provides a log of the HsvStarSchemaACM instance’s most recent call to CreateStarSchemaExtDim.</td>
</tr>
<tr>
<td>GetPersistedSettings</td>
<td><em>Deprecated</em> - use GetPersistedSettingsExtDim.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------</td>
</tr>
<tr>
<td>GetPersistedSettingsExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>QuitAsynchronousTask</td>
<td>Terminates the thread launched by CreateStarSchemaExtDim.</td>
</tr>
<tr>
<td>SetDefaultTablePrefix</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetPersistedSettings</td>
<td>Deprecated - use Set PersistedSettingsExtDim.</td>
</tr>
<tr>
<td>SetPersistedSettingsExtDim</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetSession</td>
<td>Points to the HsvSession object that represents the connection to the application. You must call SetSession before using the other HsvStarSchemaACM methods.</td>
</tr>
<tr>
<td>TestSQLConnection</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

**IHsvStarSchemaTemplates Interface Overview**

The IHHSVStarSchemaTemplates interface enables you to work with Extended Analytics templates. For information on setting IHHSVStarSchemaTemplates object references and details on the interface’s methods, see “IHHSVStarSchemaTemplates Interface” on page 732.

The following table lists the IHHSVStarSchemaTemplates interface’s methods.

**Table 40  IHHSVStarSchemaTemplates Interface Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DeleteTemplate</td>
<td>Deletes a Extended Analytics template.</td>
</tr>
<tr>
<td>EnumTemplates</td>
<td>Returns the names of the application’s Extended Analytics templates.</td>
</tr>
<tr>
<td>GetTemplate</td>
<td>Returns an XML string that contains an Extended Analytics template’s definition.</td>
</tr>
<tr>
<td>SetTemplate</td>
<td>Creates an Extended Analytics template, using an XML string that contains the template definition.</td>
</tr>
</tbody>
</table>

**HsvICM Type Library**

The HsvICM type library exposes intercompany transaction processing and administrative features, and contains the HsvICM object and the IHsvAdminICM interface. See “HsvICM Object Overview” on page 121 and “IHsvAdminICM Interface Overview” on page 123.

**HsvICM Object Overview**

The HsvICM object enables you to work with intercompany transactions. For example, you can create, match, and unmatch intercompany transactions. For information on setting HsvICM object references and details on the object’s methods, see “HsvICM Object Methods” on page 737.

The following table summarizes the HsvICM object’s methods.
<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AllMVTransPosted</td>
<td>Indicates whether all transactions for a scenario, year, period, and entity that are in a matched state are posted.</td>
</tr>
<tr>
<td>CheckReportSecurity</td>
<td>For internal use.</td>
</tr>
<tr>
<td>CreateICTransaction</td>
<td>Deprecated - use CreateICTransactionExtDim.</td>
</tr>
<tr>
<td>CreateICTransactionExtDim</td>
<td>Creates an intercompany transaction for a cell.</td>
</tr>
<tr>
<td>DeleteICReasonCode</td>
<td>Deletes a reason code.</td>
</tr>
<tr>
<td>DoesCellSupportICTransactionDetail</td>
<td>Deprecated - use DoesCellSupportICTransactionDetailExtDim.</td>
</tr>
<tr>
<td>DoesCellSupportICTransactionDetailExtDim</td>
<td>Indicates whether a cell supports intercompany transactions.</td>
</tr>
<tr>
<td>GetColumnFilter</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetCurrencyInfo</td>
<td>Returns a currency's label, translation operator, and scale.</td>
</tr>
<tr>
<td>GetEntityCurrencyID</td>
<td>Returns the currency ID of an entity's default currency.</td>
</tr>
<tr>
<td>GetEntitiesContacts</td>
<td>Returns the usernames and security identifiers of the users to be alerted for a scenario and intercompany transaction-related event for the specified entities.</td>
</tr>
<tr>
<td>GetICReasonCodeID</td>
<td>Gets the ID of a reason code.</td>
</tr>
<tr>
<td>GetICReasonCodeLabel</td>
<td>Returns a reason code from the code's internal ID.</td>
</tr>
<tr>
<td>GetICReasonCodes</td>
<td>Returns an application's reason codes and their corresponding IDs and descriptions.</td>
</tr>
<tr>
<td>GetICTransactions</td>
<td>Populates an HsvICTransactionsData object reference with intercompany transactions for the scenario, year, and period specified with the HsvICTransactionsData method Initialize.</td>
</tr>
<tr>
<td>GetMonitorICDetails</td>
<td>Returns counts of intercompany transactions that have various posting and matching statuses for the specified Scenario, Year, Period, and Entity dimension members.</td>
</tr>
<tr>
<td>GetMonitorICSummary</td>
<td>Returns the number of Entity dimension members that have various combinations of locking and process statuses for the Scenario, Year, Period, and Entity dimension members.</td>
</tr>
<tr>
<td>GetMonitorICTransactions</td>
<td>Indicates whether one or more Entity dimension members have intercompany transactions for 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.</td>
</tr>
<tr>
<td>GetRowFilter</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetTransCurrencyID</td>
<td>Returns the currency ID of a currency.</td>
</tr>
<tr>
<td>GetUnMatchTransactions</td>
<td>Populates an HsvICTransactionsData object reference with unmatched intercompany transactions for the scenario, year, and period specified with the HsvICTransactionsData method Initialize.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IsOneSideOfTransactionGroupWriteable</td>
<td>Indicates whether the user has write access to either the entity or Intercompany Partner cell of all transactions in a range of an HsvICTransactionsData object reference’s transactions.</td>
</tr>
<tr>
<td>MatchAutoAccounts</td>
<td>Matches intercompany transactions by account.</td>
</tr>
<tr>
<td>MatchAutoIDs</td>
<td>Matches intercompany transactions by Transaction ID or Reference ID.</td>
</tr>
<tr>
<td>NoTransMatchedOrPosted</td>
<td>Indicates whether matched or posted transactions exist for a scenario, year, and period.</td>
</tr>
<tr>
<td>ProcessAllICTrans</td>
<td>Deletes, posts, unposts, or unmatches all transactions for a scenario, year, and period.</td>
</tr>
<tr>
<td>ProcessICTransactions</td>
<td>Deprecated - use ProcessICTransactionsExtDim</td>
</tr>
<tr>
<td>ProcessICTransactionsExtDim</td>
<td>Deletes, posts, unposts, or unmatches the specified intercompany transactions for a scenario, year, and period.</td>
</tr>
<tr>
<td>SaveColumnFilter</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SaveICReasonCode</td>
<td>Creates a reason code.</td>
</tr>
<tr>
<td>SaveICTransaction</td>
<td>Deprecated - use SaveICTransactionExtDim</td>
</tr>
<tr>
<td>SaveICTransactionExtDim</td>
<td>Updates an existing transaction.</td>
</tr>
<tr>
<td>SaveRowFilter</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetReasonCodeToICTransactions</td>
<td>Sets the reason code for one or more intercompany transactions.</td>
</tr>
<tr>
<td>UnMatchICTransactions</td>
<td>Unmatches transactions to which the specified match codes are assigned.</td>
</tr>
</tbody>
</table>

**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 755.**

The following table summarizes the IHsvAdminICM interface’s methods.

**Table 42  IHsvAdminICM Interface Methods**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CheckSecurityForICExtract</td>
<td>Indicates whether the connected user has security rights for extracting intercompany transactions.</td>
</tr>
<tr>
<td>CloseICPeriod</td>
<td>Closes a period for intercompany transactions.</td>
</tr>
<tr>
<td>GetICEntitiesLockStatus</td>
<td>Indicates whether any entities are locked for a scenario, year, and period.</td>
</tr>
<tr>
<td>GetLockStatusICEntities</td>
<td>Deprecated - use GetICEntitiesLockStatus.</td>
</tr>
<tr>
<td>GetLockStatusICEntity</td>
<td>Indicates whether an entity is locked or unlocked for a scenario, period, and year.</td>
</tr>
<tr>
<td>Method</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>GetSettingsICPeriod</td>
<td>Returns the period status, matching tolerance, and Match/Validate Before Post settings for a scenario, year, and period.</td>
</tr>
<tr>
<td>GetSettingsICPeriods</td>
<td>Returns the period status, matching tolerance, and Match/Validate Before Post settings of multiple periods for a scenario and year.</td>
</tr>
<tr>
<td>LoadICTransactions</td>
<td>Deprecated - use LoadICTransactionsExtDim</td>
</tr>
<tr>
<td>LoadICTransactionsExtDim</td>
<td>Loads or scans intercompany transactions from arrays that specify the transactions’ dimension members and data.</td>
</tr>
<tr>
<td>LockICEntity</td>
<td>Locks an entity for a scenario, year, and period.</td>
</tr>
<tr>
<td>OpenICPeriod</td>
<td>Opens a period for a scenario and year, and specifies the period’s matching tolerance and Match/Validate Before Post settings.</td>
</tr>
<tr>
<td>OpenICPeriod2</td>
<td>Opens a period for a scenario and year, and specifies the period’s Match/Validate Before Post settings and absolute or percentage matching tolerance.</td>
</tr>
<tr>
<td>SavePeriodsSettings</td>
<td>For internal use.</td>
</tr>
<tr>
<td>UnlockICEntity</td>
<td>Unlocks an entity for a scenario, year, and period.</td>
</tr>
<tr>
<td>UpdatePeriodSettings</td>
<td>Updates an open period’s matching tolerance and Match/Validate Before Post settings for a scenario and year.</td>
</tr>
<tr>
<td>UpdatePeriodSettings2</td>
<td>Updates an open period’s Match/Validate Before Post settings and absolute or percentage matching tolerance for a scenario and year.</td>
</tr>
</tbody>
</table>

### 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>
<thead>
<tr>
<th>Information Being Loaded/Extracted</th>
<th>Type Library</th>
</tr>
</thead>
<tbody>
<tr>
<td>Security</td>
<td>HsvSecurityLoadACV</td>
</tr>
<tr>
<td>Metadata (extracts only)</td>
<td>HsvMetadataLoadACV</td>
</tr>
<tr>
<td>Rules and member lists</td>
<td>HsvRulesLoadACV</td>
</tr>
<tr>
<td>Data</td>
<td>HsvcDataLoad</td>
</tr>
<tr>
<td>Journals and journal templates</td>
<td>HsvJournalLoadACV</td>
</tr>
</tbody>
</table>

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. 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 125
- “IHsvLoadExtractOptions Interface Overview” on page 125
- “IHsvLoadExtractOption Interface Overview” on page 126
- “HsvSecurityLoadACV Type Library Overview” on page 126
- “HsvMetadataLoadACV Type Library Overview” on page 127
- “HsvRulesLoadACV Type Library Overview” on page 128
- “HsvcDataLoad Type Library Overview” on page 129
- “HsvJournalLoadACV Type Library Overview” on page 130

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 load or extract operation.

**Tip:** For a more detailed introduction to the IHsvLoadExtractOptions and IHsvLoadExtractOption interfaces, see “Load and Extract Option Interfaces” on page 765.

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 44  IHsvLoadExtractOptions Interface Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Count</td>
<td>Returns a count of the IHsvLoadExtractOption instances available for setting load or extract options. This property returns a count of the available load or extract options.</td>
</tr>
<tr>
<td>Item</td>
<td>Returns object references to instances of the IHsvLoadExtractOption interface. This property provides access to the available load or extract options.</td>
</tr>
</tbody>
</table>

For detailed descriptions of these properties, see “IHsvLoadExtractOptions Interface Properties” on page 766.

**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. See “Item” on page 766.

The following table lists the properties of the IHsvLoadExtractOption interface.

Table 45  IHsvLoadExtractOption Interface Properties

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

For detailed descriptions of these properties, see “IHsvLoadExtractOption Interface Properties” on page 767.

**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 775
- “Extracting Security Information” on page 776

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

The following table lists the HsvSecurityLoadACV object’s properties, and Table 47 lists the HsvSecurityLoadACV object’s methods.

**Table 46** HsvSecurityLoadACV Object Properties

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

The above properties are also members of other type libraries, and are described in detail in “Common Load and Extract Properties” on page 770.

The following table summarizes the HsvSecurityLoadACV object’s methods.

**Table 47** HsvSecurityLoadACV Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract</td>
<td>Extracts security information to a text file on the client computer.</td>
</tr>
<tr>
<td>Load</td>
<td>Loads security information from a text file on a client computer.</td>
</tr>
<tr>
<td>SetSession</td>
<td>Points to the HsvSession object for the application in which security information is being loaded or extracted.</td>
</tr>
</tbody>
</table>

**Caution!** You must call `SetSession` before using any of the HsvSecurityLoadACV 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 771. The HsvSecurityLoadACV’s `Load` method is slightly different than the `Load` method of other load- and extract-related type libraries; see “Load” on page 779.

**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 782.
Note: Assign HsvMetadataLoadACV object references with the `Set` keyword.

The following table lists the HsvMetadataLoadACV object’s properties, and Table 49 lists the HsvMetadataLoadACV object’s methods.

Table 48  HsvMetadataLoadACV Object Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtractOptions</td>
<td>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>.</td>
</tr>
<tr>
<td>LoadOptions</td>
<td>Returns an object reference to an instance of the IHsvLoadExtractOptions interface for a load operation. You must set this property before calling the <code>Load</code> method for any of the applicable objects.</td>
</tr>
</tbody>
</table>

The previously mentioned properties are also members of other type libraries, and are described in detail in “Common Load and Extract Properties” on page 770.

The following table summarizes the HsvMetadataLoadACV object’s methods.

Table 49  HsvMetadataLoadACV Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract</td>
<td>Extracts metadata to a text file on the client computer.</td>
</tr>
<tr>
<td>Load</td>
<td>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.</td>
</tr>
<tr>
<td>LoadWithAccessCode</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SetSession</td>
<td>Points to the HsvSession object for the application in which metadata is being extracted. You must call <code>SetSession</code> before using any of the HsvMetadataLoadACV object’s other properties or methods.</td>
</tr>
</tbody>
</table>

The previously mentioned methods are also members of other type libraries, and are described in “Common Load and Extract Methods” on page 771.

HsvRulesLoadACV Type Library Overview

The HsvRulesLoadACV type library contains 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 50  HsvRulesLoadACV Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtractCalcRules</td>
<td>Extracts rules from an application into a text file on the client computer.</td>
</tr>
</tbody>
</table>
### HsvcDataLoad Object Methods

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


### 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 797
- “Extracting Data” on page 798

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

The following table lists the HsvcDataLoad object’s properties, and Table 52 lists the HsvcDataLoad object’s methods.

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtractOptions</td>
<td>Returns an object reference to an instance of the IHsvLoadExtractOptions interface for an extraction operation. You must set this property before extracting data with Extract.</td>
</tr>
</tbody>
</table>
The above properties are also members of other type libraries, and are described in detail in “Common Load and Extract Properties” on page 770.

The following table summarizes the HsvcDataLoad object’s methods.

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

Extract, Load, and SetSession are also members of other type libraries, and are described in “Common Load and Extract Methods” on page 771. The other methods are described in “HsvcDataLoad Object Methods” on page 805.

**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 807
- “Extracting Journals” on page 807

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

The following table lists the HsvJournalLoadACV object’s properties, and Table 54 lists the HsvJournalLoadACV object’s methods.
Table 53  HsvJournalLoadACV Object Properties

<table>
<thead>
<tr>
<th>Property</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ExtractOptions</td>
<td>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 Extract.</td>
</tr>
<tr>
<td>ExtractOptionsEx</td>
<td>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.</td>
</tr>
<tr>
<td>LoadOptions</td>
<td>Returns an object reference to an instance of the IHsvLoadExtractOptions interface for a load operation. You must set this property before loading journals with Load.</td>
</tr>
</tbody>
</table>

The above properties are also members of other type libraries, and are described in detail in “Common Load and Extract Properties” on page 770.

The following table summarizes the HsvJournalLoadACV object’s methods.

Table 54  HsvJournalLoadACV Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract</td>
<td>Extracts journals and journal templates to a text file on the client computer.</td>
</tr>
<tr>
<td>ExtractEx</td>
<td>Extracts journals from an application into a text file on the client computer, applying the filtering criteria set with ExtractOptionsEx.</td>
</tr>
<tr>
<td>Load</td>
<td>Loads journals and templates from a text file on the client computer.</td>
</tr>
</tbody>
</table>
| SetSession | Points to the HsvSession object for the application in which data is being loaded or extracted.  
   **Caution!** You must call SetSession 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 771.

**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 55 on page 131 and Table 56 on page 132.

The following table lists the HsvPOVSelection object’s methods. See “HsvPOVSelection Object Methods” on page 818.

Table 55  HsvPOVSelection Object Methods

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

The following table lists the HsvPOVSelection control’s events. See “HsvPOVSelection Object Events” on page 829.

**Table 56  HsvPOVSelection Object Events**

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CheckBoxChanged</td>
<td>Launches when the user selects or clears a member’s check box.</td>
</tr>
<tr>
<td>DimensionWasSelected</td>
<td>Launches when the tab for a dimension is selected by either the user or a call to <code>SelectDimension</code>.</td>
</tr>
<tr>
<td>MultiCheckBoxChanged</td>
<td>Launches when the user clicks either the button to select all check boxes or the button to clear all.</td>
</tr>
<tr>
<td>SelectionChanged</td>
<td>Launches when the user selects a member’s label.</td>
</tr>
</tbody>
</table>
HsvResourceManager Type Library Overview

The HsvResourceManager type library provides an interface to the Resource Manager. The Resource Manager exposes error message strings. If a release is localized, error message strings are localized.

For information on setting HsvResourceManager object references and details on the object’s methods, see “HsvResourceManager Object Methods” on page 834.

The following table lists the HsvResourceManager object’s methods.

Table 57  HsvResourceManager Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetAvailableLanguages</td>
<td>Returns arrays that identify the languages for which resources are available.</td>
</tr>
<tr>
<td>GetCurrentVersion</td>
<td>Returns the version number of Financial Management that is displayed to end-users.</td>
</tr>
<tr>
<td>GetCurrentVersionInUserDisplayFormat</td>
<td>Returns the version number of Financial Management that is displayed to end-users.</td>
</tr>
<tr>
<td>GetFormattedDateTime</td>
<td>Converts a timestamp to a formatted string.</td>
</tr>
<tr>
<td>GetFormattedDateTimeForLanguage</td>
<td>Returns a string representation of a double-byte date/time value.</td>
</tr>
<tr>
<td>GetFormattedError</td>
<td>Returns two strings in a 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.</td>
</tr>
<tr>
<td>GetFormattedErrorWithLineFeed</td>
<td>Returns two strings in a 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.</td>
</tr>
<tr>
<td>GetFormattedResourceString</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetHelpDirectoryForLanguageID</td>
<td>Returns the name of the subdirectory that contains the Financial Management help system for a language.</td>
</tr>
<tr>
<td>GetHFMLanguageIdFromUserLanguages</td>
<td>Returns the Financial Management ID for the language identified by a language code.</td>
</tr>
<tr>
<td>GetLanguageCountryCodeFromLanguageId</td>
<td>Returns the two-letter language code for a language ID.</td>
</tr>
<tr>
<td>GetLocaleIdFromLanguageId</td>
<td>Returns a Windows locale ID for the Financial Management language ID.</td>
</tr>
<tr>
<td>GetResourceString</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetResourceStringFromHR</td>
<td>Returns the resource string for an HRESULT in the specified language.</td>
</tr>
<tr>
<td>GetUserDisplayDateTimeFormats</td>
<td>Returns an array of strings that describe the date and time formats into which Financial Management can convert timestamps.</td>
</tr>
<tr>
<td>GetUserLanguageFromHFMLanguageId</td>
<td>Returns the two-letter language code for the language represented by a Financial Management language ID.</td>
</tr>
<tr>
<td>GetWindowsDateFormat</td>
<td>Returns the Windows date separator character and short date format for the computer.</td>
</tr>
<tr>
<td>GetWindowsDateFormatForLocale</td>
<td>Returns the Windows date format for the locale.</td>
</tr>
</tbody>
</table>
**Method** | **Description**
---|---
Initialize | Specifies the tier for which resources are needed.

**Caution!** You must call `Initialize` before calling any of the other `HsvResourceManager` methods.

### HfmSliceCOM Type Library

The HfmSliceCOM library provides objects needed to support configurable dimensionality. It contains four objects for representing single or multiple cell coordinates:

- **HfmPovCOM** - Represents a single cell coordinate using member IDs
- **HfmPovStrCOM** - Internal use.
- **HfmSliceCOM** - Represents multiple cell coordinates, using member IDs
- **HfmSliceStrCOM** - Internal use.

#### Table 58  HfmPovCOM Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>Clears the contents of the object. Custom members are erased, and all other dimension members are reset to MEMBERNOTUSED.</td>
</tr>
<tr>
<td>ConvertToFmHfmPovStrCOM</td>
<td>For internal use.</td>
</tr>
<tr>
<td>GetAllDims</td>
<td>Returns all of the dimension members.</td>
</tr>
<tr>
<td>GetCustomMemberAt</td>
<td>Returns the custom dimension ID and member ID at a specified index.</td>
</tr>
<tr>
<td>GetCustoms</td>
<td>Returns the dimension IDs and member IDs for the custom dimensions.</td>
</tr>
<tr>
<td>GetDimensionMember</td>
<td>Returns the member ID for a specified dimension (only for non-custom dimensions).</td>
</tr>
<tr>
<td>GetLeftHandDims</td>
<td>Returns the left-hand dimension members (Scenario, Year, Period, View, Entity, and Value).</td>
</tr>
<tr>
<td>GetMemberForCustomDim</td>
<td>Retrieves the member ID for a specified custom dimension ID.</td>
</tr>
<tr>
<td>GetRightHandDims</td>
<td>Returns the right-hand dimension members (Account, ICP, Customs).</td>
</tr>
<tr>
<td>SetAllDims</td>
<td>Sets all of the dimension members.</td>
</tr>
<tr>
<td>SetCustomMemberAt</td>
<td>Sets the custom dimension ID and member ID at a specified index.</td>
</tr>
<tr>
<td>SetCustoms</td>
<td>Sets the dimension IDs and member IDs for the custom dimensions.</td>
</tr>
<tr>
<td>SetDimensionMember</td>
<td>Sets the member ID for a specified dimension (only for non-custom dimensions).</td>
</tr>
<tr>
<td>SetLeftHandDims</td>
<td>Sets the left-hand dimension members (Scenario, Year, Period, View, Entity, and Value).</td>
</tr>
<tr>
<td>SetRightHandDims</td>
<td>Sets the right-hand dimension members (Account, ICP, Customs).</td>
</tr>
</tbody>
</table>
### Table 59  HfmSliceCOM Object Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Clear</td>
<td>Clears the contents of the slice object.</td>
</tr>
<tr>
<td>ConvertToHfmSliceStrCOM</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>EnumCustomDimIds</td>
<td>Returns a list of custom dimensions that are currently represented in the slice object.</td>
</tr>
<tr>
<td>GenerateCartesianProduct</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>GetFixedMember</td>
<td>Returns the member ID for a fixed dimension.</td>
</tr>
<tr>
<td>GetMemberArrayForDim</td>
<td>Returns a list of member IDs in the slice for a specified dimension.</td>
</tr>
<tr>
<td>GetMemberAt</td>
<td>Returns the member ID at a specific index in the slice for a dimension.</td>
</tr>
<tr>
<td>GetNumCells</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>GetNumMembers</td>
<td>Returns the number of members for a specified dimension.</td>
</tr>
<tr>
<td>GetPovAtIndex</td>
<td>Returns an HfmPovCOM object representing the cell coordinates at a specific index in the slice.</td>
</tr>
<tr>
<td>InitializeFromPovString</td>
<td>Initializes the object from a string containing a POV in standard notation.</td>
</tr>
<tr>
<td>SetFixedMember</td>
<td>Sets the member ID for a fixed dimension.</td>
</tr>
<tr>
<td>SetMemberArrayForDim</td>
<td>Sets a list of member IDs in the slice for specified dimension.</td>
</tr>
<tr>
<td>SetMemberAt</td>
<td>Sets the member ID at a specific index in the slice for a dimension.</td>
</tr>
</tbody>
</table>
This chapter provides links to examples of some basic tasks that you can perform.

**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. See “HsvSession Object Properties” on page 191.

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 registered for a client.
- `HsxServer.EnumDataSources` returns the names of the applications on a 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 dimension, dimension member attributes, a member’s label and member ID. See Chapter 9, “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 dimension.
  - Note: To specify the dimension for which the IHsvTreeInfo interface obtains information, use the HsvMetadata object’s properties. If the dimension to be queried is not known at design-time, use the Dimension property, demonstrated in the GetItemID example.
  - EnumAllMemberLabels returns the labels of a dimension’s members.
  - EnumAncestors returns a member’s ancestors.

- HsvMetadata.IsCustomMemberValidForAccount indicates whether a Custom dimension member is valid for a 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.
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, see “Error Handling and the HsvResourceManager Type Library” on page 833. 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 is localized.

The example for **GetFormattedError** shows how to get an error message localized into a specified language.
Tip: Financial Management also provides a utility with which you can look up descriptions of error numbers. See “Error Message Lookup Utility” on page 843.

### Loading and Extracting Application Information

The object model provides objects for loading and extracting security, member lists, rules, data, journals, and metadata. See Chapter 22, “Type Libraries for Loading and Extracting Information,” which includes these examples:

- “Example for Loading Security” on page 775
- “Example for Extracting Security” on page 777
- “Example for Loading Metadata” on page 782
- “Example for Extracting Metadata” on page 783
- “Example for Loading Data” on page 798
- “Example for Extracting Data” on page 799
- “Example for Loading Journals” on page 807
- “Example for Extracting Journals Without Filters” on page 808
- “Example for Filtered Journal Extractions” on page 810
The HfmSliceCOM library is new for Release 11.1.2.2, and provides objects needed to support configurable dimensionality.

The HfmSliceCOM library contains four objects for representing single or multiple cell coordinates:

- **HfmPovCOM** – Represents a single cell coordinate using member IDs
- **HfmPovStrCOM** – Internal use.
- **HfmSliceCOM** – Represents multiple cell coordinates, using member IDs
- **HfmSliceStrCOM** – Internal use.

### HfmPovCOM Object Properties

Represents a single cell coordinate using member IDs.

**Scenario**
Read/write property: The member ID of the Scenario dimension. Type: Long

**Year**
Read/write property: The member ID of the Year dimension. Type: Long

**Period**
Read/write property: The member ID of the Period dimension. Type: Long

**View**
Read/write property: The member ID of the View dimension. Type: Long
Entity
Read/write property: The member ID of the Entity dimension. Type: Long

Parent
Read/write property: The member ID of the parent Entity, if applicable. Type: Long

Value
Read/write property: The member ID of the Value dimension. Type: Long

Account
Read/write property: The member ID of the Account dimension. Type: Long

ICP
Read/write property: The member ID of the ICP dimension. Type: Long

NumCustomDims
Property: Returns the number of custom dimensions in the POV. Type: Long

HfmPovCOM Object Methods

Clear
Clears the contents of the object. Custom dimension members are erased, and all other
dimension members are reset to MEMBERNOTUSED.

Syntax
<br /> <HfmPovCOM>.Clear

Return Value
None.

Example
<br /> cHfmPovCOM.Clear

ConvertToHfmPovStrCOM

For internal use.
GetAllDims

Gets all of the dimension members.

Syntax

```csharp
<HfmPovCOM>.GetAllDims(plScenarioId, plYearId, plPeriodId, plViewId, plEntityId, plParentId, plValueId, plAccountId, plICPId, pvaralCustomDimIds, pvaralCustomMemberIds)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plScenarioId</td>
<td>Long. Returns the Scenario member</td>
</tr>
<tr>
<td>plYearId</td>
<td>Long. Returns the Year member</td>
</tr>
<tr>
<td>plPeriodId</td>
<td>Long. Returns the Period member</td>
</tr>
<tr>
<td>plViewId</td>
<td>Long. Returns the View member</td>
</tr>
<tr>
<td>plEntityId</td>
<td>Long. Returns the Entity member</td>
</tr>
<tr>
<td>plParentId</td>
<td>Long. Returns the Entity Parent member</td>
</tr>
<tr>
<td>plValueId</td>
<td>Long. Returns the Value member</td>
</tr>
<tr>
<td>plAccountId</td>
<td>Long. Returns the Account member</td>
</tr>
<tr>
<td>plICPId</td>
<td>Long. Returns the ICP member</td>
</tr>
<tr>
<td>pvaralCustomIds</td>
<td>Long Array. Returns the Custom dimension IDs</td>
</tr>
<tr>
<td>pvaralCustomMemberIds</td>
<td>Long Array. Returns the Custom member IDs</td>
</tr>
</tbody>
</table>

Return Value

Long. Number of custom dimensions.

Example

```csharp
Dim 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
Dim varalCustomIds As Variant, varalCustomMemberIds As Variant
Dim numCustoms As Long
numCustoms = cHfmPovCOM.GetAllDims(lScenario, lYear, lPeriod, lView, _
  lPeriod, lView, lEntity, lParent, lValue, lAccount, lICP, _
  varalCustomIds, varalCustomMemberIds
```

GetCustomMemberAt

Returns the custom dimension ID and member ID at a specified index.

Syntax

```csharp
<HfmPovCOM>.GetCustomMemberAt lIndex, plDimId, plMemberId
```
Arguments | Description 
---|---
Index | Long. Base-0 index 
plDimId | Long. Returns the custom dimension ID at the specified index 
plMemberId | Long. Returns the custom member ID at the specified index 

Return Value
None.

Example

```vbnet
Long lDimId As Long, lMemberId As Long
lMemberId = cHfmPovCOM.GetCustomMemberAt(0, lDimId)
```

**GetCustoms**

Gets the dimension IDs and member IDs for the custom dimensions.

Syntax

```vbnet
<HfmPovCOM>.GetCustoms pvaralDimIds, pvaralMemberIds
```

Arguments | Description 
---|---
pvaralDimIds | Long Array. Returns an array of custom dimension IDs. 
pvaralMemberIds | Long Array. Returns an array of custom member IDs. 

Return Value
Long. Number of custom dimensions.

Example

```vbnet
Dim n as Long
Dim aDimIds, aMbrIds
n = cPov.GetCustoms( aDimIds, aMbrIds)
```

**GetDimensionMember**

Gets the member ID for a specified dimension (only for non-custom dimensions).

Syntax

```vbnet
<HfmPovCOM>.GetDimensionMember lDimId
```
Argument Description

DimId Long. The desired dimension ID

Return Value
None. The member ID for the specified dimension.

Example

Dim lMemberId as Long
lMemberId = cHFmPovCOM.GetDimensionMember( DIMID_ENTITY)

GetLeftHandDims

Gets the left-hand dimension members (Scenario, Year, Period, View, Entity, Value).

Syntax

<HfmPovCOM>.GetLeftHandDims plScenario, plYear, plPeriod, plView, plEntity, plParent, plValue

Argument Description

lScenarioId Long. Scenario member
lYearId Long. Year member
lPeriodId Long. Period member
lViewId Long. View member
lEntityId Long. Entity member
lParentId Long. Entity Parent member
lValueId Long. Value member

Return Value
None.

Example

Dim lScenario As Long, lYear As Long, lPeriod As Long, lView As Long, _
lEntity As Long, lParent As Long, lValue As Long
cHFmPovCOM.GetLeftHandDims lScenario, lYear, lPeriod, lView, _
lPeriod, lView, lEntity, lParent, lValue

GetMemberForCustomDim

Retrieves the member ID for a specified custom dimension ID.
**GetMemberForCustomDim**

Gets the member ID for the specified custom dimension.

**Syntax**

```<HfmPovCOM>.GetMemberForCustomDim(lDimId)```

**Argument Description**

- `lDimId`: Long. The custom dimension ID

**Return Value**

Long. The member ID for the specified custom dimension.

**Example**

```Dim lCustomMemberId As Long
' See HsvMetadata.EnumCustomDimsForApp for how to get a list of valid custom dim ids
lCustomMemberId = cHfmPovCOM.GetMemberForCustomDim(DIMID_CUSTOMBASE) 'C1```

**GetRightHandDims**

Gets the right-hand dimension members (Account, ICP, Customs).

**Syntax**

```<HfmPovCOM>.GetRightHandDims(plAccount, plICP, pvaralCustDimIds, pvaralCustMemberIds)```

**Argument Description**

- `lAccountId`: Long. Returns the Account member
- `plICP`: Long. Returns the ICP member
- `pvaralCustDimIds`: Long Array. Returns the Custom dimension IDs
- `pvaralCustMemberIds`: Long Array. Returns the Custom member IDs

**Return Value**

Long. The number of custom dimensions.

**Example**

```Dim lAccount As Long, lICP As Long
Dim varalCustomIds As Variant, varalCustomMemberIds As Variant
Dim numCustoms As Long
numCustoms = cHfmPovCOM.GetRightHandDims(lAccount, lICP, _
varalCustomIds, varalCustomMemberIds)```

**SetAllDims**

Sets all of the dimension members.
Syntax

```
<HfmPovCOM>.SetAllDims lScenarioId, lYearId, lPeriodId, lViewId, lEntityId, lParentId,
lValueId, lAccountId, lICPId, varalCustomDimIds, varalCustomMemberIds
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenarioId</td>
<td>Long. Scenario member</td>
</tr>
<tr>
<td>lYearId</td>
<td>Long. Year member</td>
</tr>
<tr>
<td>lPeriodId</td>
<td>Long. Period member</td>
</tr>
<tr>
<td>lViewId</td>
<td>Long. View member</td>
</tr>
<tr>
<td>lEntityId</td>
<td>Long. Entity member</td>
</tr>
<tr>
<td>lParentId</td>
<td>Long. Entity Parent member</td>
</tr>
<tr>
<td>lValueId</td>
<td>Long. Value member</td>
</tr>
<tr>
<td>lAccountId</td>
<td>Long. Account member</td>
</tr>
<tr>
<td>lICPId</td>
<td>Long. ICP member</td>
</tr>
<tr>
<td>varalCustomDimIds</td>
<td>Long Array. Custom dimension IDs</td>
</tr>
<tr>
<td>varalCustomMemberIds</td>
<td>Long Array. Custom member IDs</td>
</tr>
</tbody>
</table>

Return Value
None.

Example

```vba
Dim lScenario As Long, lPeriod As Long, lEntity As Long
Dim varalCustomDimIds(3) as Variant, varalCustomMemberIds As Variant
Dim i As Integer

lScenario = cScenarioTI.GetItemID("Actual")
lPeriod = cPeriodTI.GetItemID("January")
lEntity = cEntityTI.GetItemID("Regional")
' See HsvMetadata.EnumCustomDimsForApp for how to get a list of valid custom dim ids
varalCustomDimIds(0) = DIMID_CUSTOMBASE    ' C1
varalCustomDimIds(1) = DIMID_CUSTOMBASE+1    ' C2
varalCustomDimIds(2) = DIMID_CUSTOMBASE+2    ' C3
varalCustomDimIds(3) = DIMID_CUSTOMBASE+3    ' C4
for i=0 To 3
    varalCustomMemberIds(i) = MEMBERNONE
next
chfmPovCom.SetAllDims lScenario, 2012, lPeriod, MEMBERSCENARIOVIEW, lEntity,
MEMBERNOTUSED, MEMBERENTITYCURRENCY, MEMBERNONE, MEMBERICPNONE, varalCustomIds,
varalCustomMemberIds
```
**SetCustomMemberAt**

Sets the custom dimension ID and member ID at a specified index.

Syntax

```<HfmPovCOM>.SetCustomMemberAt lIndex, lDimId, lMemberId```

**Argument**  
Description

- `lIndex`  | Long. Base-0 index
- `lDimId`  | Long. Custom dimension ID
- `lMemberId`  | Long. Custom member ID

**Return Value**

None.

**Example**

```
' See HsvMetadata.EnumCustomDimsForApp for how to get a list of valid custom dim ids
cHfmPovCOM.SetCustomMemberAt 0, DIMID_CUSTOMBASE, MEMBERNONE
```

**SetCustoms**

Sets the dimension IDs and member IDs for the custom dimensions.

Syntax

```<HfmPovCOM>.SetCustoms(varalDimIds, ,varalMemberIds)```

**Argument**  
Description

- `varalDimIds`  | Long Array. Array of custom dimension IDs
- `varalMemberIds`  | Long Array. Array of custom dimension members

**Example**

Initialize all custom members for the POV to MEMBERNONE.

```
Dim cPov As New HfmPovCOM
Dim aDimIds, aDimNames, aDimAliases, aMembers()
Dim i As Integer
cMetadata.EnumCustomDimsForApp aDimIds, aDimNames, aDimAliases
Redim aMembers(UBound(aDimIds))
For i = LBound(aDimIds) To UBound(aDimIds)
    aMembers(i) = MEMBERNONE
Next
cPov.SetCustoms aDimIds, aMembers
```
SetDimensionMember

Sets the member ID for a specified dimension (only for non-custom dimensions).

Syntax

```csharp
<HfmPovCOM>.SetDimensionMember lDimId, lMemberId
```

**Argument** | **Description**
---|---
`lDimId` | Long. The dimension ID
`lMemberId` | Long. The member ID

Return Value
None.

Example

```csharp
Dim lScenario As Long
lScenario = cScenarioTI.GetItemID("Actual")
cHfmPovCOM.SetDimensionMember( DIMID_SCENARIO, lScenario)
```

SetLeftHandDims

Sets the left-hand dimension members (Scenario, Year, Period, View, Entity, Value).

Syntax

```csharp
<HfmPovCOM>.SetLeftHandDims lScenarioId, lYearId, lPeriodId, lViewId, lEntityId, lParentId, lValueId
```

**Argument** | **Description**
---|---
`lScenarioId` | Long. Scenario member
`lYearId` | Long. Year member
`lPeriodId` | Long. Period member
`lViewId` | Long. View member
`lEntityId` | Long. Entity member
`lParentId` | Entity parent member (specify MEMBERNOTUSED if not applicable)
`lValueId` | Long. Value member

Return Value
None.
Example

Dim lScenario As Long, lPeriod As Long, lEntity As Long
lScenario = cScenarioTI.GetItemID("Actual")
lPeriod = cPeriodTI.GetItemID("January")
lEntity = cEntityTI.GetItemID("Regional")
cHfmPovCOM.SetLeftHandDims lScenario, 2012, lPeriod, MEMBERSCENARIOVIEW, lEntity,
MEMBERNOTUSED, MEMBERENTITYCURRENCY

SetRightHandDims
Sets the right-hand dimension members (Account, ICP, Customs).

Syntax

<HfmPovCOM>.SetRightHandDims lAccountId, lICPId, varalCustomDimIds, varalCustomMemberIds

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lAccountId</td>
<td>Long. Account member</td>
</tr>
<tr>
<td>lICPId</td>
<td>Long. ICP member</td>
</tr>
<tr>
<td>varalCustomDimIds</td>
<td>Long Array. Custom dimension IDs</td>
</tr>
<tr>
<td>varalCustomMemberIds</td>
<td>Long Array. Custom member IDs</td>
</tr>
</tbody>
</table>

Return Value
None.

Example

' See HsvMetadata.EnumCustomDimsForApp for how to get a list of valid custom dim ids
Dim varalCustomDimIds(3) as Variant, varalCustomMemberIds As Variant
varalCustomDimIds(0) = DIMID_CUSTOMBASE ' C1
varalCustomDimIds(1) = DIMID_CUSTOMBASE+1 ' C2
varalCustomDimIds(2) = DIMID_CUSTOMBASE+2 ' C3
varalCustomDimIds(3) = DIMID_CUSTOMBASE+3 ' C4
for i=0 To 3
    varalCustomMemberIds(i) = MEMBERNONE
next
cHfmPovCom.SetRightHandDims MEMBERNONE, MEMBERICPNONE, varalCustomIds,
varalCustomMemberIds

HfmSliceCOM Object Properties

The HfmSliceCOM object represents multiple cell coordinates or slices. Some of the dimensions can be fixed to only specify a single member, while other dimensions can have multiple members specified.
ArrayReserveSize (Read/Write property)
Used to determine how much space to reserve when increasing the sizes of arrays. By default, this value is set to 1. Setting it to a larger value can potentially improve performance. Type: Long.

NumCustomDims (Read-only property)
The number of custom dimensions in the slice object. Type: Long

HfmSliceCOM Object Methods

Clear
Clears the contents of the slice object.

Syntax
<HfmSliceCOM>.Clear
No arguments.

Return Value
None.

Example

cHfmSlice.Clear

ConvertToHfmSliceStrCOM
For internal use.

EnumCustomDimIds
Returns a list of custom dimensions that are currently represented in the slice object.

Syntax
<HfmSliceCOM>.EnumCustomDimIds()
No arguments.

Return Value
Long array. A list of custom dimension IDs.
Example

Dim aDimIds
daDimIds = cSlice.EnumCustomDimIds()

**GenerateCartesianProduct**

*For internal use.*

**GetFixedMember**

Gets the member ID for a fixed dimension.

Syntax

```vbnet
<HfmSliceCOM>.GetFixedMember lDimId
```

**Argument Description**

<table>
<thead>
<tr>
<th>lDimId</th>
<th>Long, Dimension ID</th>
</tr>
</thead>
</table>

**Return Value**

Long. The member ID associated with the specified dimension.

Example

```vbnet
Dim lMemberId as Long
lMemberId = cHfmSliceCOM.GetFixedMember DIMID_SCENARIO
```

**GetMemberArrayForDim**

Gets a list of member IDs in the slice for a specified dimension.

Syntax

```vbnet
<HfmSliceCOM>.GetMemberArrayForDim(lDimId)
```

**Argument Description**

<table>
<thead>
<tr>
<th>lDimId</th>
<th>Long, Dimension ID</th>
</tr>
</thead>
</table>

**Return Value**

Long array. An array of member IDs for the specified dimension.

Example

```vbnet
Dim aMembers
aMembers = cSlice.GetMemberArrayForDim( DIMID_ENTITY )
```
**GetMemberAt**

Gets the member ID at a specified index in the slice, for a specified dimension.

**Syntax**

\[\text{cHfmSlice COM}.\text{GetMemberAt}(\text{dimId}, \text{index})\]

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimId</td>
<td>Long. Dimension ID</td>
</tr>
<tr>
<td>Index</td>
<td>Long. Base-0 index</td>
</tr>
</tbody>
</table>

**Return Value**

Long. The member ID located at the specified index for the specified dimension.

**Example**

```vba
Dim lMemberId as Long
lMemberId = cHfmSlice.GetMemberAt( DIMID_ACCOUNT, 0 )
```

**GetNumCells**

*For internal use.*

**GetNumMembers**

Returns the number of members for a specified dimension.

**Syntax**

\[\text{cHfmSlice COM}.\text{GetNumMembers}(\text{dimId})\]

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimId</td>
<td>Long. Dimension ID</td>
</tr>
</tbody>
</table>

**Return Value**

Long. The number of members in the slice for the specified dimension.

**Example**

```vba
Dim lCount As Long
lCount = cHfmSlice.GetNumMembers( DIMID_ENTITY)
```

**GetPovAtIndex**

Returns an HfmPovCOM object representing the cell coordinates at a specified index in the slice.
Syntax

```
<HfmSliceCOM>.GetPovIndex lIndex, pHfmPovCom
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lIndex</td>
<td>Long</td>
<td>Long, Base-0 index</td>
</tr>
<tr>
<td>pHfmPovCom</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell coordinates at the requested index</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```vbs
Dim cHfmPovCOM as HfmPovCOM
cHfmSliceCOM.GetPovAtIndex 0, cHfmPovCOM
```

### InitializeFromPovString

Initializes the object from a string containing a POV in standard Financial Management notation.

**Syntax**

```
<HfmSliceCOM>.InitializeFromPovString cMetadata, sPov
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>cMetadata</td>
<td>HsvMetadata. An HsvMetadata object</td>
</tr>
<tr>
<td>sPov</td>
<td>String. A POV in standard HFM notation, similar to that used in Web Data Entry forms</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```vbs
Dim cMetadata As HsvMetadata
cMetadata = cHsvSession.Metadata
cSlice.InitializeFromPovString cMetadata, "S#Actual;Budget.A#Sales.E{Europe.[base]}")
```

### SetFixedMember

Sets the member ID for a fixed dimension.

**Syntax**

```
<HfmSliceCOM>.SetFixedMember lDimId, lMemberId
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lDimId</td>
<td>Long. Dimension ID</td>
</tr>
<tr>
<td>lMemberId</td>
<td>Long. Member ID</td>
</tr>
</tbody>
</table>

Return Value
None.

Example

cSliceCOM.SetFixedMember DIMID_ICP, MEMBERICPNONE

SetMemberArrayForDim
Sets a list of member IDs in the slice for a specified dimension.

Syntax

```plaintext
<HfmSliceCOM>.SetMemberArrayForDim lDimId, varalMemberIds
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lDimId</td>
<td>Long. Dimension ID</td>
</tr>
<tr>
<td>varalMemberIds</td>
<td>Long Array. Member IDs</td>
</tr>
</tbody>
</table>

Return Value
None.

Example

```plaintext
Dim aMembers(1)
aMembers(0) = MEMBERICPNONE
aMembers(1) = MEMBERICPTOP
cSlice.SetMemberArrayForDim DIMID_ICP, aMembers
```

SetMemberAt
Sets the member ID at a specified index in the slice, for a specified dimension.

Syntax

```plaintext
<HfmSliceCOM>.SetMemberAt dimId, index, lMemberId
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dimId</td>
<td>Long. Dimension ID</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Index</td>
<td>Long, Base-0 index</td>
</tr>
<tr>
<td>MemberId</td>
<td>Long, Member ID</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

cHfmSliceCOM.SetMemberAt DIMID_ICP, 0, MEMBERICPNONE
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 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 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 60, 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.

AuthenticateUserOnClient

Provides the authentication functionality to user logon information. This method allows the authentication operation to be done on the Web server without the need to rely on the application server, thus eliminating the need for an input cluster name.

Syntax

```csharp
<HsxClient>.AuthenticateUserOnClient(bstrDomain, pbstrUser, bstrPassword, pbstrCSSToken)
```

**Argument**  
**Description**

- `bstrDomain`  
  String, [in] User Domain

- `pbstrUser`  
  String, [in/out] Accepts/Returns User Name

- `bstrPassword`  
  String, [in] User Password

- `pbstrCSSToken`  
  String, [in/out] Accepts/Returns CSS token

**Return Value**

Boolean. True if the authentication was successful and false otherwise. Additionally, if the authentication is done using user name-password combination, the CSS token is also returned as an out parameter. If the authentication is done using the CSS token, the user name is also returned as an out parameter.

**Example**

**C# Example:**

```csharp
comHsxClient = new HsxClientClass();
bAuthenticated = comHsxClient.AuthenticateUserOnClient(string.Empty, ref userID, password, ref ssoToken);
```

**VB6 Example:**

```vb
158
```
Dim hsxClient As New HsxClient
Dim bAuthenticated As Boolean
Dim userID As String, password As String, sso Token As String
userID = "admin"
password = "password"
bAuthenticated=comHsxClient.AuthenticateUserOnClient("",userID,password,ssoToken)

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 Enterprise Performance Management System User Security Administration Guide*. You can test whether the connected user is a member of the Creator group with DoesUserHaveCreateApplicationRights.

**Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the application server cluster on which to create the application.</td>
</tr>
</tbody>
</table>
**Argument** | **Description**
---|---
*bstrProduct* | String (ByVal). The product name. Enter the string “Financial Management” to create a Financial Management application.
*bstrApp* | String (ByVal). The name of the application.
*bstrAppDesc* | String (ByVal). A description of the application.
*bstrDSN* | String (ByVal). Future use. However, the argument is required—specify an empty string.
*varParam1* | 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).
*varParam2* | Variant (ByVal). Future use. However, the argument is required—specify a null Variant.
*varParam3* | Variant (ByVal). Future use. However, the argument is required—specify a null Variant.
*varParam4* | Variant (ByVal). Future use. However, the argument is required—specify a null Variant.
*varParam5* | Variant (ByVal). Future use. However, the argument is required—specify a null Variant.
*bstrProject* | 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*.

*bstrAppWebServerUrl* | 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.

```vbnet
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
```

160  HsxClient Type Library
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 is open, use the HsvSession CreateObject method.

Syntax

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the cluster or application server.</td>
</tr>
<tr>
<td>bstrClassIDOrProgID</td>
<td>String (ByVal). The name of the object to be instantiated.</td>
</tr>
</tbody>
</table>

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! This method fails if used against an application created with Oracle Hyperion EPM Architect.

Syntax

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the application server cluster.</td>
</tr>
</tbody>
</table>
Argument | Description
---|---
bstrApp | 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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the cluster.</td>
</tr>
<tr>
<td>vbDeleteAll</td>
<td>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 varabstrErrorReference argument.</td>
</tr>
<tr>
<td>varabstrErrorReference</td>
<td>String array (ByVal). The error reference numbers that identify the system errors to be deleted if the vbDeleteAll argument is set to FALSE.</td>
</tr>
</tbody>
</table>

**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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbstrDomainName</td>
<td>String. Returns the domain name.</td>
</tr>
<tr>
<td>pbstrUserName</td>
<td>String. Returns the username.</td>
</tr>
</tbody>
</table>

**DisableNewConnections**

Disables new Financial Management connections to a cluster for the specified application and application server criteria.
Tip: To disable new connections for a user, use `HsvSystemInfo.DisableNewConnections`.

To enable new connections, use `EnableNewConnections`.

Syntax

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

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

**DoesUserHaveCreateApplicationRights**

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

*Note:* The Creator group identifies the users who can create Financial Management applications, and is specified with the Configuration Utility. For instructions, see the *Oracle Enterprise Performance Management System Installation and Configuration Guide*.

Syntax

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

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

**DoesUserHaveSystemAdminRights**

Indicates whether the connected user is a member of the Administrator group for an application server cluster.
Note: The Administrator group is specified with the Configuration Utility. For instructions, see the Oracle Enterprise Performance Management System Installation and Configuration Guide.

Syntax

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

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

EnableNewConnections

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

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

To disable new connections, use DisableNewConnections.

Syntax

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the cluster.</td>
</tr>
<tr>
<td>vbAllApplications</td>
<td>Boolean (ByVal). Specifies whether to enable connections for all applications. Pass TRUE for all applications, FALSE to specify an application with the bstrAppName argument.</td>
</tr>
<tr>
<td>bstrAppName</td>
<td>String (ByVal). The name of the application for which to enable connections. This argument is used only if the vbAllApplications argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>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 bstrServer argument.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String (ByVal). The name of the application server for which to enable connections. This argument is used only if the vbAllServers argument is set to FALSE.</td>
</tr>
</tbody>
</table>
EnumProhibitConnections

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

Syntax

\[ <HsxClient>.EnumProhibitConnections bstrClusterName, pvaravbAllApps, pvarabstrAppNames, pvaravbAllServers, pvarabstrServerNames, pvaravbAllUsers, pvaralActivityUserIDs, pvaralActivityUserNames \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the cluster.</td>
</tr>
</tbody>
</table>
| pvaravbAllApps  | Variant array. Indicates whether connections are disabled for all applications. Array items can contain 0 or -1:  
|                | • -1 indicates connections to all applications are disabled.                |
|                | • 0 indicates that only connections to the application returned by the corresponding pvarabstrAppNames argument's array item are disabled. |
|                | The array is returned as a Long subtype.                                    |
| pvarabstrAppNames | Variant array. Returns the application names for which connections are disabled. Application names are returned only when the corresponding pvaravbAllApps argument's array item contains 0. |
|                | Note: If the corresponding pvaravbAllApps argument’s array item contains -1, this array item contains the string “AllApps”. |
|                | The array is returned as a String subtype.                                  |
| pvaravbAllServers | Variant array. Indicates whether connections are disabled for all application servers. Array items can contain 0 or -1:  
|                | • -1 indicates connections to all application servers are disabled.        |
|                | • 0 indicates that only connections to the application server returned by the corresponding pvarabstrServerNames argument's array item are disabled. |
|                | The array is returned as a Long subtype.                                    |
| pvarabstrServerNames | Variant array. Returns the application server names for which connections are disabled. Application server names are returned only when the corresponding pvaravbAllServers argument’s array item contains 0. |
|                | Note: If the corresponding pvaravbAllServers argument’s array item contains -1, this array item contains the string “AllServers”. |
|                | The array is returned as a String subtype.                                  |
| pvaravbAllUsers  | Variant array. Indicates whether connections are disabled for all users. Array items can contain 0 or -1:  
|                | • -1 indicates connections for all users are disabled.                     |
|                | • 0 indicates that only connections for the user represented by the corresponding pvaralActivityUserIDs and pvaralActivityUserNames arguments’ array items are disabled. |
|                | The array is returned as a Long subtype.                                    |
### EnumProvisioningProjects

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

**Syntax**

```
%HsxClient%.EnumProvisioningProjects(bstrClusterName)
```

**Argument**

- **bstrClusterName**  
  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 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.

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

**EnumRegisteredClustersOrServers**

*For internal use.*

**EnumRegisteredServerNames**

*Deprecated - use** [EnumRegisteredClusterNames](#).

**EnumUserAppPreferences**

*For internal use.*

**EnumUsersOnSystem**

Returns the usernames of and other information applicable to the users logged on to a 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 language, use [EnumUsersOnSystemEx](#).

**Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrClusterName</code></td>
<td>String (ByVal). The name of the cluster.</td>
</tr>
<tr>
<td><code>vbAllApplications</code></td>
<td>Boolean (ByVal). Specifies whether information for all applications is returned. Pass TRUE for all applications, FALSE to specify an application with the <code>bstrAppName</code> argument.</td>
</tr>
<tr>
<td><code>bstrAppName</code></td>
<td>String (ByVal). The name of the application server for which to return information. This argument is used only if the <code>vbAllApplications</code> argument is set to FALSE.</td>
</tr>
<tr>
<td><code>vbAllServers</code></td>
<td>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 <code>bstrServer</code> argument.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>bstrServer</code></td>
<td>String (ByVal). The name of the application server for which to return information. This argument is used only if the <code>vbAllServers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td><code>vbAllUsers</code></td>
<td>Boolean (ByVal). Specifies whether information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <code>lActivityUserID</code> argument.</td>
</tr>
<tr>
<td><code>lActivityUserID</code></td>
<td>Long (ByVal). The activity user ID of the user for whom to return information. This argument is used only if the <code>vbAllUsers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td></td>
<td>To get a user's activity user ID, use <code>HsvSystemInfo.GetActivityUserID</code>.</td>
</tr>
<tr>
<td><code>pvaralActivitySessionIDs</code></td>
<td>Variant array. Returns the internal IDs of the user sessions. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pbstrAppNames</code></td>
<td>Variant array. Returns the names of the applications to which users are logged on. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pbstrServerNames</code></td>
<td>Variant array. Returns the names of the application servers on which users are logged on. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pvaralActivityUserIDs</code></td>
<td>Variant array. Returns the activity user IDs of the users who are logged on. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pbstrActivityUserNames</code></td>
<td>Variant array. Returns the usernames of the users who are logged on. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pvaralCurrentActivity</code></td>
<td>Variant array. Returns the IDs of the users' current activities. Valid values are represented by the HFMConstants type library constants listed in &quot;User Activity Constants&quot; on page 895. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pvarabstrModuleNames</code></td>
<td>Variant array. Returns the names of the modules in which the users are active. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pbaradTimeStarted</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>pvarabstrDesc</code></td>
<td>Variant array. Future use. An array is returned, but you can ignore it.</td>
</tr>
</tbody>
</table>

**Example**

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

```vba
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)
```

168  HsxClient Type Library
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 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the cluster.</td>
</tr>
<tr>
<td>vbAllApplications</td>
<td>Boolean (ByVal). Specifies whether information for all applications is returned. Pass TRUE for all applications, FALSE to specify an application with the bstrAppName argument.</td>
</tr>
<tr>
<td>bstrAppName</td>
<td>String (ByVal). The name of the application server for which to return information. This argument is used only if the vbAllApplications argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>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 bstrServer argument.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String (ByVal). The name of the application server for which to return information. This argument is used only if the vbAllServers argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>Boolean (ByVal). Specifies whether information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the lActivityUserID argument.</td>
</tr>
<tr>
<td>lActivityUserID</td>
<td>Long (ByVal). The activity user ID of the user for whom to return information. This argument is used only if the vbAllUsers argument is set to FALSE. To get a user’s activity user ID, use HsvSystemInfo.GetActivityUserID.</td>
</tr>
<tr>
<td>lLanguageID</td>
<td>Long (ByVal). The ID of the language in which to return module names.</td>
</tr>
</tbody>
</table>

Tip: You can use the HsvResourceManager method GetAvailableLanguages to obtain the IDs of the languages valid for a release.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvaralActivitySessionIDs</td>
<td>Variant array. Returns the internal IDs of the user sessions. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pbstrAppNames</td>
<td>Variant array. Returns the names of the applications to which users are logged on. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pbstrServerNames</td>
<td>Variant array. Returns the names of the application servers on which users are logged on. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralActivityUserIDs</td>
<td>Variant array. Returns the activity user IDs of the users who are logged on. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pbstrActivityUserNames</td>
<td>Variant array. Returns the usernames of the users who are logged on. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralCurrentActivity</td>
<td>Variant array. Returns the IDs of the users’ current activities. Valid values are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 895. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarabstrModuleName</td>
<td>Variant array. Returns the names of the modules in which the users are active. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaradTimeStarted</td>
<td>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.</td>
</tr>
<tr>
<td>pvarabstrDesc</td>
<td>Variant array. <em>Future use.</em> An array is returned, but you can ignore it.</td>
</tr>
</tbody>
</table>

**EnumUsersOnSystemEx2**

Returns the usernames and other information applicable to all users logged on to / logged out from a 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**

```xml
dom::EnumUsersOnSystemEx2 bstrClusterName, vbAllApplications, bstrAppName, vbAllServers, bstrServer, vbAllUsers, lActivityUserID, lLanguageID, pvaralActivitySessionIDs, pbstrAppNames, pbstrServerNames, pvaralActivityUserIDs, pbstrActivityUserNames, pvaralCurrentActivity, pvarabstrModuleName, pvaradTimeStarted, pvarabstrDesc, pvaralSessionStatus
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the cluster.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>vbAllApplications</td>
<td>Boolean (ByVal). Specifies whether information for all applications is returned. Pass TRUE for all applications, FALSE to specify an application with the bstrAppName argument.</td>
</tr>
<tr>
<td>bstrAppName</td>
<td>String (ByVal). The name of the application server for which to return information. This argument is used only if the vbAllApplications argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>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 bstrServer argument.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String (ByVal). The name of the application server for which to return information. This argument is used only if the vbAllServers argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>Boolean (ByVal). Specifies whether information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the lActivityUserID argument</td>
</tr>
<tr>
<td>lActivityUserID</td>
<td>Long (ByVal). The activity user ID of the user for whom to return information. This argument is used only if the vbAllUsers argument is set to FALSE.</td>
</tr>
<tr>
<td></td>
<td>To get a user's activity user ID, use HsvSystemInfo.GetActivityUserID.</td>
</tr>
<tr>
<td>lLanguageID</td>
<td>Long (ByVal). The ID of the language in which to return module names.</td>
</tr>
<tr>
<td></td>
<td>Tip: You can use the HsvResourceManager method GetAvailableLanguages to obtain the IDs of the languages valid for a release.</td>
</tr>
<tr>
<td>pvaralActivitySessionIDs</td>
<td>Variant array. Returns the internal IDs of the user sessions.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pbstrAppNames</td>
<td>Variant array. Returns the names of the applications to which users are logged on.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pbstrServerNames</td>
<td>Variant array. Returns the names of the application servers on which users are logged on.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralActivityUserIDs</td>
<td>Variant array. Returns the activity user IDs of the users who are logged on.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pbstrActivityUserNames</td>
<td>Variant array. Returns the usernames of the users who are logged on.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralCurrentActivity</td>
<td>Variant array. Variant array. Returns the IDs of the users’ current activities. Valid values are represented by the HFMConstants type library constants listed in &quot;User Activity Constants&quot; on page 895.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarabstrModuleNames</td>
<td>Variant array. Returns the names of the modules in which the users are active.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaradTimeStarted</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a Double subtype.</td>
</tr>
<tr>
<td>pvarabstrDesc</td>
<td>Variant array. Future use. An array is returned, but you can ignore it.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pvParaSessionStatus</td>
<td>Variant array. Returns an array of status IDs that identify the user’s task status. Valid values are represented by the HFMCConstants type library constants listed in “Task Status Constants” on page 897.</td>
</tr>
</tbody>
</table>

**GetApplicationFolder**

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

**Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the application server cluster.</td>
</tr>
<tr>
<td>bstrProduct</td>
<td>String (ByVal). The name of the product. For Financial Management applications, enter the string “Financial Management”.</td>
</tr>
<tr>
<td>bstrApp</td>
<td>String (ByVal). The name of the application.</td>
</tr>
</tbody>
</table>

**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("Gszabo1", "Financial Management", "SimpleEx")```

**GetClusterInfo**

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

**Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrServerToGetClusterInfo</td>
<td>String (ByVal). The name of the application server.</td>
</tr>
<tr>
<td>vbUseAutomaticLoadBalancing</td>
<td>Boolean (ByVal). Specifies whether the application server is registered for automatic load balancing. Pass TRUE if load balancing is used, FALSE otherwise.</td>
</tr>
<tr>
<td>pbstrClusterName</td>
<td>String. Returns the name of the cluster to which the application server is assigned.</td>
</tr>
</tbody>
</table>
GetClustersAndServers
Enumerate clusters and application servers.

Syntax
<HsxClient>.GetClustersAndServers pvarabstrClusters, pvarabstrServers

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvarabstrClusters</td>
<td>String Array. Returns Cluster names.</td>
</tr>
<tr>
<td>pvarabstrServers</td>
<td>String Array. Returns Server names.</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
Dim varabstrClusters, varabstrServers
HsxClient.GetClustersAndServers varabstrClusters, varabstrServers

GetHFMErrorLogRecordSet
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 is open.

Syntax
<HsxClient>.GetLogonInfoSSO(pbstrDomain, pbstrUser)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbstrDomain</td>
<td>String. Returns the name of the user's domain.</td>
</tr>
<tr>
<td>pbstrUser</td>
<td>String. Returns the name of the user.</td>
</tr>
</tbody>
</table>
Return Value
String. Returns the external authentication token.

Example
GetLogonInfoSSO is used in the example for the HsvSecurityAccess method GetUserInfoFromUniqueID2.

GetModifyApplicationInfo
For internal use.

GetModifyApplicationStatus
For internal use.

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
bstrClusterName 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 7, “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)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrCluster</td>
<td>String (ByVal). The cluster name.</td>
</tr>
<tr>
<td>bstrAppName</td>
<td>String (ByVal). The application name.</td>
</tr>
</tbody>
</table>

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 applications and application servers.

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

Syntax

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the cluster.</td>
</tr>
<tr>
<td>vbAllApplications</td>
<td>Boolean (ByVal). Determines whether to log off users from all applications.</td>
</tr>
<tr>
<td>bstrAppName</td>
<td>String (ByVal). The name of the application for which to log users off.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>Boolean (ByVal). Determines whether to log off users from all application servers.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String (ByVal). The name of the application server for which to log users off.</td>
</tr>
</tbody>
</table>
ModifyApplication

For internal use.

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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrClusterName</code></td>
<td>String (ByVal). The name of the application server cluster.</td>
</tr>
<tr>
<td><code>bstrProduct</code></td>
<td>String (ByVal). The product name. Enter &quot;Financial Management&quot; to open a Financial Management application.</td>
</tr>
<tr>
<td><code>bstrApp</code></td>
<td>String (ByVal). The name of the application.</td>
</tr>
<tr>
<td><code>ppIUnkServer</code></td>
<td>HsxServer object. Returns an object reference to the HsxServer object. For information on this object, see Chapter 7, “HsxServer Type Library.”</td>
</tr>
<tr>
<td><code>ppIUnkSession</code></td>
<td>HsvSession object. Returns an object reference to an HsvSession object.</td>
</tr>
</tbody>
</table>

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! This method fails if used against an application created with Performance Management Architect.

Syntax

```vba
<HsxClient>.RegisterApplicationCAS bstrClusterName, bstrProjectName, bstrApp, bstrHFMWebServerURL
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrClusterName</code></td>
<td>String (ByVal). The name of the application's server cluster.</td>
</tr>
<tr>
<td><code>bstrProjectName</code></td>
<td>String (ByVal). The name of the Shared Services provisioning project for the application.</td>
</tr>
<tr>
<td><code>bstrApp</code></td>
<td>String (ByVal). The name of the application.</td>
</tr>
<tr>
<td><code>bstrHFMWebServerURL</code></td>
<td>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.</td>
</tr>
</tbody>
</table>

**Tip:** To get the names of the provisioning projects associated with an application server cluster, use `EnumProvisioningProjects`.

RegisterApplicationCASWithAccessCode

*For internal use.*

RegisterCluster

Registers a cluster or application server for a client.

Syntax

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

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

Example

The following example registers an application server named ACME.

```vba
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`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrClusterName</code></td>
<td>String (ByVal). The name of the cluster or application server.</td>
</tr>
<tr>
<td><code>bstrProduct</code></td>
<td>String (ByVal). The product name. Enter the string “Financial Management” to set a Financial Management application’s local storage folder.</td>
</tr>
<tr>
<td><code>bstrApp</code></td>
<td>String (ByVal). The name of the application.</td>
</tr>
<tr>
<td><code>bstrFolder</code></td>
<td>String (ByVal). The path on the client computer to set as the local storage folder.</td>
</tr>
</tbody>
</table>

**Example**

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

```c
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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrDomain</td>
<td>String (ByVal). The name of the user's domain.</td>
</tr>
<tr>
<td>bstrUser</td>
<td>String (ByVal). The user's username.</td>
</tr>
<tr>
<td>bstrToken</td>
<td>String (ByVal). The single sign-on token.</td>
</tr>
<tr>
<td>bstrPassword</td>
<td>String (ByVal). The user's password.</td>
</tr>
</tbody>
</table>

Example

See the example for OpenApplication.

UnregisterAllClusters

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

Syntax

```
<HasxClient>.UnregisterAllClusters
```

UnregisterCluster

Unregisters the specified cluster or application server.

Syntax

```
<HasxClient>.UnregisterCluster bstrClusterName
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClusterName</td>
<td>String (ByVal). The name of the cluster or application server.</td>
</tr>
</tbody>
</table>
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.
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 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 62, 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 occurs.

**Syntax**

```<HsxClientUI>.Initialize pIUnkHsxClient```

**Argument**  
**Description**


**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

\(<\texttt{HsxClientUI}.\texttt{Logon(vbProvideOptionToUseWindowsLoggedOnUser)}\\)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{vbProvideOptionToUseWindowsLoggedOnUser}</td>
<td>Boolean (ByVal). The value passed is ignored by the system; however, you must pass either TRUE or FALSE.</td>
</tr>
</tbody>
</table>

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 was 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 specified in 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

\(<\texttt{HsxClientUI}.\texttt{OpenApplication(ppIUnkServer, ppIUnkSession, pbstrServer, pbstrProduct, pbstrApplication)}\\)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{ppIUnkServer}</td>
<td>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 7, “HsxServer Type Library.”</td>
</tr>
<tr>
<td>\texttt{ppIUnkSession}</td>
<td>HsvSession object. Returns an object reference to an HsvSession object.</td>
</tr>
<tr>
<td>\texttt{pbstrServer}</td>
<td>String. Returns the name of the application server that the user selects.</td>
</tr>
</tbody>
</table>
**Argument**    **Description**


`pbstrApplication` 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 was set with the code in the Example for `Initialize`.

```vba
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**

```vba```
<HsxClientUI>.RegisterServer
```

**Example**

The following example displays the Register Cluster dialog box. The example assumes that the `m_cClientUI` object reference was set with the code in the Example for `Initialize`.

```vba```
```
```
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 is enabled for a cluster.

To use the HsxServer type library, you must reference HsxServer.exe in your project. The HsxServer type library contains 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. See “Application Server Tier Type Libraries” on page 51.

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 63, and are described in detail in the following topics.

**DeleteSystemErrors**
*For internal use.*

**DeleteSystemErrorsEx**
*For internal use.*

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

**Syntax**

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

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvarabstrProducts</code></td>
<td>Variant array. Returns the product names for the applications. For Financial Management applications, the array items consist of the string Financial Management. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pvarabstrApps</code></td>
<td>Variant array. Returns the names of the applications on the application server. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pvarabstrDescs</code></td>
<td>Variant array. Returns the descriptions of the applications. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pvarabstrDSNs</code></td>
<td>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.</td>
</tr>
</tbody>
</table>

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

```vbnet
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**

```vba
<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.

```vba
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**

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

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvarabstrClusters</code></td>
</tr>
<tr>
<td><code>pvarabstrServers</code></td>
</tr>
</tbody>
</table>

**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.

GetHFMErrorLogRecordSet
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, using the system error’s reference number.

Syntax

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

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrErrorReference</code></td>
<td>String (ByVal). The system error’s reference number.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dStartTimeStamp</code></td>
<td>Double (ByVal). The start of the date range to filter by, or 0 to omit filtering by timestamp.</td>
</tr>
<tr>
<td><code>dEndTimeStamp</code></td>
<td>Double (ByVal). The end of the date range to filter by, or 0 to omit filtering by timestamp.</td>
</tr>
</tbody>
</table>
### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrServerName</td>
<td>String (ByVal). The name of the application server to filter by, or an empty string to omit filtering by server.</td>
</tr>
<tr>
<td>bstrApplicationName</td>
<td>String (ByVal). The name of the application to filter by or an empty string to omit filtering by server.</td>
</tr>
<tr>
<td>pvarabstrReference</td>
<td>Variant. Returns an array of strings containing the reference numbers of the system errors that match the search criteria.</td>
</tr>
<tr>
<td>pvarabstrLogType</td>
<td>Variant. Returns an array of strings containing the errors’ log types. The valid return values are represented by the constants in &quot;Log Severity Constants&quot; on page 898.</td>
</tr>
<tr>
<td>pvarabstrTimeStamp</td>
<td>Variant. Returns an array of strings containing the errors’ timestamps.</td>
</tr>
<tr>
<td>pvarabstrServerName</td>
<td>Variant. Returns an array of strings containing the errors’ application server names.</td>
</tr>
<tr>
<td>pvarabstrApplicationName</td>
<td>Variant. Returns an array of strings containing the errors’ application names.</td>
</tr>
</tbody>
</table>

### 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.
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 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 64, 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:

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

**Data**

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

```vbelike
Set cData = m_cSession.Data
```
ICM

Returns an object reference to the HsvICM object or the IHsvAdminICM interface. See “HsvICM Object Methods” on page 737 and “IHsvAdminICM Interface Methods” on page 755.

Journals

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

```vbasic
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:

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

ProcessFlow

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

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

Reports

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

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

Security

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

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

SystemInfo

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

```vbasic
Set cSysInfo = m_cSession.SystemInfo
```
HsvSession Object Methods

The HsvSession object’s methods are summarized in Table 8 on page 65, and are described in detail in the following topics.

CreateObject

Instantiates an object on the application server on which an application is open. 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 HsxClient.CreateObjectOnServer, which takes the application server name as an argument. See “CreateObjectOnServer” on page 161.

**Syntax**

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

**Argument** | **Description**
--- | ---
*bstrClassIDOrProgID* | 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.

GetECID

Returns the Execution Context ID (ECID) for ODL logging. The ECID helps connect multiple log files in an ODL log.

**Syntax**

```
<HsvSession>.GetECID bstrEcid
```

**Argument** | **Description**
--- | ---
*bstrEcid* | String (ByRef). The Execution Context ID (ECID).
**get_HsvDQI**

Returns the DQI object reference from a valid session, that can be used to create, define, and retrieve data and status in the form of Grids from an application.

Example

```vba
Dim cDQI As HsvDQI
Set cDQI = m_cSession.get_HsvDQI()
```

**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 was loaded after the last consolidation.

Syntax

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvbMajorChange</code></td>
<td>Boolean. Returns <code>TRUE</code> if the application’s system information has changed, <code>FALSE</code> otherwise.</td>
</tr>
<tr>
<td><code>pvbMinorChange</code></td>
<td>Boolean. For internal use.</td>
</tr>
</tbody>
</table>

**HasUserStatusChanged**

Indicates whether the connected user was logged off by an administrator.

Syntax

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvbKill</code></td>
<td>Boolean. Indicates whether the user was logged off. Returns <code>TRUE</code> if the user was logged off, <code>FALSE</code> otherwise.</td>
</tr>
<tr>
<td><code>pvbWarn</code></td>
<td>Boolean. For internal use.</td>
</tr>
</tbody>
</table>

**IsBusy**

*For internal use.*
IsRunningTasks

For internal use.

LockMetadataLoadWithSystemChangeCheck

For internal use.

UnlockMetadataLoad

For internal use.
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 65.

**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 67.

Tip: The HsvMetadata object also contains several properties. See “HsvMetadata Object Properties” on page 230.

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

```vba
Dim cHsvMetadata as HsvMetadata
Set cHsvMetadata = m_cHsvSession.Metada
```

### EnumCellTextLabels

Enumerates the cell text labels for an application.

**Syntax**

```vba
<HsvMetadata>.EnumCellTextLabels pvaralIds, pvarabstrLabels
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvaralIds</td>
<td>Long array. Returns a list of IDs associated with the cell text labels.</td>
</tr>
<tr>
<td>pvarabstrLabels</td>
<td>String array. Returns a list of the cell text labels.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```vba
Dim aIds, aLabels
cMetadata.EnumCellTextLabels aIds, aLabels
```
**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.

**EnumCustomDimsForApp**

Enumerates the valid custom dimensions for an application.

**Syntax**

```
<HsvMetadata>.EnumCustomDimsForApp pvaralDimIds, pvarabstrDimNames, pvarabstrDimAliases
```

**Argument Description**

- `pvaralDimIds` Long array. Returns an array of custom dimension IDs.
- `pvarabstrDimNames` String array. Returns an array of custom dimension short names.
- `pvarabstrDimAliases` String array. Returns an array of custom dimension aliases.

**Return Value**

None.

**Example**

Dim aDimIds, aShortNames, aAliases
cMetadata.EnumCustomDimsForApp aDimIds, aShortNames, aAliases

**EnumCustomDimsForAppEx**

Enumerates the valid custom dimensions and custom dimension size for an application.
Syntax

```csharp
<HsvMetadata>.EnumCustomDimsForAppEx pvaralDimIds, pvarabstrDimNames, pvarabstrDimAliases, pvarasSizes
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvaralDimIds</td>
<td>Long array. Returns an array of custom dimension IDs.</td>
</tr>
<tr>
<td>pvarabstrDimNames</td>
<td>String array. Returns an array of custom dimension short names.</td>
</tr>
<tr>
<td>pvarabstrDimAliases</td>
<td>String array. Returns an array of custom dimension aliases.</td>
</tr>
<tr>
<td>pvarasSizes</td>
<td>Integer array. Returns an array of custom dimension sizes. 1=Small, 2=Medium, 4=Large.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

```csharp
Dim aDimIds, aShortNames, aAliases
cMetadata.EnumCustomDimsForAppEx aDimIds, aShortNames, aAliases
```

**EnumDimProperties**

```csharp
<HsvMetadata>.EnumConsolidationMethodIDs varalIDs
```

Syntax

```csharp
<HsvMetadata>.EnumDimProperties lDimID, varlIDs, varbstrNames, varlDataTypes
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lDimID</td>
<td>Long(byVal). Dimension ID for which attribute names need to be extracted.</td>
</tr>
<tr>
<td>varlIDs</td>
<td>Variant(byRef). A Variant array of attributes IDs.</td>
</tr>
<tr>
<td>varbstrNames</td>
<td>Variant(byRef). A Variant array of attributes names.</td>
</tr>
<tr>
<td>varlDataTypes</td>
<td>Variant(byRef). A Variant array of attributes data types</td>
</tr>
</tbody>
</table>

Example

```csharp
metadata = session.Metadata
metadata.EnumDimProperties(  lDimID, varlIDs, varbstrNames, varlDataTypes  )
```

**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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvar2daOptions</td>
<td>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 60 on page 204. 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 return 0 for items 3 and 4 of the second dimension.</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvaralIDs</td>
<td>Variant array. Returns the IDs of the application’s languages. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarabstrLabels</td>
<td>Variant array. Returns the labels of the application’s languages. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

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

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvar2daOptions</td>
<td>Variant array. Returns a two-dimensional safe array that represents the metadata load options.</td>
</tr>
</tbody>
</table>

The first dimension identifies the options, and is indexed from 1 to 29. The indexes and corresponding load options are listed in Table 61 on page 226.

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 seventeenth item in the first dimension, array item (17, 2) stores the system’s default delimiter.

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

Example

See the example for Load.
**EnumMemberProperties**

Extracts member properties IDs and values for a dimension member.

**Syntax**

```csharp
<HsvMetadata>.EnumMemberProperties lDimID, lMemberID, varAttributeIDs, varAttributeValues, varAttributeValuesForDisplay
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IDimID</td>
<td>Long(byVal). Dimension ID for which attribute names need to be extracted.</td>
</tr>
<tr>
<td>IMemberID</td>
<td>Long(byVal). Member ID for which properties need to be extracted.</td>
</tr>
<tr>
<td>varAttributeIDs</td>
<td>Variant(byRef). A Variant array of attributes IDs.</td>
</tr>
<tr>
<td>varAttributeValues</td>
<td>Variant(byRef). A Variant array of attributes values.</td>
</tr>
<tr>
<td>varAttributeValuesForDisplay</td>
<td>Variant(byRef). A Variant array of attributes values for display</td>
</tr>
</tbody>
</table>

**Example**

```csharp
metadata = session.Metadata

metadata.EnumMemberProperties(lDimID, lMemberID, varAttributeIDs, varAttributeValues, varAttributeValuesForDisplay)
```

**Extract**

Extracts metadata into a text file. The file is 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. See “Extracting Metadata” on page 782.

**Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrServerFilename</td>
<td>String (ByVal). The name and path of the metadata extract file. The path must exist on the application server.</td>
</tr>
<tr>
<td>bstrServerLogFilename</td>
<td>String (ByVal). The name and path of the log file. The path must exist on the application server.</td>
</tr>
<tr>
<td>varavSettings</td>
<td>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 60 on page 204.</td>
</tr>
</tbody>
</table>

**Tip:** Use `EnumExtractOptions` to return information about the valid extract options.

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

---

**HsvMetadata Object Methods** 203
Table 60  Metadata Extract Options

<table>
<thead>
<tr>
<th>Index</th>
<th>Extract Option Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Option:</strong> Currencies</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether currencies are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Boolean — TRUE to extract currencies, otherwise FALSE.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Option:</strong> Scenarios</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether scenarios are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Boolean — TRUE to extract scenarios, otherwise FALSE.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Option:</strong> Years</td>
</tr>
<tr>
<td></td>
<td>For internal use.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Option:</strong> Periods</td>
</tr>
<tr>
<td></td>
<td>For internal use.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Option:</strong> Views</td>
</tr>
<tr>
<td></td>
<td>For internal use.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Option:</strong> Entities</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether entities are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Boolean — TRUE to extract entities, otherwise FALSE.</td>
</tr>
<tr>
<td>7</td>
<td><strong>Option:</strong> Accounts</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether accounts are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Boolean — TRUE to extract accounts, otherwise FALSE.</td>
</tr>
<tr>
<td>8</td>
<td><strong>Option:</strong> Custom1 -</td>
</tr>
<tr>
<td></td>
<td>Deprecated - use CustomDims</td>
</tr>
<tr>
<td>9</td>
<td><strong>Option:</strong> Custom2</td>
</tr>
<tr>
<td></td>
<td>Deprecated - use CustomDims</td>
</tr>
<tr>
<td>10</td>
<td><strong>Option:</strong> Custom3</td>
</tr>
<tr>
<td></td>
<td>Deprecated - use CustomDims</td>
</tr>
<tr>
<td>11</td>
<td><strong>Option:</strong> Custom4</td>
</tr>
<tr>
<td></td>
<td>Deprecated - use CustomDims</td>
</tr>
<tr>
<td>12</td>
<td><strong>Option:</strong> CustomDims</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies which custom dimensions to extract. The position within the array indicates which custom dimension as follows: first position = Custom1, second position = Custom2, and so on.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Boolean — TRUE to extract Custom members, otherwise FALSE.</td>
</tr>
<tr>
<td>13</td>
<td><strong>Option:</strong> Delimiter</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies an extract file’s delimiter.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> String — a valid delimiter character. EnumExtractOptions returns the valid delimiters.</td>
</tr>
</tbody>
</table>

204  HsvMetadata Type Library
<table>
<thead>
<tr>
<th>Index</th>
<th>Extract Option Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>14</td>
<td><strong>Option:</strong> AppSettings</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether application settings are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Boolean — TRUE to extract application settings, otherwise FALSE.</td>
</tr>
<tr>
<td>15</td>
<td><strong>Option:</strong> FileFormat</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether the metadata extract file is in an ASCII text or an XML format.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Long — 0 to extract into a text file, 1 to extract into an XML file.</td>
</tr>
<tr>
<td>16</td>
<td><strong>Option:</strong> ConsolMethods</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether consolidation methods are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Boolean — TRUE to extract consolidation methods, otherwise FALSE.</td>
</tr>
<tr>
<td>17</td>
<td><strong>Option:</strong> ExtractSystemMembers</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether system members are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Boolean — TRUE to extract system members, otherwise FALSE. <strong>Note:</strong> If this option is set to TRUE, you must also set to TRUE the options for the system members to be extracted.</td>
</tr>
<tr>
<td>18</td>
<td><strong>Option:</strong> SystemAccounts</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether system-generated Account dimension members are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> Boolean — TRUE to extract system accounts, otherwise FALSE. If this option is set to TRUE you must also set the ExtractSystemMembers option to TRUE.</td>
</tr>
<tr>
<td>19</td>
<td><strong>Option:</strong> Values</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether Value dimension members are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> 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.</td>
</tr>
<tr>
<td>20</td>
<td><strong>Option:</strong> ICPs</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether Intercompany Partner dimension members are extracted.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Extract:</strong> 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.</td>
</tr>
</tbody>
</table>

**Example**

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

```vbnet
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
```

HsvMetadata Object Methods 205
ExtractModuleConfigurations

Extracts the definition file for the module configurations.

Syntax

```<HsvMetadata>.ExtractModuleConfigurations bstrClientFileName, bstrLogFileName```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClientFileName</td>
<td>String (ByVal). The name of the module configuration file.</td>
</tr>
<tr>
<td>bstrLogFileName</td>
<td>String (ByVal). The name of the log file.</td>
</tr>
</tbody>
</table>

GetApplicationAttribute

Deprecated - use GetApplicationAttributeExtDim.

GetApplicationAttributeExtDim

Returns the raw value of an application setting attribute. The attribute ID parameter is a long, and must be one of the ATTRIBEX_enums. Supersedes GetApplicationAttribute.

**Tip:** To obtain a user-readable string representing the attribute’s value, pass the raw value to `TranslateApplicationAttributeForDisplay`.

Syntax

```<HsvMetadata>.GetApplicationAttributeExtDim(lAttributeId)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lAttributeId</td>
<td>Long, ATTRIBEX_enum value that specifies the desired application attribute.</td>
</tr>
</tbody>
</table>

Return Value

Variant. The attribute value.

**Example 1**

Dim value As Variant

' Get the FDM app name attribute value

value = cMetadata.GetApplicationAttributeExtDim(ATTRIBEX_APPSETTING_FDMAPPNAME)
Example 2

' Enumerate the security settings for all custom dimensions
' This example shows the use of an attribute that requires specifying the custom dimension

Dim value as Variant
Dim maxDimNum As Long, iCust As Long
maxDimNum = cMetadata.GetMaxCustomDimNumber()
For iCust = 1 to maxDimNum
    value =
    cMetadata.GetApplicationAttributeExtDim(ATTRIBEX_APPSETTING_USESECURITYFORCUSTOM OR iCust)
    Debug.Print "Use Security for Custom " + CStr(iCust) + "="
    if CBool(Value) = True then
        Debug.Print "True"
        else
        Debug.Print "False"
    Next

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.

    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

GetAppProfileInfo

For internal use.

GetByIndexValidationAccount

Returns the member ID of a validation account, using 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

Index  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
GetCellLevelAccountType

Deprecated - use GetCellLevelAccountTypeExtDim.

GetCellLevelAccountTypeExtDim

Returns the account type of a cell, using 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. Supersedes GetCellLevelAccountType.

Tip: HsvAccounts.GetAccountType returns an Account dimension member’s account type. See “GetAccountType” on page 275.

Syntax

```<HsvMetadata>.GetCellLevelAccountType pIUnkPovCOM```

Argument Description

`pIUnkPovCOM` HfmPovCOM. Object representing the account and custom dimension members.

`psAccountType` Integer. Returns the cell’s account type.

Return Value

The account type.

GetCellTextIdFromLabel

Retrieves the ID associated with a cell text label.

Syntax

```<HsvMetadata>.GetCellTextLabelFromId lId, pbstrLabel```

Argument Description

`Cell text ID` Long. Cell text ID.

`pbstrLabel` String. Returns the label associated with the specified ID.
### GetCellTextIdsFromLabels

**Retrieves the 2-D array of cell text label IDs from 2-D array of cell text labels.**

**Syntax**

```plaintext
<HsvMetadata>.GetCellTextIdsFromLabels(vara2dCellTextLabels, pvar2daIDsOfLabels)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vara2dCellTextLabels</td>
<td>2-D String array. 2-D array of Cell text labels.</td>
</tr>
<tr>
<td>pvar2daIDsOfLabels</td>
<td>2-D String array. Returns 2-D array of cell text label IDs.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```vbscript
Dim aCellTextLabels(), alCellTextLabelIds
CMetadata.GetCellTextIdsFromLabels aCellTextLabels, alCellTextLabelIds
```

### GetCellTextLabelDescription

**Retrieves the description for a specified cell text ID and language.**

**Syntax**

```plaintext
<HsvMetadata>.GetCellTextLabelDescription lId, lLangId, pbstrDesc
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lId</td>
<td>Long. The cell text label's ID</td>
</tr>
<tr>
<td>lLangId</td>
<td>Long. The desired language</td>
</tr>
<tr>
<td>pbstrDesc</td>
<td>String. Returns the description of the specified label and language IDs.</td>
</tr>
</tbody>
</table>

**Example**

```vbscript
Dim aCellTextIds, aLabels
```
Dim llLangId As Long, i As Integer
cMetadata.EnumCellTextLabels aCellTextIds, aLabels
cSystemInfo.GetLanguageUserParameters llLangId
' Output the descriptions for all cell text labels, for the current
' user’s preferred language
For i=LBound(aCellTextIds) to UBound(aCellTextIds)
    cMetadata.GetCellTextLabelDescription aCellTextIds(i), llLangId, sDesc
    Debug.Print aCellTextIds(i) + " " + aLabels(i) + " " + sDesc
Next

GetCellTextLabelFromId
Retrieves the cell text label for a cell text ID.

Syntax
<HsvMetadata>.GetCellTextLabelFromId lId, pbstrLabel

Argument Description
Cell text ID Long. Cell text ID.

pbstrLabel String. Returns the label associated with the specified ID.

Return Value
None.

Example
Dim sLabel as String
cMetadata.GetCellTextLabelFromId 1, sLabel

GetConsolidationMethodDescription
Returns a consolidation method’s description for the specified language.

Syntax
<HsvMetadata>.GetConsolidationMethodDescription lMethodID, llLangID, pbstrDescription

Argument Description
lMethodID Long (ByVal). The ID of the consolidation method.

Tip: You can get consolidation method IDs with EnumConsolidationMethodIDs.
### GetConsolidationMethodInfo

Returns the attributes of a consolidation method, using the method’s ID.

**Note:** To get a consolidation method’s description, use `GetConsolidationMethodDescription`.

**Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lConsolMethodID</code></td>
<td>Long (ByVal). The consolidation method’s ID.</td>
</tr>
<tr>
<td><strong>Tip:</strong> You can get consolidation method IDs with <code>EnumConsolidationMethodIDs</code>.</td>
<td></td>
</tr>
<tr>
<td><code>pbstrLabel</code></td>
<td>String. The method’s label.</td>
</tr>
<tr>
<td><code>pbUsedByCalcRoutine</code></td>
<td>Boolean. Returns the method’s <code>UsedByCalcRoutine</code> attribute.</td>
</tr>
<tr>
<td><code>pbIsHoldingMethod</code></td>
<td>Boolean. Returns the method’s <code>IsHoldingMethod</code> attribute.</td>
</tr>
<tr>
<td><code>psToPercentControlOp</code></td>
<td>Integer. Returns the method’s <code>ToPercentControlComp</code> attribute. Constants that represent the valid return values are listed in Table 97 on page 853.</td>
</tr>
<tr>
<td><code>pdToPercentControl</code></td>
<td>Double. Returns the method’s <code>ToPercentControl</code> attribute.</td>
</tr>
<tr>
<td><code>pdPercentConsol</code></td>
<td>Double. Returns the method’s <code>PercentConsol</code> attribute.</td>
</tr>
<tr>
<td><strong>Note:</strong> There are predefined values that can be used for this attribute. Constants that represent these predefined values are listed in Table 98 on page 853.</td>
<td></td>
</tr>
<tr>
<td><code>psControl</code></td>
<td>Integer. Returns the method’s <code>Control</code> attribute. Constants that represent the valid return values are listed in Table 96 on page 853.</td>
</tr>
</tbody>
</table>

**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
    HsvMetadata Object Methods  213
'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**

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

**Argument**

- **pdTimeStamp**
  - Type: 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.

```vba
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**

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

**Argument**

- **lEntity**
  - Type: Long (ByVal). The member ID of the Entity dimension.
- **lParent**
  - Type: Long (ByVal). The member ID of the Parent dimension.
- **lValue**
  - Type: Long (ByVal). The member ID of the Value dimension.

**Return Value**

- Type: Long. Returns the currency value ID.
**GetCurrencyValueIDsForEntityValueCombinations**

Returns the currency value IDs for the entity value combinations.

**Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varalEntities</td>
<td>Long Array (ByVal). The member IDs of the Entity dimension.</td>
</tr>
<tr>
<td>varalParents</td>
<td>Long Array (ByVal). The member IDs of the Parent dimension</td>
</tr>
<tr>
<td>varalValues</td>
<td>Long Array (ByVal). The member IDs of the Value dimension</td>
</tr>
</tbody>
</table>

**Return Value**

Variant. Returns the currency value IDs.

**GetCustomX**

Returns the Custom dimension object for the specified custom dimension ID. Supersedes the Custom1, Custom2, Custom3 and Custom4 properties.

**Syntax**

```
<HsvMetadata>.GetCustomX(lDimId)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lDimId</td>
<td>Long. The custom dimension ID.</td>
</tr>
</tbody>
</table>

**Return Value**

HsvCustom. The Custom dimension object associated with the specified ID.

**Example**

```vbscript
Dim cCustom As HsvCustom
‘ Get the custom dim object for Custom1
Set cCustom = cMetadata.GetCustomX(DIMID_CUSTOMBASE)
```

**GetDefaultScaleAndNumDecimal**

Returns the default value for the scale setting and number of decimal places.
Syntax

```<HsvMetadata>.GetDefaultScaleAndNumDecimal, lEntity, lParent, lValue, lAccount, sNumDecimals, sScale```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the cell's Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent of the lEntity argument's entity.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the cell's Value dimension member.</td>
</tr>
<tr>
<td>lAccount</td>
<td>Long (ByVal). The member ID of the cell's Account dimension member.</td>
</tr>
<tr>
<td>sNumDecimals</td>
<td>Short(ByRef). The value for the specified number of decimals.</td>
</tr>
<tr>
<td>sScale</td>
<td>Short(ByRef). The value for the specified scale.</td>
</tr>
</tbody>
</table>

### GetDefaultValueOfActiveStatusAccount

Returns the default value for an application’s active status account. This method returns the DefaultValueForActive application setting’s value.

Syntax

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pdValue</td>
<td>Double. Returns the active status account’s default value, which is either 0 or 1.</td>
</tr>
</tbody>
</table>

### GetDimension

Returns an object reference to the specified dimension. This works the same as the Dimension property, except that the dimension ID parameter is a Long instead of Integer. Supersedes the Dimension property.

Syntax

```<HsvMetadata>.GetDimension(lDimId)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lDimId</td>
<td>Long. The desired dimension ID.</td>
</tr>
</tbody>
</table>

**Return Value**

IHsvTreeInfo interface for the dimension.

**Example**

```Dim cTI As IHsvTreeInfo```
Set cTI = cMetadata.GetDimension( DIMID_SCENARIO )

**GetDimensionIdFromName**

Converts a dimension name to its associated ID. For custom dimensions, the name can be either the short name or alias.

**Syntax**

```vba
<HsvMetadata>.GetDimensionIdFromName bstrDimName, plDimId
```

**Argument**

- `bstrDimName` String. The dimension name. For custom dimensions, this can be either the short name or alias.
- `plDimId` Long. The dimension ID associated with the specified name.

**Return Value**

None.

**Example**

```vba
Dim lID As Long
Dim lID As Long
lID = GetDimensionIdFromName "Entity", lID
```

**GetFdmAppName**

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

**Syntax**

```vba
<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**

```vba
<HsvMetadata>.GetFrequencyID bstrFrequencyLabel, plFrequencyID
```
### GetICPEntitiesAggregationWeight

*For internal use.*

### GetICPEntitiesAggregationWeightEx

*For internal use.*

### GetMaxCustomDimNumber

Returns the maximum custom dimension number for the application, in the range 1-n

#### Syntax

```csharp
<HsvMetadata>.GetMaxCustomDimNumber()
```

#### Return Value

Long. The highest custom dimension number, ranging from 1-n, where n=the maximum for the application.

#### Example

```csharp
Long lMaxNum As Long
lMaxNum = cMetadata.GetMaxCustomDimNumber()
```

### GetMaxDimId

Returns the maximum valid dimension ID for an application.

#### Syntax

```csharp
<HsvMetadata>.GetMaxDimId()
```

#### Return Value

Long. The maximum dimension ID.

#### Example

```csharp
Dim lMaxID As Long
lMaxID = cMetadata.GetMaxDimId()
```
**GetNumDimensions**

Returns the total number of dimensions in the application.

**Syntax**

```csharp
<HsvMetadata>.GetNumDimensions(lNumDims)
```

**Argument** **Description**

`lNumDims` Long (byRef).

**Example**

```csharp
Dims lNumDims
CMetadata.GetNumDimension(lNumDims)
```

**GetSupportSubmissionPhaseForAccountFlag**

Indicates whether phased submissions are enabled for the Account dimension.

**Syntax**

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

**Argument** **Description**

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

**Example**

GetSupportSubmissionPhaseForAccountFlag is used in the example for GetUseSubmissionPhaseFlag.

**GetSupportSubmissionPhaseForCustom1Flag**

*Deprecated* - use GetSupportSubmissionPhaseForCustomXFlag.

**GetSupportSubmissionPhaseForCustom2Flag**

*Deprecated* - use GetSupportSubmissionPhaseForCustomXFlag.

**GetSupportSubmissionPhaseForCustom3Flag**

*Deprecated* - use GetSupportSubmissionPhaseForCustomXFlag.
### GetSupportSubmissionPhaseForCustom4Flag

*Deprecated* - use GetSupportSubmissionPhaseForCustomXFlag.

### GetSupportSubmissionPhaseForCustomXFlag

Indicates whether phased submission is enabled for the specified custom dimension. Note: This method supersedes the GetSupportSubmissionPhaseForCustom1Flag, GetSupportSubmissionPhaseForCustom2Flag, GetSupportSubmissionPhaseForCustom3Flag and GetSupportSubmissionPhaseForCustom4Flag methods.

**Syntax**

```csharp
<HsvMetadata>.GetSupportSubmissionPhaseForCustomXFlag  lCustomDimID, pbSupportSubPhase
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lCustomDimID</code></td>
<td>Long. The desired custom dimension ID.</td>
</tr>
<tr>
<td><code>pbSupportSubPhase</code></td>
<td>Boolean. Returns True if phased submission is supported, False otherwise.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```csharp
Function CheckSuppPhasedSub( dimId As Long) As Boolean
    Dim bSupported As Boolean
    cMetadata.GetSupportSubmissionPhaseForCustomXFlag(dimId, bSupported)
    CheckSuppPhasedSub = bSupported
End Function
```

### GetSupportSubmissionPhaseForICPFlag

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

**Syntax**

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pbSupportSubmissionPhaseForICP</code></td>
<td>Boolean. Returns TRUE if phased submissions are enabled for the dimension.</td>
</tr>
</tbody>
</table>

**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.

```vbnet
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
  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
End Sub
```
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 property, use `GetByIndexValidationAccount`.

**Syntax**

```vba
<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.

```vba
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
```
GetValidationAccountMembers

Return all of the validation Account members for an application. Returns 0-based arrays, where index 0 = Validation Account 1, 1=Validation Account 2, and so on.

Syntax

`<HsvMetadata>.GetValidationAccountMembers pvaralMemberIds, pvarabstrLabels`

**Argument** | **Description**
---|---
`pvaralMemberIds` | Long array. Returns an array of validation account member IDs. An array element will contain MEMBERNOTUSED if there is no member for this account.

`pvarabstrLabels` | String array. Returns an array of validation account member labels. An array element will contain an empty string if there is no member for this account.

**Return Value**

None.

**Example**

```vbnet
Dim aIds, aLabels
Dim i

Dim cMetadata.GetValidationAccountMembers aIds, aLabels

For i=0 to UBound(aIds)
    Debug.Print “ValidationAccount” + CStr(i+1)
    if ( aIds(i) = MEMBERNOTUSED ) then
        Debug.Print “not specified”
    else
        Debug.Print CStr(aIds(i)) + “ “ aLabels(i)
    End If
Next
```

IsCustomDimValidForApp

Given a dimension ID, determines whether the custom dimension is valid for an application.

Syntax

`<HsvMetadata>.IsCustomDimValidForApp lCustomDimID, pvbIsValid`

**Argument** | **Description**
---|---
`lCustomDimID` |  
`pvbIsValid` | .
**Return Value**

None.

**Example**

```vba
Dim i As Long
Dim bIsValid As Boolean
for i=DIMID_CUSTOMBASE to DIMID_CUSTOMBASE + 10
    cMetadata.IsCustomDimValidForApp i, bIsValid
next
```

**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**

```vba
<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 857.
- **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 is executed.

```vba
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. This method indicates whether the Organization by Period
feature is enabled.

Syntax

     <HsvMetadata>.IsOrgByPeriodApplication pvarbIsEnabled

Argument    Description

   pvarbIsEnabled  Boolean. Returns TRUE if the OrgByPeriodApplication application setting is on, FALSE if it is off.

Example

The following example tests the OrgByPeriodApplication application setting. If the setting is
enabled, any code placed within the If structure is 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! This method fails if used 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 781.

Syntax

     <HsvMetadata>.Load bstrServerFilename, bstrServerLogFilename, varavSettings
**Argument**

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

**Tip:** If the load file is in an obsolete file format, error 80040526 (hexadecimal) occurs.

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

*varavSettings*   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 61 on page 226.

**Tip:** Use *EnumLoadOptions* to return information about the valid load options.

The following table describes the metadata load options. 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 data is erased. These are the options indexed from 9 through 16.

---

**Table 61  Metadata Load Options**

<table>
<thead>
<tr>
<th>Index</th>
<th>Load Option Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Option:</strong> Currencies</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether currencies are loaded.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to load currencies, otherwise FALSE.</td>
</tr>
<tr>
<td>2</td>
<td><strong>Option:</strong> Scenarios</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether scenarios are loaded.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to load scenarios, otherwise FALSE.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Option:</strong> Entities</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether entities are loaded.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to load entities, otherwise FALSE.</td>
</tr>
<tr>
<td>4</td>
<td><strong>Option:</strong> Accounts</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether accounts are loaded.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to load accounts, otherwise FALSE.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Option:</strong> Custom1</td>
</tr>
<tr>
<td></td>
<td>Deprecated - use CustomDims</td>
</tr>
<tr>
<td>6</td>
<td><strong>Option:</strong> Custom2</td>
</tr>
<tr>
<td></td>
<td>Deprecated - use CustomDims</td>
</tr>
<tr>
<td>7</td>
<td><strong>Option:</strong> Custom3</td>
</tr>
<tr>
<td></td>
<td>Deprecated - use CustomDims</td>
</tr>
<tr>
<td>Index</td>
<td>Load Option Information</td>
</tr>
<tr>
<td>-------</td>
<td>-------------------------</td>
</tr>
</tbody>
</table>
| 8     | **Option**: Custom4  
     | *Deprecated* - use CustomDims |
| 9     | **Option**: CustomDims  
     | **Usage**: Specifies which custom dimensions to load. The position within the array indicates which custom dimension as follows: first position = Custom1, second position = Custom2, and so on.  
     | **Pass to Load**: Boolean – TRUE to load custom members, otherwise FALSE. |
| 10    | **Option**: ClearCurrencies  
     | *Deprecated* - use ClearAll. |
| 11    | **Option**: ClearScenarios  
     | *Deprecated* - use ClearAll. |
| 12    | **Option**: ClearEntities  
     | *Deprecated* - use ClearAll. |
| 13    | **Option**: ClearAccounts  
     | *Deprecated* - use ClearAll. |
| 14    | **Option**: ClearCustom1  
     | *Deprecated* - use ClearAll. |
| 15    | **Option**: ClearCustom2  
     | *Deprecated* - use ClearAll. |
| 16    | **Option**: ClearCustom3  
     | *Deprecated* - use ClearAll. |
| 17    | **Option**: ClearCustom4  
     | *Deprecated* - use ClearAll. |
| 18    | **Option**: ClearAll  
     | **Usage**: Specifies whether previously loaded dimension members are deleted before metadata is loaded.  
     | **Pass to Load**: Boolean – TRUE to delete all previously loaded dimension members. Setting this to TRUE erases all of the application's data. |
| 19    | **Option**: Delimiter  
     | **Usage**: Specifies a load file’s delimiter.  
     | **Pass to Load**: String – a valid delimiter character. EnumLoadOptions returns the valid delimiters. |
| 20    | **Option**: Prescan  
     | **Usage**: Specifies whether a load file is loaded or is merely scanned for syntax accuracy when Load is called.  
<pre><code> | **Pass to Load**: Boolean – TRUE to scan without loading, FALSE to load the metadata. By default, this option is set to FALSE. |
</code></pre>
<table>
<thead>
<tr>
<th>Index</th>
<th>Load Option Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>21</td>
<td><strong>Option:</strong> AppSettings</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether application settings are loaded.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to load application settings, otherwise FALSE.</td>
</tr>
<tr>
<td>22</td>
<td><strong>Option:</strong> FileFormat</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether the metadata load file is in an ASCII text or an XML format.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Long — 0 to load a text file, 1 to load an XML file.</td>
</tr>
<tr>
<td>23</td>
<td><strong>Option:</strong> UseReplaceMode</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether the metadata replaces or is merged with existing metadata.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to replace existing metadata, FALSE to merge with existing metadata.</td>
</tr>
<tr>
<td>24</td>
<td><strong>Option:</strong> ConsolMethods</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether consolidation methods are loaded.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to load consolidation methods, otherwise FALSE.</td>
</tr>
<tr>
<td>25</td>
<td><strong>Option:</strong> ClearConsolMethods</td>
</tr>
<tr>
<td></td>
<td><em>Deprecated</em> - use ClearAll.</td>
</tr>
<tr>
<td>26</td>
<td><strong>Option:</strong> LoadSystemMembers</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether system members can be loaded.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to load system members, otherwise FALSE.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If this option is set to TRUE, you must also set to TRUE the options for the system members to be loaded.</td>
</tr>
<tr>
<td>27</td>
<td><strong>Option:</strong> SystemAccounts</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether to load system accounts.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to load system accounts, otherwise FALSE.</td>
</tr>
<tr>
<td>28</td>
<td><strong>Option:</strong> Values</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether Value dimension members are loaded.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> 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.</td>
</tr>
<tr>
<td>29</td>
<td><strong>Option:</strong> ICPs</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether Intercompany Partner dimension members are loaded.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> 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.</td>
</tr>
<tr>
<td>30</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>31</td>
<td><strong>Option:</strong> CheckIntegrity</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether to validate the integrity of the metadata file against the metadata in the current application.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If integrity errors occur, they are noted in the log file and no portion of the file is loaded into the application.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to check integrity, otherwise FALSE.</td>
</tr>
</tbody>
</table>
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.

```vbscript
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
```

**LoadModuleConfigurations**

Loads the definition file for the module configurations.

**Syntax**

```vbscript
<HsvMetadata>.LoadModuleConfigurations bstrClientFileName, bstrLogFileName
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrClientFileName</code></td>
<td>String (ByVal). The name of the module configuration file.</td>
</tr>
<tr>
<td><code>bstrLogFileName</code></td>
<td>String (ByVal). The name of the log file.</td>
</tr>
</tbody>
</table>

**LoadWithAccessCode**

*For internal use.*

**TranslateApplicationAttributeForDisplay**

*Deprecated* - use `TranslateApplicationAttributeForDisplayExtDim`.

**TranslateApplicationAttributeForDisplayExtDim**

Returns a user-readable string that represents the value of an application setting attribute. You must pass the raw value of the attribute. Supersedes `TranslateApplicationAttributeForDisplay`.

**Syntax**

```vbscript
<HsvMetadata>.TranslateApplicationAttributeForDisplay(lAttributeId, varAttribValue)
```
### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lAttributeId</code></td>
<td>Long, <code>ATTRIBEX_</code> enum value specifying the desired attribute.</td>
</tr>
<tr>
<td><code>varAttribValue</code></td>
<td>Variant (ByVal). The raw value of the attribute.</td>
</tr>
</tbody>
</table>

### Return Value

String. The user-readable attribute value.

### Example

```vba
Dim value As Variant, sValue As String
value = cMetadata.GetApplicationAttributeExtDim(ATTRIBEX_APPSETTING_DEFAULTCURRENCY)
sValue = cMetadata.TranslateApplicationAttributeForDisplayExtDim(ATTRIBEX_APPSETTING_DEFAULTCURRENCY, value)
Debug.Print sValue
```

### 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 66, 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 198.

#### 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 857.

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:

```vbnet
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; see “HsvMetadata Type Library Overview” on page 65. 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 75, 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. See “Dimension” on page 231.

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**

```csharp
<IHsvTreeInfo>.EnumAllMemberLabels pvaralMemberLabels
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvaralMemberLabels</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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`.

```csharp
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**

```csharp
<IHsvTreeInfo>.EnumAllParentAndChildIDs pvaralParentIDs, pvaralChildIDs
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvaralParentIDs</td>
<td>Variant array. Returns the IDs of the parent members. The array is returned as a Long subtype.</td>
</tr>
</tbody>
</table>
Argument Description

pvarChildIDs 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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvarabstrParents</td>
<td>Variant array. Returns the labels of the parent members. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarabstrChildren</td>
<td>Variant array. Returns the labels of the child members. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

**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. 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:

```text
!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` returns for this set of entities:

<table>
<thead>
<tr>
<th>Index #</th>
<th>Parent array element</th>
<th>Child array element</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>[None]</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>Regional</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Regional</td>
<td>UnitedStates</td>
</tr>
<tr>
<td>3</td>
<td>UnitedStates</td>
<td>California</td>
</tr>
<tr>
<td>4</td>
<td>California</td>
<td>Sunnyvale</td>
</tr>
<tr>
<td>5</td>
<td>California</td>
<td>FosterCity</td>
</tr>
</tbody>
</table>

When an entity has no child, the corresponding parent array element is 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.

```vba
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 member’s ancestors.

**Syntax**

```vba
<IHsvTreeInfo>.EnumAncestors lItemID, bIgnoreDups, pvaralIDs
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lItemID</em></td>
<td>Long (ByVal). The member ID.</td>
</tr>
</tbody>
</table>
Argument | Description
--- | ---
**bIgnoreDups** | Boolean (ByVal). Specifies whether duplicate member IDs are returned. Pass TRUE to filter out duplicates, FALSE to return duplicates.

**pvaralIDs** | 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.

```vba
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 parent’s base-level members. You can also use EnumBaseMemberIDs to get all of a dimension’s base-level members.

**Syntax**

```vba
<IHsvTreeInfo>.EnumBaseMemberIDs lParentID, bIgnoreDuplicates, pvaralBaseMemberIDs
```

**Argument**

- **lParentID** | Long (ByVal). Identifies whether member IDs are returned for a node or for the entire dimension:
  - To return a node’s base-level member IDs, pass the member ID of the node’s parent.
  - To return all of a dimension’s base-level members, pass -1.

- **bIgnoreDuplicates** | Boolean (ByVal). Specifies whether duplicate member IDs are returned. Pass TRUE to filter out duplicates, FALSE to return duplicates.

- **pvaralBaseMemberIDs** | 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`.

```vba
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 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**

```vba
<IHsvTreeInfo>.EnumDefaultAncestors lMemberID, pvaralAncestorsIDs
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lMemberID</code></td>
<td>Long (ByVal). The member ID of the dimension member.</td>
</tr>
<tr>
<td><code>pvaralAncestorsIDs</code></td>
<td>Variant. Returns an array of the default ancestors’ member IDs, or an empty Variant if a top-level member is passed to the <code>lMemberID</code> argument.</td>
</tr>
</tbody>
</table>

If an array is returned, it has a subtype of Long.

**EnumDefaultAncestorsLabels**

Returns the labels of a 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

\texttt{<IHsvTreeInfo>.EnumDefaultAncestorsLabels lMemberID, pvarabstrAncestorsLabels}

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{lMemberID}</td>
<td>Long (ByVal). The member ID of the dimension member.</td>
</tr>
<tr>
<td>\textit{pvarabstrAncestorsLabels}</td>
<td>Variant. Returns an array of the default ancestors' labels, or an empty Variant if a top-level member is passed to the \textit{lMemberID} argument. If an array is returned, it has a subtype of String.</td>
</tr>
</tbody>
</table>

Example

The following function returns the labels of an Entity dimension member’s default ancestors.

\begin{verbatim}
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
\end{verbatim}

\textbf{EnumDescendants}

Returns an array containing the member IDs of a member’s descendants.

Syntax

\texttt{<IHsvTreeInfo>.EnumDescendants lItemID, bIgnoreDups, pvaralIDs}

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\textit{lItemID}</td>
<td>Long (ByVal). The member ID of the dimension member.</td>
</tr>
<tr>
<td>\textit{bIgnoreDups}</td>
<td>Boolean (ByVal). Specifies whether duplicate member IDs are returned. Pass TRUE to filter out duplicates, FALSE to return duplicates.</td>
</tr>
<tr>
<td>\textit{pvaralIDs}</td>
<td>Variant array. Returns the array of the descendants' member IDs. The array is returned as a Long subtype.</td>
</tr>
</tbody>
</table>

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 \texttt{EnumDescendants}, passing the member IDs to \texttt{GetLabel}.

\begin{verbatim}
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
End Sub
\end{verbatim}
Member = cTreeInfo.GetItemID("UnitedStates")
cTreeInfo.EnumDescendants Member, 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**

```xml
<IHsvTreeInfo>.EnumIDsOfChildren(lListTopMemberID, lItemID, pvarChildIDArray)
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lListTopMemberID</td>
<td>Long (ByVal). Pass the HFMConstants type library constant TREE_ROOT to this argument.</td>
</tr>
<tr>
<td>lItemID</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>● To return the top members of a hierarchy, pass -1.</td>
</tr>
<tr>
<td></td>
<td>● To return the children of a member, pass the parent’s member ID.</td>
</tr>
<tr>
<td>pvarChildIDArray</td>
<td>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 lItemID.</td>
</tr>
</tbody>
</table>

**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.

```vba
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 that calls `EnumIDsOfChildren` with the following lines, the example prints the children of the California entity:

```vbnet
lPar = cTreeInfo.GetItemID("California")
iSuccess = cTreeInfo.\_EnumIDsOfChildren\_(TREERoot, lPar, \_
vaChildIDs)
```

**EnumIDsOfIChildren**

Returns children and the parent item of the requested children.

**Syntax**

```
<IVsTreeInfo>.EnumIDsOfIChildren(lListTopMemberID, lItemID, pvarChildIDArray)
```

**Argument Description**

- **lListTopMemberID** Long (ByVal). Pass the HFMConstants type library constant `TREE_ROOT` to this argument.
- **lItemID** 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:
  - 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.
- **pvarChildIDArray** 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 `lItemID`.

**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.

```vbnet
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.\_EnumIDsOfICChildren\_(TREERoot, 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**

`pvarabstrListNames` 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`.

```vba
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 `pvaralItemIDs` argument and the corresponding parent member IDs in the `pvaralParentIDs` argument. For the other dimensions, `EnumMembers` returns members in the `pvaralItemIDs` 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 node.

**Syntax**

```
<IHsvTreeInfo>.EnumMembers lListID, lListTopMemberID, pvaralItemIDs, pvaralParentIDs
```
**Argument** | **Description**
--- | ---
`lListID` | Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the HFMConstants type library constant `MEMBER_LIST_ALL_HIERARCHY` 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 `GetMemberListID`. See “GetMemberListID” on page 265.

`lListTopMemberID` | Long (ByVal). The usage of this argument depends on what you pass to the `lListID` argument:
- If you pass `MEMBER_LIST_ALL_HIERARCHY` to `lListID`, you can return either member IDs for all of the dimension’s members by passing the `TREE_ROOT` 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 `MEMBER_LIST_ALL_HIERARCHY` to `lListID`, the `lListTopMemberID` argument is ignored. Because this argument is not optional, you must still pass a valid Long.

`pvaralItemIDs` | 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.

`pvaralParentIDs` | Variant array. For the Entity dimension, this argument returns the member IDs of the parents of the members returned in the `pvaralItemIDs` 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.

```vba
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 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**

```vba
<IHsvTreeInfo>.EnumMembers2 lListID, lListTopMemberID, lScenario, lYear, lPeriod, lEntity, pvaralItemIDs, pvaralParentIDs
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lListID</strong></td>
<td>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 MEMBER_LIST_ALL_HIERARCHY to return the members in the default hierarchy. <strong>Tip:</strong> You can get member list IDs with GetMemberListID.</td>
</tr>
</tbody>
</table>
| **lListTopMemberID** | Long (ByVal). The usage of this argument depends on what you pass to the lListID argument:  
  - If you pass MEMBER_LIST_ALL_HIERARCHY to lListID, you can return either member IDs for all of the dimension’s members by passing the TREE_ROOT 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 MEMBER_LIST_ALL_HIERARCHY to lListID, the lListTopMemberID argument is ignored. Because this argument is not optional, you must still pass a valid Long. |
| **lScenario**    | Long (ByVal). The member ID of the Scenario dimension member for a dynamic member list.                                                      |
| **lYear**        | Long (ByVal). The member ID of the Year dimension member for a dynamic member list.                                                         |
| **lPeriod**      | Long (ByVal). The member ID of the Period dimension member for a dynamic member list.                                                       |
| **lEntity**      | Long (ByVal). The member ID of the Entity dimension member for a dynamic member list.                                                      |
| **pvaralItemIDs**| 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. |
| **pvaralParentIDs**| Variant array. For the Entity dimension, this argument returns the member IDs of the parents of the members returned in the pvaralItemIDs 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 scenario, year, and period. In the array, the parent and child entities are delimited by periods.

```vba
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.EnumMembers 2 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 an attribute. For example, a user can enumerate all of the entities with ATTFIB_ENTITY_IS_ICP = true.

**Syntax**

```
<HsvTreeInfo>.EnumMembersWithAttribValue attribId, varAttribValue, flags, pvaralIDs, pvarabstrLabels
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>attribId</td>
<td>Integer (ByVal). Attribute ID of member.</td>
</tr>
<tr>
<td>varAttribValue</td>
<td>Variant (ByVal). The raw value of the attribute. To obtain an attribute's raw value, use <code>GetApplicationAttribute</code>.</td>
</tr>
<tr>
<td>flags</td>
<td>Long (ByVal). Used to refine the search. Example:</td>
</tr>
<tr>
<td></td>
<td>ATTFIB_FILTER_FLAG_INCLUDE_SYS_MEMBERS: returns members that include system members if 0 is passed, system members are not returned although they match the searched value <code>varAttribValue</code>.</td>
</tr>
<tr>
<td></td>
<td>ATTFIB_FILTER_FLAG_CASE_SENSITIVE: performs case-sensitive search of <code>varAttribValue</code>.</td>
</tr>
<tr>
<td>pvaralIDs</td>
<td>Variant array. Returns the array of the member IDs. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarabstrLabels</td>
<td>Variant array. Returns an array of member labels. The array is returned as a String subtype</td>
</tr>
</tbody>
</table>

**EnumParents**

Returns an array containing the member IDs of a member’s parent members.

**Syntax**

```
<IHsvTreeInfo>.EnumParents lItemID, pvaralIDs
```
Argument | Description
---|---
`lItemID` | Long (ByVal). The member ID of the member.
`pvaralIDs` | 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`.

```vba
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**

```vba
<IHsvTreeInfo>.EnumSortedIDsOfChildren(lListTopMemberID, lItemID, lSortOptions, pvarChildIDArray)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lListTopMemberID</code></td>
<td>Long (ByVal). Pass the HFMConstants type library constant <code>TREE_ROOT</code> to this argument.</td>
</tr>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the member for which to return children.</td>
</tr>
</tbody>
</table>
| `lSortOptions` | Long (ByVal). The member information by which to sort. You can pass any combination of the HFMConstants type library constants that are listed in "Metadata Information Constants" on page 892 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.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvarChildIDArray</code></td>
<td>Variant. Returns an array of the children’s member IDs, with the array sorted as specified by the <code>lSortOptions</code> argument.</td>
</tr>
</tbody>
</table>
Return Value

Integer. Indicates whether the call succeeded. Returns 0 upon success, or a non-zero error number upon failure.

**EnumSortedIDsOfIChildren**

Returns sorted children and the parent item of the specified children.

Syntax

```
<HsvTreeInfo>.EnumSortedIDsOfIChildren(lListTopMemberID, lItemID, lSortOptions, pvarChildIDArray)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lListTopMemberID</td>
<td>Long (ByVal). Pass the HFMConstants type library constant TREE_ROOT to this argument.</td>
</tr>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the member for which to return children.</td>
</tr>
<tr>
<td>lSortOptions</td>
<td>Long (ByVal). The member information by which to sort. You can pass any combination of the HFMConstants type library constants that are listed in &quot;Metadata Information Constants&quot; on page 892 and that include the word &quot;SORT.&quot;</td>
</tr>
<tr>
<td>Note:</td>
<td>By default, EnumSortedMembers sorts in ascending order, so there in no constant for ascending order. You can specify multiple sorting options by using the Or operator with these constants.</td>
</tr>
<tr>
<td>pvarChildIDArray</td>
<td>Variant. Returns an array of the children’s member IDs, and the parent itself, with the array sorted as specified by the lSortOptions argument.</td>
</tr>
</tbody>
</table>

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
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lListID</td>
<td>Long (ByVal). The ID of the member list.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> You can get member list IDs with <code>GetMemberListID</code>.</td>
</tr>
<tr>
<td>lListTopMemberID</td>
<td>Long (ByVal). Specifies whether to return all members on the list or only a</td>
</tr>
<tr>
<td></td>
<td>member and its descendants. Valid values are described below:</td>
</tr>
<tr>
<td></td>
<td>• To return all members, pass the HFMConstants type library constant MEMBERNOTUSED.</td>
</tr>
<tr>
<td></td>
<td>• To return only a member and its descendants, pass the member ID of the</td>
</tr>
<tr>
<td></td>
<td>member.</td>
</tr>
<tr>
<td>/SortOptions</td>
<td>Long (ByVal). The member information by which to sort. You can pass any</td>
</tr>
<tr>
<td></td>
<td>combination of the HFMConstants type library constants that are listed in</td>
</tr>
<tr>
<td></td>
<td>“Metadata Information Constants” on page 892 and that include the word “SORT.”</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> By default, EnumSortedMembers sorts in ascending order, so there</td>
</tr>
<tr>
<td></td>
<td>in no constant for ascending order.</td>
</tr>
<tr>
<td></td>
<td>You can specify multiple sorting options by using the Or operator with these</td>
</tr>
<tr>
<td></td>
<td>constants.</td>
</tr>
<tr>
<td>pvaralItemIDs</td>
<td>Variant. Returns an array of member IDs that are sorted as specified by the</td>
</tr>
<tr>
<td></td>
<td>lSortOptions argument.</td>
</tr>
<tr>
<td>pvaralParentIDs</td>
<td>Variant. Returns an array of member IDs of the parents of the members</td>
</tr>
<tr>
<td></td>
<td>returned by the pvaralItemIDs argument.</td>
</tr>
</tbody>
</table>

**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.

```vba
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**
--- | ---
**lListID** | Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the HFMConstants type library constant MEMBER_LIST_ALL_HIERARCHY to search the default hierarchy, or a valid list ID to search a member list. You can get member list IDs with GetMemberListID.

**lTopMemberID** | Long (ByVal). The usage of this argument depends on what you pass to the lListID argument:
- If you pass MEMBER_LIST_ALL_HIERARCHY to lListID, you can either search the entire dimension hierarchy by passing the HFMConstants type library constant TREE_ROOT, or search a node’s members by passing the member ID of the node’s parent.
- If you pass anything other than MEMBER_LIST_ALL_HIERARCHY to lListID, the lListTopMemberID argument is ignored. Since this argument is not optional, you must still pass a valid Long.

**bstrSearchText** | 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, Find searches for all members that begin with the passed string.

**bForward** | Boolean (ByVal). Indicates whether the search goes forwards or backwards in the dimension hierarchy or member list. Pass TRUE to search forwards, FALSE to search backwards.

**plPos** | 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 Find. Use this argument as follows:
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 Find (as shown in the example).

**plMemberID** | Long. If a member label is found that matches the search string, this argument returns the member ID of the member.

**plParentID** | Long. If a member label is found that matches the search string, this argument returns the member ID of the member’s parent.

**pbFoundMatch** | 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 dimension and member list that partially match the specified string. The example also prints labels of parent members.

```vbnet
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
   Debug.Print sParLabel
   bRet = False
   lPos = -1
   lPar = -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
         Debug.Print sParLabel
   Loop
   bRet = False
Loop
```

IHsvTreeInfo Interface Methods 249
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 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**

```vbc
<IHsvTreeInfo>.FindByDesc lListID, lTopMemberID, bstrSearchText, bForward, lLanguageID, plPos, plMemberID, plParentID, pbFoundMatch
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lListID</em></td>
<td>Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the HFMConstants type library constant MEMBER_LIST_ALL_HIERARCHY to search the default hierarchy, or a valid list ID to search a member list. You can get member list IDs with GetMemberListID.</td>
</tr>
</tbody>
</table>
| *lTopMemberID*| Long (ByVal). The usage of this argument depends on what you pass to the *lListID* argument:  
- If you pass MEMBER_LIST_ALL_HIERARCHY to *lListID*, you can either search the entire dimension hierarchy by passing the HFMConstants type library constant TREE_ROOT, or search a node's members by passing the member ID of the node's parent.  
- If you pass anything other than MEMBER_LIST_ALL_HIERARCHY to *lListID*, the *lTopMemberID* argument is ignored. Since this argument is not optional, you must still pass a valid Long. |
| *bstrSearchText* | 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 begin with the passed string. |
| *bForward*    | Boolean (ByVal). Indicates whether the search goes forwards or backwards in the dimension hierarchy or member list. Pass TRUE to search forwards, FALSE to search backwards. |
| *lLanguageID* | Long (ByVal). The ID of the language. You can get this ID with the EnumLanguages method of the HsvMetadata object. |
| *plPos*       | 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 Find. Use this argument as follows:  
  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 FindByDesc (as shown in the example). |
| *plMemberID*  | Long. If a description is found that matches the search string, this argument returns the member ID of the member. |
| *plParentID*  | Long. If a description is found that matches the search string, this argument returns the member ID of the member’s parent. |
| *pbFoundMatch*| 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 dimension and member list that have descriptions partially matching the specified string. The example also prints labels of parent members.

```vba
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

```vba
<IHsvTreeInfo>.FindMatchingMembersFromHierarchy bstrSearchText, lTopMemberID, varbExactMatch, pvaravaralPaths
```

Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrSearchText</td>
<td>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 begin with the passed string.</td>
</tr>
<tr>
<td>lTopMemberID</td>
<td>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 TREE_ROOT.</td>
</tr>
<tr>
<td>varbExactMatch</td>
<td>Boolean (ByVal). Specifies whether to search for an exact or partial match. Pass TRUE for an exact match search, FALSE for a partial match search.</td>
</tr>
<tr>
<td>pvaravaralPaths</td>
<td>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 is similar to this list:</td>
</tr>
<tr>
<td></td>
<td>• (0) (0) = UnitedStates</td>
</tr>
<tr>
<td></td>
<td>• (0) (1) = California</td>
</tr>
<tr>
<td></td>
<td>• (1) (0) = Canada</td>
</tr>
</tbody>
</table>
The following subroutine prints to Visual Basic’s Immediate window the labels of all members of a dimension with labels that partially match the specified string.

```vba
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 language that match a search string.

**Syntax**

```
<IHsvTreeInfo>.FindMatchingMembersFromHierarchyByDesc bstrSearchText, lTopMemberID, varbExactMatch, lLanguageID, pvaravaralPaths
```

**Argument Description**

- **bstrSearchText**: 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 begin with the passed string.

- **lTopMemberID**: 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 `TREE_ROOT`.

- **varbExactMatch**: Boolean (ByVal). Specifies whether to search for an exact or partial match. Pass TRUE for an exact match search, FALSE for a partial match search.

- **lLanguageID**: Long (ByVal). The ID of the language. You can get this ID with the `EnumLanguages` method of the HsvMetadata object.

- **pvaravaralPaths**: 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 is similar to this list:

- `l(0)(0) = UnitedStates`
- `l(0)(1) = California`
- `l(1)(0) = Canada`
Example

The following subroutine prints to Visual Basic’s Immediate window the labels of all members of a dimension with descriptions that partially match the specified string.

```vbscript
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 language.

Syntax

```vbscript
<IHsvTreeInfo>.FindMatchingMembersFromHierarchyWildCard bstrSearchText, lTopMemberID, _
   llLanguageID, pvaravaralPaths
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrSearchText</td>
<td>String (ByVal). String (ByVal). The search string. You can use asterisks (*) as wildcard characters. The following list describes the rules for wildcard searching:</td>
</tr>
<tr>
<td></td>
<td>● You can use a wildcard at the beginning of the search string.</td>
</tr>
<tr>
<td></td>
<td>● You can use a wildcard at the end of the search string.</td>
</tr>
<tr>
<td></td>
<td>● You can use wildcards at both the beginning and end of the search string.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Asterisks placed anywhere other than the beginning or end are treated as literal characters, not as wildcards.</td>
</tr>
<tr>
<td></td>
<td>● You can omit wildcards to perform an exact match search.</td>
</tr>
<tr>
<td>lTopMemberID</td>
<td>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 TREE_ROOT.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>LanguageID</strong></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>• To search for labels, pass the HFMConstants type library constant HFM_NO_LANGUAGE.</td>
</tr>
<tr>
<td></td>
<td>• 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.</td>
</tr>
<tr>
<td><strong>pvaravaralPaths</strong></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>For example, suppose that a search returns two members: Canada (with no parent) and UnitedStates.California. The indexes to the array of arrays is similar to this list:</td>
</tr>
<tr>
<td></td>
<td>• (0)(0) = UnitedStates</td>
</tr>
<tr>
<td></td>
<td>• (0)(1) = California</td>
</tr>
<tr>
<td></td>
<td>• (1)(0) = Canada</td>
</tr>
</tbody>
</table>

**Example**

The following subroutine prints to Visual Basic’s Immediate window the labels of all members of a dimension with labels that match the specified string.

```vbscript
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 member.

**Syntax**

`<IHsvTreeInfo>.GetAllPathsToMember lMemberID, varabstrPaths`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lMemberID</strong></td>
<td>Long (ByVal). The member ID of the member for which you want to return the hierarchical paths.</td>
</tr>
</tbody>
</table>
**GetAttributeHierarchy**

Returns an array of strings that represent the hierarchical paths to the member. The array contains an item for each path to the member.

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.

```vba
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 member’s metadata attribute.

**Syntax**

```vba
<IHsvTreeInfo>.GetAttributeValue lItemID, iAttribute, vValue
```

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the dimension member.</td>
</tr>
<tr>
<td>iAttribute</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>- “Scenario Attribute Constants” on page 861</td>
</tr>
<tr>
<td></td>
<td>- “Entity Attribute Constants” on page 854</td>
</tr>
<tr>
<td></td>
<td>- “Account Attribute Constants” on page 864</td>
</tr>
<tr>
<td></td>
<td>- “Account Attribute Constants” on page 846</td>
</tr>
<tr>
<td></td>
<td>- “Custom Dimension Attributes” on page 851</td>
</tr>
<tr>
<td></td>
<td>- “Intercompany Partner Attribute Constants” on page 859</td>
</tr>
<tr>
<td>vValue</td>
<td>Variant. Returns the attribute’s value.</td>
</tr>
</tbody>
</table>

**Example**

The following function returns the UserDefined1 attribute of a Scenario dimension member.

**Note:** You can also get this attribute with HsvScenarios.GetUserDefined1.

```vba
Function getScenUserDef1(lId As Long) As Variant
```

---

**IHSvTreInfo Interface Methods** 255
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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lMemberID</em></td>
<td>Long (ByVal). The member ID of the member.</td>
</tr>
<tr>
<td><em>plPos</em></td>
<td>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 returns 3 and Management returns 4.</td>
</tr>
</tbody>
</table>

```
[None]  
  ▼ UnitedStates
    ▼ NewYork
    ▼ California
  ▼ Management
    ▼ Acme
    ▼ Warner
```

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)`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lListID</strong></td>
<td>Long (ByVal). Pass the HFMConstants type library constant <code>MEMBER_LIST_ALL_HIERARCHY</code> to use the default dimension hierarchy.</td>
</tr>
<tr>
<td><strong>Caution!</strong></td>
<td>If you pass a value other than <code>MEMBER_LIST_ALL_HIERARCHY</code>, the <code>lListTopMemberID</code> and <code>lParentID</code> arguments are ignored, and only the member ID passed in the <code>lItemID</code> argument is returned.</td>
</tr>
<tr>
<td><strong>lListTopMemberID</strong></td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>* To return the entire path of ancestors, pass the HFMConstants type library constant <code>TREE_ROOT</code>.</td>
</tr>
<tr>
<td></td>
<td>* To return a portion of the path, pass the member ID of the topmost member that you want to return.</td>
</tr>
<tr>
<td><strong>lItemID</strong></td>
<td>Long (ByVal). The member ID of the member for which you want to return the path.</td>
</tr>
<tr>
<td><strong>lParentID</strong></td>
<td>Long (ByVal). The member ID of the parent of the <code>lItemID</code> member.</td>
</tr>
<tr>
<td><strong>pvarItemIDHierarchy</strong></td>
<td>Variant array. Returns the member IDs for the path of ancestors. The <code>lItemID</code> argument’s member ID is included at the bottom of the path. The array is returned as a Long subtype.</td>
</tr>
</tbody>
</table>

**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.

```vba
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**

`plDefaultMemberID` Long. Returns the member ID of the default member.

**Example**

The following function returns the label of a 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`.

```vba
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 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**

`lMemberID` Long (ByVal). The member ID of the dimension member.

`plParentID` Long. Returns the member ID of the default parent. If the `lMemberID` argument identifies a top-level member of the dimension’s hierarchy, -1 is returned.

**GetDefaultParentLabel**

Returns the label of a dimension member’s default parent.

**Tip:** You can get the ID of a member’s default parent with `GetDefaultParent`.

Syntax

`<IHsvTreeInfo>.GetDefaultParentLabel lMemberID, pbstrParentLabel`
**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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the member.</td>
</tr>
<tr>
<td>lLanguageID</td>
<td>Long (ByVal). The ID number of the language. You can get this ID with the EnumLanguages method of the HsvMetadata object.</td>
</tr>
<tr>
<td>pbstrDesc</td>
<td>String. Returns the description of the member.</td>
</tr>
</tbody>
</table>

**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

lListID Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the HFMConstants type library constant MEMBER_LIST_ALL_HIERARCHY 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 GetMemberListID. See “GetMemberListID” on page 265.

lItemID Long (ByVal). The member ID of the dimension member for which you want to return information.

lParentID Long (ByVal). For hierarchical dimensions such as the Entity dimension, pass the member ID of the parent of the lItemID argument’s dimension member.

For non-hierarchical dimensions such as the Scenario dimension, pass the HFMConstants type library constant MEMBERNOTUSED.

pbHasChildren Boolean. If you pass MEMBER_LIST_ALL_HIERARCHY to the lListID argument, pbHasChildren returns TRUE if the lItemID argument’s member has children, otherwise FALSE.

If you pass anything other than MEMBER_LIST_ALL_HIERARCHY to the lListID argument, pbHasChildren always returns FALSE, regardless of whether the lItemID argument’s member has children.

pbstrLabel String. Returns the label of the lItemID argument’s member. For hierarchical dimensions, the label of the lParentID argument’s member is prefixed to the label of the lItemID argument’s member, with a period delimiting the parent and child labels.

pbstrDesc String. Returns the description of the lItemID 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lListID</td>
<td>Long (ByVal). Identifies either the default dimension hierarchy or a member list. Pass the HFMConstants type library constant MEMBER_LIST_ALL_HIERARCHY to return the members in the default hierarchy, or pass a valid list ID to return the members in a member list.</td>
</tr>
<tr>
<td>varalItemIDs</td>
<td>Variant array (ByVal). The member IDs of the dimension members for which you want to return information.</td>
</tr>
<tr>
<td>varalParentIDs</td>
<td>Variant array (ByVal). For hierarchical dimensions such as the Entity dimension, pass the member IDs of the parents of the varalItemIDs argument's dimension members.</td>
</tr>
<tr>
<td></td>
<td>For non-hierarchical dimensions such as the Scenario dimension, pass an array in which each element is set to the HFMConstants type library constant MEMBERNOTUSED.</td>
</tr>
<tr>
<td></td>
<td>Caution! This array must contain the same number of elements as the array passed to the varalItemIDs argument.</td>
</tr>
<tr>
<td>pvaravbHasChildren</td>
<td>Variant array. If you pass MEMBER_LIST_ALL_HIERARCHY to the lListID argument, each array element returns TRUE if the corresponding varalItemIDs array element has children, otherwise FALSE.</td>
</tr>
<tr>
<td></td>
<td>If you pass anything other than MEMBER_LIST_ALL_HIERARCHY to the lListID argument, each array element returns FALSE, regardless of whether the corresponding varalItemIDs array element has children.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a Boolean subtype.</td>
</tr>
</tbody>
</table>
**Argument** | **Description**
--- | ---
pvarabstrLabels | Variant array. Returns the labels of the members. For hierarchical dimensions, the labels of the varalParentIDs array’s members are prefixed to the labels of the varalItemIDs array’s member, with periods delimiting the parent and child labels.
The array is returned as a String subtype.
pvarabstrDescs | Variant array. Returns the descriptions of the members. The descriptions are returned in the default language.
The array is returned as a String subtype.

**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.

```vba
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 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**

```vba```
<IHsvTreeInfo>.GetItemGeneration lMemberID, plGeneration
```

**Argument** | **Description**
--- | ---
lMemberID | Long (ByVal). The member ID of the dimension member.
plGeneration | 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**

```csharp
<IHsvTreeInfo>.GetItemID(bstrLabel)
```

**Argument Description**

- `bstrLabel` 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 `lDimID` 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 857.)
- 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.

```csharp
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.

```csharp
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 GetCellExtDim.

---

### GetItemIDQL

*For internal use.*

### GetItemLevel

Returns the level of a 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**

```csharp
<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 Period dimension member is a leaf member.

```csharp
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**

```csharp
<IHsvTreeInfo>.GetLabel(lItemID, pbstrLabel)
```
**Argument**  
**Description**

*ItemID*  
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 *ItemID* 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**

```c
<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.

```c
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**

```c
<IHsvTreeInfo>.GetMemberListName lListID, pbstrListName
```

**Argument**  
**Description**

*lListID*  
Long (ByVal). The ID of the member list. Get this ID from a method that returns member list IDs.

*pbstrListName*  
String. Returns the name of the member list identified by the *lListID* 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lParentID</td>
<td>Long (ByVal). The member ID of the member for which you want to return the count.</td>
</tr>
<tr>
<td>bIgnoreDuplicates</td>
<td>Boolean (ByVal). Specifies whether duplicate member IDs are included in the count. Pass TRUE to filter out duplicates, FALSE to count duplicates.</td>
</tr>
<tr>
<td>plNumBaseMembers</td>
<td>Long. Returns the count of base-level members.</td>
</tr>
</tbody>
</table>

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 is 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the member.</td>
</tr>
</tbody>
</table>
Argument Description

plNum 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 is executed.

```vbscript
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, using the member’s ID.

Syntax

```vbscript
<IHsvTreeInfo>.GetNumDescendants lParentID, plNum
```

Argument Description

lParentID Long (ByVal). The member’s ID.

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.

```vbscript
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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>plNumMembers</code></td>
<td>Long. Returns the number of members in the dimension to which the IHsvTreeInfo object reference is initialized.</td>
</tr>
</tbody>
</table>

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, using the member’s ID.

Syntax

```<IHsvTreeInfo>.GetNumParents lMemberID, plNum```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lMemberID</code></td>
<td>Long (ByVal). The ID of the dimension member.</td>
</tr>
<tr>
<td><code>plNum</code></td>
<td>Long. Returns the number of parents for the member.</td>
</tr>
</tbody>
</table>

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
gentityParentCount = lNumParents
End Function

GetQualifiedLabel
For internal use.

GetTreeCapabilities
Returns various properties of a dimension.

Syntax
<IHsvTreeInfo>.GetTreeCapabilities pbReadWrite, pbMultiLevels, pbMultiParents, pbDataVariesByParent

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbReadWrite</td>
<td>Boolean. Returns TRUE if the dimension can be edited with Metadata Manager, otherwise FALSE.</td>
</tr>
<tr>
<td>pbMultiLevels</td>
<td>Boolean. Returns TRUE if the dimension allows parent-child relationships, otherwise FALSE.</td>
</tr>
<tr>
<td>pbMultiParents</td>
<td>Boolean. Returns TRUE if the dimension’s members can have multiple parents, otherwise FALSE.</td>
</tr>
<tr>
<td>pbDataVariesByParent</td>
<td>Boolean. Returns TRUE if the dimension’s members data varies by parent, otherwise FALSE. This argument should return TRUE only for the Entity dimension.</td>
</tr>
</tbody>
</table>

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 is 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 is set for the Entity dimension. If it is not, an error message is displayed, and the procedure is exited.

Dim sDim As String
sDim = m_cIHsvTreeInfo.GetTreeName()
If sDim <> "Entity" Then
    MsgBox "Error: The IHsvTreeInfo object reference 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

pdTimeStamp 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 857 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**

`lItemID` Long (ByVal). The member ID of the member.

`pbHasChildren` 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 is executed.

```vbs
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**

`lMemberID` Long (ByVal). The member ID of the possible base-level dimension member.

`lParentID` Long (ByVal). The member ID of the possible higher-level dimension member.

`pvarbIsBaseOf` Boolean. Returns TRUE if the member identified by the `lMemberID` argument is a base-level member beneath the member identified by the `lParentID` 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
lMemberID Long (ByVal). The member ID of the possible child.
lParentID Long (ByVal). The member ID of the possible parent.
pbIsChildOf Boolean. Returns TRUE if the lMemberID member is a child of the lParentID 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
lMemberID Long (ByVal). The member ID of the possible descendant.
Argument Description

lParentID Long (ByVal). The member ID of the possible higher-level dimension member.

pvarbIsDescendantOf Boolean. Returns TRUE if the member identified by the lMemberID argument is a descendant beneath the member identified by the lParentID 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. Using 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

lListID Long (ByVal). Identifies whether a member list or the default dimension hierarchy is used to filter and sort. Pass a valid list ID to use a member list, or pass the HFMConstants type library constant MEMBER_LIST_ALL_HIERARCHY to use the default hierarchy.

Tip: You can get member list IDs with GetMemberListID. See “GetMemberListID” on page 265.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| IListTopMemberID     | Long (ByVal). The usage of this argument depends on what you pass to the IListID argument:  
|                      | • If you pass MEMBER_LIST_ALL_HIERARCHY to IListID, this determines whether you sort and filter by using all of the dimension’s members or by the members of a node. Pass the TREE_ROOT constant to use all members, or the member ID of the node’s parent to use a node.  
|                      | • If you pass anything other than MEMBER_LIST_ALL_HIERARCHY to IListID, the IListTopMemberID argument is ignored. Since this argument is not optional, you must still pass a valid Long. |
| blignoreDuplicates   | Boolean (ByVal). Specifies whether duplicate member IDs are returned. Pass TRUE to filter out duplicates, FALSE to return duplicates.     |
| varInputItemIDs      | Long array (ByVal). The member IDs to be sorted and filtered.                                                                               |
| varInputParentIDs    | Long array (ByVal). The member IDs of the parent members passed in the varInputItemIDs argument. The items in these two arrays must have a one-to-one correspondence. |
| pvaralItemIDs        | Variant array. Returns the filtered and sorted member IDs. The array is returned as a Long subtype.                                           |
| pvaralParentIDs      | Variant array. Returns the IDs of the parents of the members returned in the pvaralItemIDs 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**

```plaintext
<IHsvTreeInfo>.TranslateAttributeValueForDisplay sAttrib, vValue, bstrValue, plDimID
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| sAttrib    | Integer (ByVal). The ID of the attribute. These IDs are represented by the following groups of constants in the HFMConstants type library:  
|            | • “Account Attribute Constants” on page 846  
|            | • “Custom Dimension Attributes” on page 851  
|            | • “Entity Attribute Constants” on page 854  
|            | • “Scenario Attribute Constants” on page 861  
|            | • “Currency Attribute Constants” on page 853  
|            | • “Consolidation Method Attribute Constants” on page 852 |
| vValue     | 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 HsvAccounts.GetAccountType. |
### Argument Description

**bstrValue**  
String. Returns one of the following values:
- 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 `plDimID` argument.

For example, a blank string is returned if `TranslateAttributeValueForDisplay` is called to obtain an account’s Custom1TopMember attribute and the attribute value is a valid Custom 1 dimension member.

**plDimID**  
Long. Returns one of the following values:
- 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.

```vba
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 70, 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:

```vba
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. See “GetCellLevelAccountType” on page 209.

Syntax

```csharp
<HsvAccounts>.GetAccountType lItemID, psAccountType
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the account.</td>
</tr>
<tr>
<td>psAccountType</td>
<td>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 849.</td>
</tr>
</tbody>
</table>

Example

The following function indicates whether an account has an account type of CurrencyRate.

```csharp
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 an Account dimension member’s CalcAttribute attribute.

Syntax

```csharp
<HsvAccounts>.GetCalcAttribute lItemID, pbstrUDAttr
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the Account dimension member.</td>
</tr>
<tr>
<td>pbstrUDAttr</td>
<td>String. Returns the attribute’s value.</td>
</tr>
</tbody>
</table>

GetICPTopMember

Returns the member ID of the Intercompany Partner dimension member assigned as an Account dimension member’s ICPTopMember attribute.
Syntax

```<HsvAccounts>.GetICPTopMember lItemID, plICPTopMember```

### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the account.</td>
</tr>
<tr>
<td><code>plICPTopMember</code></td>
<td>Long. Returns the member ID of the Intercompany Member assigned as the account’s ICPTopMember attribute.</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the account.</td>
</tr>
</tbody>
</table>
| `psIsICP` | Integer. Indicates the value of the account’s IsICP attribute. Valid values are as follows:
  - `-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
    iRet = cAccounts.GetICPTopMember lAcctID, lTopID
    cTreeInfo.GetLabel lTopID, sLabel
    getIsIcpLabel = iRet
End Function```
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. GetNumDecimalPlaces returns the value to which an account’s NumDecimalPlaces attribute is set.

Syntax

<HsvAccounts>.GetNumDecimalPlaces lItemID, psNumDecimalPlaces

Argument Description
lItemID Long (ByVal). The member ID of the account.
psNumDecimalPlaces 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. GetPlugAccount returns the member ID of the account defined as an account’s PlugAcct attribute.

**Syntax**

```vbnet
<HsvAccounts>.GetPlugAccount lAccountID, plPlugAccountID
```

**Argument** | **Description**
--- | ---
`lAccountID` | Long (ByVal). The member ID of the account for which you want to return the plug account.

`plPlugAccountID` | 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.

```vbnet
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 assigned to an account, using the account’s member ID.

**Syntax**

```vbnet
<HsvAccounts>.GetSecurityClassID lItemID, plSecurityClassID
```

**Argument** | **Description**
--- | ---
`lItemID` | Long (ByVal). The account’s member ID.

`plSecurityClassID` | 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.
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.

```vba
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**

```vba
<HsvAccounts>.GetSubmissionGroup lItemID, plSubmissionGroup
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long. The member ID of the account.</td>
</tr>
<tr>
<td><code>plSubmissionGroup</code></td>
<td>Long. Returns the Submission Group value.</td>
</tr>
</tbody>
</table>

**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. This
method returns the member ID of the Custom 1 member defined as an account’s
Custom1TopMember attribute.
Syntax

<HttpResponse>.GetTopMemberOfValidCustom1Hierarchy lItemID, plTopMember

**Argument** | **Description**
--- | ---
*lItemID* | Long (ByVal). The member ID of the account.

*plTopMember* | 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.

```vbscript
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. This method returns the member ID of the Custom 2 member defined as an account’s Custom2TopMember attribute.

Syntax

<HttpResponse>.GetTopMemberOfValidCustom2Hierarchy lItemID, plTopMember

**Argument** | **Description**
--- | ---
*lItemID* | Long (ByVal). The member ID of the account.

*plTopMember* | 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. This method returns the member ID of the Custom 3 member defined as an account’s Custom3TopMember attribute.
Syntax

```
<HsvAccounts>.GetTopMemberOfValidCustom3Hierarchy lItemID, plTopMember
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the account.</td>
</tr>
<tr>
<td><code>plTopMember</code></td>
<td>Long. Returns the member ID of the top Custom 3 dimension member.</td>
</tr>
</tbody>
</table>

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. This method returns the member ID of the Custom 4 member defined as an account’s Custom4TopMember attribute.

Syntax

```
<HsvAccounts>.GetTopMemberOfValidCustom4Hierarchy lItemID, plTopMember
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the account.</td>
</tr>
<tr>
<td><code>plTopMember</code></td>
<td>Long. Returns the member ID of the top Custom 4 dimension member.</td>
</tr>
</tbody>
</table>

Example

See the Example for `GetTopMemberOfValidCustom1Hierarchy`, replacing that method with `GetTopMemberOfValidCustom4Hierarchy`.

### GetUserDefined1

Returns an account’s UserDefined1 attribute, using the account’s member ID.

Syntax

```
<HsvAccounts>.GetUserDefined1 lItemID, pbstrUDAttr
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The account’s member ID.</td>
</tr>
<tr>
<td><code>pbstrUDAttr</code></td>
<td>String. Returns the UserDefined1 attribute.</td>
</tr>
</tbody>
</table>

282  HsvMetadata Type Library
**GetUserDefined2**

Returns an account’s UserDefined2 attribute, using the account’s member ID.

Syntax

```csharp
<HsvAccounts>.GetUserDefined2 lItemID, pbstrUDAttr
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The account’s member ID.</td>
</tr>
<tr>
<td><code>pbstrUDAttr</code></td>
<td>String. Returns the UserDefined2 attribute.</td>
</tr>
</tbody>
</table>

**GetUserDefined3**

Returns an account’s UserDefined3 attribute, using the account’s member ID.

Syntax

```csharp
<HsvAccounts>.GetUserDefined3 lItemID, pbstrUDAttr
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The account’s member ID.</td>
</tr>
<tr>
<td><code>pbstrUDAttr</code></td>
<td>String. Returns the UserDefined3 attribute.</td>
</tr>
</tbody>
</table>

**GetXBRLTags**

Returns an account’s XBRLTags attribute, using the account’s member ID.

Syntax

```csharp
<HsvAccounts>.GetXBRLTags lItemID, pbstrXBRLTags
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The account’s member ID.</td>
</tr>
<tr>
<td><code>pbstrXBRLTags</code></td>
<td>String. Returns the XBRLTags attribute.</td>
</tr>
</tbody>
</table>

**IsCalculated**

Returns a Boolean that indicates whether an account’s data is calculated by the system or is manually entered. This method returns the value to which an account’s IsCalculated attribute is set.
Syntax

<HasvAccounts>.IsCalculated lItemID, pbIsCalculated

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the account.</td>
</tr>
<tr>
<td>pbIsCalculated</td>
<td>Boolean. Returns TRUE if the account’s data is calculated, FALSE if the data is manually entered.</td>
</tr>
</tbody>
</table>

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 is executed.

Dim cAccounts As HasvAccounts, 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. This method returns the value to which an account’s IsConsolidated attribute is set.

Syntax

<HasvAccounts>.IsConsolidated lItemID, pbIsConsolidated

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long. The member ID of the account.</td>
</tr>
<tr>
<td>pbIsConsolidated</td>
<td>Boolean. Returns TRUE if the account is consolidated to parents, FALSE if it is ignored during consolidation.</td>
</tr>
</tbody>
</table>

IsCustom1AggregationEnabled

Returns the value to which an account’s EnableCustom1Aggr attribute is set.

Syntax

<HasvAccounts>.IsCustom1AggregationEnabled lItemID, pbIsEnabled

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long. The member ID of the account.</td>
</tr>
</tbody>
</table>
### pbIsEnabled

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 is executed.

```vba
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 is set.

#### Syntax

```vba
<HsvAccounts>.IsCustom2AggregationEnabled lItemID, pbIsEnabled
```

#### Argument Description

- **lItemID**
  
  Long. The member ID of the account.

- **pbIsEnabled**
  
  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 is set.

#### Syntax

```vba
<HsvAccounts>.IsCustom3AggregationEnabled lItemID, pbIsEnabled
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long. The member ID of the account.</td>
</tr>
<tr>
<td>pbIsEnabled</td>
<td>Boolean. Returns TRUE if the EnableCustom3Aggr attribute is enabled, FALSE if it is disabled.</td>
</tr>
</tbody>
</table>

Example

See the Example for IsCustom1AggregationEnabled, replacing that method with IsCustom3AggregationEnabled.

**IsCustom4AggregationEnabled**

Returns the value to which an account’s EnableCustom4Aggr attribute is set.

**Syntax**

```<HsvAccounts>.IsCustom4AggregationEnabled lItemID, pbIsEnabled```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long. The member ID of the account.</td>
</tr>
<tr>
<td>pbIsEnabled</td>
<td>Boolean. Returns TRUE if the EnableCustom4Aggr attribute is enabled, FALSE if it is disabled.</td>
</tr>
</tbody>
</table>

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. This method returns the value to which an account’s IsICP attribute is set.

**Syntax**

```<HsvAccounts>.IsICP lItemID, pbIsICP```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the account.</td>
</tr>
<tr>
<td>pbIsICP</td>
<td>Boolean. Returns TRUE if the account is used in intercompany transactions, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**IsICPRestricted**

Indicates whether an Account dimension member is restricted from having Intercompany Partner transactions with itself. This method indicates whether an Account dimension member’s IsICP attribute is set to a value of R.
Syntax

`<HsvAccounts>.IsICPRestricted lItemID, pbIsICPRestricted`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the account.</td>
</tr>
<tr>
<td><code>pbIsICPRestricted</code></td>
<td>Boolean. Returns TRUE if the account’s IsICP attribute is set to R, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**UsesLineItems**

Returns a Boolean that indicates whether an account allows line items.

Syntax

`<HsvAccounts>.UsesLineItems lItemID, pvarbUsesLineItems`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the account.</td>
</tr>
<tr>
<td><code>pvarbUsesLineItems</code></td>
<td>Boolean. Returns TRUE if the account allows line items, FALSE otherwise.</td>
</tr>
</tbody>
</table>

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 71, 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:

```vbnet
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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The ID of the Custom dimension member.</td>
</tr>
<tr>
<td>plSecurityClassID</td>
<td>Long. Returns the ID of the Custom dimension member's security class. If no security class is assigned to the member, SECURITY_CLASS_NONE is returned.</td>
</tr>
</tbody>
</table>

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 is 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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long. The member ID of the Custom dimension member.</td>
</tr>
<tr>
<td>plSubmissionGroup</td>
<td>Long. Returns the Submission Group value.</td>
</tr>
</tbody>
</table>

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

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

**pbIsCalculated**  
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 Custom dimension member is calculated. The second argument takes the ID of the Custom dimension; dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 857.

```vba
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**

```vba
<HsvCustom>.IsSwitchSignEnabledForFlow lMemberID, pbFlag
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lMemberID</strong></td>
<td>Long (ByVal). The member ID of the Custom dimension member.</td>
</tr>
<tr>
<td><strong>pbFlag</strong></td>
<td>Boolean. Indicates whether the SwitchSignForFlow attribute is enabled. Returns TRUE if it is enabled, FALSE otherwise.</td>
</tr>
</tbody>
</table>

---

### IsSwitchTypeEnabledForFlow

Indicates whether a Custom dimension member’s SwitchTypeForFlow attribute is enabled.

**Syntax**

```vba
<HsvCustom>.IsSwitchTypeEnabledForFlow lMemberID, pbFlag
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lMemberID</strong></td>
<td>Long (ByVal). The member ID of the Custom dimension member.</td>
</tr>
<tr>
<td><strong>pbFlag</strong></td>
<td>Boolean. Indicates whether the SwitchTypeForFlow attribute is enabled. Returns TRUE if it is enabled, FALSE otherwise.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>lMemberID</td>
<td>Long (ByVal). The member ID of the Custom dimension member.</td>
</tr>
<tr>
<td>pbFlag</td>
<td>Boolean. Indicates whether the SwitchTypeForFlow attribute is enabled. Returns TRUE if it is enabled, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**HsvEntities Object Methods**

The HsvEntities object’s methods return attributes of Entity dimension members. These methods are summarized in Table 14 on page 72, 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:

```vbnet
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. This method returns the value to which an entity’s AllowAdjs attribute is set.

**Syntax**

```
<HsvEntities>.GetAllowAdjustments lItemID, pbOK
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the entity.</td>
</tr>
<tr>
<td>pbOK</td>
<td>Boolean. Returns TRUE is the entity allows journal entries, FALSE if it does not.</td>
</tr>
</tbody>
</table>

**GetAllowAdjustmentsFromChildren**

Returns a Boolean that indicates whether an entity allows journal postings from its children. This method returns the value to which an entity’s AllowAdjFromChildren attribute is set.

**Syntax**

```
<HsvEntities>.GetAllowAdjustmentsFromChildren lItemID, pbOK
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the entity.</td>
</tr>
<tr>
<td>pbOK</td>
<td>Boolean. Returns TRUE is the entity allows journal postings from its children, FALSE if it does not.</td>
</tr>
</tbody>
</table>
**GetDefaultValueID**

Returns the member ID of the Value dimension member that represents an entity’s default currency. This method returns the member ID of the Value dimension member specified for an entity’s DefCurrency attribute.

**Syntax**

```csharp
<HsvEntities>.GetDefaultValueID lItemID, plValueID
```

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the entity.</td>
</tr>
<tr>
<td><code>plValueID</code></td>
<td>Long. Returns the member ID of the Value dimension member that represents the entity’s default currency.</td>
</tr>
</tbody>
</table>

**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.

```csharp
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**

```csharp
<HsvEntities>.GetHoldingCompany lItemID, plHoldingCompanyEntityID
```

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the entity for which you want to get the HoldingCompany attribute.</td>
</tr>
<tr>
<td><code>plHoldingCompanyEntityID</code></td>
<td>Long. Returns the member ID of the entity specified as the HoldingCompany attribute, or -1 if the entity does not have a holding company.</td>
</tr>
</tbody>
</table>
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

lItemID Long (ByVal). The member ID of the entity.

plSecurityAsPartnerID Long. Returns the value assigned to the SecurityAsPartner attribute.

GetSecurityClassID

Returns the ID number of an entity’s security class. This method returns the ID of the security class 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; see “GetSecurityClassID” on page 482.

Syntax

<HsvEntities>.GetSecurityClassID lItemID, plSecurityClassID
**Argument**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the entity.</td>
</tr>
<tr>
<td>pILSecurityClassID</td>
<td>Long. Returns the ID number of the security class, or SECURITY_CLASS_NONE if a security class has not been assigned to the entity.</td>
</tr>
</tbody>
</table>

**Tip:** Get the label for the returned ID by passing the ID to HsvSecurityAccess.GetSecurityClassLabel. See “GetSecurityClassLabel” on page 483.

**Example**

The following function takes an Entity member’s label and returns the label of its security class.

```vba
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
    Else
        getEntSecClass = ""
    End If
End Function
```

**GetUserDefined1**

Returns the value defined for an entity’s UserDefined1 attribute.

**Syntax**

```vba
<HsvEntities>.GetUserDefined1 lItemID, pbstrUserDefined
```

**Argument**

<table>
<thead>
<tr>
<th></th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the entity.</td>
</tr>
<tr>
<td>pbstrUserDefined</td>
<td>String. Returns the value defined for the UserDefined1 attribute.</td>
</tr>
</tbody>
</table>

**GetUserDefined2**

Returns the value defined for an entity’s UserDefined2 attribute.
**Syntax**

```
<HsvEntities>.GetUserDefined2  lItemID, pbstrUserDefined
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the entity.</td>
</tr>
<tr>
<td>pbstrUserDefined</td>
<td>String. Returns the value defined for the UserDefined2 attribute.</td>
</tr>
</tbody>
</table>

**GetUserDefined3**

Returns the value defined for an entity’s UserDefined3 attribute.

**Syntax**

```
<HsvEntities>.GetUserDefined3  lItemID, pbstrUserDefined
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the entity.</td>
</tr>
<tr>
<td>pbstrUserDefined</td>
<td>String. Returns the value defined for the UserDefined3 attribute.</td>
</tr>
</tbody>
</table>

**IsChild**

Returns a Boolean that indicates whether an entity is a child of another entity.

**Syntax**

```
<HsvEntities>.IsChild  lParentID, lEntityID, pbIsChild
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lParentID</td>
<td>Long (ByVal). The member ID of the possible parent entity.</td>
</tr>
<tr>
<td>lEntityID</td>
<td>Long (ByVal). The member ID of the possible child entity.</td>
</tr>
<tr>
<td>pbIsChild</td>
<td>Boolean. Returns TRUE if the lEntityID entity is a child of the lParentID entity, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**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.

```vba
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.\texttt{IsChild} lParID, lChildId, bIsChild

If bIsChild = False Then
    MsgBox "The entities don’t have a parent-child relationship!"
    Exit Sub
End If

\textbf{IsDescendant}

Returns a Boolean that indicates whether an entity is a descendant of another entity.

\textbf{Syntax}

\texttt{<HsvEntities>.IsDescendant lParentID, lEntityID, pbIsDescendant}

\textbf{Argument} \textbf{Description}

\begin{itemize}
    \item \texttt{lParentID} Long (ByVal). The member ID of the possible higher-level entity.
    \item \texttt{lEntityID} Long (ByVal). The member ID of the possible descendant entity.
    \item \texttt{pbIsDescendant} Boolean. Returns TRUE if the \texttt{lEntityID} argument's entity is a descendant of the \texttt{lParentID} argument's entity, FALSE if it is not a descendant.
\end{itemize}

\textbf{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.

\begin{verbatim}
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.\texttt{IsDescendant} lParID, lDescID, bIsDesc
    IsEntityDescendant = bIsDesc
End Function
\end{verbatim}

\textbf{IsICP}

Returns a Boolean that indicates whether an entity is an intercompany entity. This method returns the value defined for an entity’s IsICP attribute.

\textbf{Syntax}

\texttt{<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 is executed.

```vba
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**

```vba
<HsvEntities>.IsOrgByPeriodFilteringOn pvarbFilteringOn, plScenario, plYearID, plPeriodID
```

**Argument**  
**Description**

- **pvarbFilteringOn**  
  Boolean. Returns TRUE if the entity has organization by period filtering enabled, otherwise FALSE.

- **plScenario**  
  Long. Returns the ID of the subcube’s Scenario dimension member.

- **plYearID**  
  Long. Returns the ID of the subcube’s Year dimension member.

- **plPeriodID**  
  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; see the Financial Management Javadoc.
Note: To get the member ID of an Entity dimension member’s SecurityAsPartner attribute, use `GetSecurityAsPartnerID`.

Syntax

```<HsvEntities>.IsSecurityAsPartnerEnabled pvarbEnabled```

**Argument** | **Description**
--- | ---
`pvarbEnabled` | 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 `ICPs` 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 assigned to an Intercompany Partner, using the Intercompany Partner’s member ID.

Syntax

```<HsvICPs>.GetSecurityClassID lItemID, plSecurityClassID```

**Argument** | **Description**
--- | ---
`lItemID` | Long (ByVal). The Intercompany Partner’s member ID.
`plSecurityClassID` | 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
lItemID    Long. The member ID of the Intercompany Partner member.
plSubmissionGroup 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 in a frequency’s periods.
- The number of periods in a frequency.
- The member ID of a period at an ordinal position in a frequency.
- The member IDs of a frequency’s Period dimension members.
The HsvPeriods object’s methods are summarized in Table 16 on page 73, 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:

```vba
Dim cMetadata as HsvMetadata, cPeriods as HsvPeriods
Set cMetadata = m_cSession.MetaData
Set cPeriods = cMetadata.Periods
```

### GetBaseFrequency

**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 is the base frequency.

**Syntax**

```vba
<HsvPeriods>.GetBaseFrequency plBaseFrequency
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>plBaseFrequency</code> Long. Returns the ID of the application’s base frequency.</td>
</tr>
</tbody>
</table>

**Example**

GetBaseFrequency is used in the Example for GetPeriodsInFrequency.

### GetFrequency

**GetFrequency**

Returns the ordinal position of a period within an application’s base frequency, and also returns the ID of the base frequency.

**Syntax**

```vba
<HsvPeriods>.GetFrequency lPeriod, plFreq, plPeriodNumInFreq
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lPeriod</code> Long (ByVal). The member ID of the period.</td>
</tr>
<tr>
<td><code>plFreq</code> Long. Returns the internal ID number of the application’s base frequency.</td>
</tr>
<tr>
<td><code>plPeriodNumInFreq</code> 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.</td>
</tr>
</tbody>
</table>
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 is executed.

```vba
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

```vba
<HsvPeriods>.GetNumPeriodsInFrequency lFrequency, plNumPeriods
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lFrequency</code></td>
<td>Long (ByVal). The ID of the frequency. To use the application’s base frequency, pass the ID returned by GetBaseFrequency.</td>
</tr>
<tr>
<td><code>plNumPeriods</code></td>
<td>Long. Returns the number of periods in the frequency.</td>
</tr>
</tbody>
</table>

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.

```vba
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 an ordinal position within a frequency.

Syntax

```vba
<HsvPeriods>.GetPeriodFromFrequency lFreq, lPeriodNumInFreq, plPeriod
```
### GetOrdinalPerName

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.

```vba
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

```vba
<HsvPeriods>.GetPeriodsInFrequency lFrequency, pvaralPeriodIDs
```

### Arguments

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lFrequency</code></td>
<td>Long (ByVal). The ID of the frequency. To use the application’s base frequency, pass the ID returned by <code>GetBaseFrequency</code>.</td>
</tr>
<tr>
<td><code>pvaralPeriodIDs</code></td>
<td>Variant array. Returns the member IDs of the periods in the frequency. The array is returned as a Long subtype.</td>
</tr>
</tbody>
</table>

#### 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.

```vba
Function CheckPeriodsInYear() As Long
    Dim cTreeInfo As IHsvTreeInfo, cPeriods As HsvPeriods
    Dim lFreqID As Long, lPerID As Long, sPerName As String
    Dim lPeriodsCount As Long
    Set cTreeInfo = m_cMetadata.Periods
    Set cPeriods = m_cMetadata.Periods
    cPeriods.GetBaseFrequency lFreqID
    Set pvarPeriodIDs = cPeriods.GetPeriodsInFrequency lFreqID
    lPeriodsCount = UBound(pvarPeriodIDs) + 1
    CheckPeriodsInYear = lPeriodsCount
End Function
```
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 73, 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

lItemID Long (ByVal). The member ID of the scenario.

plStartYear 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

pvaralScenarioIds Variant array. Returns an array of Scenario IDs. The array is returned as a Long subtype.

pvaralStartYears 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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>plFrequency</td>
<td>Long. Returns the default frequency’s ID number.</td>
</tr>
</tbody>
</table>

**GetDefaultView**

Returns the number that identifies the default view of a Scenario dimension member.

Syntax

```<HsvScenarios>.GetDefaultView lItemID, plView```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>plView</td>
<td>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 &quot;View Dimension Member Constants&quot; on page 856.</td>
</tr>
</tbody>
</table>

**Tip:** To get the label for the returned ID, pass the ID to `IHsvTreeInfo.GetLabel` (for the View dimension). See "GetLabel" on page 264.

**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`.

```vbnet
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```

304  HsvMetadata Type Library
GetMaximumReviewLevel

Returns the value assigned to the MaximumReviewLevel attribute of a Scenario dimension member.

Syntax

```<HsvScenarios>.GetMaximumReviewLevel lItemID, psMaximumReviewLevel```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the scenario.</td>
</tr>
<tr>
<td><code>psMaximumReviewLevel</code></td>
<td>Integer. Returns the value assigned to the MaximumReviewLevel attribute.</td>
</tr>
</tbody>
</table>

Example

The following function takes a Scenario member’s name and returns the value assigned to the member’s MaximumReviewLevel attribute.

```vbnet
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 assigned as the ZeroViewForAdj attribute of a Scenario dimension member.

Syntax

```<HsvScenarios>.GetMissingDataZeroViewForAdjValues lItemID, plView```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lItemID</code></td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td><code>plView</code></td>
<td>Long. Returns the member ID of the View dimension member assigned as the ZeroViewForAdj attribute.</td>
</tr>
</tbody>
</table>

Example

The following function returns the label of the View dimension member assigned as a Scenario member’s ZeroViewForAdj attribute.

```vbnet
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 assigned as the ZeroViewForNonadj attribute of a Scenario dimension member.

Syntax

<HsvScenarios>.GetMissingDataZeroViewForNonAdjValues 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 ZeroViewForNonadj attribute.

Example

The following function returns the label of the View dimension member assigned as a 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 assigned to a scenario, using the scenario’s member ID.

Syntax

<HsvScenarios>.GetSecurityClassID lItemID, plSecurityClassID
**Argument**  
**Description**

`lItemID`  
Long (ByVal). The scenario’s member ID.

`plSecurityClassID`  
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.

```vba
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**

```vba
<HsvScenarios>.GetUserDefined1 lItemID, pbstrUserDefined
```

**Argument**  
**Description**

`lItemID`  
Long (ByVal). The member ID of the Scenario dimension member.

`pbstrUserDefined`  
String. Returns the UserDefined1 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 is 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**

```vbnet
<HsvScenarios>.GetUserDefined2 lItemID, pbstrUserDefined
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lItemID</em></td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td><em>pbstrUserDefined</em></td>
<td>String. Returns the UserDefined2 attribute’s text.</td>
</tr>
</tbody>
</table>

**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**

```vbnet
<HsvScenarios>.GetUserDefined3 lItemID, pbstrUserDefined
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lItemID</em></td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td><em>pbstrUserDefined</em></td>
<td>String. Returns the UserDefined3 attribute’s text.</td>
</tr>
</tbody>
</table>

**Example**

See the Example for GetUserDefined1, and replace GetUserDefined1 with
GetUserDefined3.
## IsConsolidateYTD

Indicates whether a 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

```xml
<HsvScenarios>.IsConsolidateYTD lItemID, pbConsolidateYTD
```

### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lItemID</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>pbConsolidateYTD</td>
<td>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.</td>
</tr>
</tbody>
</table>

### Example

The following function indicates whether a Scenario dimension member supports year-to-date consolidation.

```vbnet
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 Scenario and Year.

### Syntax

```xml
<HsvScenarios>.IsPhasedSubmissionEnabled lScenarioId, lYear, pvbEnabled
```

### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenarioId</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>pvbEnabled</td>
<td>Boolean. Returns True if Phased Submissions is enabled, False otherwise.</td>
</tr>
</tbody>
</table>

---

**HsvScenarios Object Methods** 309
**SupportsAccountAllocations**

*For internal use.*

**SupportsEmailAlerting**

Indicates whether email alerting is enabled for a Scenario dimension member.

**Syntax**

```<HsvScenarios>.SupportsEmailAlerting lItemID, pvbSupportsAlerting```

**Argument**

- **lItemID**
  Long (ByVal). The member ID of the Scenario dimension member.

- **pvbSupportsAlerting**
  Boolean. Indicates whether the member supports email alerting. Returns TRUE if the alerting is enabled, otherwise FALSE.

**SupportsProcessFlow**

Indicates whether Process Management is enabled for a Scenario dimension member.

**Syntax**

```<HsvScenarios>.SupportsProcessFlow lItemID, pvbSupports```

**Argument**

- **lItemID**
  Long (ByVal). The member ID of the Scenario dimension member.

- **pvbSupports**
  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.

```vbnet
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```

---

310 HsvMetadata Type Library
**SupportsTargetSetting**

For internal use.

**SupportsTimePeriodAllocations**

For internal use.

**UsesLineItems**

Indicates whether a Scenario dimension member supports line items. This method indicates the value assigned to a Scenario dimension member’s UsesLineItems attribute.

**Syntax**

```
<HsvScenarios>.UsesLineItems lItemID, pvarbUsesLineItems
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lItemID</strong> Long (ByVal). The member ID of the scenario.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pvarbUsesLineItems</strong> Boolean. Returns TRUE if the scenario supports line items, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**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 74, 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 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

```vbnet
<HsvValues>.GetCurrencyIDFromValueID lValue, plCurrencyID
```

**Argument** | **Description**
---|---
`lValue` | Long (ByVal). The member ID of the Value dimension member.

`plCurrencyID` | 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 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

```vbnet
<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:

```vbnet
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

```vbnet
<HsvYears>.GetYearRange plStartYear, plEndYear
```
### 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 77, 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. See “HsvValues Object Methods” on page 311.

Assign HsvCurrencies object references with the Currencies property of the HsvMetadata object as shown in the following example:

```vba
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**

```vbscript
<HsvCurrencies>.EnumCurrencies pvaralCurrencyIDs, pvarabstrCurrencyLabels
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvaralCurrencyIDs</code></td>
<td>Variant. Returns an array of currency IDs. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pvarabstrCurrencyLabels</code></td>
<td>Variant. Returns an array of currency labels. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

**Example**

The following example prints the labels of an application’s currencies to Visual Basic’s Immediate window.

```vbscript
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**

```vbscript
<HsvCurrencies>.EnumCurrencies2 vbICTOnly, pvaralIDs, pvarabstrLabels
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vbICTOnly</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>pvaralIDs</code></td>
<td>Variant. Returns an array of currency IDs. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pvarabstrLabels</code></td>
<td>Variant. Returns an array of currency labels. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>
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
For i = LBound(vaLabels) To UBound(vaLabels)
    Debug.Print vaLabels(i)
Next i
End Sub

GetCurrencyDescription

Returns a currency’s description in a specified language. The language must be valid.

**Tip:** HsvMetadata.**EnumLanguages** returns labels and IDs of the valid languages for the application.

**Syntax**

```vba
<HsvCurrencies>.GetCurrencyDescription(lCurrencyID, lLanguageID)
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lCurrencyID</strong> Long (ByVal). The ID of the currency. You can get a currency ID with <strong>GetCurrencyID</strong> and <strong>EnumCurrencies</strong>.</td>
</tr>
<tr>
<td><strong>lLanguageID</strong> Long (ByVal). The ID of the language. You can get language IDs with HsvMetadata.<strong>EnumLanguages</strong>.</td>
</tr>
</tbody>
</table>

**Return Value**

String. Returns the currency’s description.

Example

GetCurrencyDescription is used in the example for **GetCurrencyID**.

GetCurrencyID

Returns the currency ID of a currency.

**Syntax**

```vba
<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 specified language. The currency ID obtained by GetCurrencyID is passed to GetCurrencyDescription with the language ID obtained by HsvMetadata.EnumLanguages.

```vba
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, using 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 specified currency.

**Tip:** The operator is specified with the currency’s TranslationOperator attribute.
Syntax

\(<\text{HsvCurrencies}>\).\text{GetCurrencyTranslationOperator}(\text{lCurrencyID})

Argument Description

\(\text{lCurrencyID} \) 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 \text{GetCurrencyID}.

\begin{verbatim}
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)
ggetCurrentOp = cCurrencies.GetCurrencyTranslationOperator(lCurrID)
End Function
\end{verbatim}

\textbf{GetScale}

Returns the scale of a specified currency.

Syntax

\(<\text{HsvCurrencies}>\).\text{GetScale}(\text{lCurrencyID})

Argument Description

\(\text{lCurrencyID} \) 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 \text{GetCurrencyID}.

\begin{verbatim}
Function getCurrencyScale(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)
ggetCurrentScale = cCurrencies.GetScale(lCurrID)
End Function
\end{verbatim}

\textit{HsvCurrencies Object Methods} 317
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 867.
- 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 867.
- 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 866.

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 is 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>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSV_DATA_UPDATE_ACCUMULATE</td>
<td>Accumulate mode: if a cell contains data, the data in the cell to be copied or loaded is added to the existing data.</td>
</tr>
<tr>
<td>HSV_DATA_UPDATE_MERGE</td>
<td>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.</td>
</tr>
<tr>
<td>HSV_DATA_UPDATE_REPLACE</td>
<td>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.</td>
</tr>
<tr>
<td>HSV_DATA_UPDATE_REPLACEWITHSECURITY</td>
<td>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.</td>
</tr>
</tbody>
</table>

## 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 78, 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 78.

**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 197.
AddDataToMDDataBuffer

*Deprecated* - use AddDataToMDDataBufferExtDim.

AddDataToMDDataBufferExtDim

Adds an application’s cell to an HsvMDDataBuffer or HsvMDDataBufferLite object. Any data, description, or line items for the cell will be added. Supersedes AddDataToMDDataBuffer.

**Syntax**

```c
<HsvData>.AddDataToMDDataBufferExtDim pIUnkHfmPovCOM, varbIncludeDerivedData, pIUnkDataBuffer
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell POV.</td>
</tr>
<tr>
<td><code>varbIncludeDerivedData</code></td>
<td>Boolean</td>
<td>Determines whether derived data is included.</td>
</tr>
<tr>
<td><code>pIUnkDataBuffer</code></td>
<td>HsvMDDataBuffer or HsvMDDataBufferLite object</td>
<td>The object to which the cell is being added.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See the HfmPovCOM documentation on how to set the POV.

AttachDocumentToCell

*Deprecated* - use AttachDocumentToCellExtDim.

AttachDocumentToCellExtDim

Attaches a previously loaded document to a cell. Supersedes AttachDocumentToCell.

**Note:** An additional param, `ICellTextItemId` is required to specify the cell text label.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object that represents the cell’s POV</td>
</tr>
<tr>
<td><code>bstrDocPath</code></td>
<td>String</td>
<td>The path in which the document has been loaded. Folders in the path are delimited by backslashes (<code>\</code>).</td>
</tr>
<tr>
<td><code>ICellTextItemId</code></td>
<td>Long</td>
<td>Cell text label ID to attach the document to</td>
</tr>
</tbody>
</table>
### bstrDocFile

**Type:** String

**Description:** The name of the file to attach.

---

#### Return Value

None.

#### Example

See HfmPovCOM documentation for how to set the POV.

---

## 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

```vbnet
<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.

```vbnet
Dim cHsvData as HsvData
Set cHsvData = m_cHsvSession.Data
cHsvData.ClearAllData
```

**Tip:** HsxClient.OpenApplication returns an HsvSession object reference. See “OpenApplication” on page 176.

---

## 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 53.)

#### Syntax

```vbnet
<HsvData>.ClearAllDescriptionsInSubCube lScenario, lYear, lPeriod, lEntity, lParent, lValue
```

#### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the cell's Scenario dimension member.</td>
</tr>
</tbody>
</table>
**Argument**  
**Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the cell's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the cell's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the cell's Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent of the lEntity argument's entity.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the cell's Value dimension member.</td>
</tr>
</tbody>
</table>

**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.

```vba
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, "2012")  
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 specified date and time.

**Syntax**

```
<HsvData>.ClearDataAuditItems dTimeToClearBefore
```

**Argument**  
**Description**

dTimeToClearBefore  
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.

```vba
Dim cData As HsvData  
'm_cHsvSession is an HsvSession object reference  
Set cData = m_cHsvSession.Data  
cData.ClearDataAuditItems CDbl(Now)
```
ClearDataAuditItemsEx

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

Syntax

```c
<HsvData>.ClearDataAuditItemsEx dStartTime, dEndTime,pIUnkHfmPovCOM_MbrFilter,
vbAllServers, bstrServer, vbAllUsers, bstrUserName
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dStartTime</td>
<td>Double. 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.</td>
</tr>
<tr>
<td>dEndTime</td>
<td>Double. 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.</td>
</tr>
<tr>
<td>pIUnkHfmPovCOM_MbrFilter</td>
<td>HfmPovCOM object representing the cell POV.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>Boolean. A flag that determines whether to delete data audit information for all application servers. Pass TRUE for all servers, FALSE to filter by a specific application name.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String. The name of the application server by which to filter. This argument is used only if the vbAllServers argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>Boolean. A flag that determines whether to delete data audit information for all users. Pass TRUE for all users, FALSE to filter by a specific username.</td>
</tr>
<tr>
<td>bstrUserName</td>
<td>String. The user name of the user by which to filter. This argument is used only if the vbAllUsers argument is set to FALSE.</td>
</tr>
</tbody>
</table>

ClearInputData

Deletes data from a subcube’s cells.

**Note:** To delete an application’s data, the user must be assigned an Administrator role for the application.

**Note:** For a description of subcubes, see “About Subcubes” on page 53.

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

```plaintext
<HsvData>.ClearInputData lScenario, lYear, lEntity, lParent, vbClearInputValueID, vbClearNoneValueID, vbClearNodeLevelValueID, varlaPeriodSubset, varlaAccountSubsetIncludeList, vbUseAccountSubsetIncludeList, varlaAccountSubsetExcludeList, vbUseAccountSubsetExcludeList, pvbDataExistedPriorToClear
```

#### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code></td>
<td>Long (ByVal). The member ID of the subcube’s Scenario dimension member.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the subcube’s Year dimension member.</td>
</tr>
<tr>
<td><code>lEntity</code></td>
<td>Long (ByVal). The member ID of the subcube’s Entity dimension member.</td>
</tr>
<tr>
<td><code>lParent</code></td>
<td>Long (ByVal). The member ID of the parent of the <code>lEntity</code> argument’s entity.</td>
</tr>
<tr>
<td><code>vbClearInputValueID</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>vbClearNoneValueID</code></td>
<td>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.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>[None] Value dimension members are used with system accounts and currency rate accounts.</td>
</tr>
<tr>
<td><code>vbClearNodeLevelValueID</code></td>
<td>Boolean (ByVal). This argument is obsolete; you must pass a Boolean, but the value passed has no effect.</td>
</tr>
<tr>
<td><code>varlaPeriodSubset</code></td>
<td>Long array (ByVal). The member IDs of the cells’ Period dimension members.</td>
</tr>
<tr>
<td><code>varlaAccountSubsetIncludeList</code></td>
<td>Long array (ByVal). If you set the <code>vbUseAccountSubsetIncludeList</code> 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.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>This argument is ignored if you set <code>vbUseAccountSubsetIncludeList</code> to FALSE.</td>
</tr>
<tr>
<td><code>vbUseAccountSubsetIncludeList</code></td>
<td>Boolean (ByVal). Specifies whether the <code>varlaAccountSubsetIncludeList</code> argument is used or ignored. Pass TRUE to use it, FALSE to ignore it.</td>
</tr>
<tr>
<td><code>varlaAccountSubsetExcludeList</code></td>
<td>Long array (ByVal). If you set the <code>vbUseAccountSubsetExcludeList</code> 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.</td>
</tr>
<tr>
<td><strong>Note:</strong></td>
<td>This argument is ignored if you set <code>vbUseAccountSubsetExcludeList</code> to FALSE.</td>
</tr>
<tr>
<td><code>vbUseAccountSubsetExcludeList</code></td>
<td>Boolean (ByVal). Specifies whether the <code>varlaAccountSubsetExcludeList</code> argument is used or ignored. Pass TRUE to use it, FALSE to ignore it.</td>
</tr>
<tr>
<td><code>pvbDataExistedPriorToClear</code></td>
<td>Boolean. Indicates whether any of the specified cells contained data before <code>ClearInputData</code> was called. Returns TRUE if one or more of the cells contained data, FALSE otherwise.</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Types</th>
</tr>
</thead>
<tbody>
<tr>
<td>[None]</td>
</tr>
<tr>
<td>[Contribution Total]</td>
</tr>
<tr>
<td>[Contribution Adj]</td>
</tr>
<tr>
<td>[Contribution]</td>
</tr>
<tr>
<td>[Elimination]</td>
</tr>
<tr>
<td>[Proportion]</td>
</tr>
<tr>
<td>[Parent Total]</td>
</tr>
<tr>
<td>[Parent Adj]</td>
</tr>
<tr>
<td>[Parent]</td>
</tr>
<tr>
<td>&lt;Parent Curr Total&gt;</td>
</tr>
<tr>
<td>&lt;Parent Curr Adj&gt;</td>
</tr>
<tr>
<td>&lt;Parent Currency&gt;</td>
</tr>
<tr>
<td>&lt;Entity Curr Total&gt;</td>
</tr>
<tr>
<td>&lt;Entity Curr Adj&gt;</td>
</tr>
<tr>
<td>&lt;Entity Currency&gt;</td>
</tr>
<tr>
<td>EURO Total</td>
</tr>
<tr>
<td>EURO Adj</td>
</tr>
<tr>
<td>EURO</td>
</tr>
<tr>
<td>USD Total</td>
</tr>
<tr>
<td>USD Adj</td>
</tr>
<tr>
<td>USD</td>
</tr>
</tbody>
</table>

ClearInvalidData

Scans for or deletes invalid records.

Syntax

<HsvData>.ClearInvalidData vbScanOnly, lOptions, bstrLogFileName

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vbScanOnly</td>
<td>Boolean (ByVal). A flag indicating whether to scan for or delete invalid records. Pass TRUE to scan, FALSE to delete.</td>
</tr>
<tr>
<td>lOptions</td>
<td>Long (ByVal). This argument's value is ignored in the current release. You must pass a valid Long.</td>
</tr>
<tr>
<td>bstrLogFileName</td>
<td>String (ByVal). The name and path of the log file for the operation.</td>
</tr>
</tbody>
</table>
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, vbCopyEntityCurrencyValueID, vbCopyNoneValueID, varlaAccountSubset, dFactor, vbCopyCellText, vbCopyDerived, vbIgnoreWarnings, lEnumUpdateMode, bstrLogFileName, vbEnableDetailedLogging```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lView</td>
<td>Long (ByVal). The member ID of the source and target cells’ View dimension member.</td>
</tr>
<tr>
<td>lSourceScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member for the source cells.</td>
</tr>
<tr>
<td>lDestScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member for the target cells.</td>
</tr>
<tr>
<td>lSourceYear</td>
<td>Long (ByVal). The member ID of the Year dimension member for the source cells.</td>
</tr>
<tr>
<td>lDestYear</td>
<td>Long (ByVal). The member ID of the Year dimension member for the target cells.</td>
</tr>
<tr>
<td>varlaSourcePeriodSubset</td>
<td>Long array (ByVal). The member IDs of the Period dimension members for the source cells.</td>
</tr>
<tr>
<td>varlaDestPeriodSubset</td>
<td>Long array (ByVal). The member IDs of the Period dimension members for the target cells.</td>
</tr>
<tr>
<td>varlaEntities</td>
<td>Long array (ByVal). The member IDs of the source and target cells’ Entity dimension members.</td>
</tr>
<tr>
<td>varlaParents</td>
<td>Long array (ByVal). The member IDs of the parents of the varlaEntities argument's entities.</td>
</tr>
<tr>
<td>vbCopyEntityCurrencyValueID</td>
<td>Boolean (ByVal). Specifies whether cells for the &lt;Entity Currency&gt; Value dimension member are to be included in the source and target cells. Pass TRUE to include these cells, otherwise FALSE.</td>
</tr>
<tr>
<td>vbCopyNoneValueID</td>
<td>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.</td>
</tr>
<tr>
<td>varlaAccountSubset</td>
<td>Long array (ByVal). The member IDs of the source and target cells’ Account dimension members.</td>
</tr>
<tr>
<td>dFactor</td>
<td>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.</td>
</tr>
<tr>
<td>vbCopyCellText</td>
<td>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.</td>
</tr>
</tbody>
</table>
vbCopyDerived
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.

vbIgnoreWarnings
Boolean (ByVal). Specifies whether data is copied if the system generates a warning-level error. Pass TRUE to copy data if warnings occur, otherwise FALSE.

Note: Warnings are included in the log file regardless of whether you specify TRUE or FALSE.

lEnumUpdateMode
Long (ByVal). Determines the update mode. See “Update Mode Constants” on page 320.

bstrLogFileName
String (ByVal). The name and path of the log file in which the system includes information regarding the copy data operation.

vbEnableDetailedLogging
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. 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
Deprecated - use DeleteLineItemDetailsExtDim.

DeleteLineItemDetailsExtDim
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. Supersedes DeleteLineItemDetails

Syntax
<HsvData>.DeleteLineItemDetailsExtDim plUnkHfmPovCOM, varabstrDetail

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell’s pov.</td>
</tr>
<tr>
<td>varabstrDetail</td>
<td>String array</td>
<td>An array of the line item descriptions to be deleted.</td>
</tr>
</tbody>
</table>
Return Value
None.

Example
See HfmPovCOM documentation on how to set the POV.

**DetachDocumentFromCell**

_Deprecated_ - use **DetachDocumentFromCellExtDim**.

**DetachDocumentFromCellExtDim**

Detaches a document from a cell.

Syntax

```csharp
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object that specifies the cell’s pov</td>
</tr>
<tr>
<td>bstrDocPath</td>
<td>String</td>
<td>The path in which the document has been loaded. Folders in the path are delimited by backslashes ( \ ).</td>
</tr>
<tr>
<td>bstrDocFile</td>
<td>String</td>
<td>The name of the file to detach</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
See HfmPovCOM documentation on how to set the POV.

**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

```csharp
<HsvData>.DoCellDescriptionsExist lScenario, lYear, lEntity, lParent, pvbExist
```
**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the cells’ Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the cells’ Year dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the cells’ Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent of the lEntity argument’s entity.</td>
</tr>
<tr>
<td>pvbExist</td>
<td>Boolean. Returns TRUE if one or more of the cells contain cell text descriptions, otherwise FALSE.</td>
</tr>
</tbody>
</table>

**DoesCellDescriptionExist**

*Deprecated* - use **DoesCellDescriptionExistExtDim**.

**DoesCellDescriptionExistExtDim**

Indicates whether a cell contains cell text. Supersedes DoesCellDescriptionExist.

**Syntax**

```csharp
<HsvData>.DoesCellDescriptionExistExtDim pIUnkHfmPovCOM, pvarbExists
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCom object representing the cell’s POV</td>
</tr>
<tr>
<td>pvarbExists</td>
<td>Boolean</td>
<td>Returns a value indicating whether the cell has cell text.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See the HfmPovCOM documentation on how to specify a POV.

**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**

```csharp
<HsvData>.DoesDataExist lScenario, lYear, lEntity, lParent, pvbExist
```
Argument | Description
--- | ---
Scenario | Long (ByVal). The member ID of the cell's Scenario dimension member.
Year | Long (ByVal). The member ID of the cell's Year dimension member.
Entity | Long (ByVal). The member ID of the cell's Entity dimension member.
Parent | Long (ByVal). The member ID of the cell's Period dimension member.
pvbExist | 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 is executed. The example assumes that the variables for the member ID arguments were previously set in another procedure.

m_cHsvData.

    DoesDataExist lScen, lYr, lEnt, lPar, bExists
    If bExists = False Then
        ...
    End If

**DoesSparseDataExist**

*Deprecated* - use DoesSparseDataExistExtDim.

**DoesSparseDataExistExtDim**

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. Supersedes DoesSparseDataExist

**Syntax**

< HSVData >.DoesSparseDataExist piUnkHfmPovCOM, pvbDataExists

**Argument** | **Type** | **Description**
--- | --- | ---
piUnkHfmPovCOM | HfmPovCOM | HfmPovCOM object representing the pov
p vbDataExists | Boolean | Returns TRUE if the cell contains sparse data, otherwise FALSE

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.
EnumDataAuditItems

Deprecated - use EnumDataAuditItemsExtDim.

EnumDataAuditItemsExtDim

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.

Syntax

<HasData>.EnumDataAuditItemsExtDim dStartTime, dEndTime, vbAllServers, bstrServer, vbAllUsers, bstrUserName, lStartRecord, lEndRecord, ppIUnkHfmSliceCOM, pvarabstrServers, pvarabstrUserNames, pvaradTimeModified, pvaralActivityCode, pvaradValues, pvaralNoData, plTotalNumRecords

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dStartTime</td>
<td>Double</td>
<td>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.</td>
</tr>
<tr>
<td>dEndTime</td>
<td>Double</td>
<td>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.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>Boolean</td>
<td>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.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String</td>
<td>The name of the application server by which to filter. This argument is used only if the vbAllServers argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>Boolean</td>
<td>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.</td>
</tr>
<tr>
<td>bstrUserName</td>
<td>String</td>
<td>The user name of the user by which to filter. This argument is used only if the vbAllUsers argument is set to FALSE.</td>
</tr>
<tr>
<td>lStartRecord</td>
<td>Long</td>
<td>The index of the first record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td>lEndRecord</td>
<td>Long</td>
<td>The index of the last record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td>ppIUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>Returns the POVs for the data changes' cells.</td>
</tr>
<tr>
<td>pvarabstrServers</td>
<td>String array</td>
<td>Returns the usernames of the users who made the data changes.</td>
</tr>
<tr>
<td>pvarabstrUserNames</td>
<td>String array</td>
<td>Returns the usernames of the users who made the data changes.</td>
</tr>
<tr>
<td>pvaradTimeModified</td>
<td>Double Array</td>
<td>Returns the timestamps of the data changes. These are returned as Double values that can be cast to the Date format.</td>
</tr>
</tbody>
</table>
### EnumDataAuditItems2ExtDim

For internal use.

### EnumEntitiesWithDataForScenarioYear

Returns a variant array of Entity ID's that have data for a specified scenario and year.

**Note:** Does not account for Calculated data.

**Syntax**

```csharp
<HSVData>.EnumEntitiesWithDataForScenarioYear lScenario, lYear, pvaralEntityIDs
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the cell's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the cell's Year dimension member.</td>
</tr>
<tr>
<td>pvaralEntityIDs</td>
<td>Variant. The member IDs for the entities that have data for the specified scenario and year.</td>
</tr>
</tbody>
</table>

### 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvar2davOptions</td>
<td>Variant array. Returns a two-dimensional safe array that represents the data extract options.</td>
</tr>
</tbody>
</table>

The first dimension identifies the options, and is indexed from 1 to 20. The indexes and corresponding extract options are listed in Table 63 on page 336.

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 second item in the first dimension, array item (2, 2) stores the system’s default delimiter.

Note: Options without minimum and maximum values 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.

```vba
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

\(<\text{HsvData}>\).\text{EnumLoadOptions} \ pvar2davOptions

**Argument** | **Description**
---|---

pvar2davOptions | 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 64 on page 377.

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 \text{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 return 0 for items 3 and 4 of the second dimension.

Example

\text{EnumLoadOptions} is used in the example for \text{Load}.

### Extract

Extracts data into a text file. The file is 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. See “Extracting Data” on page 798.

Syntax

\(<\text{HsvData}>\).\text{Extract} \ bstrServerFileName, bstrLogFileName, varavSettings

**Argument** | **Description**
---|---

bstrServerFileName | String (ByVal). The name and path of the data extract file. The path must exist on the application server.

bstrLogFileName | String (ByVal). The name and path of the log file. The path must exist on the application server.
**Argument** | **Description**
---|---
`varSettings` | 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 63.

**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 796.

The listed indexes apply to the array passed to `Extract` and to the first dimension of the array returned by `EnumExtractOptions`.

### Table 63 Data Extract Options

<table>
<thead>
<tr>
<th>Index</th>
<th>Extract Option Information</th>
</tr>
</thead>
</table>
| 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:  
- `HSV_DATA_VIEW_PERIODIC`. Data for the Periodic view is extracted.  
- `HSV_DATA_VIEW_YTD`. Data for the year-to-date view is extracted.  
- `HSV_DATA_VIEW_SCENARIO`. Data for the scenario’s default View dimension member is extracted.  
*Tip:* These constants correspond to the Data View options in Extract Data. |
| 2     | **Option:** Delimiter  
*Usage:* Specifies an extract file’s delimiter.  
*Pass to Extract:* A valid delimiter character (String subtype). |
| 3     | **Option:** Log file name.  
*For internal use.* |
| 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. |
| 5     | **Option:** Append to Extract File  
*For internal use.* |
| 6     | **Option:** Extract Calculated  
*Usage:* Specifies whether to extract calculated data.  
*Pass to Extract:* TRUE to extract calculated data, FALSE otherwise. |
<table>
<thead>
<tr>
<th>Index</th>
<th>Extract Option Information</th>
</tr>
</thead>
</table>
| 7     | **Option:** Scenario  
      | **Usage:** Specifies the Scenario dimension member for which data is being extracted.  
      | **Pass to Extract:** A Long containing the Scenario member's ID. |
| 8     | **Option:** Year  
      | **Usage:** Specifies the Year dimension member for which data is being extracted.  
      | **Pass to Extract:** A Long containing the Year member's ID. |
| 9     | **Option:** Period Subset  
      | **Usage:** Specifies the Period dimension members for which data is being extracted.  
      | **Pass to Extract:** A Long array containing Period member IDs. |
| 10    | **Option:** Entity Subset  
      | **Usage:** Specifies the child Entity dimension members for which data is being extracted.  
      | **Pass to Extract:** A Long array containing Entity member IDs. |
| 11    | **Option:** Parent Subset  
      | **Usage:** Specifies the parent Entity dimension members for which data is being extracted.  
      | **Pass to Extract:** A Long array containing Entity dimension IDs. |
| 12    | **Option:** Account Subset  
      | **Usage:** Specifies the Account dimension members for which data is being extracted.  
      | **Pass to Extract:** A Long array containing Account member IDs. |
| 13    | **Option:** ICP Subset  
      | *For internal use.* |
| 14    | **Option:** Value Subset  
      | *For internal use.* |
| 15    | **Option:** Custom 1 Subset  
      | *For internal use.* |
| 16    | **Option:** Custom 2 Subset  
      | *For internal use.* |
| 17    | **Option:** Custom 3 Subset  
      | *For internal use.* |
| 18    | **Option:** Custom 4 Subset  
      | *For internal use.* |
| 19    | **Option:** Extract All Data  
      | *For internal use.* |
Index | Extract Option Information
---|---
20 | **Option:** Unicode
*For internal use.*

**Example**

The following example extracts data for the dimension members listed below:

- Scenario = Actual
- Year = 2012
- 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`.

```vba
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, "2012")
'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

 Deprecated - use ExtractDataAuditItemsExtDim.

ExtractDataAuditItemsExtDim

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. Supersedes ExtractDataAuditItems.

Note: The POV specification for ExtractDataAuditItemsExtDim differs from ExtractDataAuditItems, in that the POV is specified as member IDs, not member names.

Syntax

<HsvData. >ExtractDataAuditItemsExtDim bstrServerFileName, bstrLogFileName, bstrDelimitChar, dStartTime, dEndTime, pIUnkHfmPovCOM_MemberFilter, vbAllServers, bstrServer, vbAllUsers, bstrUserName

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrServerFileName</td>
<td>String</td>
<td>The name and path of the file into which to extract audit information</td>
</tr>
<tr>
<td>bstrLogFileName</td>
<td>String</td>
<td>The name and path of the log file for the extraction</td>
</tr>
<tr>
<td>bstrDelimitChar</td>
<td>String</td>
<td>The delimiter character for the extract file</td>
</tr>
<tr>
<td>dStartTime</td>
<td>Double</td>
<td>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.</td>
</tr>
<tr>
<td>dEndTime</td>
<td>Double</td>
<td>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.</td>
</tr>
<tr>
<td>pIUnkHfmPovCOM_MemberFilter</td>
<td>HfmPovCOM</td>
<td>Contains the dimension member IDs by which to filter cells. Note: Specify MEMBERNOTUSED for dimensions you don’t wish to filter on.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>Boolean</td>
<td>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</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String</td>
<td>The name of the application server by which to filter. This argument is used only if the vbAllServers argument is set to FALSE</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>Boolean</td>
<td>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</td>
</tr>
<tr>
<td>bstrUserName</td>
<td>String</td>
<td>The username by which to filter. This argument is used only if the vbAllUsers argument is set to FALSE.</td>
</tr>
</tbody>
</table>

Return Value

None.
Example

See HfmPovCOM documentation on how to set the POV.

**ExtractDrillableRegions**

Retrieves all ERPI URL definitions.

Syntax

```csharp
<HsvData>.ExtractDrillableRegions bstrRegionsFilename, bstrLogFileName, pvbErrors, pvbWarnings
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrRegionsFilename</td>
<td>String (ByVal). The file path to the definition file on the Financial Management application server.</td>
</tr>
<tr>
<td>bstrLogFileName</td>
<td>String (ByVal). The name and path of the log file. The path must exist on the application server.</td>
</tr>
<tr>
<td>pvbErrors</td>
<td>Boolean. Returns TRUE if errors occurred during the load task. See the log file for details.</td>
</tr>
<tr>
<td>pvbWarnings</td>
<td>Boolean. Returns TRUE if warnings occurred during the load task.</td>
</tr>
</tbody>
</table>

Examples

This function is intended to be used by HFMAwbAgent. The following example extracts 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 are set to VARIANT_TRUE.

```csharp
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

```csharp
<HsvData>.ExtractDrillableRegionsByURLNames varabstrURLNames, bstrRegionsFilename, bstrLogFileName, pvbErrors, pvbWarnings
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varabstrURLNames</td>
<td>Variant (ByVal). Array of URL names.</td>
</tr>
<tr>
<td>bstrRegionsFilename</td>
<td>String (ByVal). The file path to the definition file on Financial Management application server.</td>
</tr>
<tr>
<td>bstrLogFileName</td>
<td>String (ByVal). The name and path of the log file. The path must exist on the application server.</td>
</tr>
<tr>
<td>pvbErrors</td>
<td>Boolean. Returns TRUE if errors occurred during the load task. See the log file for details.</td>
</tr>
</tbody>
</table>
Argument | Description
--- | ---
`pvbWarnings` | 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 `pvbWarnings` are set to `VARIANT_TRUE`.

```c
ExtractDrillableRegionsByURLNames(varabstrURLNames, _T("C:\temp\~RD4a6e.tmp"), _T("C:\temp\~RD9bc3.tmp"), &vbError, &pvbWarnings);
```

The input parameter `varabstrURLNames` should be an array of BSTR, containing “ERPI_USSales,” “ERPI_EUSales,” and “ERPI_Salary.”

**FilterMembersThatHaveData**

*Deprecated* - use `FilterMembersThatHaveDataExtDim`.

**FilterMembersThatHaveDataExtDim**

Returns a list of members for a specified dimension for which data exists for the specified POVs. Supersedes `FilterMembersThatHaveData`.

**Syntax**

```c
<HsvData>.FilterMembersThatHaveDataExtDim pIUnkHfmSliceCOM, lDimID, vbConsiderBaseMembersOnly, varalMembersToFilter, pvaralMembers
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmSliceCOM</code></td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object representing the cells to be analyzed</td>
</tr>
<tr>
<td><code>lDimID</code></td>
<td>Long</td>
<td>The dimension for which you want a list of members. You can use the following dimensions: Account, ICP and Custom dimensions.</td>
</tr>
<tr>
<td><code>vbConsiderBaseMembersOnly</code></td>
<td>Boolean</td>
<td>Determines if base members only are tested</td>
</tr>
<tr>
<td><code>varalMembersToFilter</code></td>
<td>Long array</td>
<td>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</td>
</tr>
<tr>
<td><code>pvaralMembers</code></td>
<td>Long array</td>
<td>Returns the filtered list of member IDs that contain data for any of the specified POVs</td>
</tr>
</tbody>
</table>

**Return Value**

None.
Example

See HfmSliceCOM documentation for how to specify multiple POVs.

**FilterMembersThatHaveData2**

*Deprecated* - use **FilterMembersThatHaveData2ExtDim**.

**FilterMembersThatHaveData2ExtDim**

For a specified dimension, returns the filtered indexes of member IDs that contain data for any of the specified POVs. Supersedes FilterMembersThatHaveData2.

**Syntax**

```<HsvData>.FilterMembersThatHaveData2ExtDim pIUnkHfmSliceCOM, lDimID, vbConsiderBaseMembersOnly, varalMembersToFilter, pvaralFilteredMemberIndexes```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object representing the POVs to be analyzed.</td>
</tr>
<tr>
<td>lDimID</td>
<td>Long</td>
<td>The dimension for which you want a list of members. You can use the following dimensions: Account, ICP, Custom 1, Custom</td>
</tr>
<tr>
<td>vbConsiderBaseMembersOnly</td>
<td>Boolean</td>
<td>Determines if base members only are tested</td>
</tr>
<tr>
<td>varalMembersToFilter</td>
<td>Long array</td>
<td>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</td>
</tr>
<tr>
<td>pvaralFilteredMemberIndexes</td>
<td>Long array</td>
<td>Returns the filtered indexes of member IDs that contain data for any of the specified POVs. The indexes are the varalMembersToFilter 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 varalMembersToFilter array.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmSliceCOM documentation on how to specify multiple POVs.

**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 *Value* argument or the *Entity* and *Parent* 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 *Account* argument.

**FormatNumberToText** also enables you to override this default formatting.

**Syntax**

```vbnet
<HsvData>.FormatNumberToText(lEntity, lParent, lValue, lAccount, dNumber, lStatus, vbUseDefaultNumDecimals, sAlternateNumDecimals, vbUseDefaultScale, sAlternateScale, vbUseTheUsersFormattingParameters)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lEntity</td>
<td>Long (ByVal). If the <em>Value</em> argument specifies the &lt;Entity Currency&gt;, &lt;Entity Curr Adjs&gt;, or &lt;Entity Curr Total&gt; Value dimension member, <strong>FormatNumberToText</strong> scales the return value by using the Scale attribute of the entity’s default currency. Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the <em>Parent</em> or <em>Value</em> argument.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent of the Entity dimension member identified by the <em>Entity</em> argument. If the <em>Value</em> argument specifies the &lt;Parent Currency&gt;, &lt;Parent Curr Adjs&gt;, or &lt;Parent Curr Total&gt; Value dimension member, <strong>FormatNumberToText</strong> scales the return value by using the Scale attribute of this parent entity’s default currency. Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the <em>Entity</em> or <em>Value</em> argument.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the Value dimension member for the currency whose Scale attribute is to be applied to the return value. If you pass the member ID of the &lt;Entity Currency&gt;, &lt;Entity Curr Adjs&gt;, &lt;Entity Curr Total&gt;, &lt;Parent Currency&gt;, &lt;Parent Curr Adjs&gt;, or &lt;Parent Curr Total&gt; Value dimension member, the scaling is determined by the <em>Entity</em> and <em>Parent</em> arguments as described above.</td>
</tr>
<tr>
<td>lAccount</td>
<td>Long (ByVal). The member ID of the Account dimension member whose NumDecimalPlaces attribute is to be applied to the return value.</td>
</tr>
<tr>
<td>dNumber</td>
<td>Double (ByVal). The number to be converted and formatted.</td>
</tr>
<tr>
<td>lStatus</td>
<td>Long (ByVal). The cell status; the status determines whether a 0 passed to the <em>Number</em> argument is returned as the string 0 or as the string NoData. The following list provides guidelines for this argument:</td>
</tr>
<tr>
<td></td>
<td>- If you pass cell data returned by another method to the <em>Number</em> argument, it is recommended that you also pass the status returned by that method to the <em>Status</em> argument.</td>
</tr>
<tr>
<td></td>
<td>- If you want to format a number that is not returned from a cell, pass 0.</td>
</tr>
<tr>
<td></td>
<td><strong>Caution!</strong> If you pass cell data to <em>Number</em> but also pass a manually-specified status to <em>Status</em>, 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.</td>
</tr>
<tr>
<td>vbUseDefaultNumDecimals</td>
<td>Boolean (ByVal). Specify TRUE to apply the NumDecimalPlaces attribute of the account identified by the <em>Account</em> argument, otherwise FALSE. If you specify FALSE, the number of decimal places is determined by the <em>sAlternateNumDecimals</em> argument.</td>
</tr>
</tbody>
</table>
Argument | Description
--- | ---
sAlternateNumDecimals | Integer (ByVal). If you set the vbUseDefaultNumDecimals argument to FALSE, this argument specifies the number of decimal places to be applied to the return value.

To use the application’s default number of decimals, pass the HFMConstants type library constant DEFAULT_NUM_DECIMALS, described in “Number Defaults Constants” on page 906.

If you set the vbUseDefaultNumDecimals argument to TRUE, this argument is ignored.

vbUseDefaultScale | Boolean (ByVal). Specify TRUE to apply the Scale attribute of the currency identified by the lValue argument, otherwise FALSE. If you specify FALSE, the sAlternateScale argument specifies the scaling to be applied.

vbUseTheUsersFormatting | 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**

```plaintext
<HsvData>.FormatNumberToText2(lEntity, lParent, lValue, lAccount, dNumber, lStatus, vbUseDefaultNumDecimals, sAlternateNumDecimals, vbUseDefaultScale, sAlternateScale, vbUseTheUsersFormattingParameters, vbRemoveTrailingZeros, nNumDecimals, nScale)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lEntity</td>
<td>Long (ByVal). If the lValue argument specifies the &lt;Entity Currency&gt;, &lt;Entity Curr Adjs&gt;, or &lt;Entity Curr Total&gt; Value dimension member, FormatNumberToText2 scales the return value by using the Scale attribute of the entity’s default currency. Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the lParent or lValue argument.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent of the Entity dimension member identified by the lEntity argument. If the lValue argument specifies the &lt;Parent Currency&gt;, &lt;Parent Curr Adjs&gt;, or &lt;Parent Curr Total&gt; Value dimension member, FormatNumberToText2 scales the return value by using the Scale attribute of this parent entity’s default currency. Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the lEntity or lValue argument.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the Value dimension member for the currency whose Scale attribute is to be applied to the return value. If you pass the member ID of the &lt;Entity Currency&gt;, &lt;Entity Curr Adjs&gt;, &lt;Entity Curr Total&gt;, &lt;Parent Currency&gt;, &lt;Parent Curr Adjs&gt;, or &lt;Parent Curr Total&gt; Value dimension member, the scaling is determined by the lEntity and lParent arguments as described above.</td>
</tr>
<tr>
<td>lAccount</td>
<td>Long (ByVal). The member ID of the Account dimension member whose NumDecimalPlaces attribute is to be applied to the return value.</td>
</tr>
<tr>
<td>dNumber</td>
<td>Double (ByVal). The number to be converted and formatted.</td>
</tr>
<tr>
<td>lStatus</td>
<td>Long (ByVal). The cell status; the status determines whether a 0 passed to the dNumber argument is returned as the string 0 or as the string NoData. The following list provides guidelines for this argument:</td>
</tr>
<tr>
<td></td>
<td>• If you pass cell data returned by another method to the dNumber argument, it is recommended that you also pass the status returned by that method to the lStatus argument.</td>
</tr>
<tr>
<td></td>
<td>• If you want to format a number that is not returned from a cell, pass 0. <strong>Caution!</strong> If you pass cell data to dNumber but also pass a manually-specified status to lStatus, 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.</td>
</tr>
<tr>
<td>vbUseDefaultNumDecimals</td>
<td>Boolean (ByVal). Specify TRUE to apply the NumDecimalPlaces attribute of the account identified by the lAccount argument, otherwise FALSE. If you specify FALSE, the number of decimal places is determined by the sAlternateNumDecimals argument.</td>
</tr>
</tbody>
</table>
### Arguments and Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>sAlternateNumDecimals</code></td>
<td>Integer (ByVal). If you set the <code>vbUseDefaultNumDecimals</code> argument to FALSE, this argument specifies the number of decimal places to be applied to the return value.</td>
</tr>
<tr>
<td></td>
<td>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 906.</td>
</tr>
<tr>
<td></td>
<td>If you set the <code>vbUseDefaultNumDecimals</code> argument to TRUE, this argument is ignored.</td>
</tr>
<tr>
<td><code>vbUseDefaultScale</code></td>
<td>Boolean (ByVal). Specify TRUE to apply the Scale attribute of the currency identified by the <code>lValue</code> argument, otherwise FALSE. If you specify FALSE, the <code>sAlternateScale</code> argument specifies the scaling to be applied.</td>
</tr>
<tr>
<td><code>sAlternateScale</code></td>
<td>Integer (ByVal). If you set the <code>vbUseDefaultScale</code> 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. Passing 1 scales by a tenth, passing 2 scales by a hundredth, and so on.</td>
</tr>
<tr>
<td></td>
<td>To use the application’s default scaling, pass the HFMConstants type library constant <code>DEFAULT_SCALE</code>, described in “Number Defaults Constants” on page 906.</td>
</tr>
<tr>
<td></td>
<td>If you set the <code>vbUseDefaultScale</code> argument to TRUE, this argument is ignored.</td>
</tr>
<tr>
<td><code>vbUseTheUsersFormatting</code></td>
<td>Parameters 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.</td>
</tr>
<tr>
<td><code>vbRemoveTrailingZeros</code></td>
<td>Boolean (ByVal). A flag that specifies whether to remove trailing zeroes. Pass TRUE to remove trailing zeroes, FALSE otherwise.</td>
</tr>
<tr>
<td><code>nNumDecimals</code></td>
<td>Integer. Returns the number of decimal places applied to the return value.</td>
</tr>
<tr>
<td><code>nScale</code></td>
<td>Integer. Returns the degree of scaling applied to the return value.</td>
</tr>
</tbody>
</table>

### 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

```vbnet
<HsvData>.FormatStatusToText(lStatus)
```

#### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lStatus</code></td>
<td>Long (ByVal). A valid calculation status.</td>
</tr>
</tbody>
</table>

### 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 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**

```csharp
<HsvData>.FormatStoredNumberToText(dNumber, lStatus, sDecimalChar, sThousandsChar)
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>dNumber</em></td>
<td>Double (ByVal). The number to be formatted.</td>
</tr>
<tr>
<td><em>lStatus</em></td>
<td>Long (ByVal). A valid cell status. The expected usage is that you pass the cell status returned by a method such as <em>GetCell</em> or <em>GetCells</em>.</td>
</tr>
<tr>
<td><em>sDecimalChar</em></td>
<td>Integer (ByVal). The double-byte Integer that identifies the decimal to be used in the return value. Use <em>AscW</em> to get the delimiter’s Integer equivalent.</td>
</tr>
<tr>
<td><em>sThousandsChar</em></td>
<td>Integer (ByVal). The double-byte Integer that identifies the thousands delimiter to be used in the return value. Use <em>AscW</em> to get the delimiter’s Integer equivalent.</td>
</tr>
</tbody>
</table>

**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.

```csharp
Dim cData As HsvData, iDec As Integer
Dim iToush As Integer, sRet As String
Set cData = m_cSession.Data
iDec = AscW(",")
```
FormatTextToNumber

Takes a number stored in a String and converts it to a Double. If the lAccount 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)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lEntity</td>
<td>Long (ByVal). If the lValue argument specifies the &lt;Entity Currency&gt;, &lt;Entity Curr Adjs&gt;, or &lt;Entity Curr Total&gt; Value dimension member, FormatNumberToText scales the return value by using the Scale attribute of the entity’s default currency. Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the lParent or lValue argument.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent of the lEntity argument’s entity. If the lValue argument specifies the &lt;Parent Currency&gt;, &lt;Parent Curr Adjs&gt;, or &lt;Parent Curr Total&gt; Value dimension member, FormatNumberToText scales the return value by using the Scale attribute of this parent entity’s default currency. Otherwise, you must pass in a valid Entity dimension member ID, but the scaling is determined by the lEntity or lValue argument.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). A valid Value dimension ID. If the lAccount argument identifies a revenue, expense, asset, or liability account, FormatTextToNumber 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. <strong>Note:</strong> If you pass the member ID of the &lt;Entity Currency&gt;, &lt;Entity Curr Adjs&gt;, &lt;Entity Curr Total&gt;, &lt;Parent Currency&gt;, &lt;Parent Curr Adjs&gt;, or &lt;Parent Curr Total&gt; Value dimension member, the scaling is determined by the lEntity and lParent arguments as described above.</td>
</tr>
<tr>
<td>lAccount</td>
<td>Long (ByVal). The member ID of an Account dimension member. If the member is a revenue, expense, asset, or liability account, FormatTextToNumber scales the return value.</td>
</tr>
<tr>
<td>bstrText</td>
<td>String (ByVal). The text to be converted to a Double. The text must be numeric; the pbIsValidNumber argument returns FALSE if this argument is non-numeric.</td>
</tr>
<tr>
<td>vbUseDefaultScale</td>
<td>Boolean (ByVal). Specify TRUE to apply the Scale attribute of the currency identified by the lValue argument, otherwise FALSE. If you specify FALSE, the sAlternateScale argument sets the scaling to be applied.</td>
</tr>
<tr>
<td>sAlternateScale</td>
<td>Integer (ByVal). If you set the vbUseDefaultScale argument to FALSE, this argument specifies the degree of scaling to be applied to the return value. If you set the vbUseDefaultScale argument to TRUE, this argument’s value is ignored.</td>
</tr>
<tr>
<td>vbUseTheUsersFormattingParameters</td>
<td>Boolean (ByVal). Determines whether FormatTextToNumber uses the user’s preferred decimal and thousands separator characters or the system default characters when it reads the bstrText argument. Pass TRUE to use the user’s preferred characters, FALSE to use the system defaults.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pbIsValidNumber</td>
<td>Boolean. Returns TRUE if the bstrText argument is a valid number that can be converted, otherwise FALSE.</td>
</tr>
<tr>
<td>pbIsNumberNoData</td>
<td>Boolean. Returns TRUE if the bstrText argument is 0, otherwise FALSE.</td>
</tr>
</tbody>
</table>

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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrText</td>
<td>String (ByVal). The String to be converted.</td>
</tr>
<tr>
<td>sDecimalChar</td>
<td>Integer (ByVal). The double-byte Integer that identifies the decimal character used in the bstrText argument. Use AscW to get the delimiter’s Integer equivalent.</td>
</tr>
<tr>
<td>sThousandsChar</td>
<td>Integer (ByVal). The double-byte Integer that identifies the thousands separator character used in the bstrText argument. Use AscW to get the delimiter’s Integer equivalent.</td>
</tr>
<tr>
<td>pbIsValidNumber</td>
<td>Boolean. Returns TRUE if the bstrText argument contains a number that can be converted to a Double, otherwise FALSE.</td>
</tr>
<tr>
<td>pbIsNumberNoData</td>
<td>Boolean. Returns TRUE if the bstrText argument is 0, otherwise FALSE.</td>
</tr>
<tr>
<td>pdNumber</td>
<td>Double. Returns the formatted Double equivalent of the String passed to the bstrText argument.</td>
</tr>
</tbody>
</table>

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

Deprecated - use GetAllDescriptionsInSubCubeExtDim.

GetAllDescriptionsInSubCubeExtDim

Retrieve cell text descriptions for multiple cells. Supersedes GetAllDescriptionsInSubCube.

Syntax

```csharp
<HsvData>.GetAllDescriptionsInSubCubeExtDim pIUnkHfmSliceCOMin, ppIUnkHfmSliceCOMout, pvarabstrCellTextLabels, pvarabstrCellText
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOMin</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object representing the desired cells.</td>
</tr>
<tr>
<td>ppIUnkHfmSliceCOMout</td>
<td>HfmSliceCOM</td>
<td>Returns an HfmSliceCOM object representing the returned cells</td>
</tr>
<tr>
<td>pvarabstrCellTextLabelIds</td>
<td>String Array</td>
<td>Returns the cell text labels of the cell text</td>
</tr>
<tr>
<td>pvarabstrCellText</td>
<td>String Array</td>
<td>Returns the cell text</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See the HfmSliceCOM documentation on how to specify multiple POVs.

Note: The input HfmSliceCOM object should specify Scenario, Year and Value as fixed members.

GetAllURLNames

Retrieves the full list of URL names currently in Financial Management.

Syntax

```csharp
<HsvData>.GetAllURLNames pvarabstrURLNames
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvarabstrURLNames</td>
<td>Variant array. Array of URL names.</td>
</tr>
</tbody>
</table>

Example

This function is intended to be used prior to calling Load_drillable_regions in “Replace by URL name” mode and Extract_drillable_regions_by_url_names. 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 extracts all existing URL names in Financial Management.

```c
fsVARIANT *pvarabstrURLNames; GetAllURLNames (pvarabstrURLNames);
```

An example of the returned `BSTR` array `pvarabstrURLNames` contains “ERPI_USSales,” “ERPI_EUSales” and “ERPI_Salary.”

**GetAttachedDocumentsToCell**

*Deprecated* - use `GetAttachedDocumentsToCellExtDim`.

**GetAttachedDocumentsToCellExtDim**

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. Supersedes `GetAttachedDocumentsToCell`.

**Note:** `GetAttachedDocumentsToCellExtDim` differs from `GetAttachedDocumentsToCell`, in that it requires an additional param for cell text label ID.

**Syntax**

```c
<HsvData>.GetAttachedDocumentsToCellExtDim pIUnkHfmPovCOM, lCellTextItemId, pvarbstraDocumentPaths, pvarbstraDocumentFiles
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the desired POV</td>
</tr>
<tr>
<td><code>lCellTextItemId</code></td>
<td>Long</td>
<td>The cell text label ID for which information is being requested.</td>
</tr>
<tr>
<td><code>pvarbstraDocumentPaths</code></td>
<td>String array</td>
<td>Returns the paths of the files attached to the cell for the given cell text label.</td>
</tr>
<tr>
<td><code>pvarbstraDocumentFiles</code></td>
<td>String array</td>
<td>Returns the names of the files attached to the cell for the given cell text label.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation for how to specify a POV.

**GetBaseDataForAccount**

*Deprecated* - use `GetBaseDataForAccountExtDim`.
GetBaseDataForAccountExtDim

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 53.)

Syntax

\[ \text{GetCalcStatus } l\text{Scenario}, l\text{Year}, l\text{Period}, l\text{Entity}, l\text{Parent}, l\text{Value}, pl\text{CalcStatus} \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the subcube’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the subcube’s Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the subcube’s Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the subcube’s Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent of the lEntity argument’s entity.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the subcube’s Value dimension member.</td>
</tr>
<tr>
<td>plCalcStatus</td>
<td>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 HFMConstants type library; for a list of constants, see “Subcube Period Calculation Status Constants” on page 868.</td>
</tr>
</tbody>
</table>

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 is executed. The example assumes that the variables in GetCalcStatus’s member ID arguments were passed from another procedure.

```vba
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

Deprecated - use GetCalcStatusExExtDim.
**GetCalcStatusExExtDim**

Returns the calculation status of a cell. Supersedes GetCalcStatusEx.

**Syntax**

```vba
<HsvData>.GetCalcStatusExExtDim pIUnkHfmPov
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPov</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell’s POV</td>
</tr>
</tbody>
</table>

**Return Value**

Long. The calc status of the cell.

**Example**

See HfmPovCOM documentation on how to set the POV.

---

**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**

```vba
<HsvData>.GetCalcStatusStatistics(lScenarioID, lYearID, varalPeriodIDs, lValueID, varalEntityIDs, varalParentIDs, vbOBPActiveOnly)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenarioID</td>
<td>Long (ByVal). The member ID of the subcube’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYearID</td>
<td>Long (ByVal). The member ID of the subcube’s Year dimension member.</td>
</tr>
<tr>
<td>varalPeriodIDs</td>
<td>Long array (ByVal). The member IDs of the Period dimension members for which to return status information.</td>
</tr>
<tr>
<td>lValueID</td>
<td>Long (ByVal). The member ID of the subcube’s Value dimension member.</td>
</tr>
<tr>
<td>varalEntityIDs</td>
<td>Long array (ByVal). The member IDs of the Entity dimension members for which to return status information.</td>
</tr>
<tr>
<td>varalParentIDs</td>
<td>Long array (ByVal). The member IDs of the parents of the varalEntityIDs argument’s entities.</td>
</tr>
<tr>
<td>vbOBPActiveOnly</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 871.

- 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.

```vbscript
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**

*Deprecated* - use `GetCellExtDim`.

**GetCellExtDim**

Returns data and cell status for a specified cell. Supersedes GetCell.

**Syntax**

```vbscript
<HsvData>.GetCellExtDim pIUnkHfmPovCOM, pdData, plStatus
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>Object that specifies the cell POV</td>
</tr>
<tr>
<td>pdData</td>
<td>Double</td>
<td>Returns the cell’s data</td>
</tr>
<tr>
<td>plStatus</td>
<td>Long</td>
<td>Returns the cell’s status</td>
</tr>
</tbody>
</table>
Return Value
Integer. Returns 1 for success, 0 for failure.

Example
See the \texttt{HfmPovCOM} documentation on how to specify the POV.

\textbf{GetCellDescription}

\emph{Deprecated} - use \texttt{GetCellDescriptionExtDim}.

\textbf{GetCellDescriptionExtDim}

Returns the cell text description of a cell.

Syntax
\begin{verbatim}
%HsvData%.GetCellDescriptionExtDim pIUnkHfmPovCOM, pbstrCellText
\end{verbatim}

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{pIUnkHfmPovCOM}</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell's pov</td>
</tr>
<tr>
<td>\texttt{pbstrCellText}</td>
<td>String</td>
<td>The cell text</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
See the \texttt{HfmPovCOM} documentation on how to set the POV.

\textbf{GetCellDescriptions}

\emph{Deprecated} - use \texttt{GetCellDescriptionsExtDim}.

\textbf{GetCellDescriptionsExtDim}

Returns the cell text descriptions of one or more cells. Supersedes \texttt{GetCellDescriptions}.

Syntax
\begin{verbatim}
%HsvData%.GetCellDescriptionsExtDim pIUnkHfmSliceCOM, pvarabstrCellText
\end{verbatim}

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\texttt{pIUnkHfmSliceCOM}</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object representing the desired cells</td>
</tr>
</tbody>
</table>
### Argument

**pvarabstrCellText**  
String array  
Array of cell text descriptions

### Return Value

None.

### Example

See the `HfmSliceCOM` documentation on specifying multiple POVs.

---

**GetCellHistory**

*Deprecated* - use `GetCellHistoryExtDim`.

---

**GetCellHistoryExtDim**

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. Supersedes `GetCellHistory`.

### Syntax

```c
<HsvData>.GetCellHistoryExtDim pIUnkHfmPovCOM, pvarabstrServers, pvarabstrUserNames, pvaradTimeModified, pvaralActivityCode, pvaradValues, pvaralNoData
```

### Argument

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the desired POV</td>
</tr>
<tr>
<td>pvarabstrServers</td>
<td>String array</td>
<td>Returns the names of the application servers on which the data changes were made</td>
</tr>
<tr>
<td>pvarabstrUserNames</td>
<td>String array</td>
<td>Returns the usernames of the users who made the data changes</td>
</tr>
<tr>
<td>pvaradTimeModified</td>
<td>Double array</td>
<td>Returns the timestamps of the data changes. These are returned as Double values that can be cast to the Date format.</td>
</tr>
<tr>
<td>pvaralActivityCode</td>
<td>Long array</td>
<td>Returns the IDs of the user activities that caused the data changes</td>
</tr>
<tr>
<td>pvaradValues</td>
<td>Double array</td>
<td>Returns the cell values that the data changes resulted in.</td>
</tr>
<tr>
<td>pvaralNoData</td>
<td>Long Array</td>
<td>Variant array. Indicates whether cells contain data or no data. Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0 - The cell contains data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>1 - The cell contains no data.</td>
</tr>
</tbody>
</table>

### Return Value

None.
Example

See HfmPovCOM documentation for how to specify the POV.

### GetCellHistory2

 Deprecated - use GetCellHistory2ExtDim.

### GetCellHistory2ExtDim

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 EnumDataAuditItemsExtDim. To return a cell's data changes with cell values returned as Doubles, use GetCellHistory2ExtDim.

**Syntax**

```c
<HsvData>.GetCellHistory2ExtDim pIUnkHfmPovCOM, pvarabstrServers, pvarabstrUserNames, pvaradTimeModified, pvaralActivityCode, pvarabstrValues, pvaralNoData
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the desired POV</td>
</tr>
<tr>
<td>pvarabstrServers</td>
<td>String array</td>
<td>Returns the names of the application servers on which the data changes were made</td>
</tr>
<tr>
<td>pvarabstrUserNames</td>
<td>String array</td>
<td>Returns the usernames of the users who made the data changes</td>
</tr>
<tr>
<td>pvaradTimeModified</td>
<td>Double array</td>
<td>Returns the timestamps of the data changes. These are returned as Double values that can be cast to the Date format.</td>
</tr>
<tr>
<td>pvaralActivityCode</td>
<td>Long array</td>
<td>Returns the IDs of the user activities that caused the data changes</td>
</tr>
<tr>
<td>pvarabstrValues</td>
<td>String array</td>
<td>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</td>
</tr>
<tr>
<td>pvaralNoData</td>
<td>Long array</td>
<td>Indicates whether cells contain data or no data. Valid values are as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>l 0 - The cell contains data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>l 1 - The cell contains no data.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation for how to set the POV.
**GetCellJournalEntries**

*Deprecated* - use `GetCellJournalEntriesExtDim`.

**GetCellJournalEntriesExtDim**

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. Supersedes `GetCellJournalEntries`.

**Syntax**

```
<HsvData>.GetCellJournalEntriesExtDim pIUnkHfmPovCOM, pvaradData, pvarabstrData, pvaralJournalIDs
```

**Argument** | **Type** | **Description**
--- | --- | ---
`pIUnkHfmPovCOM` | HfmPovCOM | HfmPovCOM object representing the cell's POV
`pvaradData` | Double Array | Returns the data in the cell's journal entries as a double type
`pvarabstrData` | String Array | Returns the data in the cell's journal entries as a string
`pvaralJournalIDs` | Long Array | Returns the IDs of the cell's journal entries.

**Return Value**

None.

**Example**

See the `HfmPovCOM` documentation on how to set the cell’s POV.

**GetCellLineItems**

*Deprecated* - use `GetCellLineItemsExtDim`.

**GetCellLineItemsExtDim**

Returns arrays of the data and descriptions for the specified cell’s line items. Data is returned in a Double array. Supersedes `GetCellLineItems`.

**Syntax**

```
<HsvData>.GetCellLineItemsExtDim pIUnkHfmPovCOM, pvardData, pvarbstrDetails
```

**Argument** | **Type** | **Description**
--- | --- | ---
`pIUnkHfmPovCOM` | HfmPovCOM | HfmPovCOM object specifying the cell's POV
### GetCells

*Deprecated* - use GetCellsExtDim.

### GetCellsExtDim

Returns the data and statuses of multiple cells. Supersedes GetCells.

**Syntax**

```plaintext
<HsvData>.GetCellsExtDim pIUnkHfmSliceCOM, pvaradData, pvaralStatus
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmPovSliceCOM</td>
<td>An HfmPovSliceCOM object representing the desired cells.</td>
</tr>
<tr>
<td>pvaradData</td>
<td>Double array</td>
<td>Returns an array of data values for each cell.</td>
</tr>
<tr>
<td>pvaralStatus</td>
<td>Long array</td>
<td>Returns an array of statuses for each cell.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See the HfmPovSliceCOM documentation on how to specify multiple cells.

### GetCellsWithRowSuppression

*Deprecated* - use GetCellsWithRowSuppressionExtDim.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvardData</td>
<td>Double array</td>
<td>Returns an array of data for each line item</td>
</tr>
<tr>
<td>pvarabstrDetails</td>
<td>String array</td>
<td>Returns an array of detail text for each line item</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation for specifying a POV.
GetCellsWithRowSuppressionExtDim

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.
Supersedes GetCellsWithRowSuppression

Syntax

<HsvData>.GetCellsWithRowSuppressionExtDim pIUnkHfmSliceCOM, lSuppressOptions,
1NumRows, pvarabNonSuppressedRowIndex, pvaradData, pvaralStatus

Argument | Type | Description
--- | --- | ---
plUnkHfmSliceCOM | HfmSliceCOM | HfmSliceCOM object representing the desired cell povs
lSuppressOptions | Long | Long (ByVal). The type of suppression to be applied. You can specify any combination of
the following hexadecimal values:
l 0x00. No suppression; data is returned for all cells specified by the dimension member
arguments.
l 0x01. Suppress rows in which all cells contain no data.
l 0x02. Suppress rows in which all cells contain derived data.
l 0x04. Suppress rows in which all cells contain a value of zero (0).
l 0x08. Suppress rows in which all cells consist of invalid intersections
1NumRows | Long | Long (ByVal). Specifies the number of rows that the point of view comprises.
Caution! 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
pvarabNonSuppressedRowIndex | Boolean array | Indicates whether a row is suppressed. Returns TRUE if the row is suppressed, FALSE
otherwise. A row is suppressed only when all of the row’s cells meet the lSuppressOptions
argument’s criteria.
pvaradData | Double array | Returns data for the rows of cells that are not suppressed.
Note: If only some of the cells in a row meet the lSuppressOptions argument’s suppression
criteria, these cells will not be suppressed. Only rows in which all the cells meet the criteria
are suppressed. You can correlate this data with the members passed in the member ID
arguments by looping through the array returned in the pvarabRowsSuppressed
pvaralStatus | Long array | Returns the statues of the cells returned in pvaradData

Return Value

Data Type and description.

Example

See the HfmSliceCOM documention on specifying POVs for multiple cells.
GetCellsWithRowSuppression2

 Deprecated - use GetCellsWithRowSuppression2ExtDim.

GetCellsWithRowSuppression2ExtDim

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. Supersedes GetCellsWithRowSuppression2.

Syntax

```<HsvData>.GetCellsWithRowSuppression2ExtDim pIUnkHfmSliceCOM, lSuppressOptions, lNumRows, pvaravbRowIsSuppressed, pvaradData, pvaralStatus```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object representing the povs of the desired cells</td>
</tr>
<tr>
<td>lSuppressOptions</td>
<td>Long</td>
<td>The type of suppression to be applied. You can specify any combination of the following hexadecimal</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x00. No suppression–data will be returned for all cells specified by the dimension member arguments.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x01. Suppress rows in which all cells contain no data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x02. Suppress rows in which all cells contain derived data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x04. Suppress rows in which all cells contain a value of zero (0).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0x08. Suppress rows in which all cells consist of invalid intersections.</td>
</tr>
<tr>
<td>lNumRows</td>
<td>Long</td>
<td>Specifies the number of rows that the point of view comprises</td>
</tr>
<tr>
<td>pvaravbRowIsSuppressed</td>
<td>Boolean array</td>
<td>Indicates whether a row is to be suppressed.</td>
</tr>
<tr>
<td>pvaradData</td>
<td>Double array</td>
<td>Returns data for the rows of cells that are not suppressed. If all rows are suppressed, returns empty Array</td>
</tr>
<tr>
<td>pvaralStatus</td>
<td>Long Array</td>
<td>Returns the statuses of the cells that are not suppressed. If all rows are suppressed, returns empty Array</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmSliceCOM documentation on how to specify multiple POVs.

GetCountOfAttachedDocumentsToCell

 Deprecated - use GetCountOfAttachedDocumentsToCellExtDim.
**GetCountOfAttachedDocumentsToCellExtDim**

Returns the count of documents attached to a cell. Supersedes GetCountOfAttachedDocumentsToCell.

**Syntax**

```
<HsvData>.GetCountOfAttachedDocumentsToCellExtDimIUnkHfmPovCOM.
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the desired POV</td>
</tr>
</tbody>
</table>

**Return Value**

Long. Returns the number of documents attached to the cell.

**Example**

See HfmPovCOM documentation on how to set the POV.

**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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the subcube’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the subcube’s Year dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the subcube’s Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the subcube’s Value dimension member.</td>
</tr>
</tbody>
</table>

**Example**

GetCurrencyCube is used in the Example for HsvCurrencyCube.GetCell.

**GetDataForAllMetadataCombinations**

For internal use.
GetDynamicAccountRulesCache

For internal use.

GetLineItems

Deprecated – used GetCellLineItemsExtDim.

GetDataGridExtDim

For internal use.

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

Deprecated - use GetMembersThatHaveDataExtDim.

GetMembersThatHaveDataExtDim

Returns an array of dimension member IDs (as specified by lDimID) that contain data for the specified POV. Supersedes GetMembersThatHaveData.

Syntax

`<HsvData>.GetMembersThatHaveDataExtDim pIUnkHfmPovCOM, lDimID, vbConsiderBaseMembersOnly, varalMembersAutomaticallyIncludedInOutput, pvaralMembers.`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object that represents the POV. Note: Period, Account, ICP and Custom dimension members can have a value of either MEMBERALL or MEMBERNOTUSED.</td>
</tr>
<tr>
<td>Argument</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>lDimID</td>
<td>Long</td>
<td>The dimension for which you want a list of members. You can use the following dimensions: Account, ICP or Custom dimensions</td>
</tr>
<tr>
<td>vbConsiderBaseMembersOnly</td>
<td>Boolean</td>
<td>Determines if base members only are tested</td>
</tr>
<tr>
<td>varalMembersAutomaticallyIncludedInOutput</td>
<td>Long array</td>
<td>A list of members to be included in the output</td>
</tr>
<tr>
<td>pvaralMembers</td>
<td>Long array</td>
<td>Returns the filtered list of member IDs that contain data for any of the specified POVs.</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
See HfmPovCOM documentation on how to specify POV with HfmPovCOM.

**GetMultiServerMaxSyncDelayForChanges**
*For internal use.*

**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```  

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the subcube’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the subcube’s Year dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the subcube’s Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent of the lEntity argument’s entity.</td>
</tr>
</tbody>
</table>

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 ID, and the returned submission group IDs, are in arrays that have a one-to-one correspondence.

Syntax

\[
\text{GetPhaseSubmissionGridForGivenScenarioPeriod}(\text{lScenario, varalPeriodIDs, varalPhaseIDs, pvarbstrGroupInfo})
\]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>varalPeriodIDs</td>
<td>Long array (ByVal). The member IDs of the Period dimension members.</td>
</tr>
<tr>
<td>varalPhaseIDs</td>
<td>Long array (ByVal). The phase IDs of the submission phases.</td>
</tr>
<tr>
<td>pvarbstrGroupInfo</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Note:** If all submission groups are assigned to a period-submission group combination, the asterisk (*) 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 scenario and period.

```vba
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

_Deprecated_ - use “GetStatusEx” on page 365..

GetStatusEx

_Deprecated_ - use GetStatusExExtDim.
**GetStatusExExtDim**

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. Supersedes GetStatusEx.

**Syntax**

```
<HsvData>.GetStatusExExtDim pIUnkHfmPovCOM, bIncludeTransType, plStatus,
plExtendedStatus
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object identifying the cell's POV</td>
</tr>
<tr>
<td><code>bIncludeTransType</code></td>
<td>Boolean</td>
<td>Determines whether the cell's transaction status will be returned in the <code>plStatus</code> argument. Specify TRUE to return the transaction status, otherwise FALSE</td>
</tr>
<tr>
<td><code>plStatus</code></td>
<td>Long</td>
<td>Returns the cell's status.</td>
</tr>
<tr>
<td><code>plExtendedStatus</code></td>
<td>Long</td>
<td>Returns additional status information</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

---

**GetStatusUsingPhaseID**

*Deprecated* - use `GetStatusUsingPhaseIDExtDim`.

**GetStatusUsingPhaseIDExtDim**

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 IPhaseID 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 IPhaseID argument.

**Note:** The `bIncludeTransType` argument enables you to exclude the transaction status from the return value.
Supersedes GetStatusUsingPhaseID.

Syntax

```csharp
<HsvData>.GetStatusUsingPhaseIDExtDim pIUnkHfmPovCOM, lPhaseID, bIncludeTransType, plStatus
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>HfmPovCOM representing the cell’s pov</td>
</tr>
<tr>
<td><code>lPhaseID</code></td>
<td>Long</td>
<td>Pass one of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● To get a cell’s status, pass the HFMConstant MEMBERNOTUSED.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● To get a submission phase’s status, pass the phase ID</td>
</tr>
<tr>
<td><code>bIncludeTransType</code></td>
<td>Boolean</td>
<td>Determines whether the cell’s transaction status will be returned in the plStatus argument. Specify TRUE to return the transaction status, otherwise FALSE</td>
</tr>
<tr>
<td><code>plStatus</code></td>
<td>Long</td>
<td>Returns the cell’s status.</td>
</tr>
</tbody>
</table>

Return Value

Integer. For internal use.

Example

See the HfmPovCOM documentation for how to specify a POV.

GetTextCell

*Deprecated* - use GetTextCellExtDim.

GetTextCellExtDim

Returns data for a cell, as a string. Supersedes GetTextCell.

Syntax

```csharp
<HsvData>.GetTextCellExtDim pIUnkHfmPovCOM, sNumDecimals, sScale, pbstrData, plStatus
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object that specifies the cell POV</td>
</tr>
<tr>
<td><code>sNumDecimals</code></td>
<td>Integer</td>
<td>The number of decimal places that the pbstrData value will contain.</td>
</tr>
<tr>
<td><code>sScale</code></td>
<td>Integer</td>
<td>The degree of scaling to be applied to the pbstrData return value.</td>
</tr>
<tr>
<td><code>pbstrData</code></td>
<td>String</td>
<td>Returns the data value</td>
</tr>
<tr>
<td><code>plStatus</code></td>
<td>Long</td>
<td>Returns the cell status</td>
</tr>
</tbody>
</table>
Return Value
None.

Example
See the HfmPovCOM documentation for how to specify the cell POV.

**GetTextCellLineItems**

*Deprecated* - use `GetTextCellLineItemsExtDim`.

**GetTextCellLineItemsExtDim**

Appends or updates line items for the specified cell. Supersedes `SetTextCellLineItems`.

Syntax

```
<HsvData>.SetTextCellLineItemsExtDim pIUnkHfmPovCOM, varabstrData, varabstrDetails
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>An HfmPovCOM object representing the cell's pov</td>
</tr>
<tr>
<td><code>varabstrData</code></td>
<td>StringArray</td>
<td>An array of line item detail data values</td>
</tr>
<tr>
<td><code>varabstrDetails</code></td>
<td>String Array</td>
<td>An array of line item detail text</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
See HfmPovCOM documentation on how to set the POV.

**GetTextCells**

*Deprecated* - use `GetTextCellsExtDim`.

**GetTextCellsExtDim**

Returns cell data and status for multiple cells. Supersedes `GetTextCells`.

Syntax

```
<HsvData>.GetTextCellsExtDim pIUnkHfmSliceCOM, sNumDecimals, sScale, pvarabstrData, pvaralStatus
```

368  HsvData Type Library
<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object representing the desired cells</td>
</tr>
<tr>
<td>sNumDecimals</td>
<td>Integer</td>
<td>The number of decimal places that the amounts in the pvarabstrData’s values will contain.</td>
</tr>
<tr>
<td>sScale</td>
<td>Integer</td>
<td>Specifies the degree of scaling to be applied to the pvarabstrData’s return values.</td>
</tr>
<tr>
<td>pvarabstrData</td>
<td>String array</td>
<td>Array of cell values.</td>
</tr>
<tr>
<td>pvaralStatus</td>
<td>Long array</td>
<td>Array of cell statuses.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See the HfmSliceCOM documentation on how to define multiple POVs.

GetTextCellsWithRowSuppression

Deprecated - use GetTextCellsWithRowSuppressionExtDim.

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.

GetTextCellsWithRowSuppressionExtDim

Syntax

```csharp
<HsvData>.GetTextCellsWithRowSuppressionExtDim plUnkHfmSliceCOM, sNumDecimals, sScale, lSuppressOptions, lNumRows, pvaravbRowIsSuppressed, pvarabstrData, VARIANT *pvaralStatus
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object specifying the desired cells</td>
</tr>
<tr>
<td>sNumDecimals</td>
<td>Integer</td>
<td>The number of decimal places that the data returned in the pvarabstrData argument will contain</td>
</tr>
<tr>
<td>sScale</td>
<td>Integer</td>
<td>Indicates the degree of scaling to apply to the data returned in the pvarabstrData argument</td>
</tr>
<tr>
<td>Argument</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>lSuppressOptions</td>
<td>Long</td>
<td>The type of suppression to be applied. You can specify any combination of the following hexadecimal values:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0x00. No suppression–data will be returned for all cells specified by the dimension member arguments.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0x01. Suppress rows in which all cells contain no data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0x02. Suppress rows in which all cells contain derived data.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0x04. Suppress rows in which all cells contain a value of zero (0).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0x08. Suppress rows in which all cells consist of invalid intersections.</td>
</tr>
<tr>
<td>NumRows</td>
<td>Long</td>
<td>Specifies the number of rows that the point of view comprises</td>
</tr>
<tr>
<td>pvarabRowIsSuppressed</td>
<td>Booleanarray</td>
<td>Indicates whether a row is suppressed</td>
</tr>
<tr>
<td>pvarabstrData</td>
<td>String array</td>
<td>Returns data for the rows of cells that are not suppressed</td>
</tr>
<tr>
<td>pvaralStatus</td>
<td>Long array</td>
<td>Returns the statuses of the cells returned in the pvarabstrData argument</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See the HfmSliceCOM documentation on how to set multiple cell povs.

**GetTextCellsWithRowSuppression2**

*Deprecated* - use [GetTextCellsWithRowSuppression2ExtDim](#).

**GetTextCellsWithRowSuppression2ExtDim**

Returns the data and status for rows that are not suppressed. Supersedes GetTextCellsWithRowSuppression2

**Syntax**

```
<HsvData>.GetTextCellsWithRowSuppression2  pIUnkHfmSliceCOM, sNumDecimals, sScale, lSuppressOptions, lNumRows, pvarabRowIsSuppressed, pvarabstrData, pvaralStatus
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object specifying the povs of the desired cells</td>
</tr>
<tr>
<td>sNumDecimals</td>
<td>Integer</td>
<td>The number of decimal places that the data returned in the pvarabstrData argument contains</td>
</tr>
<tr>
<td>sScale</td>
<td>Integer</td>
<td>Indicates the degree of scaling to apply to the data returned in the pvarabstrData argument</td>
</tr>
</tbody>
</table>
### GetTextLineItems

**Syntax**

```vbnet
<HsvData>.GetTextLineItems lScenario, lYear, lPeriod, lEntity, lParent, lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, pbvarSavedAsYTD, pvarlEntryIDs, pvarabstrData, pvarbstrDescriptions
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal).</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal).</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal).</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal).</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal).</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal).</td>
</tr>
<tr>
<td>lAccount</td>
<td>Long (ByVal).</td>
</tr>
<tr>
<td>lICP</td>
<td>Long (ByVal).</td>
</tr>
<tr>
<td>lCustom1</td>
<td>Long (ByVal).</td>
</tr>
<tr>
<td>pbvarSavedAsYTD</td>
<td>Returns data for the rows of cells that are not suppressed</td>
</tr>
<tr>
<td>pvarlEntryIDs</td>
<td>Returns the statuses of the cells returned in the pvarabstrData argument</td>
</tr>
</tbody>
</table>

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long</td>
<td>The type of suppression to be applied. You can specify any combination of the following hexadecimal</td>
</tr>
<tr>
<td>lNumRows</td>
<td>Long</td>
<td>Specifies the number of rows that the point of view comprises</td>
</tr>
<tr>
<td>pvarabRowIsSuppressed</td>
<td>Boolean array</td>
<td>Indicates whether a row is suppressed</td>
</tr>
<tr>
<td>pvarabstrData</td>
<td>String Array</td>
<td>Returns data for the rows of cells that are not suppressed</td>
</tr>
<tr>
<td>pvarabStatus</td>
<td>Long Array</td>
<td>Returns the statuses of the cells returned in the pvarabstrData argument</td>
</tr>
</tbody>
</table>

**Example**

See the HfmSliceCOM documentation on how to set up multiple POVs.
### 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 pIUnkTransactionData
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkTransactionData</code></td>
<td>HsvTransactionData object (ByVal). Pass an HsvTransactionData object for which you have already set selection criteria. For information on using <code>GetTransactionData</code> with the HsvTransactionData object’s methods, see “HsvTransactionData Object Methods” on page 657.</td>
</tr>
</tbody>
</table>

**Example**

GetTransactionData is used in HsvMDArrays 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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code>, <code>lPeriodID</code>, <code>pbstrMetaNoPhase</code>, <code>pbstrPhaseNoMeta</code></td>
<td>Submission group names that are assigned to members but not to phases, and to phases but not to members.</td>
</tr>
</tbody>
</table>
Argument | Description
--- | ---
8Scenario | Long (ByVal). The member ID of the process unit's Scenario dimension member.
8PeriodID | Long (ByVal). The member ID of the process unit's Period dimension member.
8pbstrMetaNoPhase | String. Returns the names of the submission groups assigned to dimension members but not to submission phases.
8pbstrPhaseNoMeta | 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.

```vba
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
groupsNoMembers = sRet
End Function
```

### GetURLByName

Returns a single XML block representing the specified URL.

#### Syntax

```vba
<HsvData>.GetURLByName bstrURLName, pbstrURL
```

**Argument | Description**
--- | ---
8bstrURLName | String (ByVal). The URL name.
8pbstrURL | String. A XML block representing the URL link and its multi-language display names.

#### Examples

Oracle Hyperion Smart View for Office requires an additional function call to obtain the URL because its HFMOfficeProvider module cannot cache previous queries. Financial Management server provides 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 returns a single XML block representing the URL with the specified URL name “ERPI_Sales.”

```vba
GetURLByName (_T("ERPI_Sales"), pbstrURL);
```

An example of the returned BSTR array pbstrURL:

```xml
<foldercontents path="/"
   <resource name="Sales Report" description="" type="application/x-hyperion-applicationbuilder-report">
```

---

**HsvData Object Methods** 373
GetURLsForCell

Retrieves all URLs whose regions cover a specified POV.

Syntax

```hsvdata> GetURLsForCell bstrScenario, bstrYear, bstrPeriod, bstrEntity, bstrAccount, pvarabstrURLNames, pvarabstrURLs```

**Argument**

- **bstrScenario**  
  String (ByVal). The Scenario dimension.

- **bstrYear**  
  String (ByVal). The Year dimension.

- **bstrPeriod**  
  String (ByVal). The Period dimension.

- **bstrEntity**  
  String (ByVal). The Entity dimension.

- **bstrAccount**  
  String (ByVal). The Account dimension.

- **pvarabstrURLNames**  
  Variant array. Array of URL names.

- **pvarabstrURLs**  
  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 does not support `MEMBERALL` or member lists. This function performs a cache lookup on the region cache for all URL names and URLs matching this POV. The following example returns a list of all URL names and XML blocks for URL for the specified POV scenario “Actual,” year “2012,” period “January,” entity “Stamford,” and account “Sales.”

```csharp
GetURLsForCell (_T("Actual"), _T("2012"), _T("January"), _T("Stamford"), _T("Sales"), pvarabstrURLNames, pvarabstrURLs);
```

An example of the returned BSTR array `pvarabstrURLNames` contains “ERPI_SalesReport” and “ERPI_EUSalesReport.” An example of the returned BSTR array `pvarabstrURLs` contains two BSTR. The first BSTR is as follows:

```xml
<resource name="Sales Report" description="" type="application/x-hyperion-applicationbuilder-report">
```
The second BSTR is as follows:

```
<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&amp;fdmTargetAppName=Comma95&amp;sso_token=$SSO_TOKEN
                &sso_username=$SSO_USERNAME$&sso_password=$SSO_PASSWORD$&$ATTR(id)
                &RCP_VERSION=$RCP_VERSION$</url>
        </action>
    </resource>
</foldercontents>
```

### InsertLineItemDetails

*Deprecated* - use `InsertLineItemDetailsExtDim`

### InsertLineItemDetailsExtDim

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

**Syntax**

```
<HsvData>.InsertLineItemDetailsExtDim pIUnkHfmPovCOM, varabstrDetail, varalInsertPos
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell’s pov</td>
</tr>
<tr>
<td>varabstrDetail</td>
<td>String Array</td>
<td>An array of strings that contain the line item descriptions to be added</td>
</tr>
<tr>
<td>varalInsertPos</td>
<td>Long Array</td>
<td>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 varabstrDetail argument's array.</td>
</tr>
</tbody>
</table>
Return Value
None.

Example
See HfmPovCOM documentation on how to set the POV.

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
bstrCellText 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. See “Loading Data” on page 797.

Syntax
\(<HsvData>.Load bstrServerFileName, bstrLogFileName, varavOptions\)

Argument Description
bstrServerFileName String (ByVal). The name and path of the data load file. This file must exist on the application server.

bstrLogFileName String (ByVal). The name and path of the log file. The path must exist on the application server.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varavOptions</td>
<td>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 64. <strong>Tip:</strong> Use EnumLoadOptions to return information about the valid load options.</td>
</tr>
</tbody>
</table>

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 796.

The listed indexes apply to the array passed to Load and to the first dimension of the array returned by EnumLoadOptions.

### Table 64: Data Load Options

<table>
<thead>
<tr>
<th>Index</th>
<th>Load Option Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td><strong>Option:</strong> Delimiter</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies a load file’s delimiter.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> A valid delimiter character (String subtype).</td>
</tr>
<tr>
<td>2</td>
<td><strong>Option:</strong> Append to Log File</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether log data is appended to or overwrites the existing log file.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> Boolean — TRUE to append, FALSE to overwrite.</td>
</tr>
<tr>
<td>3</td>
<td><strong>Option:</strong> Load Calculated</td>
</tr>
<tr>
<td></td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>4</td>
<td><strong>Option:</strong> Duplicates</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Sets the data load mode to either replace or merge.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> This option corresponds to the Load options in Financial Management Load Data. See the Oracle Hyperion Financial Management User’s Guide.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> One of the constants listed in “Update Mode Constants” on page 320.</td>
</tr>
<tr>
<td>5</td>
<td><strong>Option:</strong> Accumulate within file</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether multiple values for the same cell within the load file should be accumulated or allowed to overwrite each other.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Data for system accounts never gets accumulated.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> TRUE to accumulate, FALSE to overwrite.</td>
</tr>
<tr>
<td>6</td>
<td><strong>Option:</strong> Does the file contain shares data</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether the load file contains shares data such as “shares outstanding” or “voting outstanding” or “owned.”</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> TRUE if the file contains shares data, FALSE otherwise.</td>
</tr>
<tr>
<td>Index</td>
<td>Load Option Information</td>
</tr>
<tr>
<td>-------</td>
<td>--------------------------</td>
</tr>
<tr>
<td>7</td>
<td><strong>Option:</strong> Mode</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether the data file is loaded or is merely scanned when Load is called.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> One of the following HsvcDataLoad type library constants:</td>
</tr>
<tr>
<td></td>
<td>- HSV_DATALOAD_LOAD. Data is loaded.</td>
</tr>
<tr>
<td></td>
<td>- HSV_DATALOAD_SCAN. Data is scanned but not loaded.</td>
</tr>
<tr>
<td></td>
<td>The Mode option defaults to scanning, so you must set the option to HSV_DATALOAD_LOAD to load data.</td>
</tr>
<tr>
<td>8</td>
<td><strong>Option:</strong> Does the file contain submission phase data</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether the load file contains data for phased submissions.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> TRUE if the file contains phased submission data; otherwise, FALSE..</td>
</tr>
<tr>
<td>9</td>
<td><strong>Option:</strong> FileFormat</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies the load file format. Only the native file format is supported.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> The HsvcDataLoad type library constant, HSV_DATALOAD_FILE_FORMAT_NATIVE</td>
</tr>
<tr>
<td>10</td>
<td><strong>Option:</strong> DecimalChar</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies the decimal character used in the load file. By default this option is set to an empty string, which indicates the load operation uses the decimal character set as the user preference.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> To get the user preferences for the decimal and thousands separator characters, use the HsvSystemInfo method GetNumberFormattingUserParameters.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> If the load file decimal character differs from that specified as the user preference, specify the load file's decimal character. (String subtype).</td>
</tr>
<tr>
<td>11</td>
<td><strong>Option:</strong> ThousandsChar</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> 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 uses the thousands separator character set as the user preference.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to Load:</strong> If the load file decimal character differs from that specified as the user preference, specify the load file's thousands separator character. (String subtype).</td>
</tr>
</tbody>
</table>

**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`.

```vbnet
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.

**Syntax**

```xml
<HsvData>.LoadDrillableRegions bstrRegionsFilename, bstrLogFileName, lLoadMode, pvbErrors, pvbWarnings
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrRegionsFilename</td>
<td>String (ByVal). The file path to the definition file on the Financial Management application server.</td>
</tr>
<tr>
<td>bstrLogFileName</td>
<td>String (ByVal). The name and path of the log file. The path must exist on the application server.</td>
</tr>
<tr>
<td>lLoadMode</td>
<td>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.</td>
</tr>
<tr>
<td>pvbErrors</td>
<td>Boolean. Returns TRUE if errors occurred during the load task. See the log file for details.</td>
</tr>
<tr>
<td>pvbWarnings</td>
<td>Boolean. Returns TRUE if warnings occurred during the load task.</td>
</tr>
</tbody>
</table>

**lLoadMode Parameter**

- Replace all — This mode clears 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 clears and reloads a URL definition in Financial Management if a URL with the same name is in the load file. All other URL definitions remain unchanged. This mode is intended to be used by products that manage data input, either manually or automatically.

- Merge — This mode adds new region definitions to the existing URL. If a new XML block for URL is supplied for an existing URL, it replaces the prior XML block. If it is not provided, the existing XML block is kept for the specified URL. A new URL is created if there are no existing URLs with the same URL name.

- Scan Only — This mode scans the input file for invalid format in the definition. The new definition is not stored to the database. The existing definition remains unchanged.

**Note:** For the Merge mode, if a URL link is not supplied and the URL definition exists, the new region definitions is merged. However, if the URL does not exist, an error is logged and that URL definition is 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 is 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?
        &fdmAppName=CommaFDM&amp;fdmTargetAppName=Comma95&amp;sso_token=$SSO_TOKEN$&
        &sso_username=$SSO_USERNAME$&amp;sso_password=$SSO_PASSWORD$&amp;$ATTR(id)$&
        &amp;RCP_VERSION=$RCP_VERSION$</url>
    </action>
  </resource>
</foldercontents>

!Regions
Actual;2012;January;Stamford;Payroll
Actual;2012;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?
        &fdmAppName=CommaFDM&amp;fdmTargetAppName=Comma95&amp;sso_token=$SSO_TOKEN$&
        &sso_username=$SSO_USERNAME$&amp;sso_password=$SSO_PASSWORD$&amp;$ATTR(id)$&
        &amp;RCP_VERSION=$RCP_VERSION$</url>
    </action>
  </resource>
</foldercontents>

!Regions
Actual;2012;MEMBERALL;Germany; Sales
Actual;2012;MEMBERALL;England;Sales
Actual;2012;MEMBERALL;France;Sales

SetCalcStatusLocked

Locks the cells for a period in a subcube. (For information on subcubes, see “About Subcubes” on page 53.)

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
lScenario Long (ByVal). The member ID of the subcube's Scenario dimension member.
lYear Long (ByVal). The member ID of the subcube's Year dimension member.
lPeriod Long (ByVal). The member ID of the Period dimension member for which data is being locked.
lEntity Long (ByVal). The member ID of the subcube's Entity dimension member.
lParent Long (ByVal). The member ID of the parent of the lEntity argument's entity.
lValue 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
varalScenario Long array (ByVal). The member IDs of the process units' Scenario dimension members.
varalYear Long array (ByVal). The member IDs of the process units' Year dimension members.
varalPeriod Long array (ByVal). The member IDs of the process units' Period dimension members.
varalEntity Long array (ByVal). The member IDs of the process units' Entity dimension members.
varalParent Long array (ByVal). The member IDs of the parents of the varalEntity argument's entities.
varalValue 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 53.)

Syntax
<HsvData>.SetCalcStatusUnlocked lScenario, lYear, lPeriod, lEntity, lParent, lValue
**Argument Description**

- **Scenario**: Long (ByVal). The member ID of the subcube’s Scenario dimension member.
- **Year**: Long (ByVal). The member ID of the subcube’s Year dimension member.
- **Period**: Long (ByVal). The member ID of the Period dimension member for which data is being locked.
- **Entity**: Long (ByVal). The member ID of the subcube’s Entity dimension member.
- **Parent**: Long (ByVal). The member ID of the parent of the **Entity** argument’s entity.
- **Value**: 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 locked with **SetCalcStatusLockedForMultipleProcessUnits**.

**Syntax**

```csharp
<HsvData>.SetCalcStatusUnlockedForMultipleProcessUnits varalScenario, varalYear, varalPeriod, varalEntity, varalParent, varalValue
```

**Argument Description**

- **varalScenario**: Long array (ByVal). The member IDs of the process units’ Scenario dimension members.
- **varalYear**: Long array (ByVal). The member IDs of the process units’ Year dimension members.
- **varalPeriod**: Long array (ByVal). The member IDs of the process units’ Period dimension members.
- **varalEntity**: Long array (ByVal). The member IDs of the process units’ Entity dimension members.
- **varalParent**: Long array (ByVal). The member IDs of the parents of the **varalEntity** argument’s entities.
- **varalValue**: Long array (ByVal). The member IDs of the process units’ Value dimension members.

**SetCell**

*Deprecated* - use **SetCellExtDim**.

**SetCellExtDim**

Set the data value for a cell. Supersedes **SetCell**.

**Syntax**

```csharp
<HsvData>.SetCellExtDim (pIUnkHfmPovCOM, dData, vbIsNoData)
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell POV</td>
</tr>
<tr>
<td>dData</td>
<td>Double</td>
<td>Data value</td>
</tr>
<tr>
<td>vbIsNoData</td>
<td>Boolean</td>
<td>True = set cell to NODATA. Otherwise, value is set to dData.</td>
</tr>
</tbody>
</table>

**Return Value**

Integer. 1=success, 0=failure.

**Example**

See the HfmPovCom documentation on how to specify a cell POV.

### SetCellDescriptions

*Deprecated* - use SetCellDescriptionsExtDim.

### SetCellDescriptionsExtDim

Set cell text for multiple cells. Supersedes SetCellDescriptions.

#### Syntax

```
< HvData >.SetCellDescriptionsExtDim plUnkHfmSliceCOM, varabstrCellText, varlCustom3, varlCustom4, varbstrCellText
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object specifying the desired cells</td>
</tr>
<tr>
<td>varabstrCellText</td>
<td>Stringarray</td>
<td>The cell text to be written to the cells</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See the HfmSliceCOM documentation for how to specify multiple POVs.

### SetCellLineItems

*Deprecated* - use SetCellLineItemsExtDim.
**SetCellLineItemsExtDim**

Appends or updates line items for the specified cell. Supersedes SetCellLineItems.

**Syntax**

```
<HsvData>.SetCellLineItemsExtDim pIUnkHfmPovCOM, vardData, varbstrDetails
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell’s pov</td>
</tr>
<tr>
<td>vardData</td>
<td>Double array</td>
<td>Array of line item detail data</td>
</tr>
<tr>
<td>varbstrDetails</td>
<td>String Array</td>
<td>Array of line item detail text</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

**SetCells**

*Deprecated* - use **SetCellsExtDim**.

**SetCellsExtDim**

Set cell data for multiple cells. Supersedes SetCells.

**Syntax**

```
<HsvData>.SetCellsExiDim pIUnkHfmSliceCOM, varadData, varabIsNoData
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM Object representing the cells to update</td>
</tr>
<tr>
<td>varadData</td>
<td>Long array</td>
<td>Array of data values</td>
</tr>
<tr>
<td>varabIsNoData</td>
<td>Boolean array</td>
<td>Array of bool values indicating whether to set the data value to NODATA. If True, data is set to NODATA, otherwise data is set to value from varadData.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmSliceCOM documention on how to set multiple cell POVs.
**SetCells2**

*Deprecated* - use SetCells2ExtDim.

**SetCells2ExtDim**

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. Supersedes SetCells2.

**Syntax**

\[<\text{HsvData}>.\text{SetCells2ExtDim} \ p\text{UnkHfmSliceCOM}, \ \text{varadData}, \ \text{varabIsNoData}, \ \text{pvaralStatus} \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object that represents the cells you wish to update.</td>
</tr>
<tr>
<td>varadData</td>
<td>Double array</td>
<td>The data to be written</td>
</tr>
<tr>
<td>varabIsNoData</td>
<td>Boolean array</td>
<td>Determines whether the cell should be set to NODATA</td>
</tr>
<tr>
<td>pvaralStatus</td>
<td>Long array</td>
<td>Returns the cells’ statuses</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See the HfmSliceCOM documentation on how to specify multiple cell POVs.

**SetCellsLineItems**

*Deprecated* - use SetCellsLineItemsExtDim.

**SetCellsLineItemsExtDim**

Appends or updates line items for the specified cells. Supersedes SetCellsLineItems.

**Syntax**

\[<\text{HsvData}>.\text{SetCellsLineItemsExtDim} \ p\text{UnkHfmSliceCOM}, \ \text{varadData}, \ \text{varabstrDetail} \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pUnkHfmSliceCOM</td>
<td>HfmSliceCOM</td>
<td>HfmSliceCOM object specifying the the povs for multiple cells.</td>
</tr>
<tr>
<td>varadData</td>
<td>Double array</td>
<td>Array of data values for the line items</td>
</tr>
<tr>
<td>Argument</td>
<td>Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>varabstrDetail</td>
<td>String array</td>
<td>Array of line item detail text</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
See HfmSliceCOM documentation on how to set the povs.

### SetFileForLoad

*For internal use.*

### SetLineItems

*Deprecated - use SetCellLineItemsExtDim.*

### SetLineItemsExtDim

Inserts a cell’s line items into an HsvMDDataBuffer object. Supersedes SetLineItems.

**Syntax**

```
>HsvMDDataBuffer>.SetLineItemsExtDim pIUnkHfmPovCOM, vbSavedAsYTD, varadData, varabstrDescriptions
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object that represents the POV</td>
</tr>
<tr>
<td>vbSavedAsYTD</td>
<td>Boolean</td>
<td>Determines how the cell’s totals are saved. Pass TRUE if the cell is saved as year-to-date, FALSE if saved as periodic.</td>
</tr>
<tr>
<td>varadData</td>
<td>Double array</td>
<td>The cell’s line item data</td>
</tr>
<tr>
<td>varabstrDescriptions</td>
<td>String array</td>
<td>The cell’s line item descriptions</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
See HfmPovCOM documentation on how to set the POV.
**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**

```plaintext
<HsvData>.SetPhaseSubmissionGridForGivenScenarioPeriod lScenario, varalPeriodIDs, varalPhaseIDs, pvarbstrGroupInfo
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lScenario</strong></td>
</tr>
<tr>
<td><strong>varalPeriodIDs</strong></td>
</tr>
<tr>
<td><strong>varalPhaseIDs</strong></td>
</tr>
<tr>
<td><strong>pvarbstrGroupInfo</strong></td>
</tr>
</tbody>
</table>

**Example**

For the specified scenario and periods, the following method assigns all submission groups to Phase 1.

```plaintext
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**

*Deprecated - use SetTextCellExtDim.*

**SetTextCellExtDim**

Sets a cell’s data, using a string to represent the cell value. Supersedes SetTextCell.
Syntax

```<HsvData>.SetTextCellExtDim pIUnkHfmPovCOM, sScale, bstrData```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell's POV</td>
</tr>
<tr>
<td><code>sScale</code></td>
<td>Integer</td>
<td>Indicates the degree of scaling to apply to the <code>bstrData</code> return value</td>
</tr>
<tr>
<td><code>bstrData</code></td>
<td>String</td>
<td>The cell value</td>
</tr>
</tbody>
</table>

Return Value

Integer. 1=success; 0=failure.

Example

See the HfmPovCOM documentation on how to set the POV.

SetTextCellLineItems

*Deprecated* - use `SetTextCellLineItemsExtDim`.

SetTextCellLineItemsExtDim

Appends or updates line items for the specified cell. Line item data is passed in a String array. Supersedes SetTextCellLineItems.

Syntax

```<HsvData>.SetTextCellLineItemsExtDim pIUnkHfmPovCOM, varabstrData, varabstrDetails```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM</td>
<td>HfmPovCOM object representing the cell's pov</td>
</tr>
<tr>
<td><code>varabstrData</code></td>
<td>String array</td>
<td>Array of line item data values</td>
</tr>
<tr>
<td><code>varabstrDetails</code></td>
<td>String Array</td>
<td>Array of line item detail text</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

SetTextCells

*Deprecated* - use `SetTextCellsExtDim`.
**SetTextCellsExtDim**

Set cell data for multiple cells, using strings to represent the data values.

**Syntax**

```
<HsvData>.MethodName pIUnkHfmSliceCOM, varabstrData, short sScale
```

**Argument**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmSliceCOM object representing multiple cell POVs</td>
</tr>
<tr>
<td>varabstrData</td>
<td>Array of data values</td>
</tr>
<tr>
<td>sScale</td>
<td>Indicates the degree of scaling to apply to the data values.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See the HfmSliceCOM documentation for how to set multiple POVs.

**SetTextCellsLineItems**

*Deprecated* - use **SetTextCellsLineItemsExtDim**.

**SetTextCellsLineItemsExtDim**

Appends or updates line items for the specified cell. Data is specified as strings. Supersedes SetTextCellsLineItems.

**Syntax**

```
<HsvData>.SetTextCellsLineItemsExtDim pIUnkHfmSliceCOM, varabstrData, varabstrDetail
```

**Argument**

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmSliceCOM object representing the povs for the desired cells</td>
</tr>
<tr>
<td>varabstrData</td>
<td>Array of data values for each lineitem</td>
</tr>
<tr>
<td>varabstrDetail</td>
<td>Array of line item details</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmSliceCOM documentation on specifying multiple POVs.
**SetTextLineItems**

*Deprecated* – use **SetTextCellLineItems**.

**StartLoad**

*For internal use.*

**StartLoadTask**

*For internal use.*

**UpdateDataUsingMDDataBuffer**

Inserts an HsvMDDataBuffer or HsvMDDataBufferLite object’s cells into the corresponding cells of an application.

**Syntax**

```c
<HsvData>.UpdateDataUsingMDDataBuffer pIUnkDataBuffer, lEnumUpdateMode, bAccumulateWithinBuffer
```

**Argument** | **Description**
--- | ---
`pIUnkDataBuffer` | HsvMDDataBuffer or HsvMDDataBufferLite object (ByVal). The object reference for the object.
`lEnumUpdateMode` | Long (ByVal). Determines the update mode. See "Update Mode Constants" on page 320.
`bAccumulateWithinBuffer` | 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.

*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.

```c
m_cData.UpdateDataUsingMDDataBuffer cMDBuffer, _
HSV_DATA_UPDATE_ACCUMULATE, True
```

**ValidateMembersAndConvertToIDs**

*For internal use.*
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 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 85, 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 85.

**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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member for which the allocation is to be run.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent entity of the lEntity argument’s entity.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the Value dimension member.</td>
</tr>
</tbody>
</table>

**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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lStartPeriod</td>
<td>Long (ByVal). The member ID of the first Period dimension member in the range of periods to be allocated.</td>
</tr>
<tr>
<td>lEndPeriod</td>
<td>Long (ByVal). The member ID of the last Period dimension member in the range of periods to be allocated.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member for which the allocation is to be run.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent entity of the lEntity argument’s entity.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the Value dimension member.</td>
</tr>
</tbody>
</table>

**CalcEPU**

Runs the Equity Pickup for the specified Scenario, Year, and Period.

**Syntax**

`<HsvCalculate>.CalcEPU lScenario, lYear, lPeriod, varbForceEPU`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
</tbody>
</table>

`varbForceEPU` 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`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member for which the calculation is to be run.</td>
</tr>
</tbody>
</table>
**Argument**  
**Description**  

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent entity of the lEntity argument's entity.</td>
</tr>
<tr>
<td>Value</td>
<td>Long (ByVal). The member ID of the Value dimension member.</td>
</tr>
</tbody>
</table>

**varbForceChartLogic**  
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.

```vba
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, "2012")  
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**  

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lStartPeriod</td>
<td>Long (ByVal). The member ID of the first Period dimension member in the range of periods to be calculated.</td>
</tr>
<tr>
<td>lEndPeriod</td>
<td>Long (ByVal). The member ID of the last Period dimension member in the range of periods to be calculated.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member for which the calculation is to be run.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent entity of the lEntity argument's entity.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the Value dimension member.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>varbForceChartLogic</td>
<td>Boolean (ByVal). Specifies whether the calculation should be forced if the entity does not require calculation. Specify TRUE to force the calculation, otherwise FALSE.</td>
</tr>
</tbody>
</table>

**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.

```vbescript
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, "2012")
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**

```vbescript
<HsvCalculate>.Consolidate lScenario, lYear, lPeriod, lEntity, lParent, sConsolidationType
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member for which the consolidation is to be run.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent entity of the lEntity argument's entity.</td>
</tr>
<tr>
<td>sConsolidationType</td>
<td>Integer (ByVal). Identifies the type of consolidation to be run. Pass one of the HFMConstants type library constants listed in &quot;Consolidation Type Constants&quot; on page 880.</td>
</tr>
</tbody>
</table>
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, "2012")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lPar = GetMemberID(DIMENSIONENTITY, "Regional")
Set cCalculate = m_cHsvSession.Calculate

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, "2012")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lEnt = GetMemberID(DIMENSIONENTITY, "UnitedStates")
lPar = GetMemberID(DIMENSIONENTITY, "Regional")
Set cCalculate = m_cHsvSession.Calculate

Syntax

\(<HsvCalculate>.Consolidate2 lScenario, lYear, lStartPeriod, lEndPeriod, lEntity, lParent, sConsolidationType\)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lStartPeriod</td>
<td>Long (ByVal). The member ID of the first Period dimension member in the range of periods to be consolidated.</td>
</tr>
<tr>
<td>lEndPeriod</td>
<td>Long (ByVal). The member ID of the last Period dimension member in the range of periods to be consolidated.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member for which the consolidation is to be run.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent entity of the lEntity argument's entity.</td>
</tr>
<tr>
<td>sConsolidationType</td>
<td>Integer (ByVal). Identifies the type of consolidation to be run. Pass one of the HFMConstants type library constants listed in &quot;Consolidation Type Constants&quot; on page 880.</td>
</tr>
</tbody>
</table>

Tip: You can avoid executing redundant consolidations by calling FindOverlappingConsolidation before launching a consolidation.
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, "2012")
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.*

**EnumOnDemandRules**

Returns the labels for OnDemand rules.

**Syntax**

<
HsvCalculate>
.
EnumOnDemandRules
 pvarabstrOnDemandRules

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvarabstrOnDemandRules</td>
<td>String. Returns the labels of the OnDemand rules.</td>
</tr>
</tbody>
</table>

**ExecuteOnDemandRules**

Executes OnDemand rules.

**Syntax**

<
HsvCalculate>
.
ExecuteOnDemandRules
 pIUnkHfmPovCOM, bstrOnDemandRule

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the desired POV.</td>
</tr>
<tr>
<td>bstrOnDemandRule</td>
<td>String (ByVal). The label of the rule that you want to execute.</td>
</tr>
</tbody>
</table>
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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the lEntity argument's parent.</td>
</tr>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lStartPeriod</td>
<td>Long (ByVal). The member ID of the first Period dimension member in the range of periods.</td>
</tr>
<tr>
<td>lEndPeriod</td>
<td>Long (ByVal). The member ID of the last Period dimension member in the range of periods.</td>
</tr>
<tr>
<td>lConsolType</td>
<td>Long (ByVal). Identifies the type of consolidation to test for. Pass one of the constants in the HFMConstants type library enumeration tagCONSOLIDATIONTYPE, which is described in &quot;Consolidation Type Constants&quot; on page 880.</td>
</tr>
<tr>
<td>pbstrYear</td>
<td>String. Returns the label of the Year dimension member for the consolidations, or a blank string if no matching consolidations are found.</td>
</tr>
<tr>
<td>pbstrScenario</td>
<td>String. Returns the label of the Scenario dimension member for the consolidations, or a blank string if no matching consolidations are found.</td>
</tr>
<tr>
<td>pvarabstrEntity</td>
<td>Variant. Returns an array containing the labels of the consolidations' Entity dimension members, or an empty variant if no matching consolidations are found.</td>
</tr>
<tr>
<td>pvarabstrParent</td>
<td>Variant. Returns an array containing the parents of the entities returned by the pvarabstrEntity argument.</td>
</tr>
<tr>
<td>pvarabstrStartPeriod</td>
<td>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.</td>
</tr>
<tr>
<td>pvarabstrEndPeriod</td>
<td>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.</td>
</tr>
</tbody>
</table>
Argument | Description
---|---
pvaralConsolType | 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 HFMConstants type library enumeration tagCONSORTIATIONTYPE, which is described in “Consolidation Type Constants” on page 880.

plTotalOverlapRecords | 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

*Deprecated - use “GetDefaultExchangeRateExtDim” on page 399.*

GetDefaultExchangeRateExtDim

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; see the *Oracle Hyperion Financial Management User's Guide*. Supersedes GetDefaultExchangeRate.

Syntax

```<HsvCalculate>.GetDefaultExchangeRateExtDim plUnkHfmPovCOM, lFromCurrencyId, lToCurrencyId, pdRate```

Argument | Description
---|---
plUnkHfmPovCOM | HfmPovCOM. HfmPovCOM object representing the desired POV.

lFromCurrencyId | Long (ByVal). The currency ID of the source currency.

**Tip:** You can get a currency's ID from the HsvCurrencies method `GetCurrencyID`.

lToCurrencyId | Long (ByVal). The currency ID of the destination currency.

dRate | Double. Returns the exchange rate of the currencies.

Return Value

None.
Example
See HfmPovCOM documentation on how to set the POV.

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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>varalOwners</td>
<td>Variant array of Long (ByVal). Input filtered list of Owner entities. The return list only includes the input owners.</td>
</tr>
<tr>
<td>varalOwned</td>
<td>Variant array of Long (ByVal). Input filtered list of owned entities. The return list only includes the input owned entities.</td>
</tr>
<tr>
<td>lStatus</td>
<td>Long (ByVal). The filtered EPU Status. Only ownership pairs with selected status are returned.</td>
</tr>
<tr>
<td>lPctEPUOp</td>
<td>Long (ByVal). The operator to be used with the dPctEPU value.</td>
</tr>
<tr>
<td>dPctEPU</td>
<td>Double (ByVal). The filtered percentage ownership value. Only ownership pairs with percentage ownership value match the operator condition are returned.</td>
</tr>
<tr>
<td>lCirOwn</td>
<td>Long (ByVal). The filtered circular ownership status.</td>
</tr>
<tr>
<td>pvara2DvEPUInfo</td>
<td>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).</td>
</tr>
</tbody>
</table>

GetEPUInfoEx

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>.GetEPUInfoEx lScenario, lYear, lPeriod, varalOwners, varalOwned, lMinFilterLevel, lMaxFilterLevel, lStatus, lPctEPUOp, dPctEPU, lCirOwn, vara2DvEPUInfo, lMaxLevel```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>varalOwners</td>
<td>Variant array of Long (ByVal). Input filtered list of Owner entities. The return list only includes the input owners.</td>
</tr>
<tr>
<td>varalOwned</td>
<td>Variant array of Long (ByVal). Input filtered list of owned entities. The return list only includes the input owned entities.</td>
</tr>
<tr>
<td>lMinFilterLevel</td>
<td>Long (ByVal). The minimum filter level.</td>
</tr>
<tr>
<td>lMaxFilterLevel</td>
<td>Long (ByVal). The maximum filter level.</td>
</tr>
<tr>
<td>lStatus</td>
<td>Long (ByVal). The filtered EPU Status. Only ownership pairs with selected status are returned.</td>
</tr>
<tr>
<td>lPctEPUOp</td>
<td>Long (ByVal). The operator to be used with the dPctEPU value.</td>
</tr>
<tr>
<td>dPctEPU</td>
<td>Double (ByVal). The filtered percentage ownership value. Only ownership pairs with percentage ownership value match the operator condition are returned.</td>
</tr>
<tr>
<td>lCirOwn</td>
<td>Long (ByVal). The filtered circular ownership status.</td>
</tr>
<tr>
<td>vara2DvEPUInfo</td>
<td>Variant array (ByRef). Returns two-dimensional array. The values are returned as Owner (Long), Owned (Long), Level (Long), Percentage Ownership (Double), Status (Long), Circular Ownership flag (Long).</td>
</tr>
<tr>
<td>lMaxLevel</td>
<td>Long (ByRef). The maximum level.</td>
</tr>
</tbody>
</table>

**GetVBScriptRules**

Returns the rules 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) loaded into an application. If the rules file changed after the file was loaded, the updated file’s contents are not returned by *GetVBScriptRules*.

**Syntax**

```
<HsvCalculate>.GetVBScriptRules pvarabRules, pbRulesExist
```

**Argument**  **Description**

<table>
<thead>
<tr>
<th>pvarabRules</th>
<th>Variant array. Returns the application’s rules. The array is returned as a Byte subtype.</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbRulesExist</td>
<td>Boolean. Returns TRUE if a rules file was loaded into the application, FALSE if no rules file was loaded.</td>
</tr>
</tbody>
</table>

**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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member for which the query is to be run.</td>
</tr>
<tr>
<td>pvbEntityIsAnEPUOwner</td>
<td>Boolean. Returns True if specified Entity is an EPU owner, False otherwise.</td>
</tr>
</tbody>
</table>

**LoadCalcManagerRules**

*For internal use.*

**LoadCalcManagerRules2**

*For internal use.*

**SetCOMDLLRules**

*For internal use.*

**SetVBScriptRules**

Loads or scans a rules file.
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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varabRules</td>
<td>Variant (ByVal). The rules file, passed as an array of bytes.</td>
</tr>
</tbody>
</table>
| vbScanOnly                | 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. |
| pvbErrorsWereFound        | Boolean. Indicates whether `SetVBScriptRules` found any validation errors. Returns TRUE if errors were found, FALSE otherwise. |
| pvbWarningsWereFound      | Boolean. Indicates whether `SetVBScriptRules` found any validation warnings. Returns TRUE if warnings were found, FALSE otherwise. |
| pvbInfoWasProvided        | Boolean. Indicates whether `SetVBScriptRules` returns any information not classified as errors or warnings. Returns TRUE if non-error and non-warning information is returned, FALSE otherwise. |
| pvaralErrorLineNumbers    | Variant array. Returns the line numbers to which the errors, warnings, and information apply. The array is returned as a Long subtype. |
| pvaralErrorSeverity       | Variant array. Returns numbers that indicate the severity levels of the errors, warnings, and information. Valid return values are described in the following list:  
                                      • 1 = Error severity level.  
                                      • 2 = Warning severity level.  
                                      • 3 = Information severity level.  
                                      The array is returned as a Long subtype. |
| pvarabstrErrorDescriptions| Variant array. Returns descriptions of the errors, warnings, and information. The array is returned as a String subtype. |
| pvarabstrErrorVBScript     | Variant array. Returns the rules file statements to which the errors, warnings, and information apply. The array is returned as a String subtype. |
| pvarabstrErrorDetails     | 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; 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
        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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>varabRules</code></td>
<td>Variant (ByVal). The rules file, passed as an array of bytes.</td>
</tr>
<tr>
<td><code>vbScanOnly</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>vbCheckRefIntegrity</code></td>
<td>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 <code>pbstrRefIntegDiagXML</code> argument returns information regarding any intercompany transactions that are rendered invalid.</td>
</tr>
<tr>
<td><code>pvbErrorsWereFound</code></td>
<td>Boolean. Indicates whether <code>SetVBScriptRules2</code> found any validation errors. Returns TRUE if errors were found, FALSE otherwise.</td>
</tr>
<tr>
<td><code>pvbWarningsWereFound</code></td>
<td>Boolean. Indicates whether <code>SetVBScriptRules2</code> found any validation warnings. Returns TRUE if warnings were found, FALSE otherwise.</td>
</tr>
<tr>
<td><code>pvbInfoWasProvided</code></td>
<td>Boolean. Indicates whether <code>SetVBScriptRules2</code> returns any information not classified as errors or warnings. Returns TRUE if non-error and non-warning information is returned, FALSE otherwise.</td>
</tr>
<tr>
<td><code>pvaralErrorLineNumbers</code></td>
<td>Variant array. Returns the line numbers to which the errors, warnings, and information apply. The array is returned as a Long subtype.</td>
</tr>
</tbody>
</table>
| `pvaralErrorSeverity`     | Variant array. Returns numbers that indicate the severity levels of the errors, warnings, and information. Valid return values are described in the following list:
  - 1 = Error severity level.
  - 2 = Warning severity level.
  - 3 = Information severity level.
  The array is returned as a Long subtype. |
| `pvarabstrErrorDescriptions` | Variant array. Returns descriptions of the errors, warnings, and information. The array is returned as a String subtype. |
| `pvarabstrErrorVBScript`  | Variant array. Returns the rules file statements to which the errors, warnings, and information apply. The array is returned as a String subtype. |
| `pvarabstrErrorDetails`   | Variant array. Returns details that describe why the lines caused validation to fail. The array is returned as a String subtype. |
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**

```xml
<HsvCalculate>.Translate lScenario, lYear, lPeriod, lEntity, lParent, lValue, varbForceTranslate, bTranslateAll
```

**Argument**

**Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member for which the consolidation is to be run.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the parent entity of the lEntity argument’s entity.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the Value dimension member for the currency into which the amount is to be translated.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>varbForceTranslate</td>
<td>Boolean (ByVal). Determines whether translations are forced. Specify TRUE to force translations, otherwise FALSE.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> Specifying TRUE is similar to the Force Translate command.</td>
</tr>
<tr>
<td>bTranslateAll</td>
<td>Boolean (ByVal). This argument has no effect in the current release. You must pass a value, but the system ignores the value.</td>
</tr>
</tbody>
</table>

### Example

This example translates the Italy entity’s amount to the USD currency. The dimension member labels are passed to the user-defined `GetMemberID` function, which returns their member IDs. These member IDs are then passed to `Translate`.

```vba
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, "2012")
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

```vba
<HsvCalculate>.Translate2 lScenario, lYear, lStartPeriod, lEndPeriod, lEntity, lParent, lValue, varbForceTranslate
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lStartPeriod</td>
<td>Long (ByVal). The member ID of the first Period dimension member in the range of periods to be translated.</td>
</tr>
<tr>
<td>lEndPeriod</td>
<td>Long (ByVal). The member ID of the last Period dimension member in the range of periods to be translated.</td>
</tr>
</tbody>
</table>
### Argument Description

- **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 are forced. Specify TRUE to force translations, otherwise FALSE. **Tip**: Specifying TRUE is similar to Force Translate.

### Example

This example translates the Italy entity’s amount to the USD currency. The dimension member labels are passed to the user-defined GetMemberID function, which returns their member IDs. These member IDs are then passed to Translate2.

```vba
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, "2012")
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
calculate.Translate2 lScen, lYear, lStartPer, lEndPer, lEnt, _
lPar, lVal, False
```
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 86.

**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 873.

<table>
<thead>
<tr>
<th>Column</th>
<th>Return Value (Subtype and Description)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>String, The journal label.</td>
</tr>
<tr>
<td>Status</td>
<td>Integer, The journal status. The valid return values are represented by the HFMConstants type library constants listed in “Journal Status Constants” on page 875.</td>
</tr>
<tr>
<td>Type</td>
<td>Integer, The journal type. The valid return values are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 876.</td>
</tr>
</tbody>
</table>
### Columns and Return Values

<table>
<thead>
<tr>
<th>Column</th>
<th>Return Value (Subtype and Description)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Balance Type</td>
<td>Integer. The balance type. The valid return values are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 873.</td>
</tr>
<tr>
<td>Group</td>
<td>String. The name of the journal group.</td>
</tr>
<tr>
<td>Description</td>
<td>String. The journal description.</td>
</tr>
<tr>
<td>Short Description</td>
<td>String. The truncated version of the journal description.</td>
</tr>
<tr>
<td>Created By</td>
<td>String. The fully qualified username of the journal creator.</td>
</tr>
<tr>
<td>Date Created</td>
<td>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.</td>
</tr>
<tr>
<td>Approved By</td>
<td>String. The fully qualified username of the journal approver. If a journal has not been approved, a blank String is returned.</td>
</tr>
<tr>
<td>Approved On</td>
<td>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.</td>
</tr>
<tr>
<td>Posted By</td>
<td>String. The fully qualified username of the journal poster. If a journal has not been posted, a blank String is returned.</td>
</tr>
<tr>
<td>Date Posted</td>
<td>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.</td>
</tr>
<tr>
<td>Security Class</td>
<td>Long. The ID number of the journal’s security class.</td>
</tr>
<tr>
<td>Entity</td>
<td>Long. The member ID of the journal’s entity.</td>
</tr>
<tr>
<td>Parent</td>
<td>Long. The member ID of the journal’s parent entity.</td>
</tr>
</tbody>
</table>

### 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 876.

<table>
<thead>
<tr>
<th>Column</th>
<th>Return Value (Subtype and Description)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label</td>
<td>String. The template label.</td>
</tr>
</tbody>
</table>
### 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 87, 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 192.

### 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 197.

**Syntax**

```xml
<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 2012. 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.

```vba
Dim lScen As Long, lYear As Long, lPer As Long
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2012")
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**

```vba
<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**

```vba
<HsvJournals>.GetJournalTemplateItemID(bstrLabel)
```
**GetPeriodStatusList**

Returns a two-dimensional array that indicates whether the periods for a scenario and year are opened, unopened, or closed.

**Syntax**

```vba
<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 874.

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.

```vba
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, "2012")
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 414.

**Syntax**

```csharp
<HsvJournals>.IsPeriodOpen lCategory, lYear, lPeriod, pnRc
```

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lCategory</code></td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td><code>lPeriod</code></td>
<td>Long (ByVal). The member ID of the Period dimension member to be checked.</td>
</tr>
<tr>
<td><code>pnRc</code></td>
<td>Integer. Returns -1 is returned if the period is open, or 0 if the period is unopened or closed.</td>
</tr>
</tbody>
</table>

**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**

```csharp
<HsvJournals>.OpenPeriod lCategory, lYear, lPeriod
```

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lCategory</code></td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td><code>lPeriod</code></td>
<td>Long (ByVal). The member ID of the Period dimension member to be opened.</td>
</tr>
</tbody>
</table>

**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. 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 88, 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 192.

### AddJournalGroup

Creates a journal group.

**Syntax**

```
<IHsvJournalsEx>.AddJournalGroup bstrGroup, bstrDescription
```

**Argument** | **Description**
--- | ---
`bstrGroup` | String (ByVal). The name of the journal group.
`bstrDescription` | String (ByVal). The description of the journal group.

### ApproveJournals

Approves one or more journals. The *vararlJournalIDs* 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 40D (hexadecimal) occurs.

**Syntax**

```
<IHsvJournalsEx>.ApproveJournals lScenario, lYear, vararlJournalIDs, pvararnRc
```
### ApproveJournals

Approves one or more journals. The member IDs for the Scenario and Year dimension are passed along with an array of journal IDs to the function. The function returns an array of codes that identify the success or failure of the journal approvals.

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>vararlJournalIDs</td>
<td>Long array (ByVal). The IDs of the journals to be approved. Get these IDs with the HsvJournals object's GetItemID method; see &quot;GetItemID&quot; on page 412.</td>
</tr>
<tr>
<td>pvararnRc</td>
<td>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 vararlJournalIDs argument. Successfully approved journals return 0. The array is returned as an Integer subtype.</td>
</tr>
</tbody>
</table>

**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.

```vba
Dim lScen As Long, lYear As Long, lPer As Long
Dim laJnlIDs(1) As Long, vaRetVal
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2012")
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 vararlJournalIDs 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 40D (hexadecimal) occurs.

**Syntax**

```vba```
< IHsvJournalsEx >.DeleteJournals lScenario, lYear, vararlJournalIDs, pvararnRc
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
</tbody>
</table>
vararrJournalIDs  Long array (ByVal). The IDs of the journals to be deleted. Get these IDs with the HsvJournals object’s GetItemID method; see “GetItemID” on page 412.

**Caution!** You cannot delete posted or approved journals.

pvararnRc  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 vararrJournalIDs argument. Successfully deleted journals return zero.

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 was 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, "2012")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
laJnlIDs(0) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl392")
lJnlIDs(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 40D (hexadecimal) occurs.

**Syntax**

```
<IHsvJournalsEx>.DeleteTemplates vararrTemplateIDs, pvararnRc
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vararrTemplateIDs</td>
<td>Long array (ByVal). The IDs of the templates to be deleted. Use GetJournalTemplateItemID to get template IDs; see “GetJournalTemplateItemID” on page 412.</td>
</tr>
</tbody>
</table>
**Argument** | **Description**
---|---
`pvararnRc` | 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 `varTemplateIDs` 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`.

```vbnet
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**
---|---
`pvarabstrGroups` | Variant array. Returns the names of the journal groups. The array is returned as a String subtype.

`pvarabstrDescriptions` | Variant array. Returns the descriptions of the journal groups. The array is returned as a String subtype.

**EnumJournalGroupsEx**

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>.EnumJournalGroupsEx varabstrGroups, varabstrDescriptions, varadModifiedDate, varabstrModifiedBy
```

**Argument** | **Description**
---|---
`varabstrGroups` | Variant array (ByRef). Returns the names of the journal groups. The array is returned as a String subtype.
Argument | Description
---|---
varabstrDescriptions | Variant array(ByRef). Returns the descriptions of the journal groups.
  The array is returned as a String subtype.
varadModifiedDate | Variant array (ByRef). Returns the date that the journal group was modified.
varabstrModifiedBy | Variant array (ByRef). Returns the user name of the user who modified the journal group.

**EnumJournalGroupsForScenarioYear**

Returns the names and descriptions of the journal groups assigned to journals for a specified scenario and year. The information is returned in arrays that have a one-to-one correspondence.

**Syntax**

```csharp
<IHsvJournalsEx>.EnumJournalGroupsForScenarioYear lScenario, lYear, pvarabstrGroups, pvarabstrDescriptions
```

**Argument** | **Description**
---|---
lScenario | Long (ByVal). The member ID of the Scenario dimension member.
lYear | Long (ByVal). The member ID of the Year dimension member.
pvarabstrGroups | Variant array. Returns the names of the journal groups.
  The array is returned as a String subtype.
pvarabstrDescriptions | 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 lJournalTypes 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 lScenario and lYear arguments, for which valid member IDs are required.

**Syntax**

```csharp
<IHsvJournalsEx>.EnumJournalIDsForExtractFilter lScenario, lYear, varalPeriods, varalEntities, varalParents, varalValues, lJournalTypes, varabstrLabels, varabstrGroups, varalStatus, varalAutoTypes, varalBalanceTypes, pvaralJournalIDs, pvaralTemplateIDs
```
### GenerateRecurring
Generates a journal from a recurring template.

**Syntax**

```csharp
< IHsvJournalsEx >.GenerateRecurring lScenario, lYear, lPeriod, lTemplateID
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lTemplateID</td>
<td>Long (ByVal). The member ID of the Template dimension member.</td>
</tr>
<tr>
<td>varalPeriods</td>
<td>Long array (ByVal). The member IDs of the Period dimension members by which to filter.</td>
</tr>
<tr>
<td>varalEntities</td>
<td>Long array (ByVal). The member IDs of the Entity dimension members by which to filter.</td>
</tr>
<tr>
<td>varalParents</td>
<td>Long array (ByVal). The member IDs of the parents of the Entity dimension members passed to the varalEntities argument.</td>
</tr>
<tr>
<td>varalValues</td>
<td>Long array (ByVal). The member IDs of the Value dimension members by which to filter.</td>
</tr>
<tr>
<td>lJournalTypes</td>
<td>Long (ByVal). A flag that determines whether journals, standard templates, recurring templates, or some combination thereof are returned. Valid values are represented by the HFMConstants type library constants listed in &quot;Journal and Template Type Constants&quot; on page 877. To filter with more than one of these items, use or with the applicable constants.</td>
</tr>
<tr>
<td>varabstrLabels</td>
<td>String array (ByVal). The journal labels by which to filter. You can use the percent sign (%) as a wildcard.</td>
</tr>
<tr>
<td>varabstrGroups</td>
<td>String array (ByVal). The journal groups by which to filter. You can use the percent sign (%) as a wildcard.</td>
</tr>
<tr>
<td>varalStatus</td>
<td>Long array (ByVal). The journal statuses by which to filter. Valid values are represented by the HFMConstants type library constants listed in &quot;Journal Status Constants&quot; on page 875.</td>
</tr>
<tr>
<td>varalAutoTypes</td>
<td>Long array (ByVal). The journal types by which to filter. Valid values are represented by the HFMConstants type library constants listed in &quot;Journal Type Constants&quot; on page 876.</td>
</tr>
<tr>
<td>varalBalanceTypes</td>
<td>Long array (ByVal). The balance types by which to filter. Valid values are represented by the HFMConstants type library constants listed in &quot;Balance Type Constants&quot; on page 873.</td>
</tr>
<tr>
<td>pvaralJournalIDs</td>
<td>Variant array. Returns the journal IDs that match the filtering criteria, if the lJournalTypes argument specifies that journals are to be returned. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralTemplateIDs</td>
<td>Variant array. Returns the template IDs that match the filtering criteria, if the lJournalTypes argument specifies that templates are to be returned. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lTemplateID</td>
<td>Long (ByVal). The ID of the journal template.</td>
</tr>
</tbody>
</table>

**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.

```vbnet
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**

*Deprecated* - use “GetEntityJournalsExtDim” on page 421.

**GetEntityJournalsExtDim**

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**

```vbnet
<IHsvJournalsEx>.GetEntityJournalsExtDim pIUnkHfmPovCOM, pvararnDebitCreditUnit, pvaradAmount, pvarabstrLabel, ppIUnkHfmSliceCOM, pvarabstrGroup
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the cell coordinates at the requested cell.</td>
</tr>
<tr>
<td>pvararnCreditDebitUnit</td>
<td>Variant array. Indicates whether the journal entries are debits or credits. Valid return values are represented by the HFMConstants type library constants listed in “Debit/Credit Constants” on page 874. The array is returned as an Long subtype.</td>
</tr>
<tr>
<td>pvaradAmount</td>
<td>Variant array. Returns the amounts of the journal entries. The array is returned as a Double subtype.</td>
</tr>
<tr>
<td>pvarabstrLabel</td>
<td>Variant array. Returns the labels of the journal entries. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>
ppIUnkHfmSliceCOM  HfmSliceCOM. HfmSliceCOM object containing multiple cells' coordinates.

pvarabstrGroup  Variant array. Returns the labels of the journal groups for the journal entries. The array is returned as a Long subtype.

**GetEntityJournals2**

*Deprecated* - use “GetEntityJournals2ExtDim” on page 422.

**GetEntityJournals2ExtDim**

This method supports the Web UI to identify journals through internal IDs rather than labels. This resolves several issues where Unicode text label-based lookups fail to return the correct journal due to differences in the way the Windows OS and the DBMS performed uppercase conversions of some Unicode text characters.

This method is a duplicate of GetEntityJournalsExtDim with the exception that it returns an array of journal IDs and labels.

**Syntax**

```<HsvJournalsEx>.GetEntityJournals2ExtDim pIUnkHfmPovCOM, pvararnDebitCreditUnit, pvaradAmount, pvarabstrLabel, pvaralJournalIDs, ppIUnkHfmSliceCOM, pvarabstrGroup```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the cell coordinates at the requested cell.</td>
</tr>
<tr>
<td>pvararnDebitCreditUnit</td>
<td>Variant array. Indicates whether the journal entries are debits or credits. Valid return values are represented by the HFMConstants type library constants listed in “Debit/Credit Constants” on page 874. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaradAmount</td>
<td>Variant array. Returns the amounts of the journal entries. The array is returned as a Double subtype.</td>
</tr>
<tr>
<td>pvarabstrLabel</td>
<td>Variant array. Returns the labels of the journal entries. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralJournalIDs</td>
<td>Variant array. Returns the IDs of the journal entries. The array is returned as a String subtype. This is an array of integers where each number represents the internal identifier of a journal.</td>
</tr>
<tr>
<td>ppIUnkHfmSliceCOM</td>
<td>HfmSliceCOM. HfmSliceCOM object containing multiple cells' coordinates.</td>
</tr>
<tr>
<td>pvarabstrGroup</td>
<td>Variant array. Returns the labels of the journal groups for the journal entries. The array is returned as a Long subtype.</td>
</tr>
</tbody>
</table>
GetJournal

Deprecated - use “GetJournalExtDim” on page 423.

GetJournalExtDim

Returns a variety of information for a journal.

Caution! To successfully call GetJournalExtDim, 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 GetTextJournalExtDim. It is almost identical to GetJournal; the only difference is the subtype of the line item amount array.

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>.GetJournalExtDim lScenario, lYear, lJournalID, pnType, pnStatus, pnAttribute, pbstrLabel, pbstrDescription, pbstrGroup, pIISingleEntity, pIISingleParent, plSecurityClass, pvararlEntryID, pvararnDebitCreditUnit, pvaradAmount, pvararbstrDescription, ppIUnkHfmSliceCOM```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lJournalID</td>
<td>Long (ByVal). The ID of the journal. Get this ID with HsvJournals.GetItemID; see “GetItemID” on page 412.</td>
</tr>
<tr>
<td>pnType</td>
<td>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 876.</td>
</tr>
<tr>
<td>pnStatus</td>
<td>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 875.</td>
</tr>
<tr>
<td>pnAttribute</td>
<td>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 873.</td>
</tr>
<tr>
<td>pbstrLabel</td>
<td>String. Returns the label of the journal.</td>
</tr>
<tr>
<td>pbstrDescription</td>
<td>String. Returns the description of the journal.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pbstrGroup</td>
<td>String. Returns the description of the journal group to which the journal is assigned.</td>
</tr>
<tr>
<td>plSingleEntity</td>
<td>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.</td>
</tr>
<tr>
<td>plSingleParent</td>
<td>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.</td>
</tr>
<tr>
<td>plSecurityClass</td>
<td>Long. Returns the ID of the journal’s security class.</td>
</tr>
<tr>
<td>Tip:</td>
<td>To get the label of the security class, pass this ID to HsvSecurityAccess.GetSecurityClassLabel. See “GetSecurityClassLabel” on page 483.</td>
</tr>
<tr>
<td>pvararlEntryID</td>
<td>Variant array. For internal use.</td>
</tr>
<tr>
<td>pvararrayDebitCreditUnit</td>
<td>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 874.</td>
</tr>
<tr>
<td>Tip:</td>
<td>Use GetVariance to return the total debit and credit amounts and the difference between these amounts. See “GetVariance” on page 437.</td>
</tr>
<tr>
<td>pvaradAmount</td>
<td>Variant array. Returns the amounts for the journal’s line items. The array is returned as a Double subtype.</td>
</tr>
<tr>
<td>pvararstrDescription</td>
<td>Variant array. Returns the descriptions for the journal’s line items. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pplUnkHfmSliceCOM</td>
<td>HfmSliceCOM. HfmSliceCOM object containing multiple cells’ coordinates.</td>
</tr>
</tbody>
</table>

**Example**

The following example shows how to call GetJournalExtDim; after calling GetJournalExtDim, 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 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 GetJournalExtDim.

```vba
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, "2012")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
lJnlID = m_cHsvJournals.GetItemID(lScen, lYear, lPer, "Jnl993")
m_cIHsvJournalEx.GetJournal lScen, lYear, lJnlID, lJnlPer, _
lVal, iType, iStatus, iAttr, sLabel, sDescHead, sGroup, _
```

424  HsvJournals Type Library
GetJournal2

Deprecated - use “GetJournal2ExtDim” on page 425.

GetJournal2ExtDim

Returns a variety of information for a journal. This is the same as GetJournalExtDim 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 GetJournal2ExtDim, 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 GetTextJournalExtDim. It is almost identical to GetJournal2ExtDim; the only difference is the subtype of the line item amount array.

The first three arguments specify the scenario, year, and journal IDs.

The fourth argument is the vbScaleAmounts flag. The call to GetJournalExtDim calls GetJournal2ExtDim 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 occurs.

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 pvararAccount arguments’ arrays return the amount and account for the journal’s second line item.

Syntax

<HsvJournalsEx>.GetJournal2ExtDim lScenario, lYear, lJournalID, vbScaleAmounts, pnType, pnStatus, pnAttribute, pbstrLabel, pbstrDescription, pbstrGroup, plSingleEntity, plSingleParent, plSecurityClass, pvararlEntryID, pvararnDebitCreditUnit, pvaradAmount, pvararbstrDescription, ppIUnkHfmSliceCOM

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lJournalID</td>
<td>Long (ByVal). The ID of the journal. Get this ID with HsvJournals.GetItemID; see “GetItemID” on page 412.</td>
</tr>
</tbody>
</table>
**Argument** | **Description**
---|---
*vbScaleAmounts* | Boolean (ByVal). Determines whether the textual representation of the amounts are scaled or not. Setting to TRUE scales the amounts based on the cell’s scale factor. Setting the flag to FALSE results in no scaling.

*pnType* | 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 876.

*pnStatus* | 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 875.

*pnAttribute* | 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 873.

*pbstrLabel* | String. Returns the label of the journal.

*pbstrDescription* | String. Returns the description of the journal.

*pbstrGroup* | String. Returns the description of the journal group to which the journal is assigned.

*plSingleEntity* | 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.

*plSingleParent* | 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.

*plSecurityClass* | Long. Returns the ID of the journal’s security class.

**Tip:** To get the label of the security class, pass this ID to `HsvSecurityAccess.GetSecurityClassLabel`. See “GetSecurityClassLabel” on page 483.

*pvararEntryID* | Variant array. For internal use.

*pvarDebitCreditUnit* | 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 874.

The array is returned as an Integer subtype.

*pvaradAmount* | Variant array. Returns the amounts for the journal’s line items.

The array is returned as a Double subtype.

**Tip:** Use `GetVariance` to return the total debit and credit amounts and the difference between these amounts. See “GetVariance” on page 437.

*pvaradstrDescription* | Variant array. Returns the descriptions for the journal’s line items.

The array is returned as a String subtype.

*pplUnkHfmSliceCOM* | HfmSliceCOM. HfmSliceCOM object containing multiple cells’ coordinates.

---

**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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>varalColumns</td>
<td>Long array (ByVal). Specifies the display columns for which journal information returned. Use the HFMConstants type library constants listed in &quot;Journal Column Display Constants&quot; on page 873 to create this array.</td>
</tr>
<tr>
<td>varlJournalIDs</td>
<td>Long array (ByVal). The ID numbers of the journals for which you want to return information.</td>
</tr>
<tr>
<td>pvar2DvarData</td>
<td>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 varalColumns argument; in each dimension, there is one element per journal. For details on the values returned in the array elements, see &quot;Journal Column Return Values&quot; on page 409.</td>
</tr>
</tbody>
</table>

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 = 2012
- 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, 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, "2012")
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)
GetJournalLabelsForIDs

Returns the labels of the journals that contain the specified Scenario and Year dimension members and that correspond to a specified set of journal IDs.

Syntax

```vb
<IHsvJournalsEx>.GetJournalLabelsForIDs lScenario, lYear, varalIDs, pvarabstrLabels
```

**Argument** | **Description**
---|---
`lScenario` | Long (ByVal). The member ID of the Scenario dimension member.
`lYear` | Long (ByVal). The member ID of the Year dimension member.
`varalIDs` | Long array (ByVal). The IDs of the journals.
`pvarabstrLabels` | Variant array. Returns the labels of the journals. This array has a one-to-one correspondence with the array of IDs passed to the `varalIDs` 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

```vb
<IHsvJournalsEx>.GetJournalQueryDefinitionIDs lScenario, lYear, lPeriod, lValue, varalColumns, vararvFilters, var2DalColumnsSort, pvararlJournalIDs
```

**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.
`lValue` | Long (ByVal). The member ID of the Value dimension member.
`varalColumns` | 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 873.
**Argument** | **Description**
---|---
`vararvFilters` | Variant (ByVal). Enables you to filter out journals:
- To return all journal IDs, pass an empty variable.
- To filter out journals, pass this as a Variant array that consists of 32 elements.

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 67 on page 429. 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.

**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.

`var2DalColumnsSort` | 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:
- The first dimension takes the HFMConstants type library constants listed in “Journal Column Display Constants” on page 873.
- The second dimension takes 0 to sort in ascending order, 1 in descending order.

For example, suppose you want to sort by status and then by label, with both sorts in ascending order. This array is defined as follows:

\[
(0, 0) = COLUMN_JOURNALSTATUS \\
(0, 1) = 0 \\
(1, 0) = COLUMN_JOURNALLABEL \\
(1, 1) = 0
\]

`pvararJournallDs` | 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>
<thead>
<tr>
<th>Index</th>
<th>Display Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Label</td>
<td>A String containing the desired label. You can use the percentage sign ( % ) as a wildcard character.</td>
</tr>
<tr>
<td>1</td>
<td>Status</td>
<td>An array of Longs containing the journal statuses to be applied. These statuses are represented by the HFMConstants type library constants listed in “Journal Status Constants” on page 875.</td>
</tr>
<tr>
<td>2</td>
<td>Type</td>
<td>An array of Longs containing the journal types to be applied. These journal types are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 876.</td>
</tr>
<tr>
<td>Index</td>
<td>Display Column</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>3</td>
<td>Balance Type</td>
<td>An array of Longs containing the balance types to be applied. These balance types are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 873.</td>
</tr>
<tr>
<td>4</td>
<td>Group</td>
<td>A String containing the name of the desired journal group. You can use the percentage sign ( % ) as a wildcard character.</td>
</tr>
<tr>
<td>5</td>
<td>Description</td>
<td>A String containing the name of the desired description. You can use the percentage sign ( % ) as a wildcard character.</td>
</tr>
<tr>
<td>14</td>
<td>Entity</td>
<td>The member ID (Long) of the entity by which to filter.</td>
</tr>
<tr>
<td>15</td>
<td>Parent</td>
<td>The member ID (Long) of the parent entity by which to filter.</td>
</tr>
</tbody>
</table>

**Example**

GetJournalQueryDefinitionIDs is used in the Example for GetJournalDisplayData.

**GetTemplate**

*Deprecated* - use “GetTemplateExtDim” on page 430.

**GetTemplateExtDim**

Returns a variety of information for a journal template.

**Tip:** Use GetTemplateExtDim to create a journal from a template; the return values can be passed to SaveJournalExtDim.

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 GetTextTemplateExtDim. It is almost identical to GetTemplateExtDim; the only difference is the subtype of the line item amount array.

**Syntax**

```csharp
<IHsvJournalsEx>.GetTemplateExtDim lTemplateID, pnType, pnAttribute, pbstrLabel, pbstrDescription, pbstrGroup, plSecurityClass, plSingleEntity, plSingleParent, pnTemplateType, pvararlEntryID, pvararlDebitCreditUnit, pvaradAmount, pvararbstrDescription, ppIUnkHfmSliceCOM
```

430  HsvJournals Type Library
Argument | Description
--- | ---
_TemplateID | Long (ByVal). The ID of the template. Pass the template label to `GetJournalTemplateItemID` to get this ID; see “GetJournalTemplateItemID” on page 412.

_nType | 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 876.

_nAttribute | 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 873.

_pbstrLabel | String. Returns the label of the template.

_pbstrDescription | String. Returns the description of the template.

_pbstrGroup | String. Returns the journal group of the template.

_plSecurityClass | Long. Returns the ID of the security class for the template.

_Tip: To get the ID from a security class name, use `_HsvSecurityAccess.GetSecurityClassLabel_`.

_plSingleEntity | 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.

_plSingleParent | 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.

_pnTemplateType | 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 877.

_pvararlEntryID | Variant array. For internal use.

_pvararnDebitCreditUnit | 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 874.

_The array is returned as an Integer subtype._

_pvaradAmount | Variant array. Returns the amounts for the template’s line items.

_The array is returned as a Double subtype._

_pvararstrDescription | Variant array. Returns the descriptions for the template’s line items.

_The array is returned as a String subtype._

_ppIUnkHfmSliceCOM | HfmSliceCOM. HfmSliceCOM object containing multiple cells’ coordinates.

**Example**

GetTemplate is used in the Example for `SaveJournalExtDim`.

**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.
GetTemplateDisplayData

Argument Description

varalColumns Long array (ByVal). Specifies the display columns for which template information returned. To create this array, use the HFMConstants type library constants listed in "Template Column Display Constants" on page 876.

varlTemplateIDs Long array (ByVal). The ID numbers of the templates for which you want to return information.

pvar2DvarData 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 varalColumns 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 410.

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; 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 specified set of journal template IDs.

Syntax

<IHsvJournalsEx>.GetTemplateLabelsForIDs varalIDs, pvarabstrLabels
Argument | Description
--- | ---
`varalIDs` | Long array (ByVal). The journal template IDs.

`pvarabstrLabels` | Variant array. Returns the labels of the templates. This array has a one-to-one correspondence with the array of IDs passed to the `varalIDs` 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, pvararlTemplateIDs
```

**Argument** | **Description**
--- | ---
`varalColumns` | 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 876.

`vararvFilters` | Variant (ByVal). Enables you to filter out templates:
- To return all template IDs, pass an empty variable.
- To filter out templates, pass this as a Variant array that consists of 32 elements.

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 68 on page 434. 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.

**Note:** In this release only the first four elements and the sixth through eights elements of this array are supported; the remaining elements are reserved for future use.

`var2DalColumnsSort` | 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:
- The first dimension takes the HFMConstants type library constants listed in "Template Column Display Constants" on page 876.
- The second dimension takes 0 to sort in ascending order, 1 in descending order.

For example, suppose you want to sort by label and then by group, with both sorts in ascending order. This array is defined as follows:

```
(0, 0) = COLUMN_TEMPLATELABEL
(0, 1) = 0
(1, 0) = COLUMN_TEMPLATEGROUP
(1, 1) = 0
```

`pvararlTemplateIDs` | 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>
<thead>
<tr>
<th>Index</th>
<th>Display Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Label</td>
<td>A String containing the desired label. You can use the percentage sign ( % ) as a wildcard character.</td>
</tr>
<tr>
<td>1</td>
<td>Balance Type</td>
<td>An array of Longs containing the balance types to be applied. These balance types are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 873.</td>
</tr>
<tr>
<td>2</td>
<td>Group</td>
<td>A String containing the name of the desired journal group. You can use the percentage sign ( % ) as a wildcard character.</td>
</tr>
<tr>
<td>3</td>
<td>Description</td>
<td>A String containing the name of the desired description. You can use the percentage sign ( % ) as a wildcard character.</td>
</tr>
<tr>
<td>5</td>
<td>Type</td>
<td>An array of Longs containing the template types to be applied. These journal types are represented by the HFMConstants type library constants listed in “Template Type Constants” on page 877.</td>
</tr>
<tr>
<td>6</td>
<td>Entity</td>
<td>The member ID (Long) of the entity by which to filter.</td>
</tr>
<tr>
<td>7</td>
<td>Parent</td>
<td>The member ID (Long) of the parent entity by which to filter.</td>
</tr>
</tbody>
</table>

**Example**

`GetTemplateQueryDefinitionIDs` is used in the Example for `GetTemplateDisplayData`.

**GetTextJournal**

*Deprecated* - use “GetTextJournalExtDim” on page 434.

**GetTextJournalExtDim**

Returns a variety of information for a journal.

**Caution!** To successfully call `GetTextJournalExtDim`, 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 `pvararlAccount` 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 `GetJournalExtDim`. It is almost identical to `GetTextJournalExtDim`; the only difference is the subtype of the line item amount array.

**Syntax**

```
<IHsvJournalsEx>.GetTextJournalExtDim lScenario, lYear, lJournalID, pnType, pnStatus,
PnAttribute, pbstrLabel, pbstrDescription, pbstrGroup, piSingleEntity, piSingleParent,
plSecurityClass, pvararnDebitCreditUnit, pvararbstrDescription, pvararlEntryID,
pvarabstrAmount, ppIUnkHfmSliceCOM
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lJournalID</td>
<td>Long (ByVal). The ID number of the journal. Get this ID with <code>HsvJournals.GetItemID</code>; see “GetItemID” on page 412.</td>
</tr>
<tr>
<td>pnType</td>
<td>Integer. Returns a numeric constant that identifies the journal’s type. The valid return values are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 876.</td>
</tr>
<tr>
<td>pnStatus</td>
<td>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 875.</td>
</tr>
<tr>
<td>pnAttribute</td>
<td>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 873.</td>
</tr>
<tr>
<td>pbstrLabel</td>
<td>String. Returns the label of the journal.</td>
</tr>
<tr>
<td>pbstrDescription</td>
<td>String. Returns the description of the journal.</td>
</tr>
<tr>
<td>pbstrGroup</td>
<td>String. Returns the description of the journal group to which the journal is assigned.</td>
</tr>
<tr>
<td>piSingleEntity</td>
<td>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.</td>
</tr>
<tr>
<td>piSingleParent</td>
<td>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.</td>
</tr>
<tr>
<td>plSecurityClass</td>
<td>Long. Returns the ID of the journal’s security class. To get the label of the security class, pass this ID to <code>HsvSecurityAccess.GetSecurityClassLabel</code>. See “GetSecurityClassLabel” on page 483.</td>
</tr>
<tr>
<td>pvararnDebitCreditUnit</td>
<td>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 874. The array is returned as an Integer subtype.</td>
</tr>
</tbody>
</table>
### Argument Description

- **pvararbstrDescription**  
  Variant array. Returns the descriptions for the journal’s line items.  
  The array is returned as a String subtype.

- **pvararlEntryID**  
  Variant array. *For internal use.*

- **pvarabstrAmount**  
  Variant array. Returns the amounts for the journal’s line items.  
  The array is returned as a String subtype.  
  **Tip:** Use `GetVariance` to return the total debit and credit amounts and the difference between these amounts.  
  See “GetVariance” on page 437.

- **ppIUnkHfmSliceCOM**  
  HfmSliceCOM. HfmSliceCOM object containing multiple cells’ coordinates

### Example

See the [Example](#) for `GetJournalExtDim`. 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

*Deprecated* - use “GetTextTemplateExtDim” on page 436.

### GetTextTemplateExtDim

Returns a variety of information for a journal template.

**Tip:** Use `GetTextTemplateExtDim` to create a journal from a template; its 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 `GetTemplateExtDim`. It is almost identical to `GetTextTemplateExtDim`; the only difference is the subtype of the line item amount array.

### Syntax

```csharp
<IHsvJournalsEx>.GetTextTemplateExtDim lTemplateID, pnType, pnAttribute, pbstrLabel, pbstrDescription, pbstrGroup, plSecurityClass, plSingleEntity, plSingleParent, pnTemplateType, pvararlEntryID, pvararnDebitCreditUnit, pvarabstrAmount, pvararbstrDescription, ppIUnkHfmSliceCOM
```
### Argument | Description
--- | ---
ITemplateID | Long (ByVal). The ID of the template. Pass the template label to GetJournalTemplateItemID to get this ID; see “GetJournalTemplateItemID” on page 412.

**pnType** | 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 876.

**pnAttribute** | 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 873.

**pbstrLabel** | String. Returns the label of the template.

**pbstrDescription** | String. Returns the description of the template.

**pbstrGroup** | String. Returns the journal group of the template.

**plSecurityClass** | Long. Returns the ID of the security class for the template.

**plSingleEntity** | 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.

**plSingleParent** | 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.

**pnTemplateType** | Integer. Returns the template type. The template types are represented by the HFMConstants type library constants listed in “Template Type Constants” on page 877.

**pvararlEntryID** | Variant array. For internal use.

**pvararnDebitCreditUnit** | 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 874. The array is returned as an Integer subtype.

**pvarabstrAmount** | Variant array. Returns the amounts for the template’s line items. The array is returned as a Double subtype.

**pvarabstrDescription** | Variant array. Returns the descriptions for the template’s line items. The array is returned as a String subtype.

**ppIUnkHfmSliceCOM** | HfmSliceCOM. HfmSliceCOM object containing multiple cells’ coordinates.

### Example
GetTextTemplateExtDim is used in the example for SaveTextJournalExtDim.

### 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 GetJournalExtDim or GetTextJournalExtDim as shown in the example.
Syntax

```
<IHsvJournalsEx>.GetVariance lSingleParent, lSingleEntity, varalParent, varalEntity, varalAccount, lValue, varanDebitCreditUnit, varabstrAmount, pbstrTotalDebits, pbstrTotalCredits, pbstrAmount
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSingleParent</td>
<td>Long (ByVal). Pass the value returned by the plSingleParent argument of GetJournalExtDim or GetTextJournalExtDim.</td>
</tr>
<tr>
<td>lSingleEntity</td>
<td>Long (ByVal). Pass the value returned by the plSingleEntity argument of GetJournalExtDim or GetTextJournalExtDim.</td>
</tr>
<tr>
<td>varalParent</td>
<td>Variant (ByVal). Pass the array of parent entity member IDs returned by the pvararlParent argument of GetJournalExtDim or GetTextJournalExtDim.</td>
</tr>
<tr>
<td>varalEntity</td>
<td>Variant (ByVal). Pass the array of entity member IDs returned by the pvararlEntity argument of GetJournalExtDim or GetTextJournalExtDim.</td>
</tr>
<tr>
<td>varalAccount</td>
<td>Variant (ByVal). Pass the array of account member IDs returned by the pvararlAccount argument of GetJournalExtDim or GetTextJournalExtDim.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). Pass the Value dimension member ID returned by the plValue argument of GetJournalExtDim or GetTextJournalExtDim.</td>
</tr>
<tr>
<td>varanDebitCreditUnit</td>
<td>Variant (ByVal). Pass the array flagging line items as debits or credits that is returned by the pvaranDebitCreditUnit argument of GetJournalExtDim or GetTextJournalExtDim.</td>
</tr>
<tr>
<td>varabstrAmount</td>
<td>Variant (ByVal). Pass the array of line item amounts that is returned by the pvarabstrAmount argument of GetTextJournalExtDim or the pvaradAmount argument of GetJournalExtDim.</td>
</tr>
</tbody>
</table>

**Caution!** The GetJournalExtDim array needs to be converted to a String subtype.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbstrTotalDebits</td>
<td>String. Returns the total debit amount.</td>
</tr>
<tr>
<td>pbstrTotalCredits</td>
<td>String. Returns the total credit amount.</td>
</tr>
<tr>
<td>pbstrAmount</td>
<td>String. Returns the difference between the pbstrTotalDebits and pbstrTotalCredits arguments’ amounts.</td>
</tr>
</tbody>
</table>

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.

```vbnet
m_cIHsvJournalEx.GetTextJournalExtDim 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 `vararlJournalIDs` 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 40D (hexadecimal) occurs.

---

**Syntax**

```plaintext
<IHsvJournalsEx>.PostJournals lScenario, lYear, vararlJournalIDs, pvararnRc
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code></td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td><code>vararlJournalIDs</code></td>
<td>Long array (ByVal). The IDs of the journals to be posted. Get these IDs with the HsvJournals object's <code>GetItemID</code> method; see &quot;GetItemID&quot; on page 412. Note: 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.</td>
</tr>
</tbody>
</table>
| `pvararnRc`| 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 `vararlJournalIDs` argument. Successfully posted journals return 0. The following list identifies some of the likely error codes in hexadecimal form:  

- 40F
- 3F9
- 418
- 40D
- 411
- 408
- 409

The array is returned as an Integer subtype.

---

**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`.

```plaintext
Dim lScen As Long, lYear As Long, lPer As Long
Dim laJnlIDs(1) As Long, vaRetVal
lScen = GetMemberID(DIMENSIONSCENARIO, "Actual")
lYear = GetMemberID(DIMENSIONYEAR, "2012")
lPer = GetMemberID(DIMENSIONPERIOD, "July")
laJnlIDs(0) = m_cHsvJournals.GetItemID(lScen, lYear, lPer, _
"Jnl990")
```
RejectJournals

Rejects one or more journals. The vararrJournalIDs 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 40D (hexadecimal) occurs.

### Syntax

```<IHsvJournalsEx>.RejectJournals lScenario, lYear, vararrJournalIDs, pvararnRc```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>vararrJournalIDs</td>
<td>Long array (ByVal). The IDs of the journals to be rejected. Get these IDs with the HsvJournals object's GetItemID method; see “GetItemID” on page 412.</td>
</tr>
<tr>
<td>pvararnRc</td>
<td>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 vararrJournalIDs argument. Successfully rejected journals return 0. The array is returned as an Integer subtype.</td>
</tr>
</tbody>
</table>

### 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 As Long, vaRetVal lScen = GetMemberID(DIMENSIONSCENARIO, "Actual") lYear = GetMemberID(DIMENSIONYEAR, "2012") 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

Deprecated - use “SaveJournalExtDim” on page 441.

SaveJournalExtDim

Creates a new journal, or saves changes to an existing journal that has a Working or Submitted status.

Caution! To successfully call SaveJournalExtDim, 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 vararlAccount arguments’ arrays specify the amount and account for a journal’s second line item.

SaveJournalExtDim is almost identical to SaveTextJournalExtDim; the only difference is that it passes line item amounts in a Double array while SaveTextJournalExtDim passes line item amounts in a String array. See “SaveTextJournal” on page 445.

Tip: To create a journal from a template, use GetTemplateExtDim to get the template’s information, then pass the applicable GetTemplateExtDim return values to SaveJournalExtDim.
Syntax

```<IHsvJournalsEx>.SaveJournalExtDim pIUnkHfmSliceCOM, nType, nStatus, nAttribute, bstrLabel, bstrDescription, bstrGroup, lSingleEntity, lSingleParent, lSecurityClass, vararnDebitCreditUnit, varadAmount, vararbstrDescription, plJournalID>```

**Argument** | **Description**
--- | ---
`pIUnkHfmSliceCOM` | HfmSliceCOM. HfmSliceCOM object containing multiple cells' coordinates.
`nType` | Integer (ByVal). Specifies the journal's type. Pass one of the HFMCollections type library constants listed in “Journal Type Constants” on page 876.

**Note:** You cannot use this method to save journals of type `JTF_AUTOREVERSAL`, as these are system-generated journals. Attempting to save a journal as this type results in error 8004041A (hexadecimal).

`nStatus` | Integer (ByVal). Specifies the journal's status. Pass one of the HFMCollections type library constants listed in “Journal Status Constants” on page 875.
`nAttribute` | Integer (ByVal). Specifies whether the journal must be balanced. Pass one of the HFMCollections type library constants listed in “Balance Type Constants” on page 873.
`bstrLabel` | String (ByVal). The label of the journal.
`bstrDescription` | String (ByVal). The description of the journal. You can pass a blank string if there is no description.
`bstrGroup` | 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.
`lSingleEntity` | 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.
`lSingleParent` | 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.
`lSecurityClass` | Long (ByVal). The ID of the journal’s security class. To get this ID, pass the security class name to HsvSecurityAccess.GetSecurityClassID. See “GetSecurityClassID” on page 482.
`vararnDebitCreditUnit` | Integer array (ByVal). Specifies whether the line items are debits or credits. Pass one of the HFMCollections type library constants listed in “Debit/Credit Constants” on page 874.
`varadAmount` | Double array (ByVal). The amounts of the line items.
`vararbstrDescription` | String array (ByVal). The descriptions of the line items.
`plJournalID` | Long. The purpose of this argument depends upon whether you are creating a new journal or updating an existing journal:
- 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 GetItemID; see “GetItemID” on page 412.

**Example**

This example uses `SaveJournalExtDim` 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 `GetTemplateExtDim`. note how most of its return values are passed to `SaveJournalExtDim`.

```vbscript
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.GetTemplateExtDim 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.SaveJournalExtDim pIUnkHfmSliceCOM, nType, nStatus, _
        ...nAttribute, bstrLabel, bstrDescription, bstrGroup, lSingleEntity, lSingleParent, _
        ...SecurityClass, vararnDebitCreditUnit, varadAmount, vararstrDescription, plJournalID
    'Assign the return value
    CreateJournal = lJnlID
End Function
```

**SaveTemplate**

*Deprecated* - use “SaveTemplateExtDim” on page 443.

**SaveTemplateExtDim**

Creates a new journal template, or updates an existing journal template.

---

**Caution!** To successfully call `SaveTemplateExtDim`, 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.

SaveTemplateExtDim takes a Double array for the template’s line item’s amounts. To create a template by passing a String array of amounts, use SaveTextTemplateExtDim instead of SaveTemplateExtDim. The methods are almost identical; the only difference is the subtype of the line item amount array.

Syntax

<IHsvJournalsEx>.SaveTemplateExtDim nType, nStatus, nAttribute, bstrLabel, bstrDescription, bstrGroup, lSecurityClass, lSingleEntity, lSingleParent, nTemplateType, vararnDebitCreditUnit, varadAmount, vararbstrDescription, pIUnkHfmSliceCOM, plTemplateID

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nType</td>
<td>Integer (ByVal). Specifies the journal type of the journals that created from the template. Pass one of the HFMConstants type library constants listed in &quot;Journal Type Constants&quot; on page 876.</td>
</tr>
<tr>
<td>nAttribute</td>
<td>Integer (ByVal). Specifies whether journals created from the template must be balanced. Pass one of the &quot;Balance Type Constants&quot; on page 873.</td>
</tr>
<tr>
<td>bstrLabel</td>
<td>String (ByVal). The label for the template.</td>
</tr>
<tr>
<td>bstrDescription</td>
<td>String (ByVal). The description for the template. You can pass a blank string if there is no description.</td>
</tr>
<tr>
<td>bstrGroup</td>
<td>String (ByVal). The journal group to which the journals created from the template assigned. You can pass a blank string if the journals is not assigned to a journal group</td>
</tr>
<tr>
<td>lSecurityClass</td>
<td>Long. The ID of the security class for the template.</td>
</tr>
<tr>
<td>lSingleEntity</td>
<td>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.</td>
</tr>
<tr>
<td>lSingleParent</td>
<td>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.</td>
</tr>
<tr>
<td>nTemplateType</td>
<td>Integer (ByVal). Determines whether the template is standard or recurring. Pass one of the HFMConstants type library constants listed in &quot;Template Type Constants&quot; on page 877.</td>
</tr>
<tr>
<td>vararnDebitCreditUnit</td>
<td>Integer array (ByVal). Specifies whether the line items are debits or credits. Pass one of the HFMConstants type library constants listed in &quot;Debit/Credit Constants&quot; on page 874.</td>
</tr>
<tr>
<td>varadAmount</td>
<td>Double array (ByVal). The amounts of the line items.</td>
</tr>
<tr>
<td>vararbstrDescription</td>
<td>String array (ByVal). The descriptions of the line items.</td>
</tr>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmSliceCOM. HfmSliceCOM object containing multiple cells’ coordinates.</td>
</tr>
<tr>
<td>plTemplateID</td>
<td>Long. The purpose of this argument depends upon whether you are creating a new template or updating an existing template:</td>
</tr>
<tr>
<td></td>
<td>● For a newly created template, this argument returns the automatically-generated ID of the template.</td>
</tr>
</tbody>
</table>
|                         | ● To update an existing template, pass the template’s ID with this argument. You can get template IDs with GetJournalTemplateItemID; see “GetJournalTemplateItemID” on page 412.

444 HsvJournals Type Library
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 `GetTemplateExtDim` to get the existing template’s information.

```vba
Function NewTemplate(sTemplate As String, sNewLabel As String, sDesc As String, lNewEntID As Long, lNewParID As Long) As Long
    Dim cHsvJournals As HsvJournals, iTType 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 iTTempType 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.GetTemplateExtDim lTempID, iTType, iAttr, sLabel, sDescHead, sGroup, lSecID, lSingleEnt, lSinglePar, iTTempType, lVal, vaEntryIDs, vaDebCredUnit
    vaAmt, vaItemDesc, vaEnt, vaPar, vaAcct, vaICP, vaCust1
    vaCust2, vaCust3, vaCust4
    lUpBounds = UBound(vaEnt)
    'Create the entity ID arrays that passed to SaveTemplate
    ReDim laNewEnt(lUpBounds)
    ReDim laNewPar(lUpBounds)
    For i = LBound(vaEnt) To lUpBounds
        laNewEnt(i) = lNewEntID
        laNewPar(i) = lNewParID
    Next i
    cIHsvJournalEx.SaveTemplateExtDim iTType, iAttr, sNewLabel, sDesc, sGroup, lSecID, lNewEntID, lNewParID, iTTempType, vaDebCredUnit, vaAmt, vaItemDesc, pIUnkHfmSliceCOM, lNewTempID
    NewTemplate = lNewTempID
End Function
```

**SaveTextJournal**

*Deprecated* - use “SaveTextJournalExtDim” on page 445.

**SaveTextJournalExtDim**

Creates a new journal, or saves changes to an existing journal that has a Working or Submitted status.
Caution! To successfully call SaveTextJournalExtDim, 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 vararlAccount arguments’ arrays specify the amount and account for a journal’s second line item.

SaveTextJournalExtDim is almost identical to SaveJournalExtDim; the only difference is that it passes line item amounts in a String array while SaveJournalExtDim passes line item amounts in a Double array.

Syntax

```
Application.IHsvJournalsEx.SaveTextJournalExtDim pIUnkHfmSliceCOM, nType, nStatus, nAttribute, bstrLabel, bstrDescription, bstrGroup, lSingleEntity, lSingleParent, lSecurityClass, vararnDebitCreditUnit, varabstrAmount, vararbstrDescription, plJournalID
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmSliceCOM. HfmSliceCOM object containing multiple cells' coordinates.</td>
</tr>
<tr>
<td>nType</td>
<td>Integer (ByVal). Specifies the journal’s type. Pass one of the HFMConstants type library constants listed in “Journal Type Constants” on page 876.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You cannot use this method to save journals of type JTF_AUTOREVERSAL, as these are system-generated journals. Attempting to save a journal as this type produces error 8004041A (hexadecimal).</td>
</tr>
<tr>
<td>nStatus</td>
<td>Integer (ByVal). Specifies the journal’s status. Pass one of the HFMConstants type library constants listed in “Journal Status Constants” on page 875.</td>
</tr>
<tr>
<td>nAttribute</td>
<td>Integer (ByVal). Specifies whether the journal must be balanced. Pass one of the “Balance Type Constants” on page 873.</td>
</tr>
<tr>
<td>bstrLabel</td>
<td>String (ByVal). The label of the journal.</td>
</tr>
<tr>
<td>bstrDescription</td>
<td>String (ByVal). The description of the journal. You can pass a blank string if there is no description.</td>
</tr>
<tr>
<td>bstrGroup</td>
<td>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.</td>
</tr>
<tr>
<td>lSingleEntity</td>
<td>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.</td>
</tr>
<tr>
<td>lSingleParent</td>
<td>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.</td>
</tr>
<tr>
<td>lSecurityClass</td>
<td>Long (ByVal). The ID of the journal’s security class. To get this ID, pass the security class name to HsvSecurityAccess.GetSecurityClassID. See “GetSecurityClassID” on page 482.</td>
</tr>
<tr>
<td>vararnDebitCreditUnit</td>
<td>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 874.</td>
</tr>
<tr>
<td>varabstrAmount</td>
<td>String array (ByVal). The amounts of the line items.</td>
</tr>
<tr>
<td>vararbstrDescription</td>
<td>String array (ByVal). The descriptions of the line items.</td>
</tr>
</tbody>
</table>
Argument | Description
--- | ---
`pJournalID` | Long. The purpose of this argument depends upon whether you are creating a new journal or updating an existing journal:
- 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 `GetItemID`; see "GetItemID" on page 412.

Example

This example uses `SaveTextJournalExtDim` 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 `GetTextTemplateExtDim`; note how most of its return values are passed to `SaveTextJournalExtDim`.

```vba
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 GetTextTemplateExtDim - many are also passed to 'SaveTextJournalExtDim.
    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.GetTextTemplateExtDim 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.SaveTextJournalExtDim pIUnkHfmSliceCOM, _

IHsvJournalsEx Interface Methods 447
Assign the return value
CreateTextJournal = lJnlID
End Function

**SaveTextTemplate**

*Deprecated* - use “SaveTextTemplateExtDim” on page 448.

**SaveTextTemplateExtDim**

Creates a new journal template, or updates an existing journal template.

---

**Caution!** To successfully call `SaveTextTemplateExtDim`, 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.

`SaveTextTemplateExtDim` takes a String array for the template’s line item’s amounts. To create a template by passing a Double array of amounts, use `SaveTemplateExtDim` instead of `SaveTextTemplateExtDim`. The methods are almost identical; the only difference is the subtype of the line item amount array.

**Syntax**

```
<IHsvJournalsEx>.SaveTextTemplateExtDim nType, nAttribute, bstrLabel, bstrDescription, bstrGroup, lSecurityClass, lSingleEntity, lSingleParent, nTemplateType, vararnDebitCreditUnit, varabstrAmount, vararbstrDescription, pIUnkHfmSliceCOM, plTemplateID
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nType</td>
<td>Integer (ByVal). Specifies the journal type of the journals that created from the template. Pass one of the HFMCOnstants type library constants listed in “Journal Type Constants” on page 876.</td>
</tr>
<tr>
<td>nAttribute</td>
<td>Integer (ByVal). Specifies whether journals created from the template must be balanced. Pass one of the HFMCOnstants type library constants listed in “Balance Type Constants” on page 873.</td>
</tr>
<tr>
<td>bstrLabel</td>
<td>String (ByVal). The label of the template.</td>
</tr>
<tr>
<td>bstrDescription</td>
<td>String (ByVal). The description for the template. You can pass a blank string if there is no description.</td>
</tr>
<tr>
<td>bstrGroup</td>
<td>String (ByVal). The journal group to which the journals created from the template assigned. You can pass a blank string if the journals are not assigned to a journal group</td>
</tr>
</tbody>
</table>
Argument | Description
--- | ---
ISecurityClass | Long. The ID of the security class for the template. **Tip:** To get the ID from a security class name, use HsvSecurityAccess.GetSecurityClassLabel.
ISingleEntity | 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.
ISingleParent | 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.
TemplateType | 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 877.
varDebitCreditUnit | 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 874.
varAmount | String array (ByVal). The amounts of the line items.
varDescription | String array (ByVal). The descriptions of the line items.
plHfmSliceCOM | HfmSliceCOM. HfmSliceCOM object containing multiple cells’ coordinates.
plTemplateID | Long. The purpose of this argument depends upon whether you are creating a new template or updating an existing template:
- 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 GetJournalTemplateItemID; see “GetJournalTemplateItemID” on page 412.

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 GetTextTemplateExtDim to get the existing template’s information.

```vbnet
Function NewTextTemplate(sTemplate As String, sNewLabel As String, sDesc As String, lNewEntID As Long, lNewParID As Long) As Long
    Dim cHsvJournals As HsvJournals, iTType 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 iTTempType 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.GetTextTemplateExtDim lTempID, iTType, iAttr, sLabel, _
    sDescHead, sGroup, lSecID, lSingleEnt, lSinglePar, _
```

IHsvJournalsEx Interface Methods 449
iTempType, lVal, vaEntryIDs, vaDebCredUnit, vaAmt, _
vaItemDesc, vaEnt, vaPar, vaAcct, vaICP, vaCust1, vaCust2, _
vaCust3, vaCust4
lUpBounds = UBound(vaEnt)
'Create the entity ID arrays that passed to SaveTemplate
ReDim laNewEnt(lUpBounds)
ReDim laNewPar(lUpBounds)
For i = LBound(vaEnt) To lUpBounds
   laNewEnt(i) = lNewEntID
   laNewPar(i) = lNewParID
Next i
cIHsvJournalEx.SaveTextTemplateExtDim iType, iAttr, sNewLabel, _
   sDesc, sGroup, lSecID, lNewEntID, lNewParID, iTempType, _
   vaDebCredUnit, vaAmt, vaItemDesc, pIUnkHfmSliceCOM _
   lNewTempID
NewTextTemplate = lNewTempID
End Function

SubmitJournals

Submits one or more journals. The vararlJournalIDs 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 40D (hexadecimal) occurs.

Syntax

<IHsvJournalsEx>.SubmitJournals lScenario, lYear, vararlJournalIDs, pvararnRc

Argument          Description

lScenario    Long (ByVal). The member ID of the Scenario dimension member.

lYear        Long (ByVal). The member ID of the Year dimension member.

vararlJournalIDs Long array (ByVal). The IDs of the journals to be submitted. Get these IDs with the HsvJournals object’s GetItemID method; see "GetItemID" on page 412.

pvararnRc      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 vararlJournalIDs argument. Successfully submitted journals return 0. 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, "2012")
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 `vararlJournalIDs` 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 40D (hexadecimal) occurs.

---

**Syntax**

```csharp
<IHsvJournalsEx>.UnpostJournals lScenario, lYear, vararlJournalIDs, pvararnRc
```

**Argument**

- `lScenario` Long (ByVal). The member ID of the Scenario dimension member.
- `lYear` Long (ByVal). The member ID of the Year dimension member.
- `vararlJournalIDs` Long array (ByVal). The IDs of the journals to be unposted. Get these IDs with the HsvJournals object’s `GetItemID` method; see “GetItemID” on page 412.
- `pvararnRc` 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 `vararlJournalIDs` argument. Successfully unposted journals return 0. The following list identifies some of the likely error codes in hexadecimal form:
  - 40F
  - 3F9
  - 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`. 
UnsubmitJournals

Unsubmits one or more journals. The variantJournalIDs 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 40D (hexadecimal) occurs.

Syntax

<iHsvJournalsEx>.UnsubmitJournals lScenario, lYear, vararlJournalIDs, pvararnRc

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>vararlJournalIDs</td>
<td>Long array (ByVal). The IDs of the journals to be unsubmitted. Get these IDs with the HsvJournals object's GetItemID method; see &quot;GetItemID&quot; on page 412.</td>
</tr>
<tr>
<td>pvararnRc</td>
<td>Variant array. Returns codes that identify the success or failure of the journal unsubmissions. The elements in this array have a one-to-one correspondence with the journal IDs passed in the vararlJournalIDs argument. Successfully unsubmitted journals return 0. The array is returned as an Integer subtype.</td>
</tr>
</tbody>
</table>

Example

This example unsubmits 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, "2012")
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
ValidateLineItems

Deprecated - use “ValidateLineItemsExtDim” on page 453.

ValidateLineItemsExtDim

Validates whether the values passed define a valid journal.

The journal line item data is passed as arrays; these arrays must have a one-to-one correspondence. ValidateLineItemsExtDim returns an array of error numbers that corresponds to these arrays of line item data.

Syntax

```csharp
IHsvJournalsEx.ValidateLineItemsExtDim pIUnkHfmSliceCOM, lJournalID, sJournalType, sBalanceAttribute, lSingleEntity, lSingleParent, vararlLineItemID, vararnDebitCreditUnit, varabstrAmount, pvarHRLineItems
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmSliceCOM</td>
<td>HfmSliceCOM. HfmSliceCOM object containing multiple cells' coordinates.</td>
</tr>
<tr>
<td>lJournalID</td>
<td>Long (ByVal). Pass 0 (zero) to this argument.</td>
</tr>
<tr>
<td>sJournalType</td>
<td>Integer (ByVal). Specifies the journal’s type. Pass one of the HFMConstants type library constants listed in “Journal Type Constants” on page 876.</td>
</tr>
<tr>
<td>sBalanceAttribute</td>
<td>Integer (ByVal). Specifies whether the journal must be balanced. Pass one of the HFMConstants type library constants listed in “Balance Type Constants” on page 873.</td>
</tr>
<tr>
<td>lSingleEntity</td>
<td>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.</td>
</tr>
<tr>
<td>lSingleParent</td>
<td>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.</td>
</tr>
<tr>
<td>vararlLineItemID</td>
<td>Long array (ByVal). The line numbers of the journal line items.</td>
</tr>
<tr>
<td>vararnDebitCreditUnit</td>
<td>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 874.</td>
</tr>
<tr>
<td>varabstrAmount</td>
<td>String array (ByVal). The amounts of the line items.</td>
</tr>
<tr>
<td>pvarHRLineItems</td>
<td>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.</td>
</tr>
</tbody>
</table>

Tip: To look up a description of a Financial Management error number, use the utility described in “Error Message Lookup Utility” on page 843.
**ValidateValue**

Indicates whether a String resolves to a Double that can be inserted as a journal amount. To evaluate as valid, the String must not represent a negative Double, as the system does not allow negative amounts in journals.

**Syntax**

```
<IHsvJournalsEx>.ValidateValue(lEntity, lParent, lAccount, lValue, bstAmount)
```

**Argument Description**

- **lEntity** Long (ByVal). You must pass a valid Long. The value passed is ignored, this argument is intended for future use.
- **lParent** Long (ByVal). You must pass a valid Long. The value passed is ignored, this argument is intended for future use.
- **lAccount** Long (ByVal). You must pass a valid Long. The value passed is ignored, this argument is intended for future use.
- **lValue** Long (ByVal). You must pass a valid Long. The value passed is ignored, this argument is intended for future use.
- **bstAmount** 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 is executed.

```vbnet
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 192.

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 69 Journal Report Filter Array Elements

<table>
<thead>
<tr>
<th>Index</th>
<th>Display Column</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Label</td>
<td>A String containing the desired label. You can use the percentage sign ( % ) as a wildcard character.</td>
</tr>
<tr>
<td>1</td>
<td>Status</td>
<td>An array of Longs containing the journal statuses to be applied. These statuses are represented by the HFMConstants type library constants listed in &quot;Journal Status Constants&quot; on page 875.</td>
</tr>
<tr>
<td>2</td>
<td>Type</td>
<td>An array of Longs containing the journal types to be applied. These balance types are represented by the HFMConstants type library constants listed in &quot;Journal Type Constants&quot; on page 876.</td>
</tr>
<tr>
<td>3</td>
<td>Balance Type</td>
<td>An array of Longs containing the balance types to be applied. These balance types are represented by the HFMConstants type library constants listed in &quot;Balance Type Constants&quot; on page 873.</td>
</tr>
<tr>
<td>4</td>
<td>Group</td>
<td>A String containing the name of the desired journal group. You can use the percentage sign ( % ) as a wildcard character.</td>
</tr>
<tr>
<td>5</td>
<td>Description</td>
<td>A String containing the name of the desired description. You can use the percentage sign ( % ) as a wildcard character.</td>
</tr>
<tr>
<td>14</td>
<td>Entity</td>
<td>The member ID (Long) of the entity by which to filter.</td>
</tr>
<tr>
<td>15</td>
<td>Parent</td>
<td>The member ID (Long) of the parent entity by which to filter.</td>
</tr>
<tr>
<td>16</td>
<td>Account</td>
<td>The member ID (Long) of the account by which to filter.</td>
</tr>
</tbody>
</table>

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 70 Report Column Display Constants

<table>
<thead>
<tr>
<th>Constant</th>
<th>Column</th>
<th>Return Value (Subtype and Description)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Label</td>
<td>String. The journal label.</td>
</tr>
<tr>
<td>1</td>
<td>Status</td>
<td>Integer. The journal status. The valid return values are represented by the HFMConstants type library constants listed in &quot;Journal Status Constants&quot; on page 875</td>
</tr>
</tbody>
</table>

IHsvJournalsReport Interface Methods  455
<table>
<thead>
<tr>
<th>Constant</th>
<th>Column</th>
<th>Return Value (Subtype and Description)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Type</td>
<td>Integer. The journal type. The valid return values are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 876</td>
</tr>
<tr>
<td>3</td>
<td>Balance Type</td>
<td>Integer. The balance type. The valid return values are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 873.</td>
</tr>
<tr>
<td>4</td>
<td>Group</td>
<td>String. The name of the journal group.</td>
</tr>
<tr>
<td>5</td>
<td>Description</td>
<td>String. The journal description.</td>
</tr>
<tr>
<td>6</td>
<td>Short Description</td>
<td>String. The truncated version of the journal description.</td>
</tr>
<tr>
<td>7</td>
<td>Created By</td>
<td>String. The fully qualified username of the journal creator.</td>
</tr>
<tr>
<td>8</td>
<td>Date Created</td>
<td>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.</td>
</tr>
<tr>
<td>9</td>
<td>Approved By</td>
<td>String. The fully qualified username of the journal approver. If a journal has not been approved, a blank String is returned.</td>
</tr>
<tr>
<td>10</td>
<td>Approved On</td>
<td>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.</td>
</tr>
<tr>
<td>11</td>
<td>Posted By</td>
<td>String. The fully qualified username of the journal poster. If a journal has not been posted, a blank String is returned.</td>
</tr>
<tr>
<td>12</td>
<td>Date Posted</td>
<td>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.</td>
</tr>
<tr>
<td>13</td>
<td>Security Class</td>
<td>Long. The ID number of the journal’s security class.</td>
</tr>
<tr>
<td>14</td>
<td>Entity</td>
<td>Long. The member ID of the journal’s entity.</td>
</tr>
<tr>
<td>15</td>
<td>Parent</td>
<td>Long. The member ID of the journal’s parent entity.</td>
</tr>
<tr>
<td>16</td>
<td>Account</td>
<td>Long. The member ID of the journal’s account.</td>
</tr>
<tr>
<td>17</td>
<td>ICP</td>
<td>Long. The member ID of the journal’s Intercompany Partner dimension member.</td>
</tr>
<tr>
<td>18</td>
<td>Custom1</td>
<td>Long. The member ID of the journal’s Custom1 dimension member.</td>
</tr>
<tr>
<td>19</td>
<td>Custom2</td>
<td>Long. The member ID of the journal’s Custom2 dimension member.</td>
</tr>
<tr>
<td>20</td>
<td>Custom3</td>
<td>Long. The member ID of the journal’s Custom3 dimension member.</td>
</tr>
<tr>
<td>21</td>
<td>Custom4</td>
<td>Long. The member ID of the journal’s Custom4 dimension member.</td>
</tr>
</tbody>
</table>

**GetReportData**

*Deprecated* - use “GetReportDataExtDim” on page 457.
GetReportDataExtDim

Returns data for journals; only actual journal amounts are returned.

Syntax

<IHsvJournalsReport>.GetReportDataExtDim pIUnkHfmSliceCOM, varavFilters, varalColumns, varanSortFlags, pvar2DvarData, pvardScaledValues, pvardValues, pvarlStatus, pvaralDebCred

Argument | Description
--- | ---
pIUnkHfmSliceCOM | HfmSliceCOM. HfmSliceCOM object containing multiple cells' coordinates.
varavFilters | Variant (ByVal). Filters out journals:
- 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

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 69 on page 455. 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.

**Note:** In this release only, elements 1-6 and elements 14-16 of this array are supported; the remaining elements are reserved for future use.

varalColumns | Long array (ByVal). Specifies the display columns that the report data contains. To create this array, use the constants listed in Table 70 on page 455.

**Tip:** Specify the sortable columns as the top elements of this array, which has a one-to-one correspondence with the varanSortFlags argument’s array.

varanSortFlags | Integer array (ByVal). Specifies the columns on which the report data sorted, as well as the sort order. This array has a one-to-one correspondence with the varalColumns argument’s array. Use one of the HFMConstants type library constants listed in “Journal Report Sort Option Constants” on page 875.

pvar2DvarData | Variant array. Returns information for the specified columns. This is a multidimensional array, with one dimension for each column specified in the varalColumns argument; in each dimension, there is one element per journal.

SeeTable 70 on page 455.

pvardScaledValues | Variant array. Returns the debit and credit amounts for the journals in the report. This array has a one-to-one correspondence with the pvar2DvarData argument’s array, and is returned as a Double subtype.

pvardValues | Variant array. Returns the debit and credit amounts for the journals in the report. This array has a one-to-one correspondence with the pvar2DvarData argument’s array, and is returned as a Double subtype.

pvarlStatus | 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:
- 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.

This array has a one-to-one correspondence with the pvar2DvarData argument’s array, and is returned as a Long subtype.

pvaralDebCred | Variant array. Contains either JE_TYPE_DEBIT or JE_TYPE_CREDIT.
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.

```vba
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.GetReportDataExtDim 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**

*Deprecated* - use “GetReportDataExtDim” on page 457.

**GetReportData3**

Returns data for journals; scaled and actual journal amounts are returned. It does not rely on positive and negative data values to determine whether the entry is a debit or a credit which can fail to generate correct results under certain conditions.
### Syntax

```
<HsvJournalsReport>.GetReportData3 lScenario, lYear, lPeriod, lValue, varavFilters, varalColumns, varanSortFlags, pvar2DvarData, pvardScaledValues, pvardValues, pvarlStatus, pvaralDebCred
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code></td>
<td>Long (ByVal). The member ID of the Scenario dimension member for the report data.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the Year dimension member for the report data.</td>
</tr>
<tr>
<td><code>lPeriod</code></td>
<td>Long (ByVal). The member ID of the Period dimension member for the report data.</td>
</tr>
<tr>
<td><code>lValue</code></td>
<td>Long (ByVal). The member ID of the Value dimension member for the report data.</td>
</tr>
</tbody>
</table>
| `varavFilters`| Variant (ByVal). Filters out journals:  
  * 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.  

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 69 on page 455. 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.

In this release only elements 1-6 and elements 14–16 of this array are supported; the remaining elements are reserved for future use.

| `varalColumns` | Variant (ByVal). Specifies the display columns that the report data contains. To create this array, use the constants listed in Table 70 on page 455.  
**Tip:** Specify the sortable columns as the top elements of this array, which has a one-to-one correspondence with the `varanSortFlags` argument's array. |
| `varanSortFlags` | Variant (ByVal). Specifies the columns on which the report data sorted, as well as the sort order. This array has a one-to-one correspondence with the `varalColumns` argument's array. Use one of the HFMConstants type library constants listed in “Journal Report Sort Option Constants” on page 875. |
| `pvar2DvarData` | Variant array. Returns information for the specified columns. This is a multidimensional array, with one dimension for each column specified in the `varalColumns` argument; in each dimension, there is one element per journal. See Table 70 on page 455. |
| `vardScaledValues` | 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 `pvar2DvarData` argument’s array, and is returned as a Double subtype. |
| `vardValues` | 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 `pvar2DvarData` argument’s array, and is returned as a Double subtype. |
| `varlStatus` | 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:  
  * 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.  

This array has a one-to-one correspondence with the `pvar2DvarData` argument’s array, and is returned as a Long subtype. |
| `varalDebCred` | Variant array. Contains either `JE_TYPE_DEBIT` or `JE_TYPE_CREDIT`. |
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.


**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>
<thead>
<tr>
<th>Rights level</th>
<th>Enumerated</th>
</tr>
</thead>
<tbody>
<tr>
<td>None</td>
<td>HFM_ACCESS_RIGHTS_NONE</td>
</tr>
<tr>
<td>Promote</td>
<td>HFM_ACCESS_RIGHTS_READANDPROMOTE</td>
</tr>
<tr>
<td>Read</td>
<td>HFM_ACCESS_RIGHTS_READONLY</td>
</tr>
<tr>
<td>All</td>
<td>HFM_ACCESS_RIGHTS_ALL</td>
</tr>
</tbody>
</table>
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 for internal use only; if you attempt to use it, an error occurs.

### 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 tag `HFM_ROLE_ENUM`, which is described in “Role Constants” on page 881.

**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 tag `HFM_TASK_ENUM`, which is described in “Task Constants” on page 884.

### E-mail Alerting Constants
Some `HsvSecurityAccess` methods use bitmasks that represent email alerting rights. The following table lists constants that represent the bits.

<table>
<thead>
<tr>
<th>Notification Right</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>E-mail alerting disabled</td>
<td>HFM_EMAIL_ALERTING_DISABLED</td>
</tr>
<tr>
<td>E-mail alerting enabled</td>
<td>HFM_EMAIL_ALERTING_ENABLED</td>
</tr>
</tbody>
</table>

**Note:** The bits represented by `HFM_EMAIL_ALERTING_ENABLED` and `HFM_EMAIL_ALERTING_DISABLED` 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.
### User Type Flag Constants

<table>
<thead>
<tr>
<th>User Type Flag</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclude administrators</td>
<td>HFM_USER_GROUP_ENUM_EXCLUDEADMINS</td>
</tr>
<tr>
<td>User groups</td>
<td>HFM_USER_GROUP_ENUM_GROUPS</td>
</tr>
<tr>
<td>Include nested user groups</td>
<td>HFM_USER_GROUP_ENUM_RECURSIVE</td>
</tr>
<tr>
<td>Users</td>
<td>HFM_USER_GROUP_ENUM_USERS</td>
</tr>
</tbody>
</table>

### 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>
<thead>
<tr>
<th>Identity Type</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Administrator</td>
<td>HFM_IDENTITY_TYPE_APP_ADMIN</td>
</tr>
<tr>
<td>User Group</td>
<td>HFM_IDENTITY_TYPE_GROUP</td>
</tr>
<tr>
<td>User</td>
<td>HFM_IDENTITY_TYPE_USER</td>
</tr>
<tr>
<td>Invalid security identifier</td>
<td>HFM_IDENTITY_TYPE_INVALID_SID</td>
</tr>
</tbody>
</table>

### Search Filters

The following table lists constants that represent bits for bitmasks in filtered searches, such as those performed by `EnumUsersWithFilter`.

<table>
<thead>
<tr>
<th>Filter</th>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exclude application administrators</td>
<td>HFM_USER_GROUP_ENUM_EXCLUDEADMINS</td>
</tr>
<tr>
<td>Include groups</td>
<td>HFM_USER_GROUP_ENUM_GROUPS</td>
</tr>
<tr>
<td>Return groups recursively—groups within groups—if the HFM_USER_GROUP_ENUM_GROUPS flag is set.</td>
<td>HFM_USER_GROUP_ENUM_RECURSIVE</td>
</tr>
<tr>
<td>Include users</td>
<td>HFM_USER_GROUP_ENUM_FLAGS</td>
</tr>
</tbody>
</table>
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 them, 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`. See “GetSecurityClassID” on page 293.

The HsvSecurityAccess object’s methods are summarized in Table 26 on page 91, 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; see “IsApplicationAdministrator” on page 488.

AddApplicationAdministrator

Deprecated - use AddApplicationAdministrator2.

AddApplicationAdministrator2

Assigns the Application Administrator role to a user.

Syntax

```<HsvSecurityAccess>.AddApplicationAdministrator2 bstrUserSID```

Argument Description

bstrUserSID 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

| varabstrUserSIDs | String array (ByVal). The security identifier of the users being added to or removed from the Application Administrator role. This array has a one-to-one correspondence with the varabAdd argument's array. For example, the third element of the varabAdd array indicates whether the user identified by the third element of this array is added or removed. |
| varabAdd | 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.

```vba
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

```vba
<HsvSecurityAccess>.AddOrRemoveRolesFromUser2 bstrUserSID, varalRoles, varabAdd
```

### Argument Description

| bstrUserSID | String (ByVal). The security identifier of the user being assigned to or removed from the roles. |
| varalRoles | Long array (ByVal). The IDs of the roles. This array has a one-to-one correspondence with the varabAdd argument's array. For example, the third element of the varabAdd array indicates whether the user is added to or removed from the role identified by the third element of this array. |
| varabAdd | 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. |

**Tip:** To get a role's ID from its name, use GetRoleID.
Example

The following subroutine adds the specified roles to a specified user. The subroutine takes an array of role IDs and the user’s security identifier.

```vba
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 specified role.

Syntax

```
<HsvSecurityAccess>.AddOrRemoveUsersFromRole2 lRoleID, varabstrUserSIDs, varabAdd
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lRoleID</td>
<td>Long (ByVal). The ID of the role. For a list of valid role IDs, see “Role ID Constants” on page 462.</td>
</tr>
<tr>
<td>Tip</td>
<td>To get a role's ID from its name, use GetRoleID.</td>
</tr>
<tr>
<td>varabstrUserSIDs</td>
<td>String array (ByVal). The security identifier of the users being assigned to or removed from the role.</td>
</tr>
<tr>
<td></td>
<td>This array has a one-to-one correspondence with the varabAdd argument's array. For example, the third element of the varabAdd array indicates whether the user identified by the third element of this array is assigned or removed.</td>
</tr>
<tr>
<td>varabAdd</td>
<td>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.</td>
</tr>
</tbody>
</table>

Example

The following subroutine adds the users to a specified role. The subroutine takes an array of security identifiers and the name of the role.

```vba
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!** This method fails if used against an application created with Performance Management Architect.

**Syntax**

```
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bstrSecurityClass</strong> String (ByVal). The name of the security class to add.</td>
</tr>
<tr>
<td><strong>plSecurityClassID</strong> Long. Returns the ID that Financial Management assigns to the security class.</td>
</tr>
</tbody>
</table>

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**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bstrName</strong> String (ByVal). The username or user group name.</td>
</tr>
<tr>
<td>This must be a valid Windows username or user group for the application server, otherwise an error occurs.</td>
</tr>
<tr>
<td><strong>pbstrFullName</strong> String. Returns the name of the user or group.</td>
</tr>
<tr>
<td><strong>pbstrUserSID</strong> String. Returns the user’s security identifier.</td>
</tr>
</tbody>
</table>

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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lRoleID</td>
<td>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 462.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> To get a role’s ID from its name, use GetRoleID.</td>
</tr>
<tr>
<td>bstrUserSID</td>
<td>String (ByVal). The user’s security identifier.</td>
</tr>
</tbody>
</table>

**Example**

The following subroutine assigns a user to a specified 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!** This method fails if used against an application created with Performance Management Architect.

**Syntax**

```
<HsvSecurityAccess>.DeleteSecurityClass lSecurityClassID
```
Argument | Description
---|---
$lSecurityClassID$ 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.

```vbscript
Sub delectSecClass(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
```

DeleteSecurityClasses

Deletes a set of security classes from a Classic application.

**Caution!** This method fails if used against an application created with Performance Management Architect.

Syntax

`<HsvSecurityAccess>.DeleteSecurityClasses varalSecurityClassIDs, vara1HRStatuses`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varalSecurityClassIDs</td>
<td>Array of security classes that need to be deleted.</td>
</tr>
<tr>
<td>vara1HRStatuses</td>
<td>Array of statuses, as the result of security classes deletion. If deletion of a security class fails, the corresponding vara1HRStatuses entry will have the error status.</td>
</tr>
</tbody>
</table>

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**

*pvarabstrUserSIDs*  
Variant. Returns an array containing the security identifiers of the users. The array is returned as a String subtype.

*pvarabstrNames*  
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**

*pvarabstrRoles*  
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**

*bstrUserSID*  
String (ByVal). The user’s or group’s security identifiers.

*pvaralRoleIds*  
Variant array. Returns the IDs of the assigned roles. Valid values are listed in “Role ID Constants” on page 462. The array's subtype is Long.

### Example

The following function returns the roles for a username.

```vbnet
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**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bstrUser</strong></td>
</tr>
<tr>
<td><strong>bstrLangId</strong></td>
</tr>
<tr>
<td><strong>pvarabstrRoleNames</strong></td>
</tr>
</tbody>
</table>

**Example**

The following function returns the English-language names of the connected user’s roles.

```vbnet
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**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pvaralSecurityClassIDs</strong></td>
</tr>
<tr>
<td><strong>pvarabstrSecurityClasses</strong></td>
</tr>
</tbody>
</table>

**Example**

This example puts the names of an application’s security classes into a ComboBox.

```vbnet
Dim vaSecIDs, vaSecNames, lHiBound As Long
'g_cSecurity is an HsvSecurityAccess object reference
```
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 email alerting rights to security classes.

The arrays have a one-to-one correspondence.

**Syntax**

```<HsvSecurityAccess>.EnumSecurityClassRightsForPrincipal bstrUserSID, pvaralSecurityClassIDs, pvarabstrSecurityClasses, pvaralRights, pvaralEmailAlerting```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bstrUserSID</strong></td>
<td>String (ByVal). The user's security identifier.</td>
</tr>
<tr>
<td><strong>pvaralSecurityClassIDs</strong></td>
<td>Variant array. Retuns the IDs of the security classes to which the user has access rights. The array's subtype is Long.</td>
</tr>
<tr>
<td><strong>pvarabstrSecurityClasses</strong></td>
<td>Variant array. Retuns the names of the security classes. The array's subtype is String.</td>
</tr>
<tr>
<td><strong>pvaralRights</strong></td>
<td>Variant array. Returns the user’s rights to the security classes. Valid values are represented by the constants in Table 71 on page 461. The array's subtype is Long.</td>
</tr>
<tr>
<td><strong>pvaralEmailAlerting</strong></td>
<td>Variant array. Returns an array of bitmasks that represent user's email alerting rights to the security classes. Valid values for the bits are represented by the enumeration described in “E-mail Alerting Constants” on page 462. The array's subtype is Long.</td>
</tr>
</tbody>
</table>

**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
  Next i
Function```
ReDim Preserve vaRet(lCounter)
vaRet(lCounter) = vaSecNames(i)
End If
Next i
End If
GetUserSecClassesAllAccess = vaRet
End Function

**EnumUserClassAccess**

Returns the access and email alerting rights that one or more users have for the specified security classes.

**Syntax**

```vbscript
<HSVSecurityAccess>.EnumUserClassAccess varabstrUserSIDs, varabstrSecClasses, pvar2DalRights, pvar2DalAlertable
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varabstrUserSIDs</td>
<td>String array (ByVal). The user security identifiers.</td>
</tr>
<tr>
<td>varabstrSecClasses</td>
<td>String array (ByVal). The names of the security classes.</td>
</tr>
<tr>
<td>pvar2DalRights</td>
<td>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 varabstrUserSIDs argument's array, and the second dimension has a one-to-one correspondence with the varabstrSecClasses argument's array. IDs of access rights are represented by the enumeration described in &quot;Access Rights Constants&quot; on page 461.</td>
</tr>
<tr>
<td>pvar2DalAlertable</td>
<td>Variant. Returns a two-dimensional array containing bitmasks that represent the users' email alerting rights for the security classes. The first dimension has a one-to-one correspondence with the varabstrUserSIDs argument's array, and the second dimension has a one-to-one correspondence with the varabstrSecClasses argument's array. Valid values for the bits are represented by the enumeration described in &quot;E-mail Alerting Constants&quot; on page 462.</td>
</tr>
</tbody>
</table>

**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.
### EnumUsers3

Returns the security identifiers, usernames, and access types of users in the HSV Security Access object.

**Syntax**

```csharp
<HsvSecurityAccess>.EnumUsers3 AccessType, pvarabstrUserSIDs, pvarabstrNames
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessType</td>
<td>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.</td>
</tr>
<tr>
<td>pvarabstrUserSIDs</td>
<td>Variant. Returns an array containing the security identifiers. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarabstrNames</td>
<td>Variant. Returns an array containing the usernames. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

### EnumUsersInGroup

Returns the security identifiers, usernames, and identity types of the users in a user group.

**Syntax**

```csharp
<HsvSecurityAccess>.EnumUsersInGroup bstrGroupSID, lBitmaskFlags, pvarabstrUserSIDS, pvarabstrUserFullNames, pvaralIdentityTypes
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrGroupSID</td>
<td>String (ByVal). The group's security identifiers.</td>
</tr>
<tr>
<td>lBitmaskFlags</td>
<td>Long (ByVal). A bitmask that specifies the identity types to return in the pvaralIdentityTypes argument's array. Valid values for the bits are represented by the enumeration described in &quot;User Groups - User Type Flag Constants&quot; on page 462. For example, to exclude bits for application administrators from the array, pass HFM_USER_GROUP_ENUM_EXCLUDEADMINS.</td>
</tr>
<tr>
<td>pvarabstrUserSIDS</td>
<td>Variant. Returns an array that contains the user security identifiers.</td>
</tr>
<tr>
<td>pvarabstrUserFullNames</td>
<td>Variant. Returns an array that contains the users' usernames.</td>
</tr>
<tr>
<td>pvaralIdentityTypes</td>
<td>Variant. Returns an array of bitmasks that indicate the users' identity types. Valid values for the bits are represented by the enumeration described in &quot;Identity Type Constants&quot; on page 463.</td>
</tr>
</tbody>
</table>

### EnumUsersInRole

**Deprecated** - use **EnumUsersInRole2**.

### EnumUsersInRole2

Returns the security identifiers and usernames of the users and groups assigned to a 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lRoleID</td>
<td>Long (ByVal). The ID of the role. For a list of valid role IDs, see “Role ID Constants” on page 462.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> To get a role’s ID from its name, use GetRoleID.</td>
</tr>
<tr>
<td>pvarabstrUserSIDs</td>
<td>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 pvarabstrNames argument.</td>
</tr>
<tr>
<td>pvarabstrNames</td>
<td>Variant. Returns an array containing the usernames of the users and groups assigned to the role.</td>
</tr>
</tbody>
</table>

**EnumUsersInRole3**

Returns the security identifiers and usernames of the users and groups assigned to a 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lRoleID</td>
<td>Long (ByVal). The ID of the role. For a list of valid role IDs, see “Role ID Constants” on page 462.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> To get a role’s ID from its name, use GetRoleID.</td>
</tr>
<tr>
<td>vbExpandGroup</td>
<td>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.</td>
</tr>
<tr>
<td>pvarabstrUserSIDs</td>
<td>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 pvarabstrNames argument.</td>
</tr>
<tr>
<td>pvarabstrNames</td>
<td>Variant. Returns an array containing the usernames of the users and groups assigned to the role.</td>
</tr>
</tbody>
</table>

**EnumUsersInSecurityClass**

For internal use.

**EnumUsersInSecurityClass2**

For internal use.

**EnumUsersInSecurityClass3**

For internal use.
EnumUsersInSecurityClass4

For internal use.

EnumUsersOrGroups

Returns the security identifiers, usernames, and identity types of the users in a user group.

Syntax

```csharp
<HsvSecurityAccess>.EnumUsersOrGroups lBitmaskFlags, pvarabstrUserAndGroupSIDs, pvarabstrUserAndGroupFullNames, pvaralIdentityTypes
```

Argument Description

- **lBitmaskFlags** Long (ByVal). A bitmask that specifies the identity types to return in the `pvaralIdentityTypes` argument's array. Valid values for the bits are represented by the enumeration described in "User Groups - User Type Flag Constants" on page 462.
  
  For example, to exclude bits for application administrators from the array, pass `HFM_USER_GROUP_ENUM_EXCLUDEADMINS`.

- **pvarabstrUserAndGroupSIDs** Variant. Returns an array that contains the security identifiers of the users or groups.

- **pvarabstrUserAndGroupFullNames** Variant. Returns an array that contains the usernames of the users or groups.

- **pvaralIdentityTypes** 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 463.

EnumUsersWithFilter

Depreciated. 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

```csharp
<HsvSecurityAccess>.EnumUsersWithFilter2 bstrFilter, lBitFieldPrincipalsToReturn, AccessType, pvarabstrUserSIDs, pvarabstrNames
```

Argument Description

- **bstrFilter** String (ByVal). The string by which to search user and group names. For wildcard searches, use asterisks (*).

- **lBitFieldPrincipalsToReturn** Long (ByVal). A bitmask that represents the user categories by which to filter. Valid values for the bits are listed in Table 75 on page 463.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>AccessType</td>
<td>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.</td>
</tr>
<tr>
<td>pvarabstrUserSIDs</td>
<td>Variant array. Returns the security identifiers for the users and groups that match the search criteria. The array's subtype is String.</td>
</tr>
<tr>
<td>pvarabstrNames</td>
<td>Variant array. Returns the names of the users and groups that match the search criteria. The array's subtype is String.</td>
</tr>
</tbody>
</table>

Example

The following function returns an array of the user and group names that begin with the string passed.

```vba
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**

```vba
<HsvSecurityAccess>.GetAllSecurityClassRightsForConnectedUser pvaralSecClassIds, pvarabstrSecClassNames, pvaralRights
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvaralSecClassIds</td>
<td>Variant array. Returns the IDs of the application security classes. The array's subtype is Long.</td>
</tr>
<tr>
<td>pvarabstrSecClassNames</td>
<td>Variant array. Returns the names of the application’s security classes. The array's subtype is String.</td>
</tr>
<tr>
<td>pvaralRights</td>
<td>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 461. The array's subtype is Long.</td>
</tr>
</tbody>
</table>
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 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**
--- | ---
`pvarabstrUserSIDs` | Variant. Returns an array containing the security identifiers of the application’s users. The array's subtype is String.
`pvarabstrNames` | Variant. Returns an array containing the usernames of the application users. The array's subtype is String.
`pvarabHasAccess` | 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 identifiers and username of the connected user.

**Syntax**

```
<HsvSecurityAccess>.GetConnectedUser2 pbstrUserSID, pbstrName
```

**Argument** | **Description**
--- | ---
`pbstrUserSID` | String. Returns the user’s security identifiers.
`pbstrName` | String. Returns the username.

**Example**

GetConnectedUser2 is used in the example for EnumRolesForUser.
**GetIdentityTypes**

Returns the identity types of the specified users.

Syntax

```csharp
<HsvSecurityAccess>.GetIdentityTypes(varabstrSIDs)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varabstrSIDs</td>
<td>String array (ByVal). The security identifiers of the users.</td>
</tr>
</tbody>
</table>

**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 463. 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

```csharp
<HsvSecurityAccess>.GetNumRoles plNumRoles
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plNumRoles</td>
<td>Long. Returns the number of roles.</td>
</tr>
</tbody>
</table>

**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 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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lRoleID</code></td>
<td>Long (ByVal). The ID of the role. For a list of valid role IDs, see “Role ID Constants” on page 462. <strong>Tip:</strong> To get a role’s ID from its name, use <code>GetRoleID</code>.</td>
</tr>
<tr>
<td><code>pvarabstrUserSIDs</code></td>
<td>Variant. Returns an array containing the security identifiers of an application’s users.</td>
</tr>
<tr>
<td><code>pvarabstrNames</code></td>
<td>Variant. Returns an array containing the usernames of an application’s users.</td>
</tr>
<tr>
<td><code>pvarabHasAccess</code></td>
<td>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.</td>
</tr>
</tbody>
</table>

### GetRoleID

Returns the ID of a role, using the role’s name.

**Syntax**

```
<HsvSecurityAccess>.GetRoleID bstrRole, plRoleID
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrRole</code></td>
<td>String (ByVal). The name of the role.</td>
</tr>
<tr>
<td><code>plRoleID</code></td>
<td>Long. Returns the role ID. For a list of constants that represent role IDs, see “Role ID Constants” on page 462.</td>
</tr>
</tbody>
</table>

**Example**

GetRoleID is used in the example for `AddUserToRole2`.

### GetRoleLabel

Returns the name of a role, using a role ID.

**Syntax**

```
<HsvSecurityAccess>.GetRoleLabel lRoleID, pbstrRole
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lRoleID</code></td>
<td>Long (ByVal). The role ID. For a list of constants that represent role IDs, see “Role ID Constants” on page 462.</td>
</tr>
<tr>
<td><code>pbstrRole</code></td>
<td>String. Returns the role name.</td>
</tr>
</tbody>
</table>
**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**

```csharp
<HsvSecurityAccess>.GetSecurityClassAccessForAllUsers2 lSecurityClassID,
pvarabstrUserSIDs, pvarabstrNames, pvaralRights
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lSecurityClassID</code> Long (ByVal). The ID of the security class. You can get a security class's ID from its name with GetSecurityClassID.</td>
</tr>
<tr>
<td><code>pvarabstrUserSIDs</code> Variant. Returns an array containing the user security identifiers.</td>
</tr>
<tr>
<td><code>pvarabstrNames</code> Variant. Returns an array containing the users’ usernames.</td>
</tr>
<tr>
<td><code>pvaralRights</code> Variant. Returns an array containing the users’ access rights to the security class. Valid values are represented by the enumeration described in &quot;Access Rights Constants&quot; on page 461.</td>
</tr>
</tbody>
</table>

**GetSecurityClassID**

Returns the ID of a security class, using the security class’s name.

**Syntax**

```csharp
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrSecurityClass</code> String (ByVal). The name of the security class for which the ID is to be returned.</td>
</tr>
</tbody>
</table>

*Note:* If the security class does not exist, error 8004F003 (hexadecimal) occurs.

| `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, using a security class ID.

Syntax

```csharp
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSecurityClassID</td>
<td>Long (ByVal). The ID of the security class.</td>
</tr>
<tr>
<td>pbstrSecurityClass</td>
<td>String. Returns the name of the security class.</td>
</tr>
</tbody>
</table>

GetSecurityClassRightsForConnectedUser

Indicates the access rights that the connected user has to a security class.

Syntax

```csharp
'HsvSecurityAccess'.GetSecurityClassRightsForConnectedUser lSecurityClassID, plAccessRights
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSecurityClassID</td>
<td>Long (ByVal). The ID of the security class. You can get this ID with GetSecurityClassID.</td>
</tr>
<tr>
<td>plAccessRights</td>
<td>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 &quot;Access Rights Constants&quot; on page 461.</td>
</tr>
</tbody>
</table>

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 is executed.

```csharp
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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varalTaskIDs</td>
<td>Long array (ByVal). The array of task IDs. Use the IDs listed in “Task Constants” on page 462.</td>
</tr>
<tr>
<td>pvarabHasAccess</td>
<td>Variant array. Indicates whether the connected user has rights to the tasks passed in the varalTaskIDs 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.</td>
</tr>
</tbody>
</table>

**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 is 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```
### HsvSecurityAccess Object Methods

#### 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**

```vbnet
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrUserID</code></td>
<td>String (ByVal). The user security identifiers.</td>
</tr>
<tr>
<td><code>pvaralSecurityClassIDs</code></td>
<td>Variant. Returns the IDs of the security classes.</td>
</tr>
<tr>
<td><code>pvarabstrSecurityClasses</code></td>
<td>Variant. Returns the names of the security classes.</td>
</tr>
<tr>
<td><code>pvaralRights</code></td>
<td>Variant. Returns an array containing the users' access rights to the security classes. Valid values are represented by the enumeration described in &quot;Access Rights Constants&quot; on page 461.</td>
</tr>
</tbody>
</table>

#### 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, using the user’s external authentication token and security identifiers.

**Syntax**

```vbnet
<HsvSecurityAccess>.GetUserInfoFromUniqueID2 bstrSSOToken, bstrUniqueUserID, pvarbstrUserName, pbastrFullUserName, pvarbstrFirstName, pvarbstrLastName, pvarbstrDesc, pvarbstrEMail
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrSSOToken</td>
<td>String (ByVal). The user’s external authentication token.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> You can get the token with the HsxClient method <code>GetLogonInfoSSO</code>.</td>
</tr>
<tr>
<td>bstrUniqueUserID</td>
<td>String (ByVal). The user security identifiers.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> You can get a security identifiers with <code>GetUserSID</code>.</td>
</tr>
<tr>
<td>pvarbstrUserName</td>
<td>Variant. Returns the username.</td>
</tr>
<tr>
<td>pbastrFullUserName</td>
<td>Variant. Returns the user's fully qualified domain name.</td>
</tr>
<tr>
<td>pvarbstrFirstName</td>
<td>Variant. Returns the user's first name.</td>
</tr>
<tr>
<td>pvarbstrLastName</td>
<td>Variant. Returns the user's last name.</td>
</tr>
<tr>
<td>pvarbstrDesc</td>
<td>Variant. Returns the user's description.</td>
</tr>
<tr>
<td>pvarbstrEMail</td>
<td>Variant. Returns the user's email address.</td>
</tr>
</tbody>
</table>

**Example**

The following function returns the fully qualified domain name of the connected user.

```vbnet
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`.

---

486  HsvSecurityAccess Type Library
**GetUserName2**

Returns a username, using a security identifier.

Syntax

```
<HsvSecurityAccess>.GetUserName2 bstrUserSID, pbstrName
```

**Argument**    **Description**

*bstrUserSID*    String (ByVal). The security identifier.

*pbstrName*    String. Returns the username.

**GetUserNameFromSID**

Returns the username of a user, using the user’s security identifier (SID).

Syntax

```
<HsvSecurityAccess>.GetUserNameFromSID bstrSID, pbstrName
```

**Argument**    **Description**

*bstrSID*    String (ByVal). The security identifier.

*pbstrName*    String. Returns the username.

**GetUserSID**

Returns the security identifier for a user.

Syntax

```
<HsvSecurityAccess>.GetUserSID bstrName, pbstrUserSID
```

**Argument**    **Description**

*bstrName*    String (ByVal). The username of the user.

*pbstrUserSID*    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

pbIsAdmin Boolean. Returns TRUE if the connected user is assigned to the Application Administrator role, FALSE otherwise.

IsClassicHFMAplication

Indicates whether an application is a Classic application.

Syntax

< HSVSecurityAccess > . IsClassicHFMAplication()

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. See “GetTaskAccessForConnectedUserFromList” on page 483.

Syntax

< HSVSecurityAccess > . IsConnectedUserAllowedToPerformTask lTaskID, pbHasAccess

Argument Description

lTaskID Long (ByVal). The ID of the task. For a list of valid task IDs, see “Task Constants” on page 462.

pbHasAccess 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
--- | ---
lRoleID | Long (ByVal). The ID of the role to be tested. For a list of valid role IDs, see “Role ID Constants” on page 462.
pbUserIsInRole | 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 is 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```
**RemoveUser**

*Deprecated* - use **RemoveUser2**.

**RemoveUser2**

Removes a user from an application.

**Syntax**

```vba
<HsvSecurityAccess>.RemoveUser2 bstrUserSID
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrUserSID</td>
<td>String (ByVal). The security identifier of the user to remove.</td>
</tr>
</tbody>
</table>

**RemoveUserFromRole**

*Deprecated* - use **RemoveUserFromRole2**.

**RemoveUserFromRole2**

Removes a user from a role.
Syntax

<HsvSecurityAccess>.RemoveUserFromRole2 lRoleID, bstrUserSID

**Argument**  **Description**

lRoleID  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 462.

**Tip:** To get a role’s ID from its name, use GetRoleID.

bstrUserSID  String (ByVal). The user security identifier.

**Example**

The following subroutine removes a user from a 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 email 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**

bstrUserSID  String (ByVal). The user security identifier.

varalSecurityClassIDs  Long array (ByVal). The IDs of the security classes.

You can get a security class’s ID by passing its name to GetSecurityClassID.
### Argument | Description
--- | ---
`vcallRights` | 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 461.

`vcallEmailAlerting` | Long array (ByVal). An array of bitmasks that represent the email alerting rights to assign. Valid values for the bits are represented by the enumeration described in "E-mail Alerting Constants" on page 462.

### `SetRolesForUser`
Specifies one or more roles for a user. This method overwrites any previously assigned roles for the user.

**Syntax**
```csharp
<HsvSecurityAccess>.SetRolesForUser bstrUserSid, varalRoleIds
```

**Argument** | **Description**
--- | ---
`bstrUserSid` | String (ByVal). The user security identifier.

`vcallRoleIds` | Long array (ByVal). The IDs of the roles. For a list of valid role IDs, see “Role ID Constants” on page 462.

**Example**
The following example assigns the specified roles to a username. The user security identifier is obtained with `GetUserSID`.

```vbnet
Sub assignRolesByUsername(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!** This method fails if used against an application created with Performance Management Architect.
**Argument** | **Description**  
---|---  
`lSecurityClassID` | Long (ByVal). The ID of the security class. You can get this ID by passing the security class name to `GetSecurityClassID`.  
`bstrSecurityClass` | String (ByVal). The new name for the security class.  

**Example**

The following method renames a security class.

```vba
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 email alerting rights to a security class. User security identifiers, access right IDs, and email alerting flags are passed in arrays that have a one-to-one correspondence.

**Syntax**

```vba
<HsvSecurityAccess>.SetSecurityClassRightsForManyUsers2 lSecurityClassID, varabstrUserSIDs, varalRights, varalEmailAlerting
```

**Argument** | **Description**  
---|---  
`lSecurityClassID` | Long (ByVal). Long (ByVal). The ID of the security class. You can get this ID by passing the security class name to `GetSecurityClassID`.  
`varabstrUserSIDs` | String array (ByVal). The user security identifier.  
`varalRights` | 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 461.  
`varalEmailAlerting` | Long array (ByVal). An array of bitmasks that represent the email alerting rights to assign. Valid values for the bits are represented by the enumeration described in "E-mail Alerting Constants" on page 462.
SetSecurityClassRightsForUser

Deprecated - use SetSecurityClassRightsForUser2.

SetSecurityClassRightsForUser2

Sets a user’s access and email alerting rights for a security class.

Syntax

```<HsvSecurityAccess>.SetSecurityClassRightsForUser2 lSecurityClassID, bstrUserSID, lRights, lEmailAlerting```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lSecurityClassID</code></td>
<td>Long (ByVal). The ID of the security class. You can get this ID by passing the security class name to `GetSecurityClassID.</td>
</tr>
<tr>
<td><code>bstrUserSID</code></td>
<td>String (ByVal). The user security identifier.</td>
</tr>
<tr>
<td><code>lRights</code></td>
<td>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 461.</td>
</tr>
<tr>
<td><code>lEmailAlerting</code></td>
<td>Long (ByVal). An array of bitmasks that represent the email alerting rights to assign. Valid values for the bits are represented by the enumeration described in “E-mail Alerting Constants” on page 462.</td>
</tr>
</tbody>
</table>

SetUserClassAccess

Sets security class access and email alerting rights for one or more users.

Syntax

```<HsvSecurityAccess>.SetUserClassAccess varabstrUserSIDS, varabstrSecClasses, var2DalRights, var2DalAlertable```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>varabstrUserSIDS</code></td>
<td>String array (ByVal). The user security identifier.</td>
</tr>
<tr>
<td><code>varabstrSecClasses</code></td>
<td>String array (ByVal). The names of the security classes for which rights are being set.</td>
</tr>
<tr>
<td><code>var2DalRights</code></td>
<td>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. IDs of access rights are represented by the enumeration described in “Access Rights Constants” on page 461.</td>
</tr>
<tr>
<td><code>var2DalAlertable</code></td>
<td>Long array (ByVal). A two-dimensional array containing bitmasks that represent the users’ email 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. Valid values for the bits are represented by the enumeration described in “E-mail Alerting Constants” on page 462.</td>
</tr>
</tbody>
</table>
**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 95, and are described in detail in the following topics.

To assign IHsvDataSecurity interface object references, use the `Security` property of the HsvSession object.

**GetCellLevelAccessRights**

*Deprecated* - use “GetCellLevelAccessRightsExtDim” on page 495.

**GetCellLevelAccessRightsExtDim**

Returns the access rights that the connected user has to a cell. Supersedes GetCellLevelAccessRights.

**Syntax**

```plaintext
<IHsvDataSecurity>.GetCellLevelAccessRightsExtDim pIUnkHfmPovCOM, plAccessRights
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the desired POV.</td>
</tr>
<tr>
<td><code>plAccessRights</code></td>
<td>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 461.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation for how to set the POV.

**GetProcessUnitAccessRights**

Returns the access rights that the connected user has to a process unit.
**Syntax**

```c
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit's child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit's parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit's Value dimension member.</td>
</tr>
<tr>
<td>plAccessRights</td>
<td>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 <strong>&quot;Access Rights Constants&quot; on page 461.</strong></td>
</tr>
</tbody>
</table>

**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**

```c
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit's child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit's parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit's Value dimension member.</td>
</tr>
<tr>
<td>plAccessRights</td>
<td>Long. Returns a value that identifies the connected user’s access rights for the process unit. The valid return values are listed in <strong>&quot;Access Rights Constants&quot; on page 461.</strong></td>
</tr>
</tbody>
</table>

---

496  HsvSecurityAccess Type Library
### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>psProcessState</code></td>
<td>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 878.</td>
</tr>
</tbody>
</table>

**Example**

`GetProcessUnitAccessRightsAndState` is used in the example for `HsvProcessFlow.Start`.

### GetProcessUnitAccessRightsAndStateEx

*Deprecated* - use “GetProcessUnitAccessRightsAndStateExExtDim” on page 497.

### GetProcessUnitAccessRightsAndStateExExtDim

Returns the following information for a submission phase, using either the member IDs of a cell in the phase or the phase ID. Supersedes `GetProcessUnitAccessRightsAndStateEx`.

- 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**

```csharp
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing a POV. If you use Phase ID, pass MEMBERNOTUSED for the Account, ICP and Custom dimensions.</td>
</tr>
<tr>
<td><code>lPhaseID</code></td>
<td>Long (ByVal). Pass one of these values:</td>
</tr>
<tr>
<td></td>
<td>- The phase ID</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> You can obtain a phase ID with <code>GetGroupPhaseFromCell</code>.</td>
</tr>
<tr>
<td></td>
<td><strong>MEMBERNOTUSED</strong></td>
</tr>
<tr>
<td><code>plAccessRights</code></td>
<td>Long. Returns the number that identifies the connected user’s access rights for the submission phase. The valid return values are listed in Table 71, “Access Rights Constants,” on page 461.</td>
</tr>
<tr>
<td><code>psProcessState</code></td>
<td>Integer. Returns a value that identifies the submission phase review level. The valid return values are listed in Table 139, “CEnumProcessFlowStates Enumeration,” on page 878.</td>
</tr>
</tbody>
</table>
Return Value
None.

Example
See HfmPovCOM documentation for how to set the POV.

GetProcessUnitAccessRightsEx

Deprecated - use “GetProcessUnitAccessRightsExExtDim” on page 498.

GetProcessUnitAccessRightsExExtDim

Returns the access rights that the connected user has to a submission phase, using either the member IDs of a cell in the phase or the phase ID. Supersedes GetProcessUnitAccessRightsEx. You must pass one of these values:

- 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

```csharp
<IHsvDataSecurity>.GetProcessUnitAccessRightsExExtDim pIUnkHfmPovCOM, lPhaseID, plAccessRights
```

Argument Description

- `pIUnkHfmPovCOM` HfmPovCOM. HfmPovCOM object representing a POV. Specify MEMBERNOTUSED for Account, ICP and Custom dimensions if you specify a phase ID.
- `lPhaseID` Long (ByVal). Pass one of these values:
  - The phase ID.
  - **Note:** You can obtain a phase ID with `GetGroupPhaseFromCell`.
  - MEMBERNOTUSED
- `plAccessRights` Long. Returns the number that identifies the connected user’s access rights for the submission phase. The valid return values are listed in Table 71, “Access Rights Constants,” on page 461.

Return Value
None.

Example
See HfmPovCOM documentation for how to set the POV.
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 the HsvSystemInfo object.

**HsvSystemInfo Object Methods**

The HsvSystemInfo object’s methods are summarized in Table 28 on page 96, 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 96.

### AddRefToHsxServer

*For internal use.*

**Syntax**

```
<HsvSystemInfo>.AddRefToHsxServer plRefCount
```

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plRefCount</td>
<td>Long</td>
</tr>
</tbody>
</table>

### AddTaskToAudit

Adds a 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
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lActivityCode</td>
<td>Long (ByVal). The ID of the activity being added. Valid values are represented by the HFMConstants type library constants listed in &quot;User Activity Constants&quot; on page 895. However, Hyperion recommends that you pass only the USERACTIVITYCODE_EXTERNAL constant, which represents custom activities.</td>
</tr>
<tr>
<td>dStartTime</td>
<td>Double (ByVal). The timestamp of the activity’s start time. Pass a Double that can be cast to the Date format.</td>
</tr>
<tr>
<td>dEndTime</td>
<td>Double (ByVal). The timestamp of the activity’s end time. Pass a Double that can be cast to the Date format.</td>
</tr>
<tr>
<td>bstrDescription</td>
<td>String (ByVal). A description of the activity.</td>
</tr>
<tr>
<td>varabyAttachment</td>
<td>The attachment definition as an array of Bytes.</td>
</tr>
</tbody>
</table>

**AddTaskToAuditWithAttachment**

Adds a task for the user to the audit log with attachment. 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**

```plaintext
<HsvSystemInfo>.AddTaskToAuditWithAttachment lActivityCode, dStartTime, dEndTime, bstrDescription, varabyAttachment
```

**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**

```csharp
<HsvSystemInfo>.CheckAccess pvbHasAccess
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvbHasAccess</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 task, use `ClearAuditTasks2`.

**Syntax**

```csharp
<HsvSystemInfo>.ClearAuditTasks dStartTime, dEndTime, vbAllServers, bstrServer, vbAllUsers, lActivityUserID
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dStartTime</td>
<td>Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td>dEndTime</td>
<td>Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>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 <code>bstrServer</code> argument.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String (ByVal). The name of the application server for which to delete the task audit history. This argument is used only if the <code>vbAllServers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>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 <code>lActivityUserID</code> argument.</td>
</tr>
<tr>
<td>lActivityUserID</td>
<td>Long (ByVal). The activity user ID of the user for whom to delete the task audit history. This argument is used only if the <code>vbAllUsers</code> argument is set to FALSE.</td>
</tr>
</tbody>
</table>

**Example**

The following subroutine deletes the audit history for a user. The subroutine takes a username; the activity user ID required by `ClearAuditTasks` is obtained by passing the username to `GetActivityUserID`.

```csharp
Sub ClearUserTaskAudit(sUserName As String)
```

---

**CheckAccess** 503
Dim cSysInfo As HsvSystemInfo, lUserActID As Long
'm_cHsvSession is an HsvSession object reference
Set cSysInfo = m_cHsvSession.SystemInfo
Try
        cSysInfo.GetActivityUserID sUserName, lUserActID
        cSysInfo.ClearAuditTasks 0, CDb(Now()), True, "", _
                        False, lUserActID
End Try

ClearAuditTasks2

Deletes the audit history for a 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>dStartTime</td>
<td>Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td>dEndTime</td>
<td>Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>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 bstrServer argument.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String (ByVal). The name of the application server for which to delete the task audit history. This argument is used only if the vbAllServers argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>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 lActivityUserID argument.</td>
</tr>
<tr>
<td>lActivityUserID</td>
<td>Long (ByVal). The activity user ID of the user for whom to delete the task audit history. This argument is used only if the vbAllUsers argument is set to FALSE. To get a user’s activity user ID, use GetActivityUserID.</td>
</tr>
<tr>
<td>vbAllTasks</td>
<td>Boolean (ByVal). Specifies whether audit information for all tasks or a specific task is deleted. Pass TRUE for all tasks, FALSE to specify a task with the lActivityTaskID argument.</td>
</tr>
<tr>
<td>lActivityTaskID</td>
<td>Long (ByVal). The ID of the task for which to delete audit information. This argument is used only if the vbAllTasks argument is set to FALSE. Task IDs are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 895.</td>
</tr>
</tbody>
</table>

Example

The following subroutine deletes the audit history for a specific task. The subroutine takes the ID of the task to be deleted.

Sub ClearTaskAuditAllUsers(lTaskId As Long)
ClearRunningTask

For internal use.

DeleteUserParameter

Deletes a user parameter 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 cluster or application, use

HsxClient.DisableNewConnections.

To enable new connections, use EnableNewConnections.

Syntax

<
HsvSystemInfo
>.DisableNewConnections vbAllServers, bstrServer, vbAllUsers, lActivityUserID
**EnableNewConnections**

Enables new Financial Management connections for the specified user and application server criteria.

**Tip:** To enable new connections for a cluster or application, use `HsxClient.EnableNewConnections`.

To disable new connections, use `DisableNewConnections`.

**Syntax**

```<HsvSystemInfo>.EnableNewConnections vbAllServers, bstrServer, vbAllUsers, lActivityUserID```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vbAllServers</td>
<td>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 <code>bstrServer</code> argument.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String (ByVal). The name of the application server for which to enable connections. This argument is used only if the <code>vbAllServers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>Boolean (ByVal). Specifies whether to enable connections for all users. Pass TRUE for all users, FALSE to specify a user with the <code>lActivityUserID</code> argument.</td>
</tr>
<tr>
<td>lActivityUserID</td>
<td>Long (ByVal). The activity user ID of the user for whom to enable connections. This argument is used only if the <code>vbAllUsers</code> argument is set to FALSE. To get a user's activity user ID, use <code>GetActivityUserID</code>.</td>
</tr>
</tbody>
</table>

**EnumActivityServers**

Returns the names of the application servers for which there are task audit and data audit records.
Syntax

<HSVSystemInfo>.EnumActivityServers pvarabstrActivityServerNames

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvarabstrActivityServerNames</td>
<td>Variant. Returns an array containing the names of the servers. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

**EnumActivityUsers**

Returns the usernames of all users who have performed at least one activity in the application.

Syntax

<HSVSystemInfo>.EnumActivityUsers pvarabstrActivityUserNameNames

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvarabstrActivityUserNameNames</td>
<td>Variant array. Returns the usernames. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

**EnumActivityUsersForRunningTasks**

*For internal use.*

**EnumAuditTasks**

Returns task audit information from a 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 specific 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, pvarbstrModuleName, plTotalNumRecords```  

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dStartTime</code></td>
<td>Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td><code>dEndTime</code></td>
<td>Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td><code>vbAllServers</code></td>
<td>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 <code>bstrServer</code> argument.</td>
</tr>
<tr>
<td><code>bstrServer</code></td>
<td>String (ByVal). The name of the application server for which to return audit information. This argument is used only if the <code>vbAllServers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td><code>vbAllUsers</code></td>
<td>Boolean (ByVal). Specifies whether audit information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <code>lActivityUserID</code> argument.</td>
</tr>
<tr>
<td><code>lActivityUserID</code></td>
<td>Long (ByVal). The activity user ID of the user for whom to return audit information. This argument is used only if the <code>vbAllUsers</code> argument is set to FALSE. To get a user's activity user ID, use <code>GetActivityUserID</code>.</td>
</tr>
<tr>
<td><code>lStartRecord</code></td>
<td>Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td><code>lEndRecord</code></td>
<td>Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td><code>pvaralActivityUserID</code></td>
<td>Variant array. Returns the activity user IDs for the users who performed the tasks. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pvarabstrActivityUserName</code></td>
<td>Variant array. Returns the usernames of the users who performed the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pvaralActivitySessionID</code></td>
<td>Variant array. Returns the internal IDs of the sessions in which the tasks occurred. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pvarbstrServerName</code></td>
<td>Variant array. Returns the names of the application servers on which the tasks occurred. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pvaralActivityCode</code></td>
<td>Variant array. Returns the IDs of the tasks performed by the users. Valid values are represented by the HFMConstants type library constants listed in &quot;User Activity Constants&quot; on page 895. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pvaradStartTime</code></td>
<td>Variant array. Returns the starting times of the tasks. Array items are formatted as Doubles that can be cast to the Date format.</td>
</tr>
<tr>
<td><code>pvaradEndTime</code></td>
<td>Variant array. Returns the ending times of the tasks. Array items are formatted as Doubles that can be cast to the Date format.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$pvarbstrDescription$</td>
<td>Variant array. Returns the descriptions of the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>$pvarbstrModuleNames$</td>
<td>Variant array. Returns the names of the modules for the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>$plTotalNumRecords$</td>
<td>Long. Returns the total number of audit records in the database that meet the filtering criteria.</td>
</tr>
</tbody>
</table>

## EnumAuditTasks2

Returns task audit information from a 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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$dStartTime$</td>
<td>Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td>$dEndTime$</td>
<td>Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td>$vbAllServers$</td>
<td>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.</td>
</tr>
<tr>
<td>$bstrServer$</td>
<td>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.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>Boolean (ByVal). Specifies whether audit information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the lActivityUserID argument.</td>
</tr>
<tr>
<td>lActivityUserID</td>
<td>Long (ByVal). The activity user ID of the user for whom to return audit information. This argument is used only if the vbAllUsers argument is set to FALSE. To get a user’s activity user ID, use GetActivityUserID.</td>
</tr>
<tr>
<td>vbAllTasks</td>
<td>Boolean (ByVal). Specifies whether audit information for all tasks or a specific task is returned. Pass TRUE for all tasks, FALSE to specify a task with the lActivityTaskID argument.</td>
</tr>
<tr>
<td>lActivityTaskID</td>
<td>Long (ByVal). The ID of the task for which to return audit information. This argument is used only if the vbAllTasks argument is set to FALSE. Task IDs are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 895.</td>
</tr>
<tr>
<td>lStartRecord</td>
<td>Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td>lEndRecord</td>
<td>Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td>pvaralActivityUserID</td>
<td>Variant array. Returns the activity user IDs for the users who performed the tasks. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarabstrActivityUserName</td>
<td>Variant array. Returns the usernames of the users who performed the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralActivitySessionID</td>
<td>Variant array. Returns the internal IDs of the sessions in which the tasks occurred. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarbstrServerName</td>
<td>Variant array. Returns the names of the application servers on which the tasks occurred. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralActivityCode</td>
<td>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 895. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaradStartTime</td>
<td>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.</td>
</tr>
<tr>
<td>pvaradEndTime</td>
<td>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.</td>
</tr>
<tr>
<td>pvarbstrDescription</td>
<td>Variant array. Returns the descriptions of the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarbstrModuleName</td>
<td>Variant array. Returns the names of the modules for the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pTotalNumRecords</td>
<td>Long. Returns the total number of audit records in the database that meet the filtering criteria.</td>
</tr>
</tbody>
</table>
EnumAuditTasksEx

Returns task audit information from a 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 specific 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 `EnumAuditTasksEx`. For example, a user might delete some or all of the audit records.

**Syntax**

```csharp
< HvSystemInfo >.EnumAuditTasksEx dStartTime, dEndTime, vbAllServers, bstrServer, vbAllUsers, lActivityUserID, vbAllTasks, lActivityTaskID, lStartRecord, lEndRecord, pvaralActivityUserID, pvarabstrActivityUserName, pvaralActivitySessionID, pvarbstrServerName, pvaralActivityCode, pvaradStartTime, pvaradEndTime, pvarbstrDescription, pvarbstrModuleNames, pvarabstrGuids, plTotalNumRecords
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dStartTime</code></td>
<td>Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td><code>dEndTime</code></td>
<td>Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td><code>vbAllServers</code></td>
<td>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 <code>bstrServer</code> argument.</td>
</tr>
<tr>
<td><code>bstrServer</code></td>
<td>String (ByVal). The name of the application server for which to return audit information. This argument is used only if the <code>vbAllServers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td><code>vbAllUsers</code></td>
<td>Boolean (ByVal). Specifies whether audit information for all users is returned. Pass TRUE for all users, FALSE to specify a user with the <code>lActivityUserID</code> argument.</td>
</tr>
<tr>
<td><code>lActivityUserID</code></td>
<td>Long (ByVal). The activity user ID of the user for whom to return audit information. This argument is used only if the <code>vbAllUsers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td><code>vbAllTasks</code></td>
<td>Boolean (ByVal). Specifies whether audit information for all tasks or a specific task is returned. Pass TRUE for all tasks, FALSE to specify a task with the <code>lActivityTaskID</code> argument.</td>
</tr>
<tr>
<td><code>lActivityTaskID</code></td>
<td>Long (ByVal). The ID of the task for which to return audit information. This argument is used only if the <code>vbAllTasks</code> argument is set to FALSE.</td>
</tr>
</tbody>
</table>

Task IDs are represented by the HFMConstants type library constants listed in "User Activity Constants" on page 895.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lStartRecord</td>
<td>Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td>lEndRecord</td>
<td>Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td>pvaralActivityUserID</td>
<td>Variant array. Returns the activity user IDs for the users who performed the tasks. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarabstrActivityUserName</td>
<td>Variant array. Returns the usernames of the users who performed the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralActivitySessionID</td>
<td>Variant array. Returns the internal IDs of the sessions in which the tasks occurred. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarbstrServerName</td>
<td>Variant array. Returns the names of the application servers on which the tasks occurred. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralActivityCode</td>
<td>Variant array. Returns the IDs of the tasks performed by the users. Valid values are represented by the HFMConstants type library constants listed in &quot;User Activity Constants&quot; on page 895. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaradStartTime</td>
<td>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.</td>
</tr>
<tr>
<td>pvaradEndTime</td>
<td>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.</td>
</tr>
<tr>
<td>pvarbstrDescription</td>
<td>Variant array. Returns the descriptions of the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarbstrModuleNames</td>
<td>Variant array. Returns the names of the modules for the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarabstrGuids</td>
<td>Variant array. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pITotalNumRecords</td>
<td>Long. Returns the total number of audit records in the database that meet the filtering criteria.</td>
</tr>
</tbody>
</table>

**EnumProhibitConnections**

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

**Tip:** You can return information on connections disabled for a cluster with HsxClient.EnumProhibitConnections.
Syntax

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| `pvaravbAllApps`    | Variant array. Indicates whether connections are disabled for all applications. Array items can contain 0 or -1:  
  - -1 indicates connections to all applications are disabled.  
  - 0 indicates that only connections to the application returned by the corresponding `pvarabstrAppNames` argument's array item are disabled.  

  The array is returned as a Long subtype.                                                                                                                                                                                                 |
| `pvarabstrAppNames` | Variant array. Returns the application names for which connections are disabled. Application names are returned only when the corresponding `pvaravbAllApps` argument's array item contains 0.  
  Note: If the corresponding `pvaravbAllApps` argument's array item contains -1, this array item contains the string "AllApps".  

  The array is returned as a String subtype.                                                                                                                                                                                                 |
| `pvaravbAllServers` | Variant array. Indicates whether connections are disabled for all application servers. Array items can contain 0 or -1:  
  - -1 indicates connections to all application servers are disabled.  
  - 0 indicates that only connections to the application server returned by the corresponding `pvarabstrServerNames` argument's array item are disabled.  

  The array is returned as a Long subtype.                                                                                                                                                                                                 |
| `pvarabstrServerNames` | Variant array. Returns the application server names for which connections are disabled. Application server names are returned only when the corresponding `pvaravbAllServers` argument's array item contains 0.  
  Note: If the corresponding `pvaravbAllServers` argument's array item contains -1, this array item contains the string "AllServers".  

  The array is returned as a String subtype.                                                                                                                                                                                                 |
| `pvaravbAllUsers`   | Variant array. Indicates whether connections are disabled for all users. Array items can contain 0 or -1:  
  - -1 indicates connections for all users are disabled.  
  - 0 indicates that only connections for the user represented by the corresponding `pvaralActivityUserIDs` and `pvaralActivityUserNames` arguments' array items are disabled.  

  The array is returned as a Long subtype.                                                                                                                                                                                                 |
| `pvaralActivityUserIDs` | Variant array. Returns the activity user IDs of the users for whom connections are disabled. Valid IDs are returned only when the corresponding `pvaravbAllUsers` argument's array item contains 0.  
  Note: If the corresponding `pvaravbAllUsers` argument's array item contains -1, this array item contains -1.  

  The array is returned as a Long subtype.                                                                                                                                                                                                 |
**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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>vbAllTaskTypes</code></td>
<td>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 <code>lTaskType</code> argument.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| lTaskType         | Long (ByVal). The ID of the task type for which to return information. This argument is used only if the vbAllTaskTypes argument is set to FALSE. Following are the constants that represent valid values. These are from the HFMConstants enumeration tagUSERACTIVITYCODE:  
  - 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: unmatch all intercompany transactions  
  - USERACTIVITYCODE_IC_AUTOMATCHBYID: automatch intercompany transactions by ID  
  - USERACTIVITYCODE_IC_AUTOMATCHBYACCOUNT: automatch intercompany transactions by account  
  - USERACTIVITYCODE_IC_MATCHINGREPORTBYID: matching intercompany transactions by ID report  
  - USERACTIVITYCODE_IC_MATCHINGREPORTBYACCOUNT: matching intercompany transactions by account report  
  - USERACTIVITYCODE_IC_TRANSACTIONRPT: intercompany transactions report  
| vbAllUsers        | Boolean (ByVal). Specifies whether to filter by user. Pass TRUE for all users, FALSE to filter by the user specified with the lUserID argument.  
| lUserID           | Long (ByVal). The activity user ID of the user for whom to return task information. This argument is used only if the vbAllUsers argument is set to FALSE.  
| vbAllServers      | 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 bstrServerName argument.  
| bstrServerName    | String (ByVal). The name of the application server for which to return task information. This argument is used only if the vbAllServers argument is set to FALSE.  
| vbAllSessions     | 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.  
| vbAllStatus       | Boolean (ByVal). Specifies whether to filter by task status. Pass TRUE for all statuses, FALSE to filter by the status specified with the lStatus argument.  
| lStatus           | Long (ByVal). The task status by which to filter. This argument is used only if the vbAllStatus argument is set to FALSE. Valid values are represented by the HFMConstants type library constants listed in "Task Status Constants" on page 897.  
| lStartRecord      | Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.  
| lEndRecord        | Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.  
| pvaralTaskIDs     | Variant. Returns an array containing the task IDs that identify the running tasks. The array is returned as a Long subtype.  
| pvaralTaskTypes    | Variant. Returns an array containing the tasks' types. Valid values are represented by the constants listed as valid values for the lTaskType argument. The array is returned as a Long subtype. |
Argument | Description
--- | ---
pvaralTaskProgress | Variant. Returns an array containing the progress complete percentages of the running tasks. The array is returned as a Long subtype.
pvaralTaskStatus | 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 897. The array is returned as a Long subtype.
pvarabstrUserNames | Variant. Returns an array containing the usernames of the running tasks’ users. The array is returned as a String subtype.
pvarabstrServerNames | Variant. Returns an array containing the names of the application servers for the running tasks. The array is returned as a String subtype.
pvaradStartTimes | 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.
pvaradStartRunningTimes | 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.
pvaradLastUpdateTimes | 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.
pvarabstrDescriptions | Variant. Returns an array containing the descriptions for the tasks. The array is returned as a String subtype.
pvarabstrLogFiles | Variant. Returns an array containing the file names and paths of the tasks’ log files. The array is returned as a String subtype.
plTotalRecords | Long. Returns a count of the number of tasks returned by EnumRunningTasks.

**EnumRunningTasksEx**

Returns information about the running tasks that meet the selection criteria; the information returned includes an array of flags that indicate whether the tasks currently are running or stopped. Criteria include task type, user, application server, user session, and task status. Task information is returned in arrays that have a one-to-one correspondence.

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

**Note:** The number of matching records can change after you call 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**
---|---
**vbAllTaskTypes** | 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 *lTaskType* argument.

**lTaskType** | Long (ByVal). The ID of the task type for which to return information. This argument is used only if the *vbAllTaskTypes* argument is set to FALSE.

Following are the constants that represent valid values. These are from the HFMConstants enumeration tagUSERACTIVITYCODE:

- **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**: unmatch 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

**vbAllUsers** | Boolean (ByVal). Specifies whether to filter by user. Pass TRUE for all users, FALSE to filter by the user specified with the *lUserID* argument.

**lUserID** | Long (ByVal). The activity user ID of the user for whom to return task information. This argument is used only if the *vbAllUsers* argument is set to FALSE.

To get a user’s activity user ID, use *GetActivityUserID*.

**vbAllServers** | 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 *bstrServerName* argument.

**bstrServerName** | String (ByVal). The name of the application server for which to return task information. This argument is used only if the *vbAllServers* argument is set to FALSE.

**vbAllSessions** | 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.

**vbAllStatus** | Boolean (ByVal). Specifies whether to filter by task status. Pass TRUE for all statuses, FALSE to filter by the status specified with the *lStatus* argument.

**lStatus** | Long (ByVal). The task status by which to filter. This argument is used only if the *vbAllStatus* argument is set to FALSE. Valid values are represented by the HFMConstants type library constants listed in “Task Status Constants” on page 897.
### 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.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>iStartRecord</td>
<td>Long (ByVal). The index of the first record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td>iEndRecord</td>
<td>Long (ByVal). The index of the last record in the range of records to retrieve. This is a zero-based index.</td>
</tr>
<tr>
<td>pvaralTaskIDs</td>
<td>Variant. Returns an array containing the task IDs that identify the running tasks. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralTaskTypes</td>
<td>Variant. Returns an array containing the tasks' types. Valid values are represented by the constants listed as valid values for the lTaskType argument. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralTaskProgress</td>
<td>Variant. Returns an array containing the progress complete percentages of the running tasks. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralTaskStatus</td>
<td>Variant. Returns an array containing the tasks' statuses. Valid values are represented by the HFMConstants type library constants listed in &quot;Task Status Constants&quot; on page 897. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarabstrUserNames</td>
<td>Variant. Returns an array containing the usernames of the running tasks' users. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarabstrServerNames</td>
<td>Variant. Returns an array containing the names of the application servers for the running tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaradStartTimes</td>
<td>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.</td>
</tr>
<tr>
<td>pvaradStartRunningTimes</td>
<td>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.</td>
</tr>
<tr>
<td>pvaradLastUpdateTime</td>
<td>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.</td>
</tr>
<tr>
<td>pvarabstrDescriptions</td>
<td>Variant. Returns an array containing the descriptions for the tasks. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarbStopTaskFlags</td>
<td>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.</td>
</tr>
<tr>
<td>pvarabstrLogFiles</td>
<td>Variant. Returns an array containing the file names and paths of the tasks' log files. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>plTotalRecords</td>
<td>Long. Returns a count of the number of tasks returned by EnumRunningTasksEx.</td>
</tr>
</tbody>
</table>
### Syntax

```
<HsvSystemInfo>.EnumRunningTasksPOV lTaskType, vbAllUsers, lUserID, vbAllServers, bstrServerName, vbAllSessions, pvaralTaskIds, pvaralTaskProgress, pvaralTaskStatus, pvarabstrUserNames, pvarabstrServerNames, pvaralEntity, pvaralParent, pvaralScenario, pvaralYear, pvaralStartPeriod, pvaralEndPeriod, pvaralConsolType, plTotalRecords
```

### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lTaskType</td>
<td>Long (ByVal). Pass the HFMConstants type library constant <code>USERACTIVITYCODE_CONSOLIDATION</code>.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>Boolean (ByVal). Specifies whether to filter by user. Pass TRUE for all users, FALSE to filter by the user specified with the <code>lUserID</code> argument.</td>
</tr>
<tr>
<td>lUserID</td>
<td>Long (ByVal). The activity user ID of the user for whom to return task information. This argument is used only if the <code>vbAllUsers</code> argument is set to FALSE. To get a user’s activity user ID, use <code>GetActivityUserID</code>.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>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 <code>bstrServerName</code> argument.</td>
</tr>
<tr>
<td>bstrServerName</td>
<td>String (ByVal). The name of the application server for which to return task information. This argument is used only if the <code>vbAllServers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllSessions</td>
<td>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.</td>
</tr>
<tr>
<td>pvaralTaskIds</td>
<td>Variant. Returns an array containing the task IDs that identify the running tasks. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralTaskProgress</td>
<td>Variant. Returns an array containing the progress complete percentages of the consolidations. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralTaskStatus</td>
<td>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 897. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarabstrUserNames</td>
<td>Variant. Returns an array containing the usernames of the consolidations' users. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarabstrServerNames</td>
<td>Variant. Returns an array containing the names of the application servers for the consolidations. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaralEntity</td>
<td>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.</td>
</tr>
<tr>
<td>pvaralParent</td>
<td>Variant. Returns an array containing the member IDs of the parents of the entities returned by the <code>pvaralEntity</code> argument. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralScenario</td>
<td>Variant. Returns an array containing the member IDs of the Scenario dimension members for the consolidations. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralYear</td>
<td>Variant. Returns an array containing the member IDs of the Year dimension members for the consolidations. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralStartPeriod</td>
<td>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.</td>
</tr>
</tbody>
</table>
### HsvSystemInfo Type Library

**ExtractTaskAudit**

Extracts to a file the task audit records that meet the specified criteria. Criteria include date range, application server, user, and task.

**Syntax**

```plaintext
<HsvSystemInfo>.ExtractTaskAudit bstrClientFileName, bstrClientLogFileName,
bstrDelimitChar, dStartTime, dEndTime, vbAllServers, bstrServer, vbAllUsers,
lActivityUserID, vbAllTasks, lActivityTaskID
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClientFileName</td>
<td>String (ByVal). The name and path of the extract file.</td>
</tr>
<tr>
<td>bstrClientLogFileName</td>
<td>String (ByVal). The name and path of the log file for the extract operation.</td>
</tr>
<tr>
<td>bstrDelimitChar</td>
<td>String (ByVal). The delimiter for records in the extract file.</td>
</tr>
<tr>
<td>dStartTime</td>
<td>Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td>dEndTime</td>
<td>Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td>vbAllServers</td>
<td>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 <code>bstrServer</code> argument.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String (ByVal). The name of the application server for which to extract task audit records. This argument is used only if the <code>vbAllServers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>Boolean (ByVal). Specifies whether to extract task audit records for all users. Pass TRUE for all users, FALSE to specify a user with the <code>lActivityUserID</code> argument.</td>
</tr>
<tr>
<td>lActivityUserID</td>
<td>Long (ByVal). The activity user ID of the user for whom to extract task audit records. This argument is used only if the <code>vbAllUsers</code> argument is set to FALSE.</td>
</tr>
<tr>
<td>To get a user's activity user ID, use <code>GetActivityUserID</code>.</td>
<td></td>
</tr>
<tr>
<td>vbAllTasks</td>
<td>Boolean (ByVal). Specifies whether to extract task audit records for all tasks or a specific task. Pass TRUE for all tasks, FALSE to specify a task with the <code>lActivityTaskID</code> argument.</td>
</tr>
<tr>
<td>lActivityTaskID</td>
<td>Long (ByVal). The ID of the task for which to extract audit records. This argument is used only if the <code>vbAllTasks</code> argument is set to FALSE.</td>
</tr>
<tr>
<td>Task IDs are represented by the HFMConstants type library constants listed in &quot;User Activity Constants&quot; on page 895.</td>
<td></td>
</tr>
</tbody>
</table>
**Example**

The following example extracts task audit records for all logons.

```vba
Dim cSysInfo As HsvSystemInfo
'g_cSession is an HSVSession object reference
Set cSysInfo = g_cSession.SystemInfo
C SysInfo.ExtractTaskAudit "c:\temp\Audit.txt", "c:\temp\Audit.log", _
    ",", 0, CDbl(Now()), True, ",", True, -1, False, USERACTIVITYCODE_LOGON
```

**ExtractTaskAudit2**

Extracts to a file the task audit records that meet the specified criteria. Criteria include date range, application server, user, and task. ExtractTaskAudit2 improves performance over ExtractTaskAudit significantly by reducing the number of SQL queries to database.

**Syntax**

```
<HsvSystemInfo>.ExtractTaskAudit2 bstrClientFileName, bstrClientLogFileName, bstrDelimitChar, dStartTime, dEndTime, vbAllServers, bstrServer, vbAllUsers, lActivityUserID, vbAllTasks, lActivityTaskID
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>bstrClientFileName</strong></td>
<td>String (ByVal). The name and path of the extract file.</td>
</tr>
<tr>
<td><strong>bstrClientLogFileName</strong></td>
<td>String (ByVal). The name and path of the log file for the extract operation.</td>
</tr>
<tr>
<td><strong>bstrDelimitChar</strong></td>
<td>String (ByVal). The delimiter for records in the extract file.</td>
</tr>
<tr>
<td><strong>dStartTime</strong></td>
<td>Double (ByVal). The starting date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td><strong>dEndTime</strong></td>
<td>Double (ByVal). The ending date and time of the date range. Pass a Double that can be cast to a Date format.</td>
</tr>
<tr>
<td><strong>vbAllServers</strong></td>
<td>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 <strong>bstrServer</strong> argument.</td>
</tr>
<tr>
<td><strong>bstrServer</strong></td>
<td>String (ByVal). String (ByVal). The name of the application server for which to extract task audit records. This argument is used only if the <strong>vbAllServers</strong> argument is set to FALSE.</td>
</tr>
<tr>
<td><strong>vbAllUsers</strong></td>
<td>Boolean (ByVal). Specifies whether to extract task audit records for all users. Pass TRUE for all users, FALSE to specify a user with the <strong>lActivityUserID</strong> argument.</td>
</tr>
<tr>
<td><strong>lActivityUserID</strong></td>
<td>Long (ByVal). The activity user ID of the user for whom to extract task audit records. This argument is used only if the <strong>vbAllUsers</strong> argument is set to FALSE. To get a user's activity user ID, use <strong>GetActivityUserID</strong>.</td>
</tr>
<tr>
<td><strong>vbAllTasks</strong></td>
<td>Boolean (ByVal). Specifies whether to extract task audit records for all tasks or a specific task. Pass TRUE for all tasks, FALSE to specify a task with the <strong>lActivityTaskID</strong> argument.</td>
</tr>
<tr>
<td><strong>lActivityTaskID</strong></td>
<td>Long (ByVal). The ID of the task for which to extract audit records. This argument is used only if the <strong>vbAllTasks</strong> argument is set to FALSE. Task IDs are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 895.</td>
</tr>
</tbody>
</table>

ExtractTaskAudit2 521
GetActivityCodeDesc

Returns the description of a specified type of activity.

Syntax

```<HsvSystemInfo>.GetActivityCodeDesc lActivityCode, bstrDesc```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lActivityCode</code></td>
<td>Long (ByVal). The ID of the activity. Valid values are represented by the HFMConstants type library constants listed in “User Activity Constants” on page 895.</td>
</tr>
<tr>
<td><code>bstrDesc</code></td>
<td>String. The description of the activity.</td>
</tr>
</tbody>
</table>

GetActivityUserID

Returns the activity user ID for a specified 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 464.

Syntax

```<HsvSystemInfo>.GetActivityUserID bstrUserName, plActivityUserID```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrUserName</code></td>
<td>String (ByVal). The username of the user.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The username should include the domain.</td>
</tr>
<tr>
<td><code>plActivityUserID</code></td>
<td>Long. Returns the activity user ID for the user.</td>
</tr>
</tbody>
</table>

Example

GetActivityUserID is used in the example for ClearAuditTasks.

GetAllInfoForTask

*For internal use.*

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` returns this path:

`C:\Financial Management\Acme\`

**Tip:** Financial Management automatically creates an application folder when it generates items such as reports. If no such items were 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 creates when it generates items such as reports.

### Syntax

```vbnet
<HsvSystemInfo>.GetApplicationDirectory()
```

### Return Value

String. Returns the path of the application directory.

### Example

This example prints the application directory to the Immediate window.

```vbnet
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

```vbnet
<HsvSystemInfo>.GetApplicationName()
```

### Return Value

String. Returns the application name.

### Example

The following example assigns the application name to the `sApp` variable.

```vbnet
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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbstrAppName</td>
<td>String. Returns the name of the application to which the user is logged on.</td>
</tr>
<tr>
<td>pbstrServerName</td>
<td>String. Returns the name of the application server to which the user is logged on.</td>
</tr>
<tr>
<td>plActivityUserID</td>
<td>Long. Returns the activity user ID for the user.</td>
</tr>
<tr>
<td>plCurrentActivity</td>
<td>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 895.</td>
</tr>
<tr>
<td>pbstrModuleName</td>
<td>String. Returns the name of the Financial Management module in which the user is working.</td>
</tr>
<tr>
<td>pdTimeStarted</td>
<td>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.</td>
</tr>
<tr>
<td>pbstrDescription</td>
<td>String. Returns the description of the user’s current activity.</td>
</tr>
</tbody>
</table>

GetExtractFileEncoding

Returns the file encoding type for extracted files.

Syntax

```<HsvSystemInfo>.GetExtractFileEncoding plExtractFileEncoding```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plExtractFileEncoding</td>
<td>Long. Returns the type of file encoding for extracted files. For valid values, see “Extracted File Encoding Constants” on page 894.</td>
</tr>
</tbody>
</table>
**GetFormattedDateTime**

Returns a string representation of the specified double-byte date/time value using the language ID for the connected user.

Syntax

```csharp
<HsvSystemInfo>.GetFormattedDateTime dDateTime, bDateValueOnly, bTimeValueOnly, pbstrDateTime
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>dDateTime</code></td>
<td>Double (ByVal). The timestamp.</td>
</tr>
<tr>
<td><code>bDateValueOnly</code></td>
<td>Long (ByVal). TRUE to return only the date value.</td>
</tr>
<tr>
<td><code>bTimeValueOnly</code></td>
<td>Long (ByVal). TRUE to return only the time value.</td>
</tr>
<tr>
<td><code>pbstrDateTime</code></td>
<td>String. Returned string.</td>
</tr>
</tbody>
</table>

**GetFormattedResourceString**

*For internal use.*

**GetKillUsersStatus**

*For internal use.*

**GetKillUserStatus**

Indicates whether an administrator has logged off the current user.

Syntax

```csharp
<HsvSystemInfo>.GetKillUserStatus pvbKill, pvbWarn, pdTimestamp
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvbKill</code></td>
<td>Boolean. Indicates whether the current user was logged off. Returns TRUE if the user is logged off, FALSE otherwise.</td>
</tr>
<tr>
<td><code>pvbWarn</code></td>
<td>Boolean. <em>For internal use.</em></td>
</tr>
<tr>
<td><code>pdTimestamp</code></td>
<td>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.</td>
</tr>
</tbody>
</table>

**GetLanguageUserParameters**

Gets the language in which member descriptions are displayed for the connected user.
Syntax

```
<HsvSystemInfo>.GetLanguageUserParameters plLanguageID
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>GetLanguageUserParameters</td>
</tr>
</tbody>
</table>

**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.*

---

**GetLastModifiedDateAndModifiedByForArtifact**

*For internal use.*

---

**GetModuleName**

Returns the name of the module represented by a module ID.

**Syntax**

```
<HsvSystemInfo>.GetModuleName lModuleID, bstrModName
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lModuleID</td>
</tr>
</tbody>
</table>

| 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**

*pwcDecimalChar* Integer. Returns the double-byte integer equivalent of the user’s decimal character.

*pwcThousandsChar* 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.

```vba
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**

```vba
<HsvSystemInfo>.GetResourceLanguageUserParameters plResourceLanguageID
```

**Argument Description**

*plResourceLanguageID* Long. Returns the language’s ID. The valid IDs are represented by the HFMConstants type library constants listed in “Supported Language Constants” on page 856.

**GetResourceString**

*For internal use.*

**GetResourceStringFromHR**

*For internal use.*

**GetRunningTaskLogFilePathName**

Returns the name and path of the log file for a running task.

**Syntax**

```vba
<HsvSystemInfo>.GetRunningTaskLogFilePathName lTaskID, pbstrLogFilePathName
```
### Argument | Description
--- | ---

**lTaskID** | Long (ByVal). The ID of the task.  
**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 running task’s percentage complete, status, last update time, and description.

**Syntax**

\(<HsvSystemInfo>.GetRunningTaskProgress lTaskID, plProgress, plStatus, pdLastUpdateTime, pbstrDesc\)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lTaskID</strong></td>
<td>Long (ByVal). The ID of the task.</td>
</tr>
<tr>
<td><strong>Tip:</strong></td>
<td>You can obtain the IDs of all running tasks with <strong>EnumRunningTasks</strong>.</td>
</tr>
<tr>
<td><strong>plProgress</strong></td>
<td>Long. Returns the progress complete percentage of the task.</td>
</tr>
<tr>
<td><strong>plStatus</strong></td>
<td>Long. Returns the status of the task. Valid values are represented by the HFMCollections type library constants listed in “Task Status Constants” on page 897.</td>
</tr>
<tr>
<td><strong>pdLastUpdateTime</strong></td>
<td>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.</td>
</tr>
<tr>
<td><strong>pbstrDesc</strong></td>
<td>String. Returns the task’s description.</td>
</tr>
</tbody>
</table>

### GetRunningTasksCount

Returns the number of currently running tasks for the application.

**Syntax**

\(<HsvSystemInfo>.GetRunningTasksCount plRunningTasksCount\)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>plRunningTasksCount</strong></td>
<td>Long. Returns the number of tasks that are running.</td>
</tr>
</tbody>
</table>

### GetRunningTaskStatus

Returns the status of a running task.
Syntax

`<HsvSystemInfo>.GetRunningTaskStatus ltTaskID, plStatus`

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>ltTaskID</code></td>
<td>Long (ByVal). The ID of the task.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> You can obtain the IDs of all running tasks with <code>EnumRunningTasks</code>.</td>
</tr>
<tr>
<td><code>plStatus</code></td>
<td>Long. Returns the status of the task. Valid values are represented by the HFMConstants type library constants listed in &quot;Task Status Constants&quot; on page 897.</td>
</tr>
</tbody>
</table>

**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.

```vba
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
```

**GetTaskAuditAttachment**

Returns the names of the attachments for a specified task audit.

Syntax

`<HsvSystemInfo>.GetTaskAuditAttachment bstrGuid, pvarabyAttachment`

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrGuid</code></td>
<td>GUID of the attachment that needs to be retrieved. This GUID can be retrieved by calling <code>EnumAuditTasksEx</code>.</td>
</tr>
<tr>
<td><code>pvarabyAttachment</code></td>
<td>Variant array. Returns the name of the attachment.</td>
</tr>
</tbody>
</table>
**GetUserName**

Returns the username of the connected user.

**Syntax**

```vbscript
<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.

```vbscript
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 set with `SetUserParameter`. See “SetUserParameter” on page 537.

**Syntax**

```vbscript
<HsvSystemInfo>.GetUserParameter bstrKey, pvarabyData
```

**Argument**  
**Description**

- `bstrKey`  
  String (ByVal). The key that identifies the parameter.

- `pvarabyData`  
  Variant. Returns the value set for the parameter. This value is returned as a Byte subtype.

**Example**

This example tests whether a parameter named “Projects” is defined for the user. If no parameter is defined, any code placed within the If structure is executed.

```vbscript
Dim cHsvSystemInfo As HsvSystemInfo, vaGetParam
'm_cHsvSession is an HsvSession object reference
Set cHsvSystemInfo = m_cHsvSession.SystemInfo
If IsEmpty(vaGetParam) = True Then
    ...
End If
```
GetVBScriptCalcRules

Returns the rules 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) loaded into an application. If the rules file changed after the file was loaded, the updated file’s contents are not returned by GetVBScriptCalcRules.

**Syntax**

```
<HsvSystemInfo>.GetVBScriptCalcRules pvarabRules, pbRulesExist
```

**Argument**  | **Description**  
---|---
*pvarabRules* | Variant array. Returns the application’s rules. The array is returned as a Byte subtype.  
*pbRulesExist* | Boolean. Returns TRUE if a rules file was loaded into the application, FALSE if no rules file was 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.

```vbnet
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
```

GetVBScriptCalcRulesEx

*For internal use.*

GetVBScriptMemberListRules

Returns an application’s member lists as an array of bytes in the LST format.

**Syntax**

```
<HsvSystemInfo>.GetVBScriptMemberListRules pvarabRules, pbRulesExist
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvarabRules</td>
<td>Variant array. Returns the contents of the most recently-loaded member lists file. The array is returned as a Byte subtype.</td>
</tr>
<tr>
<td>pbRulesExist</td>
<td>Boolean. Returns TRUE if a member lists file was loaded into the application, FALSE if no member lists file was loaded.</td>
</tr>
</tbody>
</table>

**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.

```vbnet
define 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.*

**IsLoggingEnabled**

Returns a zero value if diagnostic logging is not enabled, and a positive value if diagnostic logging is enabled.

**Syntax**

`IHsvSystemInfo.IsLoggingEnabled IModule, ILevel`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IModule</td>
<td>Long. Internal use only. Must be 0 (zero).</td>
</tr>
<tr>
<td>ILevel</td>
<td>Long. Internal use only. Must be 0 (zero).</td>
</tr>
</tbody>
</table>

**Return Value**

*Zero = Not enabled; 1 = Enabled*

**Example**

`IHsvSystemInfo.IsLoggingEnabled(0, 0)`
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 specific users, application servers, and sessions.

Tip: To log off users from a specific cluster or application, use HsxClient.KillUsers.

Syntax

< HSVSystemInfo >.KillUsers vbAllServers, bstrServer, vbAllUsers, lActivityUserID, vbAllSessions, lActivitySessionID

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vbAllServers</td>
<td>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 bstrServer argument.</td>
</tr>
<tr>
<td>bstrServer</td>
<td>String (ByVal). The name of the application server for which to log users off. This argument is used only if the vbAllServers argument is set to FALSE.</td>
</tr>
<tr>
<td>vbAllUsers</td>
<td>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 lActivityUserID argument.</td>
</tr>
<tr>
<td>lActivityUserID</td>
<td>Long (ByVal). The activity user ID of the user to log off. This argument is used only if the vbAllUsers argument is set to FALSE.</td>
</tr>
<tr>
<td></td>
<td>To get a user’s activity user ID, use GetActivityUserID.</td>
</tr>
<tr>
<td>vbAllSessions</td>
<td>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 lActivitySessionID argument.</td>
</tr>
<tr>
<td>lActivitySessionID</td>
<td>Long (ByVal). The ID of the session to log off. This argument is used only if the vbAllSessions argument is set to FALSE.</td>
</tr>
</tbody>
</table>

OutputSystemInfo
For internal use.
ReleaseHsxServer

For internal use.

Syntax

<HsvSystemInfo>.ReleaseHsxServer plRefCount

Argument Description

plRefCount Long.

SaveLastModifiedDateAndModifiedBy

For internal use.

SetApplicationDirectory

For internal use.

SetCOMDLLCalcRules

For internal use.

SetCurrentActivity

Sets the activity for the current user.

Syntax

<HsvSystemInfo>.SetCurrentActivity lActivityCode, bstrDescription

Argument Description

lActivityCode 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 895.

Tip: If you are adding an activity for a custom module, use USERACTIVITYCODE_EXTERNAL, which represents custom activities.

bstrDescription 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

```csharp
<HsvSystemInfo>.SetCurrentModuleEx lModuleNameResource
```

Argument Description

```
lModuleNameResource   Long (ByVal). The module ID. Valid values are represented by the HFMConstants enumeration tagMODULEIDS, which is described in "Module ID Constants" on page 908.
```

SetExtractFileEncoding

Sets the file encoding type for extracted files.

Syntax

```csharp
<HsvSystemInfo>.SetExtractFileEncoding lExtractFileEncoding
```

Argument Description

```
lExtractFileEncoding   Long (ByVal). Identifies the type of file encoding for extracted files. For valid values see "Extracted File Encoding Constants" on page 894.
```

SetLanguageUserParameters

Sets the language in which member descriptions are displayed for the connected user.

Syntax

```csharp
<HsvSystemInfo>.SetLanguageUserParameters lLanguageID
```
**SetNumberFormattingUserParameters**

Sets the user’s decimal and thousands separator characters.

**Syntax**

```vba
<HsvSystemInfo>.SetNumberFormattingUserParameters wcDecimalChar, wcThousandsChar
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wcDecimalChar</td>
<td>Integer (ByVal). The double-byte Integer that identifies the decimal character to be set.</td>
</tr>
<tr>
<td>wcThousandsChar</td>
<td>Integer (ByVal). The double-byte Integer that identifies the thousands separator character to be set.</td>
</tr>
</tbody>
</table>

**Example**

The following example sets a comma as the decimal character and a period as the thousands separator character.

```vba
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 or 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**

```vba
<HsvSystemInfo>.SetResourceLanguageForCurrentSession lResourceLanguageID
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lResourceLanguageID</td>
<td>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 856.</td>
</tr>
</tbody>
</table>
**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**

*lResourceLanguageID* 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 856.

**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. See “GetUserParameter” on page 530 and “DeleteUserParameter” on page 505.

**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.

```vbnet
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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>varabRules</code></td>
<td>Variant array (ByVal). The rules file, passed as a Variant array of bytes.</td>
</tr>
<tr>
<td><code>vbScanOnly</code></td>
<td>Boolean (ByVal). Determines whether SetVBScriptCalcRules loads after validating. Pass TRUE to load after validation, FALSE to validate without loading.</td>
</tr>
<tr>
<td><code>pvbErrorsWereFound</code></td>
<td>Boolean. Indicates whether SetVBScriptCalcRules found any validation errors. Returns TRUE if errors are found, FALSE otherwise.</td>
</tr>
<tr>
<td><code>pvbWarningsWereFound</code></td>
<td>Boolean. Indicates whether SetVBScriptCalcRules found any validation warnings. Returns TRUE if warnings are found, FALSE otherwise.</td>
</tr>
<tr>
<td><code>pvbInfoWasProvided</code></td>
<td>Boolean. Indicates whether SetVBScriptCalcRules returns any information not classified as errors or warnings. Returns TRUE if non-error and non-warning information is returned, FALSE otherwise.</td>
</tr>
<tr>
<td><code>pvaralErrorLineNumbers</code></td>
<td>Variant array. Returns the line numbers to which the errors, warnings, and information apply. The array is returned as a Long subtype.</td>
</tr>
</tbody>
</table>
| `pvaralErrorSeverity`  | Variant array. Returns numbers that indicate the severity levels of the errors, warnings, and information. Valid return values are described in the following list:  
  - 1 = Error severity level.  
  - 2 = Warning severity level.  
  - 3 = Information severity level.  

  The array is returned as a Long subtype. |
| `pvarabstrErrorDescriptions` | Variant array. Returns descriptions of the errors, warnings, and information. The array is returned as a String subtype. |
| `pvarabstrErrorVBScript` | Variant array. Returns the rules file statements to which the errors, warnings, and information apply. The array is returned as a String subtype. |
**Argument** | **Description**  
---|---  
`pvarabstrErrorDetails` | 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`. `SetVBScriptCalcRules` is set to validate without loading. If the `pvbWarningsWereFound` argument returns TRUE, the example loops through the `pvaralErrorSeverity` 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.

```vbs
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
```

**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** 539
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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varabRules</td>
<td>Variant array (ByVal). The member lists file, passed as a Variant array of bytes.</td>
</tr>
<tr>
<td>vbScanOnly</td>
<td>Boolean (ByVal). Determines whether SetVBScriptMemberListRules loads after validating. Pass TRUE to load after validation, FALSE to validate without loading.</td>
</tr>
<tr>
<td>pvbErrorsWereFound</td>
<td>Boolean. Indicates whether SetVBScriptMemberListRules found any validation errors. Returns TRUE if errors are found, FALSE otherwise.</td>
</tr>
<tr>
<td>pvbWarningsWereFound</td>
<td>Boolean. Indicates whether SetVBScriptMemberListRules found any validation warnings. Returns TRUE if warnings are found, FALSE otherwise.</td>
</tr>
<tr>
<td>pvbInfoWasProvided</td>
<td>Boolean. Indicates whether SetVBScriptCalcRules returns any information not classified as errors or warnings. Returns TRUE if non-error and non-warning information is returned, FALSE otherwise.</td>
</tr>
<tr>
<td>pvaralErrorLineNumbers</td>
<td>Variant array. Returns the line numbers to which the errors, warnings, and information apply. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvaralErrorSeverity</td>
<td>Variant array. Returns numbers that indicate the severity levels of the errors, warnings, and information. Valid return values are described in the following list:</td>
</tr>
<tr>
<td></td>
<td>- 1 = Error severity level.</td>
</tr>
<tr>
<td></td>
<td>- 2 = Warning severity level.</td>
</tr>
<tr>
<td></td>
<td>- 3 = Information severity level.</td>
</tr>
<tr>
<td></td>
<td>The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarabstrErrorDescriptions</td>
<td>Variant array. Returns descriptions of the errors, warnings, and information. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarabstrErrorVBScript</td>
<td>Variant array. Returns the member lists file statements to which the errors, warnings, and information apply. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarabstrErrorDetails</td>
<td>Variant array. Returns details that describe why the lines caused validation to fail. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

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; 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, _
avasErr, 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 & 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.

TraceLog

Writes a text value to the diagnostic log if logging is enabled.

Syntax

IHsvSystemInfo.TraceLog(IModule, ILogLevel, bstrLogText)
Argument Description

IModule Long. Internal use only. Must be 0 (zero).
ILogLevel Long. Internal use only. Must be 0 (zero).
bstrLogText BSTR. Text to log.

Return Value
None.

Example
IHsvSystemInfo.TraceLog 0, 0, "This is a log entry"

**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 the HsvProcessFlow object. Use the HsvProcessFlow object to take actions for and get histories of process units. The object 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 99.

When executing actions such as promotions, the connected user must be assigned to the required security role for the action. See the Oracle Enterprise Performance Management System User Security Administration 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 submission, 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, these 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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit's child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit's parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit's Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue member are approved. Pass TRUE to approve related process units, FALSE to approve only the process unit for the lValue member.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the approval.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns PROCESS_FLOW_STATE_APPROVED, which is the level constant for the Approved level.</td>
</tr>
</tbody>
</table>
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.

```vba
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, "2012")
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

```vba
<HsvProcessFlow>.Approve2 lScenario, lYear, lPeriod, lEntity, lParent, lValue,
    vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, varabstrPaths, varabstrFiles,
    psNewProcessState
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit’s Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit’s Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit’s child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit’s parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit’s Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether process units for Value dimension members related to the lValue member are approved. Pass TRUE to approve related process units, FALSE to approve only the process unit for the lValue member.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>If you specify TRUE and an error occurs for a period, Financial Management attempts to approve the other periods’ process units.</td>
</tr>
</tbody>
</table>
### Argument Description

**bstrAnnotation**  
String (ByVal). The comment for the approval.

**varabstrPaths**  
String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes (\).  
This array has a one-to-one correspondence with the **varabstrFiles** argument’s array of filenames.

**varabstrFiles**  
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 **varabstrPaths** argument’s array, otherwise an error occurs.

**psNewProcessState**  
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.

```vba
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, "2012")
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
chHsvProcessFlow.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

```vba
<
HsvProcessFlow
>.ApproveEx lScenario, lYear, lPeriod, lEntity, lParent, lValue,
    vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, psNewProcessState
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit’s Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit’s Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit’s child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit’s parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit’s Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether process units for Value dimension members related to the lValue member are approved. Pass TRUE to approve related process units, FALSE to approve only the process unit for the lValue member.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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 attempts to approve the other periods. Note: Specifying FALSE is the same as calling Approve.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the approval.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>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 lPeriod and lValue arguments.</td>
</tr>
</tbody>
</table>

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.

```vbnet
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, "2012")
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**

*Deprecated* - use “GetGroupPhaseFromCellExtDim” on page 548.
GetGroupPhaseFromCellExtDim

Returns the IDs of the submission group and submission phase and to which a specified cell is assigned. Supersedes GetGroupPhaseFromCell.

Syntax

```
<HsvProcessFlow>.GetGroupPhaseFromCellExtDim pIUnkHfmPovCOM, pbstrGroup, pbstrPhase
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>pbstrGroup</td>
<td>String. Returns the ID of the cell's submission group.</td>
</tr>
<tr>
<td>pbstrPhase</td>
<td>String. Returns the phase ID of the cell's submission phase.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit's child Entity dimension member.</td>
</tr>
</tbody>
</table>


548 HsvProcessFlow Type Library
**Argument** | **Description**
--- | ---
`lParent` | Long (ByVal). The member ID of the process unit's parent Entity dimension member.
`lValue` | Long (ByVal). The member ID of the process unit's Value dimension member.
`pvaradTime` | 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 `CDate`.
`pvarabstrUser` | 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.
`pvarasAction` | Variant. Returns an array of the process unit's actions. The valid return values are listed in "Process Management Action Constants" on page 878. The array is returned as an Integer subtype.
`pvarasNewState` | Variant. Returns an array of the process unit's levels. The valid return values are listed in Table 139 on page 878. The array is returned as an Integer subtype.
`pvarabstrAnnotation` | 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 was rejected. If it was rejected, the subroutine displays the username of the rejector and the comment entered for the rejection. The subroutine uses 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.

```vba
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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit’s Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit’s Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit’s child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit’s parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit’s Value dimension member.</td>
</tr>
<tr>
<td>pvaradTime</td>
<td>Variant. Returns an array of the process unit’s times and dates. The array is returned as a Double subtype.</td>
</tr>
<tr>
<td>pvarabstrUser</td>
<td>Variant. Returns an array of the usernames that have performed actions for the process unit. The full usernames — the domains and the user names — are returned.</td>
</tr>
<tr>
<td>pvarasAction</td>
<td>Variant. Returns an array of the process unit’s actions. The valid return values are listed in “Process Management Action Constants” on page 878.</td>
</tr>
<tr>
<td>pvarasNewState</td>
<td>Variant. Returns an array of the process unit’s levels. The valid return values are listed in Table 139 on page 878.</td>
</tr>
<tr>
<td>pvarabstrAnnotation</td>
<td>Variant. Returns an array of the process unit’s comments. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaravarabstrPaths</td>
<td>Variant. Returns an array of arrays that contain the paths of attached documents for each action. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaravarabstrFiles</td>
<td>Variant. Returns an array of arrays that contain the names of attached documents for each action. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

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 were 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 were specified
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

Deprecated - use “GetPhasedSubmissionHistoryExtDim” on page 551.

GetPhasedSubmissionHistoryExtDim

Returns arrays containing the history of a submission phase, using the the member IDs of a cell in the phase. The arrays have a one-to-one correspondence.

Syntax

<HsvProcessFlow>.GetPhasedSubmissionHistoryExtDim pIUnkHfmPovCOM, pvaradTime, pvarabstrUser, pvarasAction, pvarasNewState, pvarabstrAnnotation

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM . HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>pvaradTime</td>
<td>Variant array. Returns an array of the submission phase 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 CDate.</td>
</tr>
<tr>
<td>pvarabstrUser</td>
<td>Variant array. Returns an array of the usernames that performed actions for the submission phase. The usernames are fully qualified. The array is returned as a String sub-type.</td>
</tr>
<tr>
<td>pvarasAction</td>
<td>Variant array. Returns an array of the submission phase actions. The valid return values are listed in Table 138 on page 878. The array is returned as an Integer sub-type.</td>
</tr>
<tr>
<td>pvarasNewState</td>
<td>Variant array. Returns an array of the submission phase review levels. The valid return values are listed in Table 138 on page 878. The array is returned as an Integer sub-type.</td>
</tr>
</tbody>
</table>
**Argument**

pvarabstrAnnotation  Variant array. Returns an array of the submission phase comments. The array is returned as a String sub-type.

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

### GetPhasedSubmissionState

*Deprecated* - use “GetPhasedSubmissionStateExtDim” on page 552.

### GetPhasedSubmissionStateExtDim

Returns the current review level of a specified cell’s submission phase.

**Syntax**

```csharp
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM . HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>psProcessState</td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase review level. Valid values are listed in Table 139 on page 878.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

### GetPhasedSubmissionStateUsingPhaseID

*Deprecated* - use “GetPhasedSubmissionStateUsingPhaseIDExtDim” on page 553.
GetPhasedSubmissionStateUsingPhaseIDExtDim

Returns the review level of a submission phase using the member IDs of a cell in the phase or the phase ID. Supersedes GetPhasedSubmissionStateUsingPhaseID. You must pass one of these values:

- 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

```csharp
```

**Argument** | **Description**
---|---
`pIUnkHfmPovCOM` | HfmPovCOM. HfmPovCOM object representing the POV. If using phase ID, set the Account, ICP and Custom dimension members to MEMBERNOTUSED.
`lPhaseID` | Long (ByVal). Pass one of these values:
- The phase ID, which you can obtain with `GetGroupPhaseFromCell`.
- MEMBERNOTUSED
`vbEnsureUptodate` | Boolean (ByVal). Forces a read from the database before returning the review level. Pass TRUE to force a read, FALSE for the system to read from the database only if necessary.
- **Note:** The Force option applies only to multi-server systems.
`psProcessState` | Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the submission phase review level. See Table 139 on page 878.

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

GetState

Returns the current level of a process unit.

Syntax

```csharp
```
Argument | Description
---|---
`lScenario` | Long (ByVal). The member ID of the process unit's Scenario dimension member.
`lYear` | Long (ByVal). The member ID of the process unit's Year dimension member.
`lPeriod` | Long (ByVal). The member ID of the process unit's Period dimension member.
`lEntity` | Long (ByVal). The member ID of the process unit's child Entity dimension member.
`lParent` | Long (ByVal). The member ID of the process unit's parent Entity dimension member.
`lValue` | Long (ByVal). The member ID of the process unit's Value dimension member.

`psProcessState` | Integer. Returns the level constant that identifies the process unit's current level. The valid return values are listed in Table 139 on page 878.

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 is executed.

```vba
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, "2012")
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**

*Deprecated - use “PhasedSubmissionApproveExtDim” on page 554.*

**PhasedSubmissionApproveExtDim**

Approves a submission phase using the member IDs of a cell in the phase. Supersedes `PhasedSubmissionApprove`.

**Note:** To approve by passing a phase ID, use `PhasedSubmissionApprove2ExtDim`. 
Syntax

```<HsvProcessFlow>.PhasedSubmissionApproveExtDim pIUnkHfmPovCOM, vbUseAllValueMembers, bstrAnnotation, psNewProcessState```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td><code>vbUseAllValueMembers</code></td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the 1Value member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td><code>bstrAnnotation</code></td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td><code>psNewProcessState</code></td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

**PhasedSubmissionApprove2**

*Deprecated* - use “PhasedSubmissionApprove2ExtDim” on page 555.

**PhasedSubmissionApprove2ExtDim**

Approves a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID. Supersedes PhasedSubmissionApprove2. You must pass of these values:

- 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>.PhasedSubmissionApprove2ExtDim pIUnkHfmPovCOM, lPhaseID, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, varabstrPaths, varabstrFiles, psNewProcessState```  

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the POV. If using Phase ID, specify MEMBERNOTUSED for the Account, ICP and Custom members.</td>
</tr>
<tr>
<td><strong>Argument</strong></td>
<td><strong>Description</strong></td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>IPhaseID</td>
<td>Long (ByVal). Pass one of these values:</td>
</tr>
<tr>
<td></td>
<td>● The phase ID, which you can obtain with <strong>GetGroupPhaseFromCell</strong>.</td>
</tr>
<tr>
<td></td>
<td>● <strong>MEMBERNOTUSED</strong></td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <strong>Value</strong> member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes (). This array has a one-to-one correspondence with the <strong>varabstrFiles</strong> argument's array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>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 <strong>varabstrPaths</strong> argument's array, otherwise an error occurs.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the <strong>CEnumProcessFlowStates</strong> enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

**PhasedSubmissionApproveEx**

*Deprecated* - use “PhasedSubmissionApproveExExtDim” on page 556.

**PhasedSubmissionApproveExExtDim**

**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, using the member IDs of a cell in the phase. Supersedes PhasedSubmissionApproveEx.

```
<HsvProcessFlow>.PhasedSubmissionApproveExExtDim pIUnkHfmPovCOM, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, psNewProcessState
```
**Argument** | **Description**
--- | ---
plUnkHfmPovCOM | HfmPovCOM. HfmPovCOM object representing the POV. If using Phase ID, specify MEMBERNOTUSED for the Account, ICP and Custom members.

vbUseAllValueMembers | Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the IValue member. Pass TRUE to apply the action to related submission phases.

vbApplyToAllPeriods | 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.

bstrAnnotation | String (ByVal). The comment for the process management action.

psNewProcessState | Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.

**Return Value**
None.

**Example**
See HfmPovCOM documentation on how to set the POV.

**PhasedSubmissionGetHistory2**

`Deprecated` - use “PhasedSubmissionGetHistory2ExtDim” on page 557.

**PhasedSubmissionGetHistory2ExtDim**

Returns arrays containing the history of a submission phase, using 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**

```xml
<HsvProcessFlow>.PhasedSubmissionGetHistory2ExtDim plUnkHfmPovCOM, pvaradTime, pvarabstrUser, pvarasAction, pvarasNewState, pvarabstrAnnotation, pvaravaraabstrPaths, pvaravaraabstrFiles
```

**Argument** | **Description**
--- | ---
plUnkHfmPovCOM | HfmPovCOM. HfmPovCOM object representing the POV. If using Phase ID, specify MEMBERNOTUSED for the Account, ICP and Custom members.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvaradTime</td>
<td>Variant array. Returns an array of the submission phase 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>.</td>
</tr>
<tr>
<td>pvarabstrUser</td>
<td>Variant array. Returns an array of the usernames that performed actions for the submission phase. The usernames are full qualified. The array is returned as a String sub-type.</td>
</tr>
<tr>
<td>pvarasAction</td>
<td>Variant array. Returns an array of the submission phase actions. The valid return values are listed in Table 138 on page 878. The array is returned as an Integer sub-type.</td>
</tr>
<tr>
<td>pvarasNewState</td>
<td>Variant array. Returns an array of the submission phase review levels. The valid return values are listed in The member ID of the cell's Custom4 dimension member. The array is returned as an Integer sub-type.</td>
</tr>
<tr>
<td>pvarabstrAnnotation</td>
<td>Variant array. Returns an array of the submission phase's comments. The array is returned as a String sub-type.</td>
</tr>
<tr>
<td>pvaravarabstrPaths</td>
<td>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.</td>
</tr>
<tr>
<td>pvaravarabstrFiles</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

**PhasedSubmissionGetHistory2UsingPhaseID**

*Deprecated* - use “PhasedSubmissionGetHistory2UsingPhaseIDExtDim” on page 558.

**PhasedSubmissionGetHistory2UsingPhaseIDExtDim**

Returns arrays containing the history of a submission phase, using either the member IDs of a cell in the phase or the phase ID. Supersedes PhasedSubmissionGetHistory2UsingPhaseID. You must pass one of these values:

- 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>.PhasedSubmissionGetHistory2UsingPhaseIDExtDim pIUnkHfmPovCOM, lPhaseID, pvaradTime, pvarabstrUser, pvarasAction, pvarasNewState, pvarabstrAnnotation, pvaravarabstrPaths, pvaravarabstrFiles
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV. If using Phase ID, specify MEMBERNOTUSED for the Account, ICP and Custom members.</td>
</tr>
<tr>
<td>lPhaseID</td>
<td>Long (ByVal). Pass one of these values: The phase ID, which you can obtain with GetGroupPhaseFromCell. MEMBERNOTUSED</td>
</tr>
<tr>
<td>pvaradTime</td>
<td>Variant array. Returns an array of the submission phase 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 CDate.</td>
</tr>
<tr>
<td>pvarabstrUser</td>
<td>Variant array. Returns an array of the usernames that performed actions for the submission phases. The usernames are fully qualified. The array is returned as a String sub-type.</td>
</tr>
<tr>
<td>pvarasAction</td>
<td>Variant array. Returns an array of the submission phase actions. The valid return values are listed in Table 138 on page 878. The array is returned as an Integer sub-type.</td>
</tr>
<tr>
<td>pvarasNewState</td>
<td>Variant array. Returns an array of the submission phase review levels. The valid return values are listed in Table 139 on page 878. The array is returned as an Integer sub-type.</td>
</tr>
<tr>
<td>pvarabstrAnnotation</td>
<td>Variant array. Returns an array of the submission phase comments The array is returned as a String sub-type.</td>
</tr>
<tr>
<td>pvaravarabstrPaths</td>
<td>Variant array. Returns an array that contains the paths of attached documents for each action. The array is returned as a String sub-type.</td>
</tr>
<tr>
<td>pvaravarabstrFiles</td>
<td>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.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.
**PhasedSubmissionProcessManagementChangeStateForMultipleEntities2**

*Deprecated* - use “PhasedSubmissionProcessManagementChangeStateForMultipleEntities2ExtDim” on page 560.

**PhasedSubmissionProcessManagementChangeStateForMultipleEntities2ExtDim**

Applies a process management action and attaches documents to submission phases for multiple Entity dimension members, using 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. Supersedes PhasedSubmissionProcessManagementChangeStateForMultipleEntities2.

**Syntax**

```<HsvProcessFlow>.PhasedSubmissionProcessManagementChangeStateForMultipleEntities2ExtDim pIUnkHfmSliceCOM, bstrAnnotation, lAction, vbUseAllValueMembers, vbApplyToAllPeriods, sProcessStateToPromoteTo, varabstrPaths, varabstrFiles, psNewProcessState```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV. If using Phase ID, specify MEMBERNOTUSED for the Account, ICP and Custom members.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>lAction</td>
<td>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 878.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the IValueIn member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases containing base periods that intersect with the specified dimension member. Pass TRUE to apply the action to all such submission phases.</td>
</tr>
<tr>
<td>sProcessStateToPromoteTo</td>
<td>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 878.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes (). This array has a one-to-one correspondence with the varabstrFiles argument's array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>String array (ByVal). The file names of the documents to attach.</td>
</tr>
</tbody>
</table>

*Note:* Files must be loaded in the folder specified by the corresponding item in the varabstrPaths argument's array, otherwise an error occurs.

---

**560 HsvProcessFlow Type Library**
**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the review level applied to the process units. Valid values are represented by the HFMConstants type library constants listed in &quot;Process Management Review Level Constants&quot; on page 878.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

---

**PhasedSubmissionPromote**

*Deprecated* - use “PhasedSubmissionPromoteExtDim” on page 561.

**PhasedSubmissionPromoteExtDim**

Promotes a submission phase to a specified review level, using the member IDs of a cell in the phase. Supersedes PhasedSubmissionPromote.

To promote by passing a phase ID, use PhasedSubmissionPromote2ExtDim.

**Syntax**

```plaintext
<HsvProcessFlow>.PhasedSubmissionPromoteExtDim pIUnkHfmPovCOM, vbUseAllValueMembers, bstrAnnotation, sProcessStateToPromoteTo, psNewProcessState
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the 1Value member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>sProcessStateToPromoteTo</td>
<td>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 139 on page 878.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.
PhasedSubmissionPromote2

*Deprecated* - use “PhasedSubmissionPromote2ExtDim” on page 562.

PhasedSubmissionPromote2ExtDim

Promotes a submission phase, and optionally attaches documents using either the member IDs of a cell in the phase or the phase ID. Supersedes PhasedSubmissionPromote2. You must pass one of these values:

- 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>.PhasedSubmissionPromote2ExtDim pIUnkHfmPovCOM, lPhaseID, vbUseAllValueMembers, bstrAnnotation, sProcessStateToPromoteTo, varabstrPaths varabstrFiles, psNewProcessState```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV. If using Phase ID, set the Account, ICP andCustom members to MEMBERNOTUSED.</td>
</tr>
<tr>
<td>lPhaseID</td>
<td>Long (ByVal). Pass one of these values:</td>
</tr>
<tr>
<td></td>
<td>- The phase ID, which you can obtain with GetGroupPhaseFromCell</td>
</tr>
<tr>
<td></td>
<td>- MEMBERNOTUSED</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the lValue member. Pass TRUE to apply to the action to related submission phases.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>sProcessStateToPromoteTo</td>
<td>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 878.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are specified by backslashes (). This array has a one-to-one correspondence with the varabstrFiles argument's array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>String array (ByVal). The file names of the documents to attach.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Files must be loaded in the folder specified by the corresponding item in the varabstrPaths argument's array, otherwise an error occurs.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.</td>
</tr>
</tbody>
</table>

Return Value

None.
Example
See HfmPovCOM documentation on how to set the POV.

**PhasedSubmissionPublish**

*Deprecated* - use “PhasedSubmissionPublishExExtDim” on page 565.

**PhasedSubmissionPublishExtDim**

Publishes a submission phase using the member IDs of a cell in the phase. Supersedes PhasedSubmissionPublishEx.

To publish by passing a phase ID, use PhasedSubmissionPublish2.

**Syntax**

```plaintext
<HsvProcessFlow>.PhasedSubmissionPublishExExtDim pIUnkHfmPovCOM, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, psNewProcessState
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the 1Value member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

Example

See HfmPovCOM documentation on how to set the POV.

**PhasedSubmissionPublish2**

*Deprecated* - use “PhasedSubmissionPublish2ExtDim” on page 564.
PhasedSubmissionPublish2ExtDim

Publishes a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID. Supersedes PhasedSubmissionPublish2. You must pass one of these values:

- 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>.
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV. If using Phase ID, set the Account, ICP and Custom members to MEMBERNOTUSED.</td>
</tr>
<tr>
<td>lPhaseID</td>
<td>Long (ByVal). Pass one of these values:</td>
</tr>
<tr>
<td></td>
<td>- The phase ID,, which you can obtain with GetGroupPhaseFromCell.</td>
</tr>
<tr>
<td></td>
<td>- MEMBERNOTUSED</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the lValue member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes (). This array has a one-to-one correspondence with the varabstrFiles argument's array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>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 varabstrPaths argument's array, otherwise an error occurs.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.
PhasedSubmissionPublishEx

Deprecated - use “PhasedSubmissionPublishExtDim” on page 563.

PhasedSubmissionPublishExExtDim

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 using the member IDs of a cell in the phase.

Syntax

```<HsvProcessFlow>.PhasedSubmissionPublishExExtDim pIUnkHfmPovCOM, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, psNewProcessState```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>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.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the CEnumProcessFlow enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

PhasedSubmissionReject

Deprecated - use “PhasedSubmissionRejectExtDim” on page 565.

PhasedSubmissionRejectExtDim

Rejects a submission phase, using the member IDs of a cell in the phase.

To reject by passing a phase ID, use PhasedSubmissionReject2ExtDim.
Syntax

```
<HSVProcessFlow>.PhasedSubmissionRejectExtDim pIUnkHfmPovCOM, vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td><code>vbUseAllValueMembers</code></td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <code>1_Value</code> member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td><code>bstrAnnotation</code></td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td><code>psNewProcessState</code></td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

**PhasedSubmissionReject2**

Deprecated - use “PhasedSubmissionReject2ExtDim” on page 566.

**PhasedSubmissionReject2ExtDim**

Rejects a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID. Supersedes PhasedSubmissionReject2. You must pass one of these values:

- 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>.PhasedSubmissionReject2ExtDim pIUnkHfmPovCOM, lPhaseID, vbUseAllValueMembers, bstrAnnotation, varabstrPaths, varabstrFiles, psNewProcessState
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
</tbody>
</table>
### Argument Description

- **lPhaseID**
  - Long (ByVal). Pass one of these values:
    - The phase ID, which you can obtain with GetGroupPhaseFromCell.
    - MEMBERNOTUSED

- **vbUseAllValueMembers**
  - Boolean (ByVal). Specifies whether to apply to the action to submission phases for Value dimension members related to the 1Value member. Pass TRUE to apply the action to related submission phases.

- **bstrAnnotation**
  - String (ByVal). The comment for the process management action.

- **varabstrPaths**
  - String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes (\). This array has a one-to-one correspondence with the varabstrFiles argument's array of filenames.

- **varabstrFiles**
  - 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 varabstrPaths argument's array, otherwise an error occurs.

- **psNewProcessState**
  - Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.

### Return Value
None.

### Example
See HfmPovCOM documentation on how to set the POV.

#### PhasedSubmissionSignOff

*Deprecated*- use “PhasedSubmissionSignOffExtDim” on page 567.

#### PhasedSubmissionSignOffExtDim

Signs off on a submission phase using the member IDs of a cell in the phase.

To sign off by passing a phase ID, use PhasedSubmissionSignOff2.

### Syntax

```xml
<HsvProcessFlow>.PhasedSubmissionSignOff lScenario, lYear, lPeriod, lEntity, lParent,
lValue, lAccount, lICP, lCustom1, lCustom2, lCustom3, lCustom4, vbUseAllValueMembers,
bstrAnnotation
```

### Argument Description

- **plUnkHfmPovCOM**
  - HfmPovCOM. HfmPovCOM object representing the POV.

- **vbUseAllValueMembers**
  - Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the 1Value member. Pass TRUE to apply the action to related submission phases.
## PhasedSubmissionSignOff2

*Deprecated* - use “PhasedSubmissionSignOff2ExtDim” on page 568.

### PhasedSubmissionSignOff2ExtDim

Signs off on a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID. Supersedes PhasedSubmissionSignoff2. You must pass one of these values:

- 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>.PhasedSubmissionSignOff2ExtDim pIUnkHfmPovCOM, lPhaseID, vbUseAllValueMembers, bstrAnnotation, varabstrPaths, VARIANT varabstrFiles```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the POV. If using Phase ID, specify MEMBERNOTUSED for the Account, ICP and Custom members.</td>
</tr>
<tr>
<td><code>lPhaseID</code></td>
<td>Long (ByVal). Pass one of these values:</td>
</tr>
<tr>
<td></td>
<td>- The phase ID, which you can obtain with GetGroupPhaseFromCell.</td>
</tr>
<tr>
<td></td>
<td>- MEMBERNOTUSED</td>
</tr>
<tr>
<td><code>vbUseAllValueMembers</code></td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the <code>lValue</code> member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td><code>bstrAnnotation</code></td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td><code>varabstrPaths</code></td>
<td>String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes (<code>\</code>). This array has a one-to-one correspondence with the <code>varabstrFiles</code> argument's array of filenames.</td>
</tr>
</tbody>
</table>
Argument Description

varabstrFiles 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 varabstrPaths argument’s array, otherwise an error occurs.

Return Value
None.

Example
See HfmPovCOM documentation on how to set the POV.

PhasedSubmissionStart

Deprecated - use “PhasedSubmissionStartExtDim” on page 569.

PhasedSubmissionStartExtDim

Starts a submission phase using the member IDs of a cell in the phase. Supersedes PhasedSubmissionStart.

To start by passing a phase ID, use PhasedSubmissionStart2ExtDim.

Syntax

<HsvProcessFlow>.PhasedSubmissionStartExtDim pIUnkHfmPovCOM, vbUseAllValueMembers, bstrAnnotation, psNewProcessState

Argument Description

pIUnkHfmPovCOM HfmPovCOM. HfmPovCOM object representing the POV.

vbUseAllValueMembers Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the 1Value member. Pass TRUE to apply the action to related submission phases.

bstrAnnotation String (ByVal). The comment for the process management action.

psNewProcessState Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878

Return Value
None.

Example
See HfmPovCOM documentation on how to set the POV.
PhasedSubmissionStart2

Deprecated - use “PhasedSubmissionStart2ExtDim” on page 570.

PhasedSubmissionStart2ExtDim

Starts a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID. Supersedes PhasedSubmissionStart2. You must pass one of these values:

- 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>.PhasedSubmissionStart2ExtDim pIUnkHfmPovCOM, lPhaseID, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, varabstrPaths, varabstrFiles, psNewProcessState```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>lPhaseID</td>
<td>Long (ByVal). Pass one of these values:</td>
</tr>
<tr>
<td></td>
<td>● The phase ID, which you can obtain with GetGroupPhaseFromCell.</td>
</tr>
<tr>
<td></td>
<td>● MEMBERNOTUSED</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the 1Value member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes. This array has a one-to-one correspondence with the varabstrFiles argument’s array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>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 varabstrPaths argument’s array, otherwise an error occurs.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878</td>
</tr>
</tbody>
</table>

Return Value

None.
Example
See HfmPovCOM documentation on how to set the POV.

**PhasedSubmissionStartEx**

*Deprecated* - use “PhasedSubmissionStartExExtDim” on page 571.

**PhasedSubmissionStartExExtDim**

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, using the member IDs of a cell in the phase. Supersedes PhasedSubmissionStartEx.

**Syntax**

```<HsvProcessFlow>.PhasedSubmissionStartExExtDim pIUnkHfmPovCOM, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, psNewProcessState```

**Argument** | **Description**
--- | ---
`pIUnkHfmPovCOM` | HfmPovCOM. HfmPovCOM object representing the POV.
`vbUseAllValueMembers` | Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the `lValue` member. Pass TRUE to apply the action to related submission phases.
`vbApplyToAllPeriods` | 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.
`bstrAnnotation` | String (ByVal). The comment for the process management action.
`psNewProcessState` | Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878

**Return Value**

None.

Example

See HfmPovCOM documentation on how to set the POV.

**PhasedSubmissionSubmit**

*Deprecated* - use “PhasedSubmissionSubmitExtDim” on page 572.
PhasedSubmissionSubmitExtDim

Submits a submission phase using the member IDs of a cell in the phase. Supersedes PhasedSubmissionSubmit.

To submit by passing a phase ID, use PhasedSubmissionSubmitExtDim.

Syntax

```<HsvProcessFlow>.PhasedSubmissionSubmitExtDim pIUnkHfmPovCOM, vbUseAllValueMembers, bstrAnnotation, psNewProcessState```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the Value member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

PhasedSubmissionSubmit2

*Deprecated* - use “PhasedSubmissionSubmit2ExtDim” on page 572.

PhasedSubmissionSubmit2ExtDim

Submits a submission phase and optionally attaches documents, using either the member IDs of a cell in the phase or the phase ID. Supersedes PhasedSubmissionSubmit2. You must pass one of these values:

- 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>.PhasedSubmissionSubmit2ExtDim pIUnkHfmPovCOM, lPhaseID, vbUseAllValueMembers, bstrAnnotation, varabstrPaths, varabstrFiles, psNewProcessState```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>lPhaseID</td>
<td>Long (ByVal). Pass one of these values:</td>
</tr>
<tr>
<td></td>
<td>● The phase ID, which you can obtain with GetGroupPhaseFromCell</td>
</tr>
<tr>
<td></td>
<td>● MEMBERNOTUSED</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the 1Value member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes. This array has a one-to-one correspondence with the varabstrFiles argument's array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>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 varabstrPaths argument's array, otherwise an error occurs.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the CEnumProcessFlowStates enumeration constant that represents the review level resulting from the action. See Table 139 on page 878.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

---

**ProcessManagementChangeStateForMultipleEntities**

Applies a 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**

```xml
<HsvProcessFlow>.ProcessManagementChangeStateForMultipleEntities lScenarioIn, lYearIn, lPeriodIn, varalEntity, varalParent, lValueIn, bstrAnnotation, lAction, vbUseAllValueMembers, vbApplyToAllPeriods, sProcessStateToPromoteTo, psNewProcessState
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenarioIn</td>
<td>Long (ByVal). The member ID of the process units’ Scenario dimension member.</td>
</tr>
<tr>
<td>lYearIn</td>
<td>Long (ByVal). The member ID of the process units’ Year dimension member.</td>
</tr>
<tr>
<td>lPeriodIn</td>
<td>Long (ByVal). The member ID of the process units’ Period dimension member.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>varalEntity</td>
<td>Long array (ByVal). The member IDs of the process units’ child Entity dimension members.</td>
</tr>
<tr>
<td>varalParent</td>
<td>Long array (ByVal). The member IDs of the process units’ parent Entity dimension members. This array has a one-to-one correspondence with the varalEntity argument's array.</td>
</tr>
<tr>
<td>lValueIn</td>
<td>Long (ByVal). The member ID of the process units’ Value dimension member.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). A comment to apply to the action.</td>
</tr>
<tr>
<td>lAction</td>
<td>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 878.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). A flag that determines whether the action applies to process units for Value dimension members related to the lValue member. Pass TRUE to apply the action to related process units, FALSE to apply the action to only the process unit for the lValue member.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td>sProcessStateToPromoteTo</td>
<td>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 878.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>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 878.</td>
</tr>
</tbody>
</table>

**ProcessManagementChangeStateForMultipleEntities 2**

*Deprecated* - use “ProcessManagementChangeStateForMultipleEntities2ExtDim” on page 574.

**ProcessManagementChangeStateForMultipleEntities 2ExtDim**

Applies a 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. Supersedes PhasedSubmissionProcessManagementChangeStateForMultipleEntities2.

**Syntax**

```plaintext
<HsvProcessFlow>.PhasedSubmissionProcessManagementChangeStateForMultipleEntities2ExtDim pIUnkHfmSliceCOM, bstrAnnotation, lAction, vbUseAllValueMembers, vbApplyToAllPeriods, sProcessStateToPromoteTo, varabstrPaths, varabstrFiles, psNewProcessState
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). A comment to apply to the action.</td>
</tr>
<tr>
<td>lAction</td>
<td>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 878.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). A flag that determines whether the action applies to process units for Value dimension members related to the <code>lValue</code> member. Pass TRUE to apply the action to related process units, FALSE to apply the action to only the process units for the <code>lValue</code> member.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td>sProcessStateToPromoteTo</td>
<td>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 878.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached are loaded. Folders in the path are delimited by backslashes ( \ ). This array has a one-to-one correspondence with the <code>varabstrFiles</code> argument's array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>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 <code>varabstrPaths</code> argument's array, otherwise an error occurs.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>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 878.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

**ProcessManagementChangeStateForMultipleEntities Ex**

*Deprecated* - use “ProcessManagementChangeStateForMultipleEntitiesExExtDim” on page 575.

**ProcessManagementChangeStateForMultipleEntities ExExtDim**

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

Syntax

```c
<HsvProcessFlow>.ProcessManagementChangeStateForMultipleEntitiesEx pIUnkHfmSliceCOM,
  bstrAnnotation, lAction, vbUseAllValueMembers, vbApplyToAllPeriods,
  sProcessStateToPromoteTo, psNewProcessState
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmSliceCOM object representing the cells’ POVs. Scenario, Year, Period and Value are fixed (1 member). The remaining dimensions can specify multiple members.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process management action.</td>
</tr>
<tr>
<td>lAction</td>
<td>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 878.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether to apply the action to submission phases for Value dimension members related to the lValueIn member. Pass TRUE to apply the action to related submission phases.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td>sProcessStateToPromoteTo</td>
<td>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 878.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>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 878.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

**Promote**

Promotes a process unit to a specified review level.

To attach documents when promoting, use `Promote2`.

Syntax

```c
<HsvProcessFlow>.Promote lScenario, lYear, lPeriod, lEntity, lParent, lValue,
  vbUseAllValueMembers, bstrAnnotation, sProcessStateToPromoteTo, psNewProcessState
```
### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit's child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit's parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit's Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue member are promoted. Pass TRUE to promote related process units, FALSE to promote only the process unit for the lValue member.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the promotion.</td>
</tr>
<tr>
<td>sProcessStateToPromoteTo</td>
<td>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 139 on page 878.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the constant that identifies the level to which the process unit was promoted. If the call to Promote succeeds, this is equal to the constant passed in the sProcessStateToPromoteTo argument.</td>
</tr>
</tbody>
</table>

### 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 was 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
    End If
    Dim cProcessFlow As HsvProcessFlow
    Dim 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
    End If
```

Promote 577
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

```
<HasProcessFlow>.Promote2 lScenario, lYear, lPeriod, lEntity, lParent, lValue,
vbUseAllValueMembers, bstrAnnotation, sProcessStateToPromoteTo, varabstrPaths,
varabstrFiles, psNewProcessState
```

### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit’s Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit’s Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit’s child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit’s parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit’s Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue member are promoted. Pass TRUE to promote related process units, FALSE to promote only the process unit for the lValue member.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the promotion.</td>
</tr>
<tr>
<td>sProcessStateToPromoteTo</td>
<td>Integer (ByVal). The constant that identifies the level to which the process unit is being promoted. Use one of the level constants in Table 139 on page 878.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached are loaded. Folders in the path are delimited by backslashes ( \ ). This array has a one-to-one correspondence with the varabstrFiles argument’s array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>String array (ByVal). The file names of the documents to attach.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the constant that identifies the level to which the process unit was promoted. If the call to Promote succeeds, this is equal to the constant passed in the sProcessStateToPromoteTo argument.</td>
</tr>
</tbody>
</table>

**Note:** Files must be loaded in the folder specified by the corresponding item in the varabstrPaths argument’s array, otherwise an error occurs.
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 was assigned to the role passed to the subroutine; if this returns TRUE, Promote2 is called.

```vba
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
```
### 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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>The member ID of the process unit’s Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>The member ID of the process unit’s Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>The member ID of the process unit’s child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>The member ID of the process unit’s parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>The member ID of the process unit’s Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue member are published. Pass TRUE to publish related process units, FALSE to publish only the process unit for the lValue member.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>The comment for the publishing of the process unit.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns PROCESS_FLOW_STATE_PUBLISHED, which is the level constant for the Published level.</td>
</tr>
</tbody>
</table>

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.
**Argument** | **Description**
--- | ---
`lYear` | Long (ByVal). The member ID of the process unit's Year dimension member.
`lPeriod` | Long (ByVal). The member ID of the process unit's Period dimension member.
`lEntity` | Long (ByVal). The member ID of the process unit's child Entity dimension member.
`lParent` | Long (ByVal). The member ID of the process unit's parent Entity dimension member.
`lValue` | Long (ByVal). The member ID of the process unit's Value dimension member.

`vbUseAllValueMembers` | Boolean (ByVal). Specifies whether process units for Value dimension members related to the `lValue` member are published. Pass TRUE to publish related process units, FALSE to publish only the process unit for the `lValue` member.

`vbApplyToAllPeriods` | 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 attempts to publish the other periods.

`bstrAnnotation` | String (ByVal). The comment for the publishing of the process unit.

`varabstrPaths` | String array (ByVal). The paths in which the documents to be attached are loaded. Folders in the path are delimited by backslashes (\). This array has a one-to-one correspondence with the `varabstrFiles` argument's array of filenames.

`varabstrFiles` | 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 `varabstrPaths` argument's array, otherwise an error occurs.

`psNewProcessState` | 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 `lPeriod` and `lValue` 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`.

```vbnet
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, "2012")
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
```

Publish2 581
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

```vba
HsvProcessFlow.PublishEx lScenario, lYear, lPeriod, lEntity, lParent, lValue, vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, psNewProcessState
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>The member ID of the process unit's child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>The member ID of the process unit's parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit's Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether process units for Value dimension members related to the <code>lValue</code> member are published. Pass TRUE to publish related process units, FALSE to publish only the process unit for the <code>lValue</code> member.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>The comment for the publishing of the process unit.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns <code>PROCESS_FLOW_STATE_PUBLISHED</code>, which is the level constant for the Published level. This value applies to the Period and Value dimension members passed in the <code>lPeriod</code> and <code>lValue</code> arguments.</td>
</tr>
</tbody>
</table>

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`.

```vba
Dim lScen As Long, lYear As Long, lPer As Long
Dim lEnt As Long, lPar As Long, lVal As Long
```

582 HsvProcessFlow Type Library
Reject

Demotes a process unit from its current level to its previous level.

To attach documents when rejecting, use Reject2.

Syntax

\(<\text{HsvProcessFlow}.\text{Reject}\ l\text{Scenario}, \ l\text{Year}, \ l\text{Period}, \ l\text{Entity}, \ l\text{Parent}, \ l\text{Value}, \ v\text{bUseAllValueMembers}, \ b\text{strAnnotation}, \ p\text{sNewProcessState}\)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit’s Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit’s Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit’s child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit’s parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit’s Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue member are demoted. Pass TRUE to demote related process units, FALSE to demote only the process unit for the lValue member.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the rejection.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the constant that identifies the level to which the process unit was demoted. For valid return values, see Table 139 on page 878.</td>
</tr>
</tbody>
</table>

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, "2012")
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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit's child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit's parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit's Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue member are demoted. Pass TRUE to demote related process units, FALSE to demote only the process unit for the lValue member.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the rejection.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached are loaded. Folders in the path are delimited by backslashes (). This array has a one-to-one correspondence with the varabstrFiles argument's array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>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 varabstrPaths argument’s array, otherwise an error occurs.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>Integer. Returns the constant that identifies the level to which the process unit was demoted. For valid return values, see Table 139 on page 878.</td>
</tr>
</tbody>
</table>

**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, "2012")
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
Set 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit's child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit's parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit's Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue 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 lValue member.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the signoff.</td>
</tr>
</tbody>
</table>

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.

```vba
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, "2012")
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**

```vba
<HsvProcessFlow>.SignOff2 lScenario, lYear, lPeriod, lEntity, lParent, lValue, vbUseAllValueMembers, bstrAnnotation, varabstrPaths, varabstrFiles
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit's child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit's parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit's Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue 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 lValue member.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the signoff.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes (). This array has a one-to-one correspondence with the varabstrFiles argument's array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>String array (ByVal). The file names of the documents to attach.</td>
</tr>
</tbody>
</table>

**Note:** Files must be loaded in the folder specified by the corresponding item in the varabstrPaths argument's array, otherwise an error occurs.
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`.

```vba
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, "2012")
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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit’s Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit’s Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit’s child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit’s parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit’s Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue member are to be started. Pass TRUE to start related process units, FALSE to start only the process unit for the lValue member.</td>
</tr>
</tbody>
</table>
### 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.

```vbscript
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
        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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit’s Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit’s Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the process unit’s Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the process unit’s child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit’s parent Entity dimension member.</td>
</tr>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the process unit’s Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Determines whether process units for Value dimension members related to the lValue member are to be started. Pass TRUE to start related process units, FALSE to start only the process unit for the lValue member.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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.</td>
</tr>
<tr>
<td>bstrAnnotation</td>
<td>String (ByVal). The comment for the process unit.</td>
</tr>
<tr>
<td>varabstrPaths</td>
<td>String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes (). This array has a one-to-one correspondence with the varabstrFiles argument’s array of filenames.</td>
</tr>
<tr>
<td>varabstrFiles</td>
<td>String array (ByVal). The file names of the documents to attach.</td>
</tr>
<tr>
<td>Note:</td>
<td>Files must be loaded in the folder specified by the corresponding item in the varabstrPaths argument’s array, otherwise an error occurs.</td>
</tr>
<tr>
<td>psNewProcessState</td>
<td>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 lPeriod and lValue arguments</td>
</tr>
</tbody>
</table>

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, "2012")
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
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, lPer, lEnt, lPar, lVal, _
    vbUseAllValueMembers, vbApplyToAllPeriods, bstrAnnotation, psNewProcessState
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the process unit's Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the process unit's Year dimension member.</td>
</tr>
<tr>
<td>lPer</td>
<td>Long (ByVal). The member ID of the process unit's Period dimension member.</td>
</tr>
<tr>
<td>lEnt</td>
<td>Long (ByVal). The member ID of the process unit's child Entity dimension member.</td>
</tr>
<tr>
<td>lParent</td>
<td>Long (ByVal). The member ID of the process unit's parent Entity dimension member.</td>
</tr>
<tr>
<td>lVal</td>
<td>Long (ByVal). The member ID of the process unit's Value dimension member.</td>
</tr>
<tr>
<td>vbUseAllValueMembers</td>
<td>Boolean (ByVal). Specifies whether process units for Value dimension members related to the lValue member are to be started. Pass TRUE to start related process units, FALSE to start only the process unit for the lValue member.</td>
</tr>
<tr>
<td>vbApplyToAllPeriods</td>
<td>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 attempts to start the other periods.</td>
</tr>
<tr>
<td>Note:</td>
<td>Specifying FALSE is the same as calling Start.</td>
</tr>
</tbody>
</table>

590 HsvProcessFlow Type Library
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrAnnotation</code></td>
<td>String (ByVal). The comment for the process unit.</td>
</tr>
<tr>
<td><code>psNewProcessState</code></td>
<td>Integer. Returns <code>PROCESS_FLOW_STATE_FIRST_PASS</code>, which is the level constant for the First Pass level. This value applies to the Period and Value dimension members passed in the <code>lPeriod</code> and <code>lValue</code> arguments.</td>
</tr>
</tbody>
</table>

**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`.

```vba
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, "2012")
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**

```vba
<HsvProcessFlow>.Submit lScenario, lYear, lPeriod, lEntity, lParent, lValue, vbUseAllValueMembers, bstrAnnotation, psNewProcessState
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code></td>
<td>Long (ByVal). The member ID of the process unit’s Scenario dimension member.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>The member ID of the process unit’s Year dimension member.</td>
</tr>
<tr>
<td><code>lPeriod</code></td>
<td>The member ID of the process unit’s Period dimension member.</td>
</tr>
<tr>
<td><code>lEntity</code></td>
<td>The member ID of the process unit’s child Entity dimension member.</td>
</tr>
<tr>
<td><code>lParent</code></td>
<td>The member ID of the process unit’s parent Entity dimension member.</td>
</tr>
<tr>
<td><code>lValue</code></td>
<td>Long (ByVal). The member ID of the process unit’s Value dimension member.</td>
</tr>
</tbody>
</table>
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
--- | ---
`lScenario` | Long (ByVal). The member ID of the process unit's Scenario dimension member.
`lYear` | Long (ByVal). The member ID of the process unit's Year dimension member.
`lPeriod` | Long (ByVal). The member ID of the process unit's Period dimension member.
`lEntity` | Long (ByVal). The member ID of the process unit's child Entity dimension member.
`lParent` | Long (ByVal). The member ID of the process unit's parent Entity dimension member.
`lValue` | Long (ByVal). The member ID of the process unit's Value dimension member.
`vbUseAllValueMembers` | Boolean (ByVal). Determines whether process units for Value dimension members related to the `lValue` member are submitted. Pass TRUE to submit related process units, FALSE to submit only the process unit for the `lValue` member.
`bstrAnnotation` | String (ByVal). The comment for the submission.
`psNewProcessState` | 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.

```vba
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, "2012")
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
```
**Argument** | **Description**
--- | ---
`varabstrPaths` | String array (ByVal). The paths in which the documents to be attached were loaded. Folders in the path are delimited by backslashes (\). This array has a one-to-one correspondence with the `varabstrFiles` argument's array of filenames.

`varabstrFiles` | 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 `varabstrPaths` argument's array, otherwise an error occurs.

`psNewProcessState` | 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`.

```vba
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, "2012")
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 the HsvReports object. The HsvReports object’s methods are summarized in Table 30 on page 103, 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 103.

### CheckReportSecurityClass

Indicates whether the connected user has Read or All access rights to a report’s security class. If the user does not have access rights, a non-zero error number occurs as an HRESULT. If the user has access rights, zero occurs 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**

```plaintext
<HsvReports>.CheckReportSecurityClass bstrName, nReportFileType, nReportType
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrName</code></td>
<td>String (ByVal). The name of the report.</td>
</tr>
<tr>
<td><code>nReportFileType</code></td>
<td>Integer (ByVal). Identifies the file type of the report. Pass one of the constants listed in “Document File Type Constants” on page 894.</td>
</tr>
<tr>
<td><code>nReportType</code></td>
<td>Integer (ByVal). Identifies the report type. Pass one of the constants listed in “Document Type Constants” on page 893 that represent report types.</td>
</tr>
</tbody>
</table>
**CheckReportSecurityClass2**

Indicates whether the connected user has a specified level of access rights to a report’s security class. If the user does not have access rights, a non-zero error number occurs as an HRESULT. If the user has access rights, zero occurs 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**

```vbscript
<HsvReports>.CheckReportSecurityClass2 bstrPath, bstrFile, nReportType, nReportFileType, lAccessLevel
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrPath</code></td>
<td>String (ByVal). The path of the folder containing the report. The folders in the path are delimited by backslashes ().</td>
</tr>
<tr>
<td><code>bstrFile</code></td>
<td>String (ByVal). The name of the report.</td>
</tr>
<tr>
<td><code>nReportType</code></td>
<td>Integer (ByVal). Identifies the report type. Pass one of the constants listed in “Document Type Constants” on page 893 that represent report types.</td>
</tr>
<tr>
<td><code>nReportFileType</code></td>
<td>Integer (ByVal). Identifies the file type of the report. Pass one of the constants listed in “Document File Type Constants” on page 894.</td>
</tr>
<tr>
<td><code>lAccessLevel</code></td>
<td>Long (ByVal). The access rights for which to check. Use one of the following constants.</td>
</tr>
<tr>
<td></td>
<td>• <code>HFM_ACCESS_RIGHTS_READONLY</code>: Read access rights.</td>
</tr>
<tr>
<td></td>
<td>• <code>HFM_ACCESS_RIGHTS_ALL</code>: All access rights.</td>
</tr>
</tbody>
</table>

**Note:** These constants are members of the HFMConstants type library enumeration described in “Access Rights Constants” on page 881.

**CheckSecurityRole**

Indicates whether the connected user is assigned to the security role that enables users to save reports of a specified report type. If the user is not assigned to this role, a non-zero error number occurs as an HRESULT; if the user is assigned to this role, zero occurs as an HRESULT.

**Tip:** In Visual Basic, check for the return value with `Err.Number`.

**Syntax**

```vbscript
<HsvReports>.CheckSecurityRole nReportType
```
**DeleteDocuments**

Deletes documents from the application server.

**Syntax**

```<HsvReports>.DeleteDocuments varabstrPaths, varabstrNames, lDocumentType, lDocumentFileType, vbIncludeSubFolders```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| `varabstrPaths`     | 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` |
| `varabstrNames`     | String array (ByVal). The names of the documents to delete.                                                                                   |
| `lDocumentType`     | Long (ByVal). Identifies the type of document to be deleted. For valid values see “Document Type Constants” on page 893.                     |
| `lDocumentFileType` | Long (ByVal). Identifies the file types of the documents to be deleted. For valid values see “Document File Type Constants” on page 894.       |
| `vbIncludeSubFolders` | Boolean (ByVal). Determines if subdirectories are included.                                                                                 |

**DeleteReport**

Deletes a report from the application server.

**Syntax**

```<HsvReports>.DeleteReport bstrName, nReportFileType, nReportType```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrName</code></td>
<td>String (ByVal). The name of the report.</td>
</tr>
<tr>
<td><code>nReportFileType</code></td>
<td>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 894.</td>
</tr>
<tr>
<td><code>nReportType</code></td>
<td>Integer (ByVal). Identifies the type of report to be deleted. Pass one of the constants listed in “Document Type Constants” on page 893 that represent report types.</td>
</tr>
</tbody>
</table>
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`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varabstrFormNames</td>
<td>String array (ByVal). The names of the reports.</td>
</tr>
<tr>
<td>nReportFileType</td>
<td>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 894.</td>
</tr>
<tr>
<td>nReportType</td>
<td>Integer (ByVal). Identifies the type of report to be deleted. Pass one of the constants listed in “Document Type Constants” on page 893 that represent report types.</td>
</tr>
<tr>
<td>pvaralRc</td>
<td>Variant array. Returns any error numbers from DeleteReports.</td>
</tr>
</tbody>
</table>

- 0x399. Indicates that the user does not have All security access.
- 0x7c6. Indicates that the user is not assigned the proper role for deleting reports.

Example

DeleteReports is used in the Example for EnumReports.

DoesDocumentExist

Checks whether a document exists.

Syntax

`<HsvReports>.DoesDocumentExist bstrDocName, IDocumentType, IDocumentFileType`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrDocName</td>
<td>String. The name of the document.</td>
</tr>
<tr>
<td>IDocumentType</td>
<td>Long. The document type (see Document Type Constants section)</td>
</tr>
<tr>
<td>IDocumentFileType</td>
<td>Long. The document file type (see Document File Type Constants section)</td>
</tr>
</tbody>
</table>

Return Value

Boolean. True if the document exists, False if it does not exist.
Example

Dim bExists as Boolean

bExists = cReports.DoesDocumentExist "MyDoc", WEBOM_DOCTYPE_WEBGRID, WEBOM_DOCFILETYPE_XML

### DoesFolderExist

Checks whether a folder exists.

#### Syntax

```
<HSVReports>.DoesFolderExist bstrFolderName
```

#### Argument  Description

`bstrFolderName` String. The name of the folder being searched for.

#### Return Value

Boolean. True if the folder exists, False if the folder does not exist.

Example

Dim bExists as Boolean

bExists = cReports.DoesFolderExist "MyFolder"

### 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

`bstrPath` String (ByVal). The path of the folder containing the documents. The folders in the path are delimited by backslashes (\).
### Argument Description

**$IdDocumentType$**
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 893.

**$IdDocumentFileType$**
Long (ByVal). Identifies the file types of the documents to return. Valid values are represented by the HFMConstants type library constants listed in "Document File Type Constants" on page 894.

**$vbFilterByCurrentUserOnly$**
Boolean (ByVal). Determines whether only documents created by the connected user are returned. Pass TRUE to return only documents created by the connected user, FALSE otherwise.

**$vbFilterByCreateTime$**
Boolean (ByVal). *For internal use.*

**$dStartTime$**
Double (ByVal). *For internal use.*

**$dStopTime$**
Double (ByVal). *For internal use.*

**$pvarabstrDescriptions$**
Variant array. Returns the descriptions of the documents.

The array is returned as a String subtype.

**$pvaradTimestamp$**
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.

**$pvaralSecurityClass$**
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 specified folder.

```vba
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**

```vba
<HSVReports>.EnumDocumentsEx(bstrPath, varalDocumentType, varalDocumentFileType, vbFilterByCreateTime, dStartTime, dStopTime, lShowPrivateDocs, pvarabstrDescriptions, pvaradTimestamp, pvaralSecurityClass, pvarabIsPrivate, pvaralFolderContentType, pvarabstrDocOwner, pvaralFileType, pvaralReportType)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrPath</code></td>
<td>String (ByVal). The path of the folder containing the documents. The folders in the path are delimited by backslashes ().</td>
</tr>
<tr>
<td><code>varalDocumentType</code></td>
<td>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 893.</td>
</tr>
<tr>
<td><code>varalDocumentFileType</code></td>
<td>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 894.</td>
</tr>
<tr>
<td><code>vbFilterByCreateTime</code></td>
<td>Boolean (ByVal). For internal use.</td>
</tr>
<tr>
<td><code>dStartTime</code></td>
<td>Double (ByVal). For internal use.</td>
</tr>
<tr>
<td><code>dStopTime</code></td>
<td>Double (ByVal). For internal use.</td>
</tr>
<tr>
<td><code>lShowPrivateDocs</code></td>
<td>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 903.</td>
</tr>
<tr>
<td><code>pvarabstrDescriptions</code></td>
<td>Variant array. Returns the descriptions of the documents. The array is returned as a String subtype.</td>
</tr>
<tr>
<td><code>pvaradTimestamp</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>pvaralSecurityClass</code></td>
<td>Variant array. Returns the documents’ security class IDs. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pvarabIsPrivate</code></td>
<td>Variant array. Indicates whether the documents are flagged as public or private. The array is returned as a Boolean subtype.</td>
</tr>
<tr>
<td><code>pvaralFolderContentType</code></td>
<td>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 893. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td><code>pvarabstrDocOwner</code></td>
<td>Variant array. Returns an array containing the usernames of the documents’ owners. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pvarFileType</td>
<td>Variant array. Returns an array of the documents' file types. File types are represented by the HFMConstants type library constants listed in &quot;Document File Type Constants&quot; on page 894. The array is returned as a Long subtype.</td>
</tr>
<tr>
<td>pvarReportType</td>
<td>Variant array. Returns an array of the documents' document types. Document types are represented by the HFMConstants type library constants listed in &quot;Document Type Constants&quot; on page 893. The array is returned as a Long subtype.</td>
</tr>
</tbody>
</table>

**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 specified folder.

```vba
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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nReportFileType</td>
<td>Integer (ByVal). Identifies the file type of the report for which you want to return information. Pass one of the constants listed in &quot;Document File Type Constants&quot; on page 894.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>vbFilterByCurrentUserOnly</td>
<td>Boolean (ByVal). Determines whether only reports created by the connected user returned. Pass TRUE to only return reports created by the connected user, FALSE otherwise.</td>
</tr>
<tr>
<td>vbFilterByCreateTime</td>
<td>Boolean (ByVal). For internal use.</td>
</tr>
<tr>
<td>dStartTime</td>
<td>Double (ByVal). For internal use.</td>
</tr>
<tr>
<td>dStopTime</td>
<td>Double (ByVal). For internal use.</td>
</tr>
<tr>
<td>vbFilterByReportType</td>
<td>Boolean (ByVal). Determines whether reports of all types returned. Pass TRUE to return only the type of report identified by the nReportType argument, FALSE to return all types of reports.</td>
</tr>
<tr>
<td>nReportType</td>
<td>Integer (ByVal). If you pass TRUE for the vbFilterByReportType argument, this argument determines the type of report for which information returned. Pass one of the report type constants listed in “Document Type Constants” on page 893 that represent report types.</td>
</tr>
<tr>
<td>pvarabstrNames</td>
<td>Variant array. Returns the names of the reports that match the criteria specified in EnumReports’ arguments. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvarabstrDescriptions</td>
<td>Variant array. Returns the descriptions of the reports. The array is returned as a String subtype.</td>
</tr>
<tr>
<td>pvaradTimestamp</td>
<td>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.</td>
</tr>
</tbody>
</table>

**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.

```vbnet
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)`
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrPath</td>
<td>String (ByVal). The path to the document.</td>
</tr>
<tr>
<td>bstrName</td>
<td>String (ByVal). The name of the document.</td>
</tr>
<tr>
<td>lDocumentType</td>
<td>Long (ByVal). Identifies the type of document to return. Valid values are represented by the HFMConstants type library constants listed in &quot;Document Type Constants&quot; on page 893.</td>
</tr>
<tr>
<td>lDocumentFileType</td>
<td>Long (ByVal). Identifies the file type of the documents to return. Valid values are represented by the HFMConstants type library constants listed in &quot;Document File Type Constants&quot; on page 894.</td>
</tr>
<tr>
<td>pbstrDescription</td>
<td>String. Returns the description of the document.</td>
</tr>
<tr>
<td>plSecurityClass</td>
<td>Long. Returns the ID of the document’s security class.</td>
</tr>
</tbody>
</table>

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)`
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrPath</td>
<td>String (ByVal). The path to the document.</td>
</tr>
<tr>
<td>bstrName</td>
<td>String (ByVal). The name of the document.</td>
</tr>
<tr>
<td>lDocumentType</td>
<td>Long (ByVal). Identifies the type of document to return. Valid values are represented by the HFMConstants type library constants listed in &quot;Document Type Constants&quot; on page 893.</td>
</tr>
<tr>
<td>lDocumentFileType</td>
<td>Long (ByVal). Identifies the file type of the documents to return. Valid values are represented by the HFMConstants type library constants listed in &quot;Document File Type Constants&quot; on page 894.</td>
</tr>
<tr>
<td>pbstrDescription</td>
<td>String. Returns the description of the document.</td>
</tr>
<tr>
<td>plSecurityClass</td>
<td>Long. Returns the ID of the document’s security class.</td>
</tr>
<tr>
<td>pvbIsPrivate</td>
<td>Boolean. Indicates whether the document is public or private. TRUE indicates private, FALSE indicates public.</td>
</tr>
<tr>
<td>pbstrDocOwner</td>
<td>String. Returns the username of the document’s owner.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>pdTimestamp</td>
<td>Double. Returns the document’s timestamp. The timestamp can be converted to a Date format; for example, in Visual Basic you can convert with CDate.</td>
</tr>
<tr>
<td>plFolderContentType</td>
<td>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 &quot;Document Type Constants&quot; on page 893.</td>
</tr>
</tbody>
</table>

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**

```vbs
<HsvReports>.GetReport bstrName, nReportFileType, nReportType, plSecurityClass, pbstrDescription, pvarabyFile
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrName</td>
<td>String (ByVal). The name of the report. The report must exist on the application server.</td>
</tr>
<tr>
<td>nReportFileType</td>
<td>Integer (ByVal). Identifies the file type of the report. Pass one of the constants listed in &quot;Document File Type Constants&quot; on page 894.</td>
</tr>
<tr>
<td>nReportType</td>
<td>Integer (ByVal). Identifies the type of report. Pass one of the constants listed in &quot;Document Type Constants&quot; on page 893 that represent report types.</td>
</tr>
<tr>
<td>plSecurityClass</td>
<td>Long. Returns the ID number of the report’s security class.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> You can get the name of the security class by passing this ID to GetSecurityClassLabel. See &quot;GetSecurityClassLabel&quot; on page 483.</td>
</tr>
<tr>
<td>pbstrDescription</td>
<td>String. Returns the description of the report.</td>
</tr>
<tr>
<td>pvarabyFile</td>
<td>Variant array. Returns the report’s definition. The array is returned as a Byte subtype.</td>
</tr>
</tbody>
</table>

**Example**

The following example prints the definition of a journal report named Monthly to Visual Basic’s Immediate window.

```vbs
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**

```vbscript
<HsvReports>.SaveDocument bstrPath, bstrName, bstrDescription, lDocumentType, lDocumentFileType, lSecurityClass, varabyDocument, vbOverwriteExisting
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrPath</code></td>
<td>String (ByVal). The path of the folder containing the documents. The folders in the path are delimited by backslashes ().</td>
</tr>
<tr>
<td><code>bstrName</code></td>
<td>String (ByVal). The name of the document.</td>
</tr>
<tr>
<td><code>bstrDescription</code></td>
<td>String (ByVal). The description of the document.</td>
</tr>
<tr>
<td><code>lDocumentType</code></td>
<td>Long (ByVal). The document type. Valid values are represented by the HFMConstants type library constants listed in &quot;Document Type Constants&quot; on page 893.</td>
</tr>
<tr>
<td><code>lDocumentFileType</code></td>
<td>Long (ByVal). The file type of the document. Valid values are represented by the HFMConstants type library constants listed in &quot;Document File Type Constants&quot; on page 894.</td>
</tr>
<tr>
<td><code>lSecurityClass</code></td>
<td>Long (ByVal). The ID of the document's security class.</td>
</tr>
<tr>
<td><code>varabyDocument</code></td>
<td>Byte array (ByVal). The document's definition as an array of Bytes.</td>
</tr>
<tr>
<td><code>vbOverwriteExisting</code></td>
<td>Boolean (ByVal). Determines whether to overwrite a document of the same name on the application server. Pass TRUE to overwrite existing documents, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**Example**

The following subroutine saves the specified journal report in the .RPT format. The subroutine uses `HsvSecurityAccess.GetSecurityClassID` to get the ID of the security class that is passed.

```vbscript
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
```

606  HsvReports Type Library
**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**

```vba
<HsvReports>.SaveDocumentEx bstrPath, bstrName, bstrDescription, lDocumentType, lDocumentFileType, lSecurityClass, varabyDocument, vbIsPrivate, lContentType, vbOverwriteExisting
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrPath</code></td>
<td>String (ByVal). The path of the folder containing the documents. The folders in the path are delimited by backslashes ().</td>
</tr>
<tr>
<td><code>bstrName</code></td>
<td>String (ByVal). The name of the document.</td>
</tr>
<tr>
<td><code>bstrDescription</code></td>
<td>String (ByVal). The description of the document.</td>
</tr>
<tr>
<td><code>lDocumentType</code></td>
<td>Long (ByVal). The document type. Valid values are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 893.</td>
</tr>
<tr>
<td><code>lDocumentFileType</code></td>
<td>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 894.</td>
</tr>
<tr>
<td><code>lSecurityClass</code></td>
<td>Long (ByVal). The ID of the document’s security class.</td>
</tr>
<tr>
<td><code>varabyDocument</code></td>
<td>Byte array (ByVal). The document’s definition as an array of Bytes.</td>
</tr>
<tr>
<td><code>vbIsPrivate</code></td>
<td>Boolean (ByVal). Specifies whether the folder is public or private. Pass TRUE for private, FALSE for public.</td>
</tr>
<tr>
<td><code>lContentType</code></td>
<td>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> ignores this argument's value in that case. Valid values are represented by the HFMConstants type library constants listed in “Document Type Constants” on page 893.</td>
</tr>
<tr>
<td><code>vbOverwriteExisting</code></td>
<td>Boolean (ByVal). Determines whether to overwrite a document of the same name on the application server. Pass TRUE to overwrite existing documents, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**SetReport**

Saves a report on the application server.

**Syntax**

```vba
<HsvReports>.SetReport bstrName, nReportFileType, nReportType, lSecurityClass, bstrDescription, varabyFile, vbOverwriteExisting
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrName</td>
<td>String (ByVal). The name of the report.</td>
</tr>
<tr>
<td>nReportFileType</td>
<td>Integer (ByVal). Identifies the file type of the report. Pass one of the constants listed in “Document File Type Constants” on page 894.</td>
</tr>
<tr>
<td>nReportType</td>
<td>Integer (ByVal). Identifies the type of report. Pass one of the constants listed in “Document Type Constants” on page 893 that represent report types.</td>
</tr>
<tr>
<td>lSecurityClass</td>
<td>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 GetSecurityClassID. See “GetSecurityClassID” on page 482.</td>
</tr>
<tr>
<td>bstrDescription</td>
<td>String (ByVal). The description of the report.</td>
</tr>
<tr>
<td>varabyFile</td>
<td>Byte array (ByVal). The definition of the report. Pass the report definition as an array of bytes. You must pass a valid report definition. See the Oracle Hyperion Financial Management Administrator’s Guide.</td>
</tr>
<tr>
<td>vbOverwriteExisting</td>
<td>Boolean (ByVal). If a report with the name passed as the bstrName argument exists, this determines whether the existing report overwritten. Pass TRUE to overwrite an existing report, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**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. 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.

```vbscript
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 483 and “GetSecurityClassID” on page 482.
ValidateMembersAgainstSlice

For internal use.
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.
    
    See “About Subcubes” on page 53.

- **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; the information in an HsvMDDataBuffer object’s cells is not stored in an 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. See “HsvMDDataBufferLite Object Methods” on page 634.
Assign HsvMDDataBuffer object references with the `Set` and `New` keywords. For an example, see “HsvMDDataBuffer Object Overview” on page 105.

The HsvMDDataBuffer object’s methods are summarized in Table 31 on page 105, 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 is not unlocked and errors could occur.

**Syntax**

```csharp
<HsvMDDataBuffer>.BeginEnumeration plNumCubes
```

**Argument**

**Description**

- `plNumCubes` Long. Returns the number of subcubes in the HsvMDDataBuffer object.

**Example**

`BeginEnumeration` is used in the Example for `GetDataAtIndex`.

**CreateDataIndexList**

*Deprecated* - use “CreateDataIndexListExtDim” on page 613.

**CreateDataIndexListExtDim**

Creates an HsvMDIndexList object based upon the items in an HsvMDDataBuffer object. Supersedes `CreateDataIndexList`.

`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 added to the HsvMDIndexList object.
- `MEMBERALL` constant. Passing `MEMBERALL` means that all items containing the dimension member added to the HsvMDIndexList object.
- `MEMBERNOTUSED` constant. Passing `MEMBERNOTUSED` means that only those items containing `MEMBERNOTUSED` for the dimension member 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 added to the HsvMDIndexList object.

Tip: These constants are in the HFMConstants type library. See “Dimension Member Constants” on page 858.

Syntax

```
<HsvMDDataBuffer>.CreateDataIndexListExtDim pIUnkHfmPovCOM, ppIMDIndexListUnk
```

Argument Description

- `pIUnkHfmPovCOM` HfmPovCOM. HfmPovCOM object representing a POV. Members may be a member ID, MEMBERALL, MEMBERNOTUSED, or MEMBERANYONE.

- `ppIMDIndexListUnk` HsvMDIndexList object. Returns an HsvMDIndexList object containing the items in the HsvMDDataBuffer object that match the criteria specified in the other arguments.

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

**CreateDataIndexListEx**

*Deprecated* - use “CreateDataIndexListExtDim” on page 613.

**EndEnumeration**

Unlocks an HsvMDDataBuffer object locked by BeginEnumeration.

Syntax

```
<HsvMDDataBuffer>.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 HsvMDDataBuffer instance. *acmeData* and *ACMEDATA* are considered case-insensitive duplicate descriptions.

**Tip:** *SetCheckLineItemDetailsForCaseInsensitiveDuplicates* enables or disables whether the system should check for case-insensitive duplicate line item descriptions.

**Syntax**

```csharp
<HsvMDDataBuffer>.GetCheckLineItemDetailsForCaseInsensitiveDuplicates pvbEnabled
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvbEnabled</code></td>
<td>Boolean. Indicates whether case-insensitive duplicate checking is enabled. Returns TRUE if enabled, FALSE otherwise.</td>
</tr>
</tbody>
</table>

GetCubeIndexFromPOV
Returns the index of a subcube in an HsvMDDataBuffer object, using the member IDs of the subcube’s dimension members.

**Syntax**

```csharp
<HsvMDDataBuffer>.GetCubeIndexFromPOV lScenario, lYear, lEntity, lParent, lValue, plCubeIndex
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code></td>
<td>Long (ByVal). The member ID of the subcube’s Scenario dimension member.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the subcube’s Year dimension member.</td>
</tr>
<tr>
<td><code>lEntity</code></td>
<td>Long (ByVal). The member ID of the subcube’s Entity dimension member.</td>
</tr>
<tr>
<td><code>lParent</code></td>
<td>Long (ByVal). The member ID of the subcube’s Parent dimension member.</td>
</tr>
<tr>
<td><code>lValue</code></td>
<td>Long (ByVal). The member ID of the subcube’s Value dimension member.</td>
</tr>
<tr>
<td><code>plCubeIndex</code></td>
<td>Long. The index of the subcube, or -1 if the HsvMDDataBuffer object does not contain the subcube.</td>
</tr>
</tbody>
</table>

**Example**

GetCubeIndexFromPOV is used in the Example for GetPeriodIndexFromPOVForData.

GetCubePOVFromIndex
Returns the member IDs of a subcube’s dimension members, using the subcube’s index.
Syntax

\(<\text{HsvMDDataBuffer}>\).\text{GetCubePOVFromIndex}\ l\text{CubeIndex},\ p\text{Scenario},\ p\text{Year},\ p\text{Entity},\ p\text{Parent},\ p\text{Value}

**Argument**  **Description**

\(l\text{CubeIndex}\)  **Long (ByVal).** The index of the subcube.  
You can use \text{BeginEnumeration} to return the number of subcubes in an \text{HsvMDDataBuffer} object.

\(p\text{Scenario}\)  **Long.** Returns the member ID of the subcube's **Scenario** dimension member.

\(p\text{Year}\)  **Long.** Returns the member ID of the subcube's **Year** dimension member.

\(p\text{Entity}\)  **Long.** Returns the member ID of the subcube's **Entity** dimension member.

\(p\text{Parent}\)  **Long.** Returns the member ID of the \(p\text{Entity}\) argument's parent entity.

\(p\text{Value}\)  **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 \text{HsvMDDataBuffer} object. \text{BeginEnumeration} returns the number of subcubes in the object. For each subcube, \text{GetCubePOVFromIndex} returns the member IDs. To use the member IDs in the loop, replace the ellipsis (...) with code.

```vba
Dim lNumCubes As Long, lScen As Long, lYear As Long
Dim lEnt As Long, lPar As Long, lVal As Long
m_cMDBuffer.\text{BeginEnumeration} lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDBuffer.\text{GetCubePOVFromIndex} i, lScen, lYear, lEnt, _, lPar, lVal
    ... 'Insert code here
Next i
m_cMDBuffer.\text{EndEnumeration}
```

**GetData**

*Deprecated* - use “\text{GetDataExtDim}” on page 616.

**GetDataExtDim**

Returns a cell’s data, using the member IDs that identify the cell. It also indicates whether the cell is null.

**Tip:** To get a cell’s data by passing an index, use \text{GetDataAtIndex}.

Syntax

\(<\text{HsvMDDataBuffer}>\).\text{GetDataExtDim}\ p\text{IUnkHfmPovCOM},\ pd\text{Data},\ p\text{vbIsNoData}
### getDataAtIndex

**Deprecated** - use “getDataAtIndexExtDim” on page 617.

### getDataAtIndexExtDim

Returns the data for a cell, using the indexes that identify the cell. It also indicates whether the cell is null, and returns the member IDs of the cell’s View, Account, Intercompany Partner, and Custom dimension members. Supersedes GetDataAtIndex.

#### Syntax

```csharp
<HsvMDDataBuffer>.GetDataAtIndexExtDim lCubeIndex, lPeriodIndex, lCellIndex, ppIUnkHfmPovCOM, pdData, pvbIsNoData
```

#### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the cell’s subcube. Use the BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
<tr>
<td>lPeriodIndex</td>
<td>Long (ByVal). The index of the cell’s period. Use the GetNumPeriodsInCubeForData or GetPeriodIndexFromPOVForData to determine valid index values.</td>
</tr>
<tr>
<td>lCellIndex</td>
<td>Long (ByVal). The index of the cell. Use GetNumCellsForData to determine valid index values.</td>
</tr>
<tr>
<td>ppIUnkHfmPovCOM</td>
<td>HfmPovCOM. Returns the POV of the cell at the specified index.</td>
</tr>
<tr>
<td>pdData</td>
<td>Double. Returns the cell’s data.</td>
</tr>
<tr>
<td>pvbIsNoData</td>
<td>Boolean. Indicates whether the cell is null. Returns TRUE if null, FALSE otherwise.</td>
</tr>
</tbody>
</table>

#### Return Value

None.
Example
See HfmPovCOM documentation on how to access the members of the POV.

**GetDescription**

*Deprecated* - use “GetDescriptionExtDim” on page 618.

**GetDescriptionExtDim**

Returns a cell’s description, using the member IDs that identify the cell.

*Tip:* To get a cell’s description by passing an index, use GetDescriptionAtIndex. See “GetDescriptionAtIndex” on page 618.

**Syntax**

```c
<HsvMDDataBuffer>.GetDescriptionExtDim pIUnkHfmPovCOM, pbstrDescription
```

**Argument**

**Description**

- `pIUnkHfmPovCOM` HfmPovCOM. HfmPovCOM object representing the POV.
- `pbstrDescription` String. Returns the cell’s description.

**Return Value**

None.

Example

See HfmPovCOM documentation on how to set the POV.

**GetDescriptionAtIndex**

*Deprecated* - use “GetDescriptionAtIndexExtDim” on page 618.

**GetDescriptionAtIndexExtDim**

Returns the description for a cell, using the indexes that identify the cell. GetDescriptionAtIndex also returns the member IDs of the cell’s Account, Intercompany Partner, and Custom dimension members. Supersedes GetDescriptionAtIndex.

**Syntax**

```c
<HsvMDDataBuffer>.GetDescriptionAtIndexExtDim lCubeIndex, lPeriodIndex, lCellIndex, ppIUnkHfmPovCOM, pbstrDesc
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the cell’s subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
<tr>
<td>lPeriodIndex</td>
<td>Long (ByVal). The index of the cell’s period. Use GetNumPeriodsInCubeForDescriptions or GetPeriodIndexFromPOVForDescriptions to determine valid index values.</td>
</tr>
<tr>
<td>lCellIndex</td>
<td>Long (ByVal). The index of the cell. Use GetNumCellsForDescriptions to determine valid index values.</td>
</tr>
<tr>
<td>ppIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV at the specified index.</td>
</tr>
<tr>
<td>pbstrDesc</td>
<td>String. Returns the cell’s description.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to access the members of the POV.

**GetLineItems**

*Deprecated* - use “GetLineItemsExtDim” on page 619.

**GetLineItemsExtDim**

Returns the data and descriptions for a cell’s line items, using 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>.GetLineItemsExtDim pIUnkHfmPovCOM, pvbSavedAsYTD, pvaradData, pvarabstrDescriptions```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>pvbSavedAsYTD</td>
<td>Boolean. Indicates how the cell’s data is saved. Returns TRUE if saved as year-to-date, FALSE if saved as periodic.</td>
</tr>
<tr>
<td>pvaradData</td>
<td>Variant array. Returns the data for the cell’s line items. The array is returned as a Double subtype.</td>
</tr>
<tr>
<td>pvarabstrDescriptions</td>
<td>Variant array. Returns the descriptions of the cell’s line items. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

Return Value

None.
Example
See HfmPovCOM documentation on how to set the POV.

**GetLineItemsAtIndex**

*Deprecated* - use “GetLineItemsAtIndexExtDim” on page 620.

**GetLineItemsAtIndexExtDim**

Returns the data and descriptions for a cell’s line items, using the indexes that identify the cell. It 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. Supersedes GetLineItemsAtIndex

**Syntax**

```cpp
<HsvMDDataBuffer>.GetLineItemsAtIndexExtDim lCubeIndex, lPeriodIndex, lCellIndex, ppIUnkHfmPovCOM, pvbSavedAsYTD, pvaradData, pvarabstrDescriptions
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the cell’s subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.</td>
</tr>
<tr>
<td>lPeriodIndex</td>
<td>Long (ByVal). The index of the cell’s period. Use <code>GetNumPeriodsInCubeForLineItems</code> or <code>GetPeriodIndexFromPOVForLineItems</code> to determine valid index values.</td>
</tr>
<tr>
<td>lCellIndex</td>
<td>Long (ByVal). The index of the cell. Use <code>GetNumCellsForLineItems</code> to determine valid index values.</td>
</tr>
<tr>
<td>ppIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV for the specified index.</td>
</tr>
<tr>
<td>pvbSavedAsYTD</td>
<td>Boolean. Indicates how the cell is saved. Returns TRUE if the cell is saved as year-to-date, FALSE if saved as periodic.</td>
</tr>
<tr>
<td>pvaradData</td>
<td>Variant array. Returns the data for the cell’s line items. The array is returned as a Double subtype.</td>
</tr>
<tr>
<td>pvarabstrDescriptions</td>
<td>Variant array. Returns the descriptions of the cell’s line items. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to access the members of the POV.

**GetNumCellsForData**

Returns the number of cells that are in a period of a subcube and that contain data.
**Syntax**

```c
<HsvMDDataBuffer>.GetNumCellsForData lCubeIndex, lPeriodIndex, plNumCells
```

**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.

- **plNumCells**: 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**

```c
<HsvMDDataBuffer>.GetNumCellsForDescriptions lCubeIndex, lPeriodIndex, plNumCells
```

**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 `GetNumPeriodsInCubeForDescriptions` or `GetPeriodIndexFromPOVForDescriptions` to determine valid index values.

- **plNumCells**: 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**

```c
<HsvMDDataBuffer>.GetNumCellsForLineItems lCubeIndex, lPeriodIndex, plNumCells
```

**Argument**  
**Description**

- **lCubeIndex**: Long (ByVal). The index of the subcube. Use `BeginEnumeration` or `GetCubeIndexFromPOV` to determine valid index values.

---

---

---
Argument | Description
--- | ---

`lPeriodIndex` | Long (ByVal). The index of the period. Use `GetNumPeriodsInCubeForLineItems` or `GetPeriodIndexFromPOVForLineItems` to determine valid index values.

`plNumCells` | 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, plNumPeriods```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lCubeIndex</code></td>
<td>Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.</td>
</tr>
<tr>
<td><code>plNumPeriods</code></td>
<td>Long. Returns the number of periods with cells that contain data.</td>
</tr>
</tbody>
</table>

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, plNumPeriods```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lCubeIndex</code></td>
<td>Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.</td>
</tr>
<tr>
<td><code>plNumPeriods</code></td>
<td>Long. Returns the number of periods with cells that contain descriptions.</td>
</tr>
</tbody>
</table>

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

```csharp
<HsvMDDataBuffer>.GetNumPeriodsInCubeForLineItems lCubeIndex, plNumPeriods
```

**Argument** | **Description**
--- | ---
`lCubeIndex` | Long (ByVal). The index of the subcube. Use `BeginEnumeration` or `GetCubeIndexFromPOV` 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, using the subcube’s index and the period’s member ID. An index is returned only for periods with cells that contain data.

Syntax

```csharp
<HsvMDDataBuffer>.GetPeriodIndexFromPOVForData 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 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 were 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.

```vbnet
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
```
GetPeriodIndexFromPOVForDescriptions

Returns the index of a period within a subcube, using 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

1CubeIndex Long (ByVal). The index of the subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.

1Period 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 were 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, using 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

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 were 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, using 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.</td>
</tr>
<tr>
<td>lPeriodIndex</td>
<td>Long (ByVal). The index of the cell's period. To determine valid index values, use <code>GetNumPeriodsInCubeForData</code>.</td>
</tr>
<tr>
<td>plPeriod</td>
<td>Long. Returns the member ID of the period.</td>
</tr>
</tbody>
</table>

### GetPeriodPOVFromIndexForDescriptions

Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain descriptions.

**Syntax**

```csharp
<HsvMDDataBuffer>.GetPeriodPOVFromIndexForDescriptions lCubeIndex, lPeriodIndex, plPeriod
```

### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.</td>
</tr>
<tr>
<td>lPeriodIndex</td>
<td>Long (ByVal). The index of the cell's period. To determine valid index values, use <code>GetNumPeriodsInCubeForData</code>.</td>
</tr>
<tr>
<td>plPeriod</td>
<td>Long. Returns the member ID of the period.</td>
</tr>
</tbody>
</table>

### GetPeriodPOVFromIndexForLineItems

Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain line items.

**Syntax**

```csharp
<HsvMDDataBuffer>.GetPeriodPOVFromIndexForLineItems lCubeIndex, lPeriodIndex, plPeriod
```

### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.</td>
</tr>
<tr>
<td>lPeriodIndex</td>
<td>Long (ByVal). The index of the cell's period. To determine valid index values, use <code>GetNumPeriodsInCubeForLineItems</code>.</td>
</tr>
<tr>
<td>plPeriod</td>
<td>Long. Returns the member ID of the period.</td>
</tr>
</tbody>
</table>

### GetPMErrorRecordCount

*For internal use.*
GetPMRecordCount
For internal use.

GetRecordFromPMBuffer
For internal use.

GetRecordFromPMErrorBuffer
For internal use.

GetSortedNature
For internal use.

InsertDataAtBeginning
Deprecated - use “InsertDataAtBeginningExtDim” on page 628.

InsertDataAtBeginningExtDim
Inserts data for a cell at the top of an HsvMDDataBuffer object.

Syntax

```cpp
<HsvMDDataBuffer>.InsertDataAtBeginningExtDim pIUnkHfmPovCOM, dData, vbIsNoData
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>dData</td>
<td>Double (ByVal). The cell’s data.</td>
</tr>
<tr>
<td>vbIsNoData</td>
<td>Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.</td>
</tr>
</tbody>
</table>

InsertDataAtEnd
Deprecated - use “InsertDataAtEndExtDim” on page 628.

InsertDataAtEndExtDim
Inserts data for a cell at the bottom of an HsvMDDataBuffer object. Supersedes InsertDataAtEnd.
**Syntax**

```csharp
<HsvMDDataBuffer>.InsertDataAtEndExtDim pIUnkHfmPovCOM, dData, vbIsNoData
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td><code>dData</code></td>
<td>Double (ByVal). The cell's data.</td>
</tr>
<tr>
<td><code>vbIsNoData</code></td>
<td>Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

---

**InsertDescriptionAtBeginning**

*Deprecated* - use “InsertDescriptionAtBeginningExtDim” on page 629.

---

**InsertDescriptionAtBeginningExtDim**

Inserts a cell description at the top of an HsvMDDataBuffer object.

**Syntax**

```csharp
<HsvMDDataBuffer>.InsertDescriptionAtBeginningExtDim pIUnkHfmPovCOM, bstrDescription
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td><code>bstrDescription</code></td>
<td>String (ByVal). The cell's description.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

---

**InsertDescriptionAtEnd**

*Deprecated* - use “InsertDescriptionAtEndExtDim” on page 630.
**InsertDescriptionAtEndExtDim**

Inserts a cell description at the bottom of an HsvMDDataBuffer object.

**Syntax**

```<HsvMDDataBuffer>.InsertDescriptionAtEndExtDim pIUnkHfmPovCOM, bstrDescription```

**Argument**

- **pIUnkHfmPovCOM**  
  HfmPovCOM object representing the POV.

- **bstrDescription**  
  String (ByVal). The cell’s description.

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

**InsertLineItemsAtBeginning**

*Deprecated* - use “InsertLineItemsAtBeginningExtDim” on page 630.

**InsertLineItemsAtBeginningExtDim**

Inserts line items for a cell at the top of an HsvMDDataBuffer object. Supersedes InsertLineItemsAtBeginning.

**Syntax**

```<HsvMDDataBuffer>.InsertLineItemsAtBeginningExtDim pIUnkHfmPovCOM, varadData, varabstrDescriptions```

**Argument**

- **pIUnkHfmPovCOM**  
  HfmPovCOM. HfmPovCOM object representing the POV.

- **varadData**  
  Double array (ByVal). The data for the cell’s line items.

- **varabstrDescriptions**  
  String array (ByVal). The descriptions of the cell’s line items.

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.
**InsertLineItemsAtEnd**

*Deprecated* - use “InsertLineItemsAtEndExtDim” on page 631.

**InsertLineItemsAtEndExtDim**

Inserts line items for a cell at the bottom of an HsvMDDataBuffer object.

**Syntax**

```
<HsvMDDataBuffer>.InsertLineItemsAtEndExtDim pIUnkHfmPovCOM, vbSavedAsYTD, varadData, varabstrDescriptions
```

**Argument**  
**Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>vbSavedAsYTD</td>
<td>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.</td>
</tr>
<tr>
<td>varadData</td>
<td>Double array (ByVal). The data for the cell's line items.</td>
</tr>
<tr>
<td>varabstrDescriptions</td>
<td>String array (ByVal). The descriptions of the cell's line items.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

**InsertRecordIntoPMBuffer**

*For internal use.*

**InsertRecordIntoPMErrorBuffer**

*For internal use.*

**RemoveAll**

Clears all the cells from an HsvMDDataBuffer object.

**Syntax**

```
<HsvMDDataBuffer>.RemoveAll
```
SetCheckLineItemDetailsForCaseInsensitiveDuplicates

Specifies whether the system checks for case-insensitive duplicate line item descriptions. For example, `acmeData` and `ACMEDATA` are 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 occurs.

Syntax

```<HsvMDDataBuffer>.SetCheckLineItemDetailsForCaseInsensitiveDuplicates vbEnable```

**Argument Description**

- `vbEnable` Boolean (ByVal). Specifies whether to check for case-insensitive duplicates. Pass TRUE to check, FALSE otherwise.

SetData

*Deprecated* - use “SetDataExtDim” on page 632.

SetDataExtDim

Inserts a cell’s data into an HsvMDDataBuffer object. Supersedes SetData.

Syntax

```<HsvMDDataBuffer>.SetDataExtDim pIUnkHfmPovCOM, dData, vbIsNoData, vbAddToExistingData, plNumElementsInDataUnit```

**Argument Description**

- `pIUnkHfmPovCOM` HfmPovCOM. HfmPovCOM object representing the POV.
- `dData` Double (ByVal). The cell’s data.
- `vbIsNoData` Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.
- `vbAddToExistingData` Boolean (ByVal). If the HsvMDDataBuffer object already contains data for the cell, this argument determines whether the existing data overwritten. Pass TRUE to add the `dData` argument’s data to the existing data, FALSE to overwrite the existing data.
- `plNumElementsInDataUnit` 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 SetData adds the cell.

**Return Value**

None.
Example
See HfmPovCOM documentation on how to set the POV.

**SetDescription**

*Deprecated* - use “SetDescriptionExtDim” on page 633.

**SetDescriptionExtDim**

 Inserts a cell’s description into an HsvMDDataBuffer object. Supersedes SetDescription.

Syntax

```csharp
<HsvMDDataBuffer>.SetDescriptionExtDim pIUnkHfmPovCOM, bstrDescription
```

**Argument**

**Description**

*pIUnkHfmPovCOM*  HfmPovCOM. HfmPovCOM object representing the POV

*bstrDescription*  String (ByVal). The cell’s description.

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

**SetGrowByAmount**

Creates additional memory for an HsvMDDataBuffer object.

Syntax

```csharp
<HsvMDDataBuffer>.SetGrowByAmount lGrowByAmount
```

**Argument**

**Description**

*lGrowByAmount*  Long (ByVal). The number of cells for which you want to create memory.

**SetLineItems**

*Deprecated* - use “SetLineItemsExtDim” on page 634.
**SetLineItemsExtDim**

Inserts a cell’s line items into an HsvMDDataBuffer object. Supersedes SetLineItems.

**Syntax**

```csharp
<HsvMDDataBuffer>.SetLineItemsExtDim pIUnkHfmPovCOM, vbSavedAsYTD, varadData, varabstrDescriptions
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td><code>vbSavedAsYTD</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>varadData</code></td>
<td>Double array (ByVal). The cell’s line item data.</td>
</tr>
<tr>
<td><code>varabstrDescriptions</code></td>
<td>String array (ByVal). The cell’s line item descriptions.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

**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; the information in an HsvMDDataBufferLite object’s cells is not stored in an 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. See “UpdateDataUsingMDDataBuffer” on page 390.

- **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. See “AddDataToMDDataBuffer” on page 321.

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 is always cached to RAM. You can change the minimum number of subcubes that are cached with `SetMinCubesInCache`.

If you want similar functionality, but with all the subcubes cached to RAM, use the HsvMDDataBuffer object. See “HsvMDDataBuffer Object Methods” on page 612.

Assign HsvMDDataBufferLite object references with the `Set` and `New` keywords. For an example, see “HsvMDDataBufferLite Object Overview” on page 108.

The HsvMDDataBufferLite object’s methods are summarized in Table 32 on page 109, 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 is not unlocked and errors could occur.

**Syntax**

```c
<HsvMDDataBufferLite>.BeginEnumeration plNumCubes
```

**Argument** | **Description**
--- | ---
`plNumCubes` | Long. Returns the number of subcubes in the HsvMDDataBufferLite object.

**Example**

`BeginEnumeration` is used in the **Example** for `GetDataAtIndex`.

### CreateDataIndexList

*Deprecated* - use “CreateDataIndexListExtDim” on page 636.
CreateDataIndexListExtDim

Creates an HsvMDIndexList object based upon the items in an HsvMDDataBufferLite object. Supersedes CreateDataIndexList.

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 added to the HsvMDIndexList object.
- MEMBERALL constant. Passing MEMBERALL means that all items containing the dimension member added to the HsvMDIndexList object.
- MEMBERNOTUSED constant. Passing MEMBERNOTUSED means that only those items containing MEMBERNOTUSED for the dimension member 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 added to the HsvMDIndexList object.

Tip: These constants are in the HFMConstants type library. See “Dimension Member Constants” on page 858.

Syntax

```cpp
<HsvMDDataBufferLite>.CreateDataIndexListExtDim pIUnkHfmPovCOM, ppIMDIndexListUnk
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV. Members may be a member ID, MEMBERALL, MEMBERNOTUSED, or MEMBERANYONE.</td>
</tr>
<tr>
<td>ppIMDIndexListUnk</td>
<td>HsvMDIndexList object. Returns an HsvMDIndexList object containing the items in the HsvMDDataBufferLite object that match the criteria specified in the other arguments.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

EraseRecordFromPMBuffer

For internal use.
**EndEnumeration**

Unlocks an HsvMDDataBufferLite object locked by BeginEnumeration.

**Syntax**

```csharp
<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` are considered case-insensitive duplicate descriptions.

**Tip:** `SetCheckLineItemDetailsForCaseInsensitiveDuplicates` enables or disables whether the system should check for case-insensitive duplicate line item descriptions.

**Syntax**

```csharp
<HsvMDDataBufferLite>.GetCheckLineItemDetailsForCaseInsensitiveDuplicates pvbEnabled
```

**Argument** | **Description**
---|---
`pvbEnabled` | 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, using the member IDs of the subcube's dimension members.

**Syntax**

```csharp
<HsvMDDataBufferLite>.GetCubeIndexFromPOV lScenario, lYear, lEntity, lParent, lValue, plCubeIndex
```

**Argument** | **Description**
---|---
`lScenario` | Long (ByVal). The member ID of the subcube's Scenario dimension member.
`lYear` | Long (ByVal). The member ID of the subcube's Year dimension member.
`lEntity` | Long (ByVal). The member ID of the subcube's Entity dimension member.
`lParent` | Long (ByVal). The member ID of the subcube's Parent dimension member.
`lValue` | Long (ByVal). The member ID of the subcube's Value dimension member.
### GetCubePOVFromIndex

Returns the member IDs of a subcube’s dimension members, using the subcube’s index.

**Syntax**

```csharp
<HsvMDDataBufferLite>.GetCubePOVFromIndex lcubeIndex, plScenario, plYear, plEntity, plParent, plValue
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lcubeIndex</code></td>
<td>Long (ByVal). The index of the subcube.</td>
</tr>
<tr>
<td><code>plScenario</code></td>
<td>Long. Returns the member ID of the subcube’s Scenario dimension member.</td>
</tr>
<tr>
<td><code>plYear</code></td>
<td>Long. Returns the member ID of the subcube’s Year dimension member.</td>
</tr>
<tr>
<td><code>plEntity</code></td>
<td>Long. Returns the member ID of the subcube’s Entity dimension member.</td>
</tr>
<tr>
<td><code>plParent</code></td>
<td>Long. Returns the member ID of the <code>plEntity</code> argument’s parent entity.</td>
</tr>
<tr>
<td><code>plValue</code></td>
<td>Long. Returns the member ID of the subcube’s Value dimension member.</td>
</tr>
</tbody>
</table>

**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.

```vbscript
Dim lNumCubes As Long, lScen As Long, lYear As Long
Dim lEnt As Long, lPar As Long, lVal As Long
m_cMDBufferLite.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDBufferLite.GetCubePOVFromIndex i, lScen, lYear, lEnt, lPar, lVal
    ... 'Insert code here
Next i
m_cMDBufferLite.EndEnumeration
```

### GetData

*Deprecated* - use “GetDataExtDim” on page 639.
**GetDataExtDim**

Returns a cell’s data, using 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. See “GetDataAtIndex” on page 639.

**Syntax**

```csharp
<HsvMDDataBufferLite>.GetDataExtDim pIUnkHfmPovCOM, pdData, pvbIsNoData
```

**Argument** | **Description**
--- | ---
`pIUnkHfmPovCOM` | HfmPovCOM. An HfmPovCOM object representing the cell’s POV.
`pdData` | Double. Returns the cell’s data.
`pvbIsNoData` | Boolean. Indicates whether the cell is null. Returns TRUE if null, FALSE otherwise.

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

---

**GetDataAtIndex**

Returns the data for a cell, using 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**

```csharp
<HsvMDDataBufferLite>.GetDataAtIndex lCubeIndex, lPeriodIndex, lCellIndex, plView, plAccount, plICP, plCustom1, plCustom2, plCustom3, plCustom4, pdData, pvbIsNoData
```

**Argument** | **Description**
--- | ---
`lCubeIndex` | Long (ByVal). The index of the cell’s subcube. Use the BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.
`lPeriodIndex` | Long (ByVal). The index of the cell’s period. Use the GetNumPeriodsInCubeForData or GetPeriodIndexFromPOVForData to determine valid index values.
`lCellIndex` | Long (ByVal). The index of the cell. Use GetNumCellsForData to determine valid index values.
`plView` | Long. Returns the member ID of the cell’s View dimension member.
`plAccount` | Long. Returns the member ID of the cell’s Account dimension member.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plICP</td>
<td>Long. Returns the member ID of the cell’s Intercompany Partner dimension member.</td>
</tr>
<tr>
<td>plCustom1</td>
<td>Long. Returns the member ID of the cell’s Custom 1 dimension member.</td>
</tr>
<tr>
<td>plCustom2</td>
<td>Long. Returns the member ID of the cell’s Custom 2 dimension member.</td>
</tr>
<tr>
<td>plCustom3</td>
<td>Long. Returns the member ID of the cell’s Custom 3 dimension member.</td>
</tr>
<tr>
<td>plCustom4</td>
<td>Long. Returns the member ID of the cell’s Custom 4 dimension member.</td>
</tr>
<tr>
<td>pdData</td>
<td>Double. Returns the cell’s data.</td>
</tr>
<tr>
<td>pvbIsNoData</td>
<td>Boolean. Indicates whether the cell is null. Returns TRUE if null, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**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 work 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.

```vbnet
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_cMDBufferLite.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDBufferLite.GetNumPeriodsInCubeForData i, lNumPers
    For j = 0 To lNumPers - 1
        m_cMDBufferLite.GetNumCellsForData i, j, lNumCells
        For k = 0 To lNumCells - 1
            m_cMDBufferLite.GetDataAtIndex i, j, k, lVw, lAccount, _
            lIntCo, lCus1, lCus2, lCus3, lCus4, dMdData, bMdNoData
            Debug.Print dMdData
        Next k
    Next j
Next i
m_cMDBufferLite.EndEnumeration
```

**GetDescription**

*Deprecated* - use “GetDescriptionExtDim” on page 641.
GetDescriptionExtDim

Returns a cell’s description, using the member IDs that identify the cell.

Tip: To get a cell’s description by passing an index, use GetDescriptionAtIndex. See “GetDescriptionAtIndex” on page 641.

Syntax

<HsvMDDataBufferLite>.GetDescriptionExtDim pIUnkHfmPovCOM, pbstrDescription

Argument Description

pIUnkHfmPovCOM HfmPovCOM object representing the POV.

pbstrDescription String. Returns the cell’s description.

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

GetDescriptionAtIndex

Returns the description for a cell, using 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

lCubeIndex Long (ByVal). The index of the cell’s subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.

lPeriodIndex Long (ByVal). The index of the cell’s period. Use GetNumPeriodsInCubeForDescriptions or GetPeriodIndexFromPOVForDescriptions to determine valid index values.

lCellIndex Long (ByVal). The index of the cell. Use GetNumCellsForDescriptions to determine valid index values.

plAccount Long. Returns the member ID of the cell’s Account dimension member.

plICP Long. Returns the member ID of the cell’s Intercompany Partner dimension member.

plCustom1 Long. Returns the member ID of the cell’s Custom 1 dimension member.
Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plCustom2</td>
<td>Long. Returns the member ID of the cell’s Custom 2 dimension member.</td>
</tr>
<tr>
<td>plCustom3</td>
<td>Long. Returns the member ID of the cell’s Custom 3 dimension member.</td>
</tr>
<tr>
<td>plCustom4</td>
<td>Long. Returns the member ID of the cell’s Custom 4 dimension member.</td>
</tr>
<tr>
<td>pbstrDesc</td>
<td>String. Returns the cell’s description.</td>
</tr>
</tbody>
</table>

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.

```vbnet
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_cMDBufferLite.BeginEnumeration lNumCubes
For i = 0 To lNumCubes - 1
    m_cMDBufferLite.GetNumPeriodsInCubeForDescriptions i, _
    lNumPers
    For j = 0 To lNumPeriods - 1
        m_cMDBufferLite.GetNumCellsForDescriptions i, j, lNumCells
        For k = 0 To lNumCells - 1
            m_cMDBufferLite.GetDescriptionAtIndex i, j, k, lAccount, _
            lIntCo, lCus1, lCus2, lCus3, lCus4, sDesc
            Debug.Print sDesc
        Next k
    Next j
Next i
m_cMDBufferLite.EndEnumeration
```

GetLineItems

*Deprecated* - use “GetLineItemsExtDim” on page 643.
**GetLineItemsExtDim**

Returns the data and descriptions for a cell’s line items, using 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**

```csharp
<HsvMDDataBufferLite>.GetLineItemsExtDim pIUnkHfmPovCOM, pvbSavedAsYTD, pvaradData, pvarabstrDescriptions
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>pvbSavedAsYTD</td>
<td>Boolean. Indicates how the cell’s data is saved. Returns TRUE if saved as year-to-date, FALSE if saved as periodic.</td>
</tr>
<tr>
<td>pvaradData</td>
<td>Variant array. Returns the data for the cell’s line items. The array is returned as a Double subtype.</td>
</tr>
<tr>
<td>pvarabstrDescriptions</td>
<td>Variant array. Returns the descriptions of the cell’s line items. The array is returned as a String subtype.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

---

**GetLineItemsAtIndex**

*Deprecated* - use “GetLineItemsAtIndexExtDim” on page 643.

**GetLineItemsAtIndexExtDim**

Returns the data and descriptions for a cell’s line items, using 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**

```csharp
<HsvMDDataBufferLite>.GetLineItemsAtIndexExtDim lCubeIndex, lPeriodIndex, lCellIndex, ppIUnkHfmPovCOM, pvbSavedAsYTD, pvaradData, pvarabstrDescriptions
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the cell’s subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
</tbody>
</table>
Argument | Description
---|---
_iPeriodIndex_ | Long (ByVal). The index of the cell’s period. Use `GetNumPeriodsInCubeForLineItems` or `GetPeriodIndexFromPOVForLineItems` to determine valid index values.
_iCellIndex_ | Long (ByVal). The index of the cell. Use `GetNumCellsForLineItems` to determine valid index values.
.ppUnkhfmPovCOM_ | HfmPovCOM. HfmPovCOM object representing the POV for the specified index.
_pvSavedAsYTD_ | Boolean. Indicates how the cell is saved. Returns TRUE if the cell is saved as year-to-date, FALSE if saved as periodic.
_pvaradData_ | Variant array. Returns the data for the cell’s line items. The array is returned as a Double subtype.
_pvarabstrDescriptions_ | Variant array. Returns the descriptions of the cell’s line items. The array is returned as a String subtype.

Return Value
None.

Example
See HfmPovCOM documentation on how to access the members of the POV.

GetNumCellsForData

Returns the number of cells that are in a period of a subcube and that contain data.

Syntax

```csharp
<HsvMDDataBufferLite>.GetNumCellsForData lCubeIndex, lPeriodIndex, plNumCells
```

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.
_plNumCells_ | 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

```csharp
<HsvMDDataBufferLite>.GetNumCellsForDescriptions lCubeIndex, lPeriodIndex, plNumCells
```
**Argument**  **Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
<tr>
<td>lPeriodIndex</td>
<td>Long (ByVal). The index of the period. Use GetNumPeriodsInCubeForDescriptions or GetPeriodIndexFromPOVForDescriptions to determine valid index values.</td>
</tr>
<tr>
<td>plNumCells</td>
<td>Long. Returns the number of cells that contain descriptions.</td>
</tr>
</tbody>
</table>

**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**

```c
<HsvMDDataBufferLite>.GetNumCellsForLineItems lCubeIndex, lPeriodIndex, plNumCells
```

**Argument**  **Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
<tr>
<td>lPeriodIndex</td>
<td>Long (ByVal). The index of the period. Use GetNumPeriodsInCubeForLineItems or GetPeriodIndexFromPOVForLineItems to determine valid index values.</td>
</tr>
<tr>
<td>plNumCells</td>
<td>Long. Returns the number of cells that contain line items.</td>
</tr>
</tbody>
</table>

**Example**

GetNumCellsForLineItems is used in the example for GetLineItemsAtIndexExtDim.

---

**GetNumPeriodsInCubeForData**

Returns the number of periods that are in a subcube and that contain cells with data.

**Syntax**

```c
<HsvMDDataBufferLite>.GetNumPeriodsInCubeForData lCubeIndex, plNumPeriods
```

**Argument**  **Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
<tr>
<td>plNumPeriods</td>
<td>Long. Returns the number of periods with cells that contain data.</td>
</tr>
</tbody>
</table>
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**

```csharp
<HsvMDDataBufferLite>.GetNumPeriodsInCubeForDescriptions lCubeIndex, plNumPeriods
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
<tr>
<td>plNumPeriods</td>
<td>Long. Returns the number of periods with cells that contain descriptions.</td>
</tr>
</tbody>
</table>

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**

```csharp
<HsvMDDataBufferLite>.GetNumPeriodsInCubeForLineItems lCubeIndex, plNumPeriods
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
<tr>
<td>plNumPeriods</td>
<td>Long. Returns the number of periods with cells that contain line items.</td>
</tr>
</tbody>
</table>

Example

GetNumPeriodsInCubeForLineItems is used in the example for GetLineItemsAtIndexExtDim.

**GetPeriodIndexFromPOVForData**

Returns the index of a period within a subcube, using 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

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 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 were 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_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.GetPeriodIndexFromPOVForData lCubeIx, _
    m_lPer, lPerIx
    ' Exit if the period does not have cells with data
    If lPerIx = -1 Then
        m_cMDBufferLite.EndEnumeration
        Exit Sub
    Else
        m_cMDBufferLite.GetNumCellsForData lCubeIx, _
        lPerIx, lNumCells
        For i = 0 To lNumCells - 1
            m_cMDBufferLite.GetDataAtIndex lCubeIx, lPerIx, i, _
                lView, lAcct, lICP, lCust1, lCust2, lCust3, _
                lCust4, dCellData, bIsNoData
            Debug.Print dCellData
        Next i
    End If
End If
m_cMDBufferLite.EndEnumeration
## GetPeriodIndexFromPOVForDescriptions

Returns the index of a period within a subcube, using 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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>plPeriodIndex</td>
<td>Long. Returns the period’s index, or -1 if the period does not contain cells with descriptions.</td>
</tr>
</tbody>
</table>

### 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 were 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.

```vba
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_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.GetPeriodIndexFromPOVForDescriptions _
    lCubeIx, m_lPer, lPerIx
    ' Exit if the period does not have cells with descriptions
    If lPerIx = -1 Then
      m_cMDBufferLite.EndEnumeration
      Exit Sub
    Else
      m_cMDBufferLite.GetNumCellsForDescriptions lCubeIx, _
      lPerIx, lNumCells
      For i = 0 To lNumCells - 1
        m_cMDBufferLite.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, using the subcube’s index and the period’s member ID. An index is returned only for periods with cells that contain line items.

Syntax

```plaintext
<HsvMDDataBufferLite>.GetPeriodIndexFromPOVForLineItems lCubeIndex, lPeriod, plPeriodIndex
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lCubeIndex</code></td>
<td>Long (ByVal). The index of the subcube. Use <code>BeginEnumeration</code> or <code>GetCubeIndexFromPOV</code> to determine valid index values.</td>
</tr>
<tr>
<td><code>lPeriod</code></td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td><code>plPeriodIndex</code></td>
<td>Long. Returns the period’s index, or -1 if the period does not contain cells with line items.</td>
</tr>
</tbody>
</table>

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 were 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, `GetLineItemsAtIndexExtDim` returns the line items’ data and descriptions, which are then printed to the Immediate window.

```vbscript
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
    Else
        GetNumCellsForLineItems lCubeIx, lPerIx
        ' Loop through each cell
        For i = 0 To lNumCells - 1
            dCellData = vdCellData(i)
            bYTD = False
            vsDescs = GetLineItemsAtIndexExtDim lCubeIx, lPerIx, i
            Debug.Print vsDescs
        Next i
    End If
End If
m_cMDBufferLite.EndEnumeration
```
Exit Sub
Else
   m_cMDBufferLite.GetNumCellsForLineItems lCubeIx, _
   lPerIx, lNumCells
   For i = 0 To lNumCells - 1
      m_cMDBufferLite.GetLineItemsAtIndexExtDim 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, using a subcube index and a period index. IDs are returned only for periods with cells that contain data.

Syntax

GetPeriodPOVFromIndexForData lCubeIndex, lPeriodIndex, plPeriod

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 cell’s period. To determine valid index values, use GetNumPeriodsInCubeForData.
plPeriod  Long. Returns the member ID of the period.

GetPeriodPOVFromIndexForDescriptions

Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain descriptions.

Syntax

GetPeriodPOVFromIndexForDescriptions lCubeIndex, lPeriodIndex, plPeriod

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 cell’s period. To determine valid index values, use GetNumPeriodsInCubeForDescriptions.
plPeriod  Long. Returns the member ID of the period.
GetPeriodPOVFromIndexForLineItems

Returns the member ID of a period, using a subcube index and a period index. IDs are returned only for periods with cells that contain line items.

Syntax

```<HsvMDDataBufferLite>.GetPeriodPOVFromIndexForLineItems lCubeIndex, lPeriodIndex, plPeriod```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lCubeIndex</td>
<td>Long (ByVal). The index of the subcube. Use BeginEnumeration or GetCubeIndexFromPOV to determine valid index values.</td>
</tr>
<tr>
<td>lPeriodIndex</td>
<td>Long (ByVal). The index of the cell’s period. To determine valid index values, use GetNumPeriodsInCubeForLineItems.</td>
</tr>
<tr>
<td>plPeriod</td>
<td>Long. Returns the member ID of the period.</td>
</tr>
</tbody>
</table>

GetPMErrorRecordCount

For internal use.

GetPMRecordCount

For internal use.

GetRecordFromPMBuffer

For internal use.

GetRecordFromPMErrorBuffer

For internal use.

GetSortedNature

For internal use.

InsertDataAtEnd

Deprecated - use “InsertDataAtEnd” on page 651.
**InsertDataAtEndExtDim**

Inserts data for a cell at the bottom of an HsvMDDataBufferLite object.

Syntax

```csharp
<HsvMDDataBufferLite>.InsertDataAtEndExtDim pIUnkHfmPovCOM, dData, vbIsNoData
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>dData</td>
<td>Double (ByVal). The cell’s data.</td>
</tr>
<tr>
<td>vbIsNoData</td>
<td>Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

**InsertDescriptionAtEnd**

*Deprecated* - use “InsertDescriptionAtEndExtDim” on page 652.

**InsertDescriptionAtEndExtDim**

Inserts a cell description at the bottom of an HsvMDDataBufferLite object.

Syntax

```csharp
<HsvMDDataBufferLite>.InsertDescriptionAtEndExtDim pIUnkHfmPovCOM, bstrDescription
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>bstrDescription</td>
<td>String (ByVal). The cell’s description.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.
InsertLineItemsAtEnd

*Deprecated* - use “InsertLineItemsAtEndExtDim” on page 653.

InsertLineItemsAtEndExtDim

Inserts line items for a cell at the bottom of an HsvMDDataBufferLite object.

Syntax

```cpp
<HsvMDDataBufferLite>.InsertLineItemsAtEndExtDim pIUnkHfmPovCOM, vbSavedAsYTD, varadData, varabstrDescriptions
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>vbSavedAsYTD</td>
<td>Boolean (ByVal). Determines how the cell’s totals are saved. Pass TRUE if</td>
</tr>
<tr>
<td></td>
<td>the cell’s line items are saved as year-to-date, FALSE if saved as periodic.</td>
</tr>
<tr>
<td>varadData</td>
<td>Double array (ByVal). The data for the cell’s line items.</td>
</tr>
<tr>
<td>varabstrDescriptions</td>
<td>String array (ByVal). The descriptions of the cell’s line items.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

InsertRecordIntoPMBuffer

*For internal use.*

InsertRecordIntoPMErrorBuffer

*For internal use.*

RemoveAll

Clears all the cells from an HsvMDDataBufferLite object.

Syntax

```cpp
<HsvMDDataBufferLite>.RemoveAll
```
**SetCheckLineItemDetailsForCaseInsensitiveDuplicates**

Specifies whether the system checks for case-insensitive duplicate line item descriptions. For example, *acmeData* and *ACMEDATA* are 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 occurs.

**Syntax**

```<HsvMDDataBufferLite>.SetCheckLineItemDetailsForCaseInsensitiveDuplicates vbEnable```

**Argument Description**

- **vbEnable**  
  Boolean (ByVal). Specifies whether to check for case-insensitive duplicates. Pass TRUE to check, FALSE otherwise.

**SetData**

*Deprecated* - use “SetDataExtDim” on page 654.

**SetDataExtDim**

Inserts a cell’s data into an HsvMDDataBufferLite object.

**Syntax**

```<HsvMDDataBuffer>.SetDataExtDim pIUnkHfmPovCOM, dData, vbIsNoData, vbAddToExistingData, plNumElementsInDataUnit```

**Argument Description**

- **pIUnkHfmPovCOM**  
  HfmPovCOM. HfmPovCOM object representing the POV.

- **dData**  
  Double (ByVal). The cell’s data.

- **vbIsNoData**  
  Boolean (ByVal). Determines whether the cell is set to null. Pass TRUE to set to null, FALSE otherwise.

- **vbAddToExistingData**  
  Boolean (ByVal). If the HsvMDDataBuffer object already contains data for the cell, this argument determines whether the existing data overwritten. Pass TRUE to add the *dData* argument’s data to the existing data, FALSE to overwrite the existing data.

- **plNumElementsInDataUnit**  
  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 *SetData* adds the cell.

**Return Value**

None.
Example
See HfmPovCOM documentation on how to set the POV.

**SetDescription**

*Deprecated* - use “SetDescriptionExtDim” on page 655.

**SetDescriptionExtDim**

Inserts a cell’s description into an HsvMDDataBufferLite object.

**Syntax**

```c
<HsvMDDataBufferLite>.SetDescriptionExtDim pIUnkHfmPovCOM, bstrDescription
```

**Argument**

- `pIUnkHfmPovCOM` HfmPovCOM. HfmPovCOM object representing the POV.
- `bstrDescription` String (ByVal). The cell’s description.

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to set the POV.

**SetGrowByAmount**

Creates additional memory for an HsvMDDataBufferLite object.

**Syntax**

```c
<HsvMDDataBufferLite>.SetGrowByAmount lGrowByAmount
```

**Argument**

- `lGrowByAmount` Long (ByVal). The number of cells for which you want to create memory.

**SetLineItems**

*Deprecated* - use “SetLineItemsExtDim” on page 656.
SetLineItemsExtDim

Inserts a cell’s line items into an HsvMDDataBufferLite object.

Syntax

<HsvMDDataBuffer>.SetLineItemsExtDim pIUnkHfmPovCOM, vbSavedAsYTD, varadData, varabstrDescriptions

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>vbSavedAsYTD</td>
<td>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.</td>
</tr>
<tr>
<td>varadData</td>
<td>Double array (ByVal). The cell’s line item data.</td>
</tr>
<tr>
<td>varabstrDescriptions</td>
<td>String array (ByVal). The cell’s line item descriptions.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation on how to set the POV.

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 634.

Assign IHsvMDDataBufferLite object references with the Set and New keywords as shown below:

```vbnet
Dim cDataCache As IHsvMDDataBufferLite
Set cDataCache = New HSVMDARRAYSLib.HsvMDDataBufferLite
```
**SetMinCubesInCache**

Sets the number of subcubes that cached to RAM for the HsvMDDataBufferLite object.

**Note:** By default, the HsvMDDataBufferLite object caches one subcube to RAM. SetMinCubesInCache overrides this default.

**Syntax**

```csharp
<IHsvMDDataBufferLite>.SetMinCubesInCache nMinCubesInCache
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nMinCubesInCache</td>
</tr>
<tr>
<td>Integer (ByVal). The number of subcubes to cache.</td>
</tr>
</tbody>
</table>

**Example**

The following example sets the number of cached subcubes to 2.

```csharp
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.

1. Set an HsvTransactionData object reference using the Set and New keywords as shown below:

   ```csharp
   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 869 along with the applicable member ID. For example, the following lines specify the current entity, parent, and period as selection criteria:

   ```csharp
   cHsvTransData.SetQueryItem TRANSACTION_DIMENSIONS_CUR_ENTITY, lEnt
   cHsvTransData.SetQueryItem TRANSACTION_DIMENSIONS_CUR_PARENT, lPar
   cHsvTransData.SetQueryItem TRANSACTION_DIMENSIONS_CUR_PERIOD, lPer
   ```
To have the HsvTransactionData object populated with an array of the transaction data, pass the HsvTransactionData object to the HsvData object’s GetTransactionData method.

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.

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.

After finishing with the data, end the enumeration by calling EndDataEnum.

The HsvTransactionData object’s methods are summarized in “HsvTransactionData Object Overview” on page 111, 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**

```csharp
<HsvTransactionData>.BeginDataEnum plNumItems
```

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plNumItems</td>
<td>Long. Returns a count of the items in the HsvTransactionData object’s array.</td>
</tr>
</tbody>
</table>

**Example**

For an example that uses this method, see “GetTransactionDataExtDim” on page 659.

**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

```csharp
```

Example

For an example that uses this method, see “GetICTransactionDataExtDim” on page 668.

EndQueryEnum

*For internal use.*

GetFixedDimensionMembers

*For internal use.*

GetQueryItem

*Deprecated* - use “GetQueryItemExtDim” on page 659.

GetQueryItemExtDim

*For internal use.*

GetTransactionData

*Deprecated* - use “GetTransactionDataExtDim” on page 659.

GetTransactionDataExtDim

Returns source and destination data for a transaction, using 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. Supersedes GetTransactionData

**Syntax**

```csharp
```

**Argument**

**Description**

- **lItem**
  - 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. See “BeginDataEnum” on page 658.
### 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

```plaintext
<HsvTransactionData>.Initialize lCurrentScenario, lCurrentYear
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lCurrentScenario</code></td>
<td>Long (ByVal). The member ID of the transactions' current scenario.</td>
</tr>
<tr>
<td><code>lCurrentYear</code></td>
<td>Long (ByVal). The member ID of the transactions' current year.</td>
</tr>
</tbody>
</table>

#### Example

For an example that uses this method, see “GetTransactionDataExtDim” on page 659.
SetQueryItem

Deprecated - use “SetQueryItemExtDim” on page 661.

SetQueryItemExtDim

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. Supersedes SetQueryItem

Syntax

```<HsvTransactionData>.SetQueryItemExtDim nTransactionDimension, lValue```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>nTransactionDimension</td>
<td>Long. TRANSACTION_DIMENSIONS_EX enum value. [Note: For the TRANSACTION_DIMENSIONS_DES_CUSTOM_EX or TRANSACTION_DIMENSIONS_SRC_CUSTOM_EX values, you must OR in the custom dimension number.]</td>
</tr>
<tr>
<td>lValue</td>
<td>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 nTransactionDimension argument.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See the Transaction Dimension Constants section for more information on the TRANSACTION_DIMENSIONS_EX enums.

SetTransactionData

Deprecated - use “SetTransactionDataExtDim” on page 661.

SetTransactionDataExtDim

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 added to the HsvMDIndexList object.
The HsvMDIndexList object’s methods are summarized in Table 34 on page 112, and are described in detail in the following topics.

**GetItem**

*Deprecated* - use “GetItemExtDim” on page 662.

**GetItemExtDim**

Returns the member IDs of the dimension members for an item in an HsvMDIndexList object.

**Syntax**

```c
<HsvMDIndexList>.GetItemExtDim lItem, ppIUnkHfmPovCOM
```

**Argument**  
**Description**

- `lItem`  
  Long (ByVal). The index for which the member IDs returned.  
  Use `GetNumItems` to get the number of indexes in the HsvMDIndexList object.

- `ppIUnkHfmPovCOM`  
  HfmPovCOM. HfmPovCOM object representing the POV at the specified index.

**Return Value**

None.

**Example**

See HfmPovCOM documentation on how to retrieve members.

**GetNumItems**

Returns the number of indexes in an HsvMDIndexList object. Use this to loop with `GetItem`.

**Syntax**

```c
<HsvMDIndexList>.GetNumItems plNumItems
```

**Argument**  
**Description**

- `plNumItems`  
  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 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.

1. To use the HsvICTransactionsData object:
   1. Set an HsvICTransactionsData object reference using the Set and New keywords as shown below:

   ```vba
   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 specific 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 113, and are described in detail in the following topics.

AddAccountCustomCombination

*Deprecated* - use “AddAccountCustomCombinationExtDim” on page 663.

AddAccountCustomCombinationExtDim

For internal use.
AddICTransactionData

Deprecated - use “AddICTransactionDataExtDim” on page 664.

AddICTransactionDataExtDim

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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plNumItems</td>
<td>Long. Returns a count of the transactions contained by the HsvICTransactionsData instance.</td>
</tr>
</tbody>
</table>
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
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lItem</em></td>
<td>Long (ByVal). The index of the transaction in the HsvICTransactionsData instance’s array of intercompany transactions.</td>
</tr>
<tr>
<td></td>
<td>You can get the upper bounds of the index by subtracting one from the count of transactions returned by <em>BeginDataEnum</em>.</td>
</tr>
<tr>
<td><em>pvbReadAccess</em></td>
<td>Boolean. Returns the user’s read access rights. TRUE indicates that the user has read access.</td>
</tr>
<tr>
<td><em>plWriteAccess</em></td>
<td>Long. Returns the user’s write access rights. The following list describes the valid values:</td>
</tr>
<tr>
<td></td>
<td>• 0 = Read-only access.</td>
</tr>
<tr>
<td></td>
<td>• 1 = User is allowed to specify reason codes for the Intercompany Partner transactions even if the user lacks write access to those transactions.</td>
</tr>
<tr>
<td></td>
<td>• 2 = Write access.</td>
</tr>
</tbody>
</table>

**GetAccountCustomCombination**

*Deprecated* - use “GetAccountCustomCombinationExtDim” on page 666.

**GetAccountCustomCombinationExtDim**

*For internal use.*

**GetAccountCustomCombinationTotal**

*For internal use.*

**GetEntityPartnerOption**

*For internal use.*

**GetErrorStatus**

Returns the HRESULT associated with a specific intercompany transaction.

**Syntax**

```vbnet
<HsvICTransactionsData>.GetErrorStatus lItemIndex, plHRStatus
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><em>lItemIndex</em></td>
<td>Long (ByVal). The index of the transaction in the HsvICTransactionsData instance’s array of intercompany transactions.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> You can get the upper bounds of the index by subtracting one from the count of transactions returned by <em>BeginDataEnum</em>.</td>
</tr>
</tbody>
</table>
Argument  Description

plHRStatus  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 24, “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 specific transaction, use GetICTransactionCell.

Syntax

<HsvICTransactionsData>.GetFixedDimensionMembers plScenario, plYear, plPeriod

Argument  Description

plScenario  Long. Returns the member ID of the Scenario dimension member.

plYear  Long. Returns the member ID of the Year dimension member.

plPeriod  Long. Returns the member ID of the Period dimension member.

GetICTransactionCell

Deprecated - use “GetICTransactionCellExtDim” on page 667.

GetICTransactionCellExtDim

Returns the member IDs of an intercompany transaction’s Entity, Intercompany Partner, Account, and Custom dimension members. Supersedes GetICTransactionCell.
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 specific transaction, use GetICTransactionData.

Syntax

```c
<HsvICTransactionsData>.GetICTransactionCellExtDim lItem, ppIUnkHfmPovCOM
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lItem</strong></td>
<td>Long. The index of the transaction in the HsvICTransactionsData instance’s array of intercompany transactions.</td>
</tr>
<tr>
<td><strong>ppIUnkHfmPovCOM</strong></td>
<td>HfmPovCOM .HfmPovCOM object representing the POV at the specified index.</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

See HfmPovCOM documentation for how to retrieve members from the POV.

**GetICTransactionData**

*Deprecated* - use “GetICTransationDataExtDim” on page 668.

**GetICTransationDataExtDim**

Returns an intercompany transaction’s details, including the transaction’s amounts, currency, dimension members, and so on. Supersedes GetICTransactionData.

Syntax

```c
<HsvICTransactionsData>.GetICTransactionDataExtDim lItem, plSequenceId, ppIUnkHfmPovCOM, plTransactionCurrency, plPostStatus, plMatchStatus, plReasonCode, plType, pdModifiedDate, pdTransactionDate, pdAmount, pdLocalAmount, pdRate, pbstrUser, pbstrId, pbstrSubId, pbstrReferenceId, pbstrMatchCode, pbstrComment1, pbstrComment2
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>lItem</strong></td>
<td>Long (ByVal). The index of the transaction in the HsvICTransactionsData instance’s array of intercompany transactions.</td>
</tr>
<tr>
<td><strong>plSequenceId</strong></td>
<td>Long. Returns the transaction’s sequence ID.</td>
</tr>
<tr>
<td><strong>ppIUnkHfmPovCOM</strong></td>
<td>HfmPovCOM .HfmPovCOM object representing the POV at the specified index.</td>
</tr>
</tbody>
</table>

Tip: You can get the upper bounds of the index by subtracting one from the count of transactions returned by BeginDataEnum.
### Argument Description

**plTransactionCurrency** Long. Returns the currency ID of the transaction’s currency.

**Tip:** You can get the currency’s label by passing this ID to the HsvCurrencies method `GetCurrencyLabel`.

**plPostStatus** Long. Returns the transaction’s posting status. Valid values are represented by the HFMCConstants type library constants listed in “Posting Status Constants” on page 900.

**plMatchStatus** Long. Returns the transaction’s matching status. Valid values are represented by the HFMCConstants type library constants listed in “Matching Status Constants” on page 900.

**plReasonCode** Long. Returns the ID of the transaction’s reason code.

**Tip:** To get the reason code, pass this ID to the HsvICM method `GetICReasonCodeLabel`.

**plType** Long. For internal use.

**pdModifiedDate** Double. Returns a timestamp that identifies when the transaction was last modified. The timestamp is formatted as a Double.

**pdTransactionDate** Double. Returns the transaction date formatted as a Double.

**pdAmount** Double. Returns the transaction amount.

**pdLocalAmount** Double. Returns the entity currency amount.

**pdRate** Double. Returns the conversion rate, which represents the transaction amount divided by the entity currency amount.

**pbstrUser** String. Returns the username of the user who last updated the transaction.

**pbstrId** String. Returns the Transaction ID.

**pbstrSubId** String. Returns the transaction’s Sub ID.

**pbstrReferenceId** String. Returns the transaction’s Reference ID.

**pbstrMatchCode** String. Returns the transaction’s match code.

**pbstrComment1** String. Returns the transaction’s first comment.

**pbstrComment2** String. Returns the transaction’s second comment.

### Return Value

None.

### Example

See HfmPovCOM documentation for how to retrieve members from the POV.

### 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 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 GetICTransactionDate 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.GetICTransactionDate i, lSeqId, lEnt, lIcp, lAcct, _
        lCust1, lCust2, lCust3, lCust4, 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

Deprecated - use “IsICTransactionValidExtDim” on page 672.

IsICTransactionValidExtDim

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 causes an error.

**Syntax**

```
<HsvICTransactionsData>.SetAccessRights lItem, vbReadAccess, lWriteAccess
```

**Argument**  **Description**

*lItem*  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.

*vbReadAccess*  Boolean (ByVal). A flag that specifies the user’s read access rights. Pass TRUE to grant read access rights, FALSE to remove them.
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>WriteAccess</code></td>
<td>Long (ByVal). A flag that specifies the user's write access rights. Pass one of the following values:</td>
</tr>
<tr>
<td></td>
<td>• 0 = Read-only access.</td>
</tr>
<tr>
<td></td>
<td>• 1 = Allow the user to specify reason codes for the Intercompany Partner transactions even if the user lacks write access to those transactions.</td>
</tr>
<tr>
<td></td>
<td>• 2 = Write access.</td>
</tr>
</tbody>
</table>

**SetEntityPartnerOption**

*For internal use.*

**SetErrorStatus**

*For internal use.*

**SetFilterOptions**

*For internal use.*

**SetICTransactionData**

*Deprecated - use “SetICTransactionDataExtDim” on page 673.*

**SetICTransactionDataExtDim**

*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.
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 53.

**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 GetOneCellProm StoredItem.

---

**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 53. The HsvCurrencyCube’s object’s methods are summarized in “HsvCurrencyCube Object Overview” on page 115, 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 675.

Tip: Remove the lock by calling EndEnumerationOfStoredData.

Syntax

\[<\text{HsvCurrencyCube}>.\text{BeginEnumerationOfStoredData \ lValue, \ plHandle, \ plTotalNumStoredItems} \]

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\lValue</td>
<td>Long (ByVal). The member ID of the subcube items' Value dimension member.</td>
</tr>
<tr>
<td>\plHandle</td>
<td>Long. Returns a handle. You must pass this handle to other HsvCurrencyCube methods that have a “handle” argument.</td>
</tr>
<tr>
<td>\plTotalNumStoredItems</td>
<td>Long. Returns a count of the subcube items for the \lValue argument's Value dimension member. Use this count to determine the valid range of index numbers for GetOneCellFromStoredItem or GetPOVFromStoredItem.</td>
</tr>
</tbody>
</table>

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.
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

Deprecated - use GetCellExtDim.

GetCellExtDim

Returns the data in and transaction status of a cell in a subcube. Supersedes GetCell.

Syntax

<'HsvCurrencyCube>.GetCellExtDim pIUnkHfmPovCOM, pdData, pbyTransType

Argument Description

pIUnkHfmPovCOM HfmPovCOM. HfmPOVCom object representing the POV.

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 871.

Return Value

None.

Example

See HfmPovCOM documentation for how to set the POV.

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
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>plScenario</td>
<td>Long. Returns the member ID of the subcube’s Scenario dimension member.</td>
</tr>
<tr>
<td>plYear</td>
<td>Long. Returns the member ID of the subcube’s Year dimension member.</td>
</tr>
<tr>
<td>plEntity</td>
<td>Long. Returns the member ID of the subcube’s Entity dimension member.</td>
</tr>
<tr>
<td>plInputValueID</td>
<td>Long. Returns the member ID of the subcube’s input Value dimension member. For example, if the HsvCurrencyCube object is initialized with a Value dimension member named USD Total, this argument returns the member ID of the Value dimension member named USD.</td>
</tr>
</tbody>
</table>

**GetOneCellFromStoredItem**

*Deprecated* - use GetOneCellFromStoredItemExtDim.

**GetOneCellFrom StoredItemExtDim**

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 675. Supersedes GetOneCellFromStoredItem.

**Syntax**

```
<HsvCurrencyCube>.GetOneCellFromStoredItemExtDim lHandle, lItem, lPeriod, lView, ppIUnkHfmPovCOM, pvbDimensionMembersAreValid, pdData, pbyTransType
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lHandle</td>
<td>Long (ByVal). The item’s handle. You must pass the handle that was returned by BeginEnumerationOfStoredData.</td>
</tr>
<tr>
<td>lItem</td>
<td>Long (ByVal). The index of the item within the count of items returned by BeginEnumerationOfStoredData.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the cell’s Period dimension member.</td>
</tr>
<tr>
<td>lView</td>
<td>Long (ByVal). The member ID of the cell’s View dimension member.</td>
</tr>
<tr>
<td>plValue</td>
<td>Long. Returns the member ID of the cell’s Value dimension member.</td>
</tr>
<tr>
<td>ppIUnkHfmPovCOM</td>
<td>HfmPovCOM. Returns an HfmPovCOM object representing the cell’s POV.</td>
</tr>
<tr>
<td>pvbDimensionMembersAreValid</td>
<td>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 was changed and data loaded in merge mode. For performance reasons, Financial Management does not delete obsolete cells from the database.</td>
</tr>
<tr>
<td>pdData</td>
<td></td>
</tr>
<tr>
<td>pbyTransType</td>
<td></td>
</tr>
</tbody>
</table>

**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.
**Argument** | **Description**
---|---
`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 871.

**Return Value**
None.

**Example**
See HfmPovCOM documentation for how to retrieve members for the POV.

**GetPOVFromStoredItem**

*Deprecated* - use `GetPOVFromStoredItemExtDim`.

**GetPOVFromStoredItemExtDim**

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 675. Supersedes GetPOVFromStoredItem.

**Syntax**

```
<HsvCurrencyCube>.GetPOVFromStoredItemExtDim lHandle, lItem, ppIUnkHfmPovCOM, pvbDimensionMembersAreValid
```

**Argument** | **Description**
---|---
`lHandle` | The item’s handle. You must pass the handle that was returned by `BeginEnumerationOfStoredData`.
`lItem` | 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.

`ppIUnkHfmPovCOM` | HfmPovCOM. Returns an HfmPovCOM object representing the cell’s POV.
`pvbDimensionMembersAreValid` | 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 was changed and data 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.

**Return Value**
None.
Example

See HfmPovCOM documentation for how to retrieve members for the POV.

HsvNodeCube Object Methods

The HsvNodeCube object provides access to node subcubes, which are described in “About Subcubes” on page 53. The HsvNodeCube’s object’s methods are summarized in “HsvNodeCube Object Overview” on page 116, 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 675.

Tip: Remove the lock by calling EndEnumerationOfStoredData.

Syntax

```
<HsvNodeCube>.BeginEnumerationOfStoredData lValue, plHandle, plTotalNumStoredItems
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lValue</td>
<td>Long (ByVal). The member ID of the subcube items’ Value dimension member.</td>
</tr>
<tr>
<td>plHandle</td>
<td>Long (x86 platform) or Double (x64 platform). Returns a handle. You must pass this handle to other HsvNodeCube methods that have a “handle” argument.</td>
</tr>
<tr>
<td>plTotalNumStoredItems</td>
<td>Long. Returns a count of the subcube items for the lValue argument’s Value dimension member. Use this count to determine the valid range of index numbers when using GetOneCellFromStoredItem or GetPOVFromStoredItem.</td>
</tr>
</tbody>
</table>

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

```csharp
<HsvNodeCube>.EndEnumerationOfStoredData lHandle
```

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lHandle</code></td>
<td>Long (ByVal) (x86 platform) or Double (ByVal) (x64 platform). The item’s handle. Pass the handle that was returned by <code>BeginEnumerationOfStoredData</code>.</td>
</tr>
</tbody>
</table>

Example

`EndEnumerationOfStoredData` is used in the Example for `GetOneCellFromStoredItem`.

**GetCell**

 Deprecated - use “GetCellExtDim” on page 681.

**GetCellExtDim**

Returns the data in and transaction status of a cell in a subcube. Supersedes GetCell.

Syntax

```csharp
<HsvNodeCube>.GetCellExtDim pIUnkHfmPovCOM, pdData, pbyTransType
```

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIUnkHfmPovCOM</code></td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td><code>pdData</code></td>
<td>Double. Returns the cell’s data.</td>
</tr>
<tr>
<td><code>pbyTransType</code></td>
<td>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 871.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

See HfmPovCom documentation for how to set the POV.

**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
plScenario  Long. Returns the member ID of the subcube’s Scenario dimension member.
plYear  Long. Returns the member ID of the subcube’s Year dimension member.
plEntity  Long. Returns the member ID of the subcube’s Entity dimension member.
plParent  Long. Returns the member ID of the parent of the plEntity argument’s entity.

GetOneCellFromStoredItem

Deprecated - use GetOneCellFromStoredItemExtDim.

GetOneCellFromStoredItemExtDim

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 675. Supersedes GetOneCellFromStoredItem.

Syntax
<HsvNodeCube>.GetOneCellFromStoredItemExtDim lHandle, lItem, lPeriod, lView, ppIUnkHfmPovCOM, pvbDimensionMembersAreValid, pdData, pbyTransType

Argument  Description
lHandle  Long (ByVal) (x86 platform) or Double (x64 platform). The item’s handle. You must pass the handle that was returned by BeginEnumerationOfStoredData.

lItem  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.

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.

ppIUnkHfmPovCOM  HfmPovCOM. Returns the POV of the specified item.

pvbDimensionMembersAreValid  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 was changed and data 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.
### Argument | Description
--- | ---
`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 871.

### Return Value
None.

### Example
See HfmPovCOM documentation for how to set the POV.

**GetPOVFromStoredItem**

*Deprecated* - use `GetPOVFromStoredItemExtDim`.

**GetPOVFromStoredItemExtDim**

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 675. Supersedes GetPOVFromStoredItem.

### Syntax

```c
<HsvNodeCube>.GetPOVFromStoredItem lHandle, lItem, ppIUnkHfmPovCOM, pvbDimensionMembersAreValid
```

### Argument | Description
--- | ---
`lHandle` | Long (ByVal) (x86 platform) or Double (x64 platform). The item’s handle. You must pass the handle that was returned by BeginEnumerationOfStoredData.

`lItem` | 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.

`ppIUnkHfmPovCOM` | HfmPovCOM. Returns an HfmPovCOM object representing the POV at the specified index.

`pvbDimensionMembersAreValid` | 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 was changed and data 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.

### Return Value
None.
Example

See HfmPovCOM documentation for how to retrieve members for the POV.
The HsvDQI type library provides methods for intelligent data retrieval.
An HsvDQI object should be obtained from the HsvSession API `get_HsvDQI()` only.

**HsvDQI Object Methods**

The HsvDQI object’s methods are summarized in the “HsvDQI Type Library Overview” on page 117, and are described in detail in these topics.

**AddEdgeDimension**

Adds a dimension to the edge definition.

**Syntax**

```c
<HsvDQI>.AddEdgeDimension(lEdgeID, bstrDimName)
```

**Argument** | **Description**
--- | ---
`lEdgeID` | Long. Edge ID.
`bstrDimName` | String. Dimension name.

**Return Value**

None.

**Example**

```c
HsvDQIobj.AddEdgeDimension( lEdgeID, ’’Entity’’);
```
**AddEdgeDimensionID**

Add a dimension to the edge definition (input as IDs).

**Syntax**

```csharp
<HsvDQI>.AddEdgeDimensionID(lEdgeID, lDimID)
```

**Argument Description**

- **lEdgeId** Long. Edge ID.
- **lDimId** Long. Dimension ID to be added to edge.

**Return Value**

None.

**Example**

```csharp
HsvDQIobj.AddEdgeDimensionID(lEdgeID, lDimId)
```

**AddEdgeSource**

Adds a source to the edge. The order and number of dimensions in edge and source should match.

**Syntax**

```csharp
<HsvDQI>.AddEdgeSource(lEdgeID, lSourceID)
```

**Argument Description**

- **lEdgeId** Long. Edge ID.
- **lSourceID** Long. Edge source to be added to the edge.

**Return Value**

None.

**Example**

```csharp
HsvDQIobj.AddEdgeSource(lEdgeID, lSourceID);
```

**AddEdgeSources**

Add a set of sources to the edge. The number and order of dimensions in the edge and sources should match.

**Syntax**

```csharp
<HsvDQI>.AddEdgeSources(lEdgeID, varalDQISrcIds)
```

---

686 HsvDQI Type Library
**AddEdgeSources**

Adds a dimension to a source object.

**Syntax**

```
<HsvDQI>.AddEdgeSources(lEdgeID, arrSourceIDs);
```

**Argument Description**

- **lEdgeID**: Long. Edge ID.
- **arrSourceIDs**: Long. A list of source IDs for sources to be added to the edge.

**Return Value**

None.

**Example**

```
HsvDQIobj.AddEdgeSources(lEdgeID, arrSourceIDs);
```

---

**AddSourceDimension**

Adds a dimension to a source object.

**Syntax**

```
<HsvDQI>.AddSourceDimension(lSourceID, bstrDimName);
```

**Argument Description**

- **lSourceID**: Long. Source ID.
- **bstrDimName**: String. Contains the dimension name to be added.

**Return Value**

None.

**Example**

```
HsvDQIobj.AddSourceDimension(lSourceID, "Entity");
```

---

**AddSourceDimensionID**

Adds a dimension to a source object definition (input as IDs).

**Syntax**

```
<HsvDQI>.AddSourceDimensionID(lSourceID, lDimId);
```

**Argument Description**

- **lSourceID**: Long. Source ID.
- **lDimId**: Long. Dimension ID to be added.

**Return Value**

None.
Example
HsvDQIobj.AddSourceDimensionID(lSourceID, DIMID_ENTITY)

**AddToSourceJoinTypesForDimensions**
Adds Join types to be used between dimensions in the source to resolve source definition.

**Syntax**

\[ <HsvDQI>.AddToSourceJoinTypesForDimensions(lSourceID, varalJoinTypes) \]

**Argument**
- **lSourceID**
  - Long. Source ID.

- **varalJoinTypes**
  - Long. Array of join types to be used across dimensions in this source. Size should be \( n-1 \) where \( n \) is number of dimensions in the source.
  - 6 - CROSS_JOIN
  - 7 - POSITIONAL_JOIN

**Return Value**
None.

Example
Dim arrJointypes As ArrayList
arrJoinTypes.Add(CROSS_JOIN)
arrJoinTypes.Add(POSITIONAL_JOIN)
HsvDQIobj.AddToSourceJoinTypesForDimensions(lSourceID, arrJoinTypes.ToArray())

**CollapseAll**
Collapses all nodes in specified edge completely for a specified grid.

**Syntax**

\[ <HsvDQI>.CollapseAll(lGridID, lEdgeType) \]

**Argument**
- **lGridID**
  - Long. Grid ID.

- **lEdgeType**
  - Long. Edge to be collapsed completely (ROW, COLUMN)

**Return Value**
None.
Example
HsvDQIobj.CollapseAll(lGridId,ROW);

**CollapseGrid**

Collapse a particular member in a specified grid.

**Syntax**

```csharp
<HsvDQI>.CollapseGrid(lGridID, lEdgeType, lIndex, bstrDimName, bstrMemName)
```

**Argument**

- **lGridID**: Long. Grid ID.
- **lEdgeType**: Long. Edge to be collapsed (ROW, COLUMN)
- **lIndex**: Long. Index to be collapsed.
- **bstrDimName**: String. Dimension name.
- **bstrMemName**: String. Member to be collapsed in specified dimension.

**Return Value**

None.

**Example**

HsvDQIobj.CollapseGrid(lGridId,ROW,0, 'Entity','UnitedStates');

**CollapseGridDimension**

Collapses a dimension in an index in the specified edge of the grid. Only for dynamic grids and where Grid Option Dimension Expansion is enabled.

**Syntax**

```csharp
<HsvDQI>.CollapseGridDimension(lGridID, lEdgeType, lIndex, bstrDimName)
```

**Argument**

- **lGridID**: Long. Grid ID.
- **lEdgeType**: Long. Edge to be collapsed (ROW, COLUMN).
- **lIndex**: Long. Index to be collapsed.
- **bstrDimName**: String. Dimension name.

**Return Value**

None.
**CollapseGridDimensionID**

Collapses a dimension in given index in the specified edge of the grid. Only for dynamic grids and where Grid Option Dimension Expansion is enabled (input as IDs).

**Syntax**

```<HsvDQI>.CollapseGridDimensionID(lGridID, lEdgeType, lIndex, lDimID)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long, Grid ID.</td>
</tr>
<tr>
<td>lEdgeType</td>
<td>Long, Edge to be collapsed (ROW, COLUMN).</td>
</tr>
<tr>
<td>lIndex</td>
<td>Long, Index to be collapsed.</td>
</tr>
<tr>
<td>bstrDimName</td>
<td>String, Dimension name.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

HsvDQIobj.CollapseGridDimensionID(lGridId, ROW, 0, ‘‘Entity’’)

**CollapseGridID**

Collapses a particular member in the specified edge of the grid (input as IDs).

**Syntax**

```<HsvDQI>.CollapseGridID(lGridID, lEdgeType, lIndex, lDimID, lMemID)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long, Grid ID.</td>
</tr>
<tr>
<td>lEdgeType</td>
<td>Long, Edge to be collapsed (ROW, COLUMN).</td>
</tr>
<tr>
<td>lIndex</td>
<td>Long, Index to be collapsed.</td>
</tr>
<tr>
<td>lDimID</td>
<td>Long, Dimension ID.</td>
</tr>
<tr>
<td>lMemID</td>
<td>Long, Member ID to be collapsed in specified dimension.</td>
</tr>
</tbody>
</table>

**Return Value**

None.
Example
HsvDQIobj.CollapseGridID(lGridId,ROW,0,DIMID_ENTITY,lMemID)

CreateEdge
Creates a DQI edge with specified type for a specified grid.

Syntax
<HsvDQI>.CreateEdge(lEdgeType, lGridId, plEdgeID)

Argument Description
lEdgeType Long. Specifies the edge type to be created.
  1 - PAGE
  2 - ROW
  3 - COLUMN
  4 - POV
lGridID Long. Grid ID.
plEdgeID Long. Returns the unique edge ID created.

Return Value
None.

Example
Dim edgeID As Long
HsvDQIobj.CreateEdge(PAGE,gridId,edgeID)

CreateGrid
Creates a DQI Grid instance in an application.

Syntax
<HsvDQI>.CreateGrid(lGridType, lClientType, plGridId)

Argument Description
lGridType Long. Input value specifies the grid type.
  0 – FORWARD_ONLY – used for reports
  1 – DYNAMIC – used for interactive grids
lClientType Long. Input value specifies the client type.
  0 – Default value for custom clients
  >1 - Internal use
**CreateGrid**

Long. Returns an integer containing a unique grid ID.

**Return Value**

None.

**Example**

Dim gridId As Long  
HsvDQIobj.CreateGrid(FORWARD_ONLY, 0, gridId)

---

**CreatePageControl**

Creates a DQI Page control object for a specified grid.

**Syntax**

```
<HsvDQI>.CreatePageControl(lGridId, plPageControlID)
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridId</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>plPageControlID</td>
<td>Long. Returns the unique page control ID created.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

Dim pageControlID as Long  
HsvDQIobj.CreatePageControl(gridId, pageControlID)

---

**CreateSource**

Creates a DQI source with a specified type.

**Syntax**

```
<HsvDQI>.CreateSource(lSrcType, lGridId, plSourceID)
```

---
Argument Description

lSrcType Long. Specifies the source type to be created.
   ● 1 - MEMBER_SOURCE – Used for ROW, COLUMN
   ● 2 - SLICE_SOURCE – Used for POV
   ● 3, 4 - Future
   ● 5 - PAGE_SOURCE – Used for Page

lGridID Long. Grid ID.

plSourceID Long. Returns the unique source ID created.

Return Value

Returns None.

Example

Dim sourceID1 As Long
Dim sourceID2 As Long
HsvDQIObj.CreateSource(MEMBER_SOURCE, gridId, sourceID1)
HsvDQIObj.CreateSource(SLICE_SOURCE_SOURCE, gridId, sourceID2)

DefineGrid

Validates and defines the grid definition, after which the grid can be used for grid operations.

Syntax

<HsvDQI>.DefineGrid(lGridID)

Argument Description

lGridID Long. Grid ID.

Return Value

Returns None.

Example

HsvDQIObj.DefineGrid(lGridId);

DeleteGrid

Deletes the grid and its related objects from DQI internally.

Syntax

<HsvDQI>.DeleteGrid(lGridID)
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
```
HsvDQIobj.DeleteGrid(lGridId);
```

**ExpandAll**
Expands all nodes in specified edge completely for a specific grid.

Syntax
```
<HsvDQI>.ExpandAll(lGridID,lEdgeType)
```

Argument Description
```
lGridID   Long. Grid ID.
lEdgeType Long. Edge to be expanded completely (ROW, COLUMN).
```

Return Value
None.

Example
```
HsvDQIobj.ExpandAll(lGridId,ROW);
```

**ExpandGrid**
Expands a particular member in a specified grid.

Syntax
```
<HsvDQI>.ExpandGrid(lGridID,lEdgeType,lIndex,bstrDimName,bstrMemName)
```

Argument Description
```
lGridID   Long. Grid ID.
lEdgeType Long. Edge to be expanded (ROW, COLUMN)
lIndex    Long. Index to be expanded.
bstrDimName String. Dimension name.
bstrMemName String. Member to be expanded in specified dimension.
```
Return Value
None.

Example
HsvDQIobj.ExpandGrid(lGridId,ROW,0, ‘’Entity’’,’’UnitedStates’’);

**ExpandGridDimension**
Expands a dimension in an index in the specified edge of the grid. Only for dynamic grids and where the Grid Option Dimension Expansion is enabled.

Syntax
<HsvDQI>.ExpandGridDimension(lGridID,lEdgeType,lIndex,bstrDimName)

**Argument**  **Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>lEdgeType</td>
<td>Long. Edge to be expanded (ROW, COLUMN).</td>
</tr>
<tr>
<td>lIndex</td>
<td>Long. Index to be expanded.</td>
</tr>
<tr>
<td>bstrDimName</td>
<td>String. Dimension name.</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
HsvDQIobj.ExpandGridDimension(lGridId,ROW,0, ‘’Entity’’)

**ExpandGridDimensionID**
Expand a dimension in an index in the specified edge of the grid. Only for dynamic grids and where Grid Option Dimension Expansion is enabled.

Syntax
<HsvDQI>.ExpandGridDimensionID(lGridID,lEdgeType,lIndex,lDimID))

**Argument**  **Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>lEdgeType</td>
<td>Long. Edge to be expanded (ROW, COLUMN).</td>
</tr>
<tr>
<td>lIndex</td>
<td>Long. Index to be expanded.</td>
</tr>
<tr>
<td>bstrDimID</td>
<td>String. Dimension name.</td>
</tr>
</tbody>
</table>
**Return Value**

None.

**Example**

HsvDQIobj.ExpandGridDimensionID(lGridId,ROW,0, DIMID_ENTITY)

---

**ExpandGridID**

Expands a particular member in the mentioned edge of the grid (input as IDs).

**Syntax**

```csharp
<HsvDQI>.ExpandGridID(lGridID,lEdgeType,lIndex,lDimID,lMemId)
```

**Argument Description**

- `lGridID`: Long. Grid ID.
- `lEdgeType`: Long. Edge to be expanded (ROW, COLUMN).
- `lIndex`: Long. Index to be expanded.
- `lDimID`: Long. Dimension ID.
- `lMemId`: Long. Member ID to be expanded in specified dimension.

**Return Value**

None.

**Example**

HsvDQIobj.ExpandGridID(lGridId,ROW,0,DIMID_ENTITY,lMemID)

---

**FreezeEdgeDefinition**

Freezes the edge definition, after which dimensions cannot be added to edge.

**Syntax**

```csharp
<HsvDQI>.FreezeEdgeDefinition(lEdgeID)
```

**Argument Description**

- `lEdgeId`: Long. Edge ID to be frozen for definition.

**Return Value**

None.

**Example**

HsvDQIobj.FreezeEdgeDefinition(lEdgeID);
**GetAllDataForGridPageDimensionIndex**

Gets the data for the specified PageDimension based on the current state of the grid.

**Syntax**

```csharp
<HsvDQI>.GetAllDataForGridPageDimensionIndex(lGridID,lPageIndex,pvaralXIndexes,pvaralYIndexes,pvaralDataValues,pvaralStatusValues)
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>lPageIndex</td>
<td>Long. Page dimension index to be selected.</td>
</tr>
<tr>
<td>pvaralXIndexes</td>
<td>LongArray. Returns set of xIndexes.</td>
</tr>
<tr>
<td>pvaralYIndexes</td>
<td>LongArray. Returns set of yindexes in order with x.</td>
</tr>
<tr>
<td>pvaralDataValues</td>
<td>LongArray. Returns set of data values in order with x,y.</td>
</tr>
<tr>
<td>pvaralStatusValues</td>
<td>LongArray. Returns set of status values in order with x,y.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```csharp
HsvDQIobj.GetAllDataForGridPageDimensionsIndex(lGridId,lPage,arrOutXindexes,arrOutYIndexes,arrOutData,arrOutStatus)
```

---

**GetAllVisibleDimensionDetails**

Returns all the visible headers (beyond this physical page) for the requested edge of grid, with IDs and labels in output for members. The 2D array output parameters can be accessed by using (index,dimensionorder) as 2D access combination for values in output.

**Syntax**

```csharp
<HsvDQI>.GetAllVisibleDimensionDetails(lGridID,lEdgeType,pvaralIndexes,plNoOfDimensions,pvaralDimIds,pvaralDimStrings,pvaral2DMemIds,pvarabstr2DMemStrings,pvaral2DExpansionFlags,pvaral2DDimExpansionFlags,pvarabstr2DDescriptions,pvaral2DDimGenerations,vbFillParentInformation)
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>lEdgeType</td>
<td>Long. Edge type.</td>
</tr>
<tr>
<td>pvaralIndexes</td>
<td>LongArray. Returns set of indexes on this edge for display.</td>
</tr>
<tr>
<td>plNoOfDimensions</td>
<td>Long. Returns number of dimensions on this edge.</td>
</tr>
</tbody>
</table>
### GetVisibleDimensionDetails

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pvaralDimIDs</td>
<td>LongArray. Returns the 1D Dim IDs in order on this edge.</td>
</tr>
<tr>
<td>pvarabstrDimStrings</td>
<td>StringArray. Returns the 1D Dim Names in order on this edge.</td>
</tr>
<tr>
<td>pvar2DMemIDs</td>
<td>LongArray. Returns the 2D Member IDs for all dimensions on this edge.</td>
</tr>
<tr>
<td>pvarabstr2DMemStrings</td>
<td>StringArray. Returns the 2D Member Names for all dimensions on this edge.</td>
</tr>
<tr>
<td>pvaral2DExpansionFlags</td>
<td>LongArray. Returns 2D array of expansion state of members for this edge.</td>
</tr>
<tr>
<td>pvaral2DDimExpansionFlags</td>
<td>LongArray. Returns 2D array of expansion state of dimension for this edge.</td>
</tr>
<tr>
<td>pvarabstr2dDescriptions</td>
<td>StringArray. Returns the 2D Member Descriptions for all dimensions on this edge.</td>
</tr>
<tr>
<td>pvaral2DDimGenerations</td>
<td>LongArray. Returns the 2D array of Generation offset number used for by client for display.</td>
</tr>
<tr>
<td>vbFillParentInformation</td>
<td>Boolean. Input flag, which denotes parent dimension information is to be filled in 2D array outputs for all output values, if Entity is part of this edge.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```c
HsvDQIobj.GetAllVisibleDimensionDetails(lGridID,lEdgeType,arrOutIndexes,lOutNofdims,arrOutDimIds,arrOutDimNames,arrOut2DMemIDs,arrOut2DMemStrings,arrOut2DExpansionFlags,arrOut2DDimExpansionFlags,arrOur2DDescriptions,arrOur2DDimGenerations,false)
```

### GetGridCompleteDimensionDetails

Returns all the visible and hidden headers (beyond physical page) for the requested edge of grid, with IDs and labels in output for members. The 2D array output parameters can be accessed by using (index,dimensionorder) as 2D access combination for values in output.

**Syntax**

```c
<HsvDQI>.GetGridCompleteDimensionDetails(lGridID, lEdgeType, pvaralIndexes, plNoOfDimensions, pvaralDimIds, pvaralDimStrings, pvar2DMemIds, pvarabstr2DMemStrings, pvaral2DExpansionFlags, pvaral2DDimExpansionFlags, pvarabstr2dDescriptions, pvaral2DDimGenerations, vbFillParentInformation)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>lEdgeType</td>
<td>Long. Edge type.</td>
</tr>
<tr>
<td>pvaralIndexes</td>
<td>LongArray. Returns Set of indexes on this edge for display.</td>
</tr>
<tr>
<td>plNoOfDimensions</td>
<td>Long. Returns Number of dimensions on this edge.</td>
</tr>
</tbody>
</table>
Argument | Description
---|---
pvaralDimIDs | LongArray. Returns the 1D Dim IDs in order on this edge.
pvarabstrDimStrings | StringArray. Returns the 1D Dim Names in order on this edge.
pvaral2DMemIDs | LongArray. Returns the 2D Member IDs for all dimensions on this edge.
pvarabstr2DMemStrings | StringArray. Returns the 2D Member Names for all dimensions on this edge.
pvaral2DExpansionFlags | LongArray. Returns 2D array of expansion state of members for this edge.
pvaral2DDimExpansionFlags | LongArray. Returns 2D array of expansion state of dimension for this edge.
pvarabstr2dDescriptions | StringArray. Returns the 2D Member Descriptions for all dimensions on this edge.
Pvaral2DDimGenerations | LongArray. Returns the 2D array of Generation offset number used for by client for display.
vbFillParentInformation | Boolean. Input flag, which denotes parent dimension information is to be filled in 2D array outputs for all output values, if Entity is part of this edge.

Return Value
None.

Example

HsvDQIobj.GetGridCompleteDimensionDetails
(lGridID, lEdgeType, arrOutIndexes, lOutNofdims, arrOutDimIDs, arrOutDimNames, arrOut2DMemIDs, arrOut2DMemStrings, arrOut2DExpansionFlags, arrOut2DDimExpansionFlags, arrour2DDescriptions, arrOur2DDimGenerations, false)

GetGridDetails

Returns the number of rows, columns, and page dimension entries in the grid. The grid must already be defined already for this method to return values.

Syntax

<HsvDQI>.GetGridDetails(lGridID, plNoOfRows, plNoOfCols, plNoOfPageDimEntries)

Argument | Description
---|---
lGridID | Long. Grid ID.
plNoOfRows | Long. Returns the number of rows.
plNoOfCols | Long. Returns the number of columns.
plNoOfPageDimEntries | Long. Returns the number of pagedim entries.

Return Value
None.
Example

HsvDQIobj.GetGridDetails(lGridId, lNoRows, lNoCols, lNoPageDimEntries)

**GetGridDimensionDetails**

Gets the headers for the requested edge of grid, subject to physical page limitations. The 2d array output parameters can be accessed by using (index,dimensionorder) as 2D access combination for values in output.

**Syntax**

```hsvdqi.GetGridDimensionDetails(lGridID, lEdgeType, pvaralIndexes, plNoOfdimensions, pvarabstr2DMemStrings, pvaral2DExpansionFlags, pvaral2DDimExpansionFlags, pvarabstr2DDescr iptions, pvaral2DDimGenerations)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>lEdgeType</td>
<td>Long. Edge type.</td>
</tr>
<tr>
<td>pvaralIndexes</td>
<td>LongArray. Returns set of indexes on the edge for display.</td>
</tr>
<tr>
<td>plNoOfDimensions</td>
<td>Long. Returns number of dimensions on the edge.</td>
</tr>
<tr>
<td>pvarabstr2DMemStrings</td>
<td>StringArray. Returns the 2D member names for all dimensions on this edge.</td>
</tr>
<tr>
<td>pvaral2DExpansionFlags</td>
<td>LongArray. Returns 2D array of expansion state of members for this edge.</td>
</tr>
<tr>
<td>pvaral2DDimExpansionFlags</td>
<td>LongArray. Returns 2D array of expansion state of dimension for this edge.</td>
</tr>
<tr>
<td>pvarabstr2dDescriptions</td>
<td>StringArray. Returns the 2D member descriptions for all dimensions on this edge.</td>
</tr>
<tr>
<td>pvaral2DDimGenerations</td>
<td>LongArray Returns the 2D array of generation offset number used by client for display.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

HsvDQIobj. GetGridDimensionDetails(lGridID, lEdgeType, arrOutIndexes, lOutNofdims, arrOut2DMemStrings, arrOut2DExpansionFlags, arrOut2DDimExpansionFlags, arrour2DDescriptions, arrOur2DDimGenerations)

**GetGridDimensionDetailsIDsAndLabels**

Gets the Headers for the requested edge of grid, with IDs and labels in output for members, subject to current physical page limitations. The 2D array output parameters can be accessed by using (index,dimensionorder) as 2D access combination for values in output.
Syntax

`<HsvDQI>.GetGridDimensionDetailsIDsAndLabels(lGridID, lEdgeType, pvaralIndexes, plNoOfDimensions, pvaral2DMemIDs, pvarabstr2DMemStrings, pvaral2DExpansionFlags, pvaral2DDimExpansionFlags, pvarabstr2DDescriptions, pvaral2DDimGenerations, vbFillParentInformation)`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>lEdgeType</td>
<td>Long. Edge type.</td>
</tr>
<tr>
<td>pvaralIndexes</td>
<td>Long array. Returns set of indexes on this edge for display.</td>
</tr>
<tr>
<td>plNoOfDimensions</td>
<td>Long. Returns number of dimensions on this edge.</td>
</tr>
<tr>
<td>pvaral2DMemIDs</td>
<td>Long array. Returns the 2D member IDs for all dimensions on this edge.</td>
</tr>
<tr>
<td>pvarabstr2DMemStrings</td>
<td>String array. Returns the 2D member names for all dimensions on this edge.</td>
</tr>
<tr>
<td>pvaral2DExpansionFlags</td>
<td>Long array. Returns 2D array of expansion state of members for this edge.</td>
</tr>
<tr>
<td>pvaral2DDimExpansionFlags</td>
<td>Long array. Returns 2D array of expansion state of dimension for this edge.</td>
</tr>
<tr>
<td>pvarabstr2dDescriptions</td>
<td>String array. Returns the 2D member descriptions for all dimensions on this edge.</td>
</tr>
<tr>
<td>pvaral2DDimGenerations</td>
<td>Long array. Returns the 2D array of generation offset number used for by client for display.</td>
</tr>
<tr>
<td>vbFillParentInformation</td>
<td>Boolean. Input flag, which denotes parent dimension information is to be filled in 2D array outputs for all output values, if Entity is part of this edge.</td>
</tr>
</tbody>
</table>

Return Value
None.

Example

`HsvDQIobj.GetGridDimensionDetailsIDsAndLabels(lGridID, lEdgeType, arrOutIndexes, lOutNofDimensions, arrOut2DMemIDs, arrOut2DMemStrings, arrOut2DExpansionFlags, arrOut2DDimExpansionFlags, arrOut2DDescriptions, arrOut2DDimGenerations, false)`

**GetIndexesRangeForSourcesOnEdge**

 Gets start and end indexes to figure out the range of indexes for every logical source in the specified edge.

Syntax

`<HsvDQI>.GetIndexesRangeForSourcesOnEdge( lGridId, lEdgeType, pvaralSourceIds, pvaralStartIndexes, pvaralEndIndexes)`

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridId</td>
<td>Long. Grid ID.</td>
</tr>
</tbody>
</table>

HsvDQI Object Methods 701
**Argument** | **Description**
---|---
EdgeType | Long. Edge type for which details are requested.
pvaralSourceIds | LongArray. Returns set of source IDs on the edge.
pvaralStartIndexes | LongArray. Returns set of start indexes for sources in order.
pvaralEndIndexes | LongArray. Returns set of end indexes for sources in order.

**Return Value**
None.

**Example**
HsvDQIobj.GetIndexesRangeForSourcesOnEdge(lGridID, ROW, arrOutSrcIds, arrOutStartIndexes, arrOutEndIndexes)

### GetPageControlDetails

Gets the page directions to be enabled for the page control in UI for a specified grid.

**Syntax**

```<HsvDQI> .GetPageControlDetails(lGridId, pvaralPageControls)```

**Argument** | **Description**
---|---
PageID | Long. Grid ID.
pvaralPageControls | Long. Returns array of page directions to be enabled in UI.

**Return Value**
None.

**Example**
HsvDQIobj.GetPageControlDetails(lGridId, arrOutPageDirections)

The output would have values such as SCROLL_UP, SCROLL_LEFT values as array, indicating those directions can be enabled in page control of client for valid page movement.

### GetPOVForIndexes

Gets the POV for the specified combination of x,y indexes in the grid.

**Syntax**

```<HsvDQI> .GetPOVForIndexes(lGridId, lPageIndex, lRowIndex, lColIndex, pvarabstrDims, pvarabstrMems)```
**Argument** | **Description**
---|---
`lGridId` | Long. Grid ID.
`lPageIndex` | Long. PageIndex selected.
`lRowIndex` | Long. X value of cell of which POV is requested
`lColIndex` | Long. Y value of cell of which POV is requested
`pvarabstrDims` | StringArray. Returns array of dimension names.
`pvarabstrMems` | StringArray. Returns array of member names in order of dimensions as in param pvarabstrDims.

**Return Value**
None.

**Example**

HsvDQIobj.GetPOVForIndexes(lGridID, pageIndex, x, y, arrOutDims, arrOutMems);

---

**GetPOVIDsForIndexes**

Gets the POV for the specified cell x,y in the grid (output as IDs).

**Syntax**

```<HsvDQI>.GetPOVIDsForIndexes(lGridId, lPageIndex, lRowIndex, lColIndex, pvaralDimIDs, pvaralMemIDs)```

**Argument** | **Description**
---|---
`lGridId` | Long. Grid ID.
`lPageIndex` | Long. PageIndex selected.
`lRowIndex` | Long. X value of cell of which POV is requested.
`lColIndex` | Long. Y value of cell of which POV is requested.
`pvaralDimIDs` | LongArray. Returns array of dimension IDs.
`pvaralMemIDs` | LongArray. Returns array of member IDs in order of dimensions as in parameter pvaralDimIDs.

**Return Value**
None.

**Example**

HsvDQIobj.GetPOVIDsForIndexes(GridID,pageIndex,x,y,arrOutDimIds,arrOutMemIds)
### GetSourceIDForIndexes

Gets the row and column logical source IDs for the specified set of x and y coordinates.

**Syntax**

```csharp
GetSourceIDForIndexes(lGridID, varalXIndexes, varalYIndexes, pvaralLogicalRowSourceIDs, pvaralLogicalColSourceIDs)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>varalXIndexes</td>
<td>LongArray. Array of x indexes.</td>
</tr>
<tr>
<td>varalYIndexes</td>
<td>LongArray. Array of y indexes.</td>
</tr>
<tr>
<td>pvaralLogicalRowSourceIDs</td>
<td>LongArray. Returns the set of logical row source IDs.</td>
</tr>
<tr>
<td>pvaralLogicalColSourceIDs</td>
<td>LongArray. Returns the set of logical column source IDs.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

```csharp
HsvDQIobj.GetSourceIDForIndexes(lGridID, arrXIndexes, arrYIndexes, arrOutRowSrcs, arrOutColSrcs)
```

### JoinSources

Join two sources with mentioned Join type to form the output source definition. CROSS/POSITION Joins are allowed when input sources have different dimensions. UNION_ALL and REMOVE are used when input sources have the same dimensions and order.

**Syntax**

```csharp
<HsvDQI>.JoinSources(lInputSource1ID, lInputSource2ID, lJoinType, lOutputSourceID)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lInputSource1ID</td>
<td>Long. SourceID of source 1.</td>
</tr>
<tr>
<td>lInputSource2ID</td>
<td>Long. SourceID of source 2.</td>
</tr>
<tr>
<td>lJoinType</td>
<td>Long. Join type to be performed.</td>
</tr>
<tr>
<td></td>
<td>● 2 - UNION_ALL</td>
</tr>
<tr>
<td></td>
<td>● 4 - REMOVE</td>
</tr>
<tr>
<td></td>
<td>● 6 - CROSS_JOIN</td>
</tr>
<tr>
<td></td>
<td>● 7 - POSITIONAL_JOIN</td>
</tr>
<tr>
<td></td>
<td>● 1, 3 - Not used</td>
</tr>
<tr>
<td>lOutputSource1ID</td>
<td>Long. Source ID of the resultant source.</td>
</tr>
</tbody>
</table>
Return Value
None.

Example

lJointype = UNION_ALL  'used if input sources are of same dim order
HsvDQIobj.JoinSources(lSource1, lSource2 ,lJointype, lOutputSource1) ;
lJointype = CROSS_JOIN
HsvDQIobj.JoinSources(lSource3, lSource4 ,lJointype, lOutputSource2) ;

JoinSourcesInSequence
Joins two or more sources in order based on join types specified to form the output source definition.

Syntax

<HsvDQI>.JoinSourcesInSequence(varalInputSourceIds, varalJoinTypes, lOutputSourceID)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>varalInputSourceIds</td>
<td>LongArray. SourceIDs in sequence to be joined.</td>
</tr>
<tr>
<td>varalJoinTypes</td>
<td>Jointype in sequence to be used (n-1 for n source ID in parameter 1).</td>
</tr>
<tr>
<td>lOutputSourceID</td>
<td>SourceID to OutputSource.</td>
</tr>
</tbody>
</table>

Return Value
None.

Example

Dim arrSources as ArrayList
Dim arrJoinTypes as ArrayList
arrSource.Add(lSource1)
arrSource.Add(lSource2)
arrSource.Add(lSource3)
arrJoinTypes.Add(CROSS_JOIN)
arrJoinTypes.Add(POSITIONAL_JOIN) ' size n-1
HsvDQIobj.JoinSourcesInSequence( arrSources.ToArray(),
arrJoinTypes.ToArray(), lOutputSource) ;
**SelectGridPageDimensionIndex**

Selects the active page dimension index to determine the active page dimension.

**Syntax**

```<HsvDQI>.SelectGridPageDimensionIndex(lGridID,lPageIndex)```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>lPageIndex</td>
<td>Long. Active page dimension index to be selected.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```HsvDQIobj.SelectGridPageDimensionIndex(lGridId,lPageIndex);```  

**SelectMemberSourceAncestors**

Selects ancestors of a specified member to a source of type member source for the active dimension or last added dimension.

**Syntax**

```<HsvDQI>.SelectMemberSourceAncestors(lSourceID,bstrTopMemberName, vbInclusive)```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>bstrTopMemberName</td>
<td>String. Member whose ancestors are to be added.</td>
</tr>
<tr>
<td>vbInclusive</td>
<td>Boolean. Not used.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```HsvDQIobj.SelectMemberSourceAncestors(lSourceID, ''Account1'', false)```  

**SelectMemberSourceAncestorsID**

Selects ancestors of a specified member to member source for the active dimension (input as IDs).
Syntax

```<HsvDQI>.SelectMemberSourceAncestorsID(lSourceID, lMemID, vbInclusive)```

**Argument** | **Description**
---|---
`lSourceID` | Long. Source ID.
`lMemID` | Long. Member ID whose ancestors are to be added.
`vbInclusive` | Boolean. Not used.

**Return Value**

None.

**Example**

```HsvDQIobj.SelectMemberSourceAncestorsID(lSourceID, lmemID, false)```

### SelectMemberSourceBase

Selects base members of a specified member to a source of type membersource for the active dimension or last added dimension.

**Syntax**

```<HsvDQI>.SelectMemberSourceBase(lSourceID, bstrTopMemberName, vbInclusive)```

**Argument** | **Description**
---|---
`lSourceID` | Long. Source ID.
`bstrTopMemberName` | String. Member whose base members are to be added.
`vbInclusive` | Boolean. Not used.

**Return Value**

None.

**Example**

```HsvDQIobj.SelectMemberSourceBase(lSourceID, ''Account1'', true)```

### SelectMemberSourceBaseID

Selects base members of a specified member to member source for the active dimension (input as IDs).

**Syntax**

```<HsvDQI>.SelectMemberSourceBase(lSourceID, lMemID, vbInclusive)```
**SelectMemberSourceBaseID**

Selects a member and its base members to a member source for the active dimension.

**Syntax**

```vbnet
<HsvDQI>.SelectMemberSourceBaseID(lSourceID, lMemID, false)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lSourceID</code></td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td><code>lMemID</code></td>
<td>Long. Member ID whose base members are to be added.</td>
</tr>
<tr>
<td><code>vbInclusive</code></td>
<td>Boolean. Not used.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```
HsvDQIobj.SelectMemberSourceBaseID(lSourceID, lMemID, false)
```

---

**SelectMemberSourceChildren**

Selects a member and its children to a member source for the active dimension.

**Syntax**

```vbnet
<HsvDQI>.SelectMemberSourceChildren(lSourceID, bstrTopMemberName, vbInclusive)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lSourceID</code></td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td><code>bstrTopMemberName</code></td>
<td>String. Member name.</td>
</tr>
<tr>
<td><code>vbInclusive</code></td>
<td>Boolean. Not used.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```
HsvDQIobj.SelectMemberSourceChildren(lSourceID, ''UnitedStates'', false)
```

---

**SelectMemberSourceChildrenID**

Selects a member and its children to a member source for the active dimension (input as IDs).

**Syntax**

```vbnet
<HsvDQI>.SelectMemberSourceChildrenID(lSourceID, lMemID, vbInclusive)
```

**Arguments**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lSourceID</code></td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td><code>lMemID</code></td>
<td>Long. Top member ID whose children are to be selected.</td>
</tr>
<tr>
<td><code>vbInclusive</code></td>
<td>Boolean. Not used.</td>
</tr>
</tbody>
</table>
SelectMemberSourceDescendants

Selects descendants of specified member to a source of type member source for the active
dimension or last added dimension.

Syntax

```
<HsvDQI>.SelectMemberSourceDescendants( lSourceID, bstrTopMemberName, vbInclusive)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID</td>
</tr>
<tr>
<td>bstrTopMemberName</td>
<td>String. Member whose descendants are to be added.</td>
</tr>
<tr>
<td>vbInclusive</td>
<td>Boolean. Not used.</td>
</tr>
</tbody>
</table>

Example

```
HsvDQIobj.SelectMemberSourceDescendants(lSourceID, 'Account1', false)
```

SelectMemberSourceHierarchy

Selects a hierarchy of members to a member source for the active dimension.

Syntax

```
<HsvDQI>.SelectMemberSourceHierarchy(lSourceID, bstrTopMemberName, vbInclusive)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID</td>
</tr>
<tr>
<td>bstrTopMemberName</td>
<td>String. Member name.</td>
</tr>
<tr>
<td>vbInclusive</td>
<td>Boolean. Not used.</td>
</tr>
</tbody>
</table>
Return Value
None.

Example
HsvDQI.SelectMemberSourceHierarchy(lSourceID, ‘‘UnitedStates’’, false)

SelectMemberSourceHierarchyExpanded
Selects a hierarchy of member to member source for the active dimension and sets them to expanded state.

Syntax
<HsvDQI>.SelectMemberSourceHierarchyExpanded(lSourceID,bstrTopMemberName, vbInclusive)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>bstrTopMemberName</td>
<td>String. Top member of the hierarchy to be selected.</td>
</tr>
<tr>
<td>vbInclusive</td>
<td>Boolean. Not used.</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
HsvDQIobj.SelectMemberSourceHierarchyExpanded(lSourceID,strTopMember,false)

SelectMemberSourceHierarchyExpandedID
Selects a hierarchy of members to a membersource for active dimension with expanded state (input as IDs).

Syntax
<HsvDQI>.SelectMemberSourceHierarchyExpandedID(lSourceID,lMemID, vbInclusive)

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>lMemID</td>
<td>Long. Top member of the hierarchy to be selected.</td>
</tr>
<tr>
<td>vbInclusive</td>
<td>Long. Not used.</td>
</tr>
</tbody>
</table>

Return Value
None.
Example
HsvDQIobj.SelectMemberSourceHierarchyExpandedID(lSourceID, lMembId, false)

**SelectMemberSourceHierarchyID**

Selects a hierarchy of members to a member source for the active dimension (input as IDs).

**Syntax**

\[<\text{HsvDQI}>.\text{SelectMemberSourceHierarchyID}(\text{lSourceID, lMemID, vbInclusive})\]

**Argument**  **Description**

\(\text{lSourceID}\)  
Long. Source ID.

\(\text{lMemID}\)  
Long. Top member ID of the hierarchy to be selected.

\(\text{vbInclusive}\)  
Long. Not used.

**Return Value**

None.

**Example**

HsvDQIObj.SelectMemberSourceHierarchyID(lSourceID, lMemberId, false);

**SelectMemberSourceMemberID**

Selects a single member to member source for the active dimension (input as ID).

**Syntax**

\[<\text{HsvDQI}>.\text{SelectMemberSourceMemberID}(\text{lSourceID, lMemID, lParentID})\]

**Argument**  **Description**

\(\text{lSourceID}\)  
Long. Source ID.

\(\text{lMemID}\)  
Long. Member ID to be added to the source.

\(\text{lParentID}\)  
Long. Parent ID to be added to source (only for Entity dimension).

**Return Value**

None.

**Example**

HsvDQIobj.SelectMemberSourceMemberID(lSourceID, lMemberId, lParentId)
SelectMemberSourceMemberIDs

Selects a set of members to a source of type member source for the active dimension (input as IDs).

Syntax

\(<\text{HsvDQI}\.\text{SelectMemberSourceMemberIDs}(1\text{SourceID}, \text{varalMemIDs}, \text{varalParentIDs})\>

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1SourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>varalMemIDs</td>
<td>LongArray. Array of member IDs to be added to the source.</td>
</tr>
<tr>
<td>varalParentIDs</td>
<td>LongArray. Array of parent IDs to be added to source (Only for Entity dimension).</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

HsvDQIobj.SelectMemberSourceMemberIDs(1SourceID, arrMemberIds, arrParentIds)

SelectMemberSourceMembers

Selects a set of members to a source of type member source for the active dimension.

Syntax

\(<\text{HsvDQI}\.\text{SelectMemberSourceMembers}(1\text{SourceID}, \text{varabstrMemberNames}, \text{varabstrParentNames})\>

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1SourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>varabstrMemberNames</td>
<td>StringArray. Array of Member names to be added to the source.</td>
</tr>
<tr>
<td>varabstrParentNames</td>
<td>StringArray. Array of Parent names to be added to source (used for active entity dimension).</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

Dim arrMemberNames as Array List
Dim arrParentNames as Array List
arrMemberNames.Add('Connecticut')
arrParentNames.Add('United States')
HsvDQIobj.SelectMemberSourceMembers(1SourceID,
arrMemberNames.ToArray(), arrParentNames.ToArray())

**SelectMemberSourceParentsID**

Selects parents of a specified member to member source for the active dimension (input as IDs).

**Syntax**

```csharp
<HsvDQI>.SelectMemberSourceParentsID(lSourceID, lMemID, vbInclusive)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>lMemID</td>
<td>Long. Member ID whose parents are to be added.</td>
</tr>
<tr>
<td>vbInclusive</td>
<td>Boolean. Not used.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

HsvDQIobj.SelectMemberSourceParentsID(lSourceID, lMemID, false)

**SelectMemberSourceParents**

Selects parents of specified member to a source of type member source for the active dimension or last added dimension.

**Syntax**

```csharp
<HsvDQI>.SelectMemberSourceParents(lSourceID, bstrTopMemberName, vbInclusive)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>bstrTopMemberName</td>
<td>String. Member whose parents are to be added.</td>
</tr>
<tr>
<td>vbInclusive</td>
<td>Boolean. Not used.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

HsvDQIobj.SelectMemberSourceParents(lSourceID, ‘Account1’, true)
**SelectMemberSourceUserDefinedMemberList**

Selects the members in a given list into a member source for the active dimension.

**Syntax**

```csharp
<HsvDQI>.SelectMemberSourceUserDefinedMemberList(lSourceID, bstrMemberListName)
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>bstrMemberListName</td>
<td>String. Memberlist name.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

```csharp
HsvDQIobj.SelectMemberSourceUserDefinedMemberList(lSourceID, ''.List1'')
```

**SelectMemberSourceUserDefinedMemberList2**

Selects the members in a given list into a member source for the active dimension.

**Syntax**

```csharp
<HsvDQI>.SelectMemberSourceUserDefinedMemberList(lSourceID, lScenario, lYear, lPeriod, lEntity, bstrMemberListName)
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long (ByVal). Source ID.</td>
</tr>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member.</td>
</tr>
<tr>
<td>bstrMemberListName</td>
<td>String (ByVal). Memberlist name.</td>
</tr>
</tbody>
</table>

**SelectPageSourceMemberIDsForDim**

Adds the dimension and set of members (input as IDs) to a page source definition.

**Syntax**

```csharp
<HsvDQI>.SelectPageSourceMemberIDsForDim(lSourceID, lDimID, varalMemIDs, varalParentIDs)
```

---

714  HsvDQI Type Library
### SelectPageSourceMemberIDsForDim

Adds the dimension and set of members to a page source.

**Syntax**

```csharp
<HsvDQI>.SelectPageSourceMemberIDsForDim(lSourceID, DIMID_SCENARIO, lMemID, lParentID;
```

**Argument**

- **lSourceID**: Long. Source ID.
- **lDimID**: Long. Dimension name to be added.
- **varalMemIDs**: LongArray. Set of member IDs to be selected for the dimension.
- **varalParentIDs**: LongArray. Set of parent IDs to be selected (only for Entity dimension).

**Return Value**

None.

**Example**

```csharp
HsvDQIobj.SelectPageSourceMemberIDsForDim(lSourceID, DIMID_SCENARIO, lMemID, lParentID;
```

### SelectPageSourceMembersForDim

Adds the dimension and set of members to a page source.

**Syntax**

```csharp
<HsvDQI>.SelectPageSourceMembersForDim(lSourceID, bstrDimName, varabstrMemberNames, varabstrParentNames)
```

**Argument**

- **lSourceID**: Long. Source ID.
- **bstrDimName**: String. Dimension name to be added.
- **varabstrMemberName**: String array. Set of Member names to be selected against the dimension.
- **varabstrMemberNames**: String Array. Set of Parent names to be selected against the dimension (only for Entity dimension).

**Return Value**

None.

**Example**

```csharp
Dim arrMemberNames as ArrayList
Dim arrParentNames as ArrayList
arrMemberNames.Add('Connecticut')
arrParentNames.Add('United States')
HsvDQIobj.SelectPageSourceMembersForDim(lSourceID, bstrDimName, arrMemberNames.ToArray(), arrParentNames.ToArray())
```
SelectSliceSourceMemberIDForDim

Adds the dimension and member to a slice source definition (inputs as IDs).

Syntax

```csharp
<HsvDQI>.SelectSliceSourceMemberIDForDim(lSourceID, lDimId, lMemID, lParentID)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>lDimId</td>
<td>Long. Dimension ID to be added.</td>
</tr>
<tr>
<td>lMemID</td>
<td>Long. Member ID to be added for dimension.</td>
</tr>
<tr>
<td>lParentID</td>
<td>Long. Parent ID of the selected member (only for Entity dimension).</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

```csharp
HsvDQIobj.SelectSliceSourceMemberIDForDim(lSourceID, lDimId, lMemberID, lParentID);
```

SelectSliceSourceMemberForDim

Adds the dimension and member to a slice source definition.

Syntax

```csharp
<HsvDQI>.SelectSliceSourceMemberForDim(lSourceID, bstrDimName, bstrMemberName, bstrParentName)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>bstrDimName</td>
<td>String. Dimension name to be added.</td>
</tr>
<tr>
<td>bstrMemberName</td>
<td>String. Member name.</td>
</tr>
<tr>
<td>bstrParentName</td>
<td>String. Parent name of the member selected (only for Entity dimension).</td>
</tr>
</tbody>
</table>

Return Value

None.

Example

```csharp
HsvDQIobj.SelectSliceSourceMemberForDim(lSourceID, ''Entity'',''Connecticut'',''United States'')
HsvDQIobj.SelectSliceSourceMemberForDim(lSourceID, ''Accounts'',''Account1'',''')
```
**SetColsForPageControl**
Sets the number of columns for the page control object.

Syntax

```<HsvDQI>.SetColsForPageControl (lPageID, lNoOfCols)```

**Argument Description**

- **lPageID**  Long. Page control ID.
- **lNoOfCols**  Long. Number of columns.

**Return Value**

None.

**Example**

```
HsvDQIobj.SetColsForPageControl(lPageID, lNoOfCols)
```

**SetEdgeRepeatedMetaDataSuppression**
Sets the suppression of repeating metadata headers in the specified edge to on or off.

Syntax

```<HsvDQI>.SetEdgeRepeatedMetaDataSuppression(lEdgeID, vbFlag)```

**Argument Description**

- **lEdgeID**  Long. Edge ID.
- **vbFlag**  Long. True or false.

**Return Value**

None.

**Example**

```
HsvDQIobj.SetEdgeRepeatedMetaDataSuppression(lEdgeID, true)
```

**SetGridEdge**
Sets the edge to a grid definition.

Syntax

```<HsvDQI>.SetGridEdge(lGridID, lEdgeID)```
Argument | Description
--- | ---
`lGridID` | Long, Grid ID.

`lEdgeId` | Long, Edge ID to be added to grid.

Return Value
None.

Example
```
HsvDQIobj.SetGridEdge(lGridId, lEdgeID);
```

**SetGridNavigationType**
Sets the navigation type for a grid. Applies only to FORWARD_ONLY grids. By default, a grid is of COLUMN_FIRST navigational mode.

Syntax
```
<HsvDQI>.SetGridNavigationType(lGridId, lNavigationType)
```

Argument | Description
--- | ---
`lGridId` | Long, Grid ID.

`lNavigationType` | Long, Type of navigation to be followed by the grid:

  1 - COLUMN_FIRST
  2 - ROW_FIRST

Return Value
None.

Example
```
HsvDQIobj.SetGridNavigationType(lGridID, COLUMN_FIRST)
```

**SetGridOptionDescriptionEnabled**
Enables or disables the description in headers being returned for a specified grid.

Syntax
```
<HsvDQI>.SetGridOptionDescriptionEnabled(lGridID, vbFlag)
```

Argument | Description
--- | ---
`lGridID` | Long, Grid ID.

`vbFlag` | Boolean, True/False set to on or off.
Return Value
None.

Example
HsvDQIobj.SetGridOptionDescriptionEnabled(lGridId, true)

**SetGridOptionDimensionExpansion**
Set the dimension expansion feature on or off for a specified grid. This must be set before a grid is defined.

Syntax

```
<HsvDQI>.SetGridOptionDimensionExpansion(lGridId,vbFlag)
```

**Argument  Description**

- **lGridId** Long. Grid ID.
- **vbFlag** Boolean. True or False.

Return Value
None.

Example
HsvDQIobj.SetGridOptionDimensionExpansion(lGridID, true)

**SetGridOptionDisplayParentForEntityHierarchy**
Used to disable the Parent.Child format in output for Entity members in hierarchy selection in output headers. By default, this option is enabled.

Syntax

```
<HsvDQI>.SetGridOptionDisplayParentForEntityHierarchy(lGridId,vbFlag)
```

**Argument  Description**

- **lGridId** Long. Grid ID.
- **vbFlag** Boolean. True/False set to on/off.

Return Value
None.

Example
HsvDQIobj.SetGridOptionDisplayParentForEntityHierarchy(lGridId, false)
SetGridPageControl
Sets a PageControl to a grid definition.

Syntax
<HsvDQI>.SetGridPageControl(lGridID, lPageControlID)

Argument Description
lGridID Long Grid ID.
lPageControlID Long PageControl ID to be added to grid.

Return Value
None.

Example
HsvDQIobj.SetPageControl(lGridId, lPageControlID)

SetGridSuppressionType
Sets a particular type of suppression on a row or column edge to on or off. Not applicable for a Slice/Page edge.

Syntax
<HsvDQI>.SetGridSuppressionType(lGridID, lEdgeType, tSuppressionType, flag)

Argument Description
lGridID Long Grid ID.
lEdgeType Long Type of edge (ROW, COLUMN).
tSuppression Long Suppression type:
  ● 1 = SUPPRESS_NODATA
  ● 2 = SUPPRESS_DERIVED
  ● 4 = SUPPRESS_ZERO
  ● 8 = SUPPRESS_INVALID
  ● 16 = SUPPRESS_NOACCESS
  ● 32 = SUPPRESS_ERROR
Flag Boolean True/False

Return Value
None.
Example
HsvDQIobj.SetGridSuppressionType(lGridId,ROW,SUPPRESS_INVALID,true);
HsvDQIobj.SetGridSuppressionType(lGridId,COLUMN,SUPPRESS_NODATA,false);

**SetPageMovement**
Sets the direction of page movement. The next data fetch will fetch data based on this movement.

Syntax

```<HsvDQI>.SetPageMovement(lGridID, lPageAction)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lGridID</td>
<td>Long. Grid ID.</td>
</tr>
<tr>
<td>lPageAction</td>
<td>Long. Direction of movement. (Dynamic grids)</td>
</tr>
<tr>
<td></td>
<td>- 1 - SCROLL_UP</td>
</tr>
<tr>
<td></td>
<td>- 2 - SCROLL_DOWN</td>
</tr>
<tr>
<td></td>
<td>- 3 - SCROLL_RIGHT</td>
</tr>
<tr>
<td></td>
<td>- 4 - SCROLL_LEFT (FORWARD_ONLY grids)</td>
</tr>
<tr>
<td></td>
<td>- 5 - SCROLL_NEXT</td>
</tr>
</tbody>
</table>

Return Value
None.

Example
HsvDQIobj.SetPageMovement(lGridID,SCROLL_UP)

**SetRowsForPageControl**
Sets the number of rows for the page control object.

Syntax

```<HsvDQI>.SetRowsForPageControl(lPageID, lNoOfRows)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lPageID</td>
<td>Long. Page control ID.</td>
</tr>
<tr>
<td>lNoOfRows</td>
<td>Long. Number of rows.</td>
</tr>
</tbody>
</table>

Return Value
None.
Example
HsvDQIobj.SetRowsForPageControl(lPageID, lNoofRows)

**SetSourceNotSuppressible**
Disables or overrides the suppression options set at grid or edge level for rows or columns resulting from this edge source. Applicable only for member source objects (row or column).

**Syntax**

```
<HsvDQI>.SetSourceNotSuppressible(lSourceID, vbFlag)
```

**Argument Description**

- **lSourceID**: Long. Source ID.
- **vbFlag**: Boolean. Enable or disable the suppression option.

**Return Value**
None.

**Example**
HsvDQIobj.SetSourceNotSuppressible(lSrcId, true);

**SetSourceOptionFillDefaultParentForEntity**
Used to set default parent for members during entity member selection in this particular source. Disables the need to always specify parent members.

**Syntax**

```
<HsvDQI>.SetSourceOptionFillDefaultParentForEntity( lSourceID, vbFlag)
```

**Argument Description**

- **lSourceId**: Long. Source ID.
- **vbFlag**: Boolean. True/False set to on/off.

**Return Value**
None.

**Example**
HsvDQIobj.SetSourceOptionFillDefaultParentForEntity(lSourceID, false)

**SetSourceSubType**
Sets the subtype for a source.
Syntax

\(<\text{HsvDQI}>\).SetSourceSubType(lSourceID, lSubType)

**Argument Description**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lSourceID</td>
<td>Long. Source ID.</td>
</tr>
<tr>
<td>lSubType</td>
<td>Long. Subtype to be set.</td>
</tr>
<tr>
<td></td>
<td>● 1 - SIMPLE_SOURCE - source that has member definitions.</td>
</tr>
<tr>
<td></td>
<td>● 2 - AGGREGATE_SOURCE - source that is formed from one or more sources by joins.</td>
</tr>
</tbody>
</table>

**Return Value**

None.

**Example**

HsvDQIobj.SetSourceSubtype(lSourceID, SIMPLE_SOURCE)
This chapter describes the HsvStarSchemaACM type library, which exposes Financial Management’s Extract Data Extended Analytics features.

**Note:** For Extended Analytics information, see the *Oracle Hyperion Financial Management Administrator’s Guide*.


The HsvStarSchemaACM object also contains the following enumerations:

- SS_PUSH_OPTIONS. see CreateStarSchema ssPushType argument.
- EA_EXTRACT_TYPE_FLAGS. See CreateStarSchema eaExtractType argument.
- EA_TASK_STATUS_FLAGS. See GetAsynchronousTaskStatus 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 39 on page 120, 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**

*Deprecated* - use “CreateStarSchemaExtDim” on page 726.

**CreateStarSchemaExtDim**

Creates or updates a star schema by exporting data for all cells that intersect the specified dimension members. `CreateStarSchemaExtDim` launches an asynchronous thread. Supersedes `CreateStarSchema` and `CreateStarSchemaAndReturnTaskID`.

`CreateStarSchemaExtDim` 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 `CreateStarSchemaExtDim`. For example, you can specify both member IDs and labels within an array for a dimension.

To export data with `CreateStarSchemaExtDim`, 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**

```vbnet
'HsvStarSchemaACM'.CreateStarSchemaExtDim bstrDSN, bstrPrefix, ssPushType, eaExtractType, vbIncludeDynamicAccts, vbIncludeCalculatedData, vbIncludeDerivedData, eaLineItemOption, vbIncludeCellDesc, vbIncludePhaseGroup, bstrDelimiter, pIUnkHfmSliceCOM, plTaskId
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrDSN</code></td>
<td>String (ByVal). The Extended Analytics Data Source Name that points to the database for the star schema. <strong>Tip:</strong> For the Extended Analytics Data Source Names registered on an application server, use <code>EnumRegisteredDSNs</code>.</td>
</tr>
<tr>
<td><code>bstrTablePrefix</code></td>
<td>String (ByVal). The Relational Table Prefix, which is the prefix that identifies the star schema's tables. <strong>If</strong> a star schema identified by this prefix does not exist, <code>CreateStarSchemaExtDim</code> creates a new set of tables. <strong>If</strong> the star schema exists, the tables are updated; the <code>ssPushType</code> argument determines whether all data is deleted from the <code>&lt;PREFIX&gt;_FACT</code> table.</td>
</tr>
</tbody>
</table>
**ssPushType**  
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):
- ssCREATE - Create a star schema
- ssUPDATE - Update a star schema

**eaExtractType**  
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):
- 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 - For internal use.
- EA_EXTRACT_TYPE_ESSBASE - Oracle Essbase
- EA_EXTRACT_TYPE_WAREHOUSE - Data Warehouse
- EA_EXTRACT_TYPE_FLATFILE
- EA_EXTRACT_TYPE_FLATFILE_NOHEADER
- EA_EXTRACT_TYPE_MAX

**vbIncludeDynamicAccts**  
Boolean (ByVal). A flag that specifies whether to include dynamic accounts. Pass TRUE to include these accounts, pass FALSE to exclude them.

**vbIncludeCalculatedData**  

**vbIncludeDerivedData**  

**eaLineItemOption**  
EA_LINEITEM_OPTIONS. Internal use. Pass EA_LINEITEM_SUMMARY.

**vbIncludeCellDesc**  

**vbIncludePhaseGroup**  

**bstrDelimiter**  
String. Delimiter to use for flat file extracts. (eaExtractType = EA_EXTRACT_TYPE_FLATFILE)

**pIUnkHfmSliceCOM**  
HfmSliceCOM or HfmSliceStrCOM. An HfmSliceCOM or HfmSliceStrCOM object representing the desired POVs.

**plTaskId**  
Long. Returns a unique ID assigned to this extract request.

---

**Return Value**
None.

**Example**
See HfmSliceCOM and HfmSliceStrCOM documentation on how to set multiple POVs.

**CreateStarSchemaAndReturnTaskID**

**Deprecated** - use CreateStarSchemaExtDim.
CreateStarSchemaFromHTTP
For internal use.

DeleteStarSchema
Deletes a star schema from a 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
bstrDSN String (ByVal). The Extended Analytics Data Source Name for the database that contains the star schema to be deleted.
Tip: For the Extended Analytics Data Source Names registered on an application server, use EnumRegisteredDSNs.

bstrTablePrefix 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 registered on the application server.

Note: To add Data Source Names for Extended Analytics, use the Financial Management Configuration Utility. See the Oracle Hyperion Financial Management 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 box with the Data Source Names registered on an application server. The HsvStarSchemaACM instance is created on the application server by \texttt{HsvSession.CreateObject}. The application server is identified by the \texttt{HsvSession} object reference passed to \texttt{SetSession}.

```vba
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 \texttt{CreateStarSchema}. \texttt{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.

\textbf{Tip:} You can use \texttt{GetAsynchronousTaskStatus} to display the status of the thread. For example, you can periodically update a progress bar to display the number of items that were processed.

Syntax

```
<HsvStarSchemaACM>.GetAsynchronousTaskStatus plCurrentTask, pdNumRecords,
pdNumCompletedRecords, pvbIsRunning, plErrorCode
```
## Argument Description

### plCurrentTask
EA_TASK_STATUS_FLAGS. Returns the identity of the task currently being processed by the CreateStarSchemaExtDim thread. Valid values represented by the following constants (from Enum EA_TASK_STATUS_FLAGS):

- EA_TASK_STATUS_BLOCKED - The task is waiting to be queued.
- EA_TASK_STATUS_CANCELLED - The task was cancelled.
- EA_TASK_STATUS_COMPLETE - The task is complete.
- EA_TASK_STATUS_COMPLETE_W_ERRORS - The task is complete, but an error occurred. The error number is returned by the plErrorCode 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 - For internal use.

**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.

### pdNumRecords
Double. Returns the total number of items to be processed for the task. This does not apply to the Initializing task.

**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.

### pdNumCompletedRecords
Double. Returns the total number of items processed for the task.

### pvbIsRunning
Boolean. Indicates whether the CreateStarSchemaExtDim thread is still running. Returns TRUE if the thread is running, FALSE if it finished.

### plErrorCode
Long. Returns the HRESULT for the thread’s final processing status. This value is valid only when the thread finishes running.

---

### GetExtractLogData

Returns a string that provides a log of the HsvStarSchemaACM instance’s most recent call to CreateStarSchema.

**Note:** The system periodically deletes information on performed tasks. If the log information or an extraction was deleted, GetExtractLogData does not return log information. The pvbHadData argument indicates whether GetExtractLogData found log data.

**Syntax**

```c
<HsvStarSchemaACM>.GetExtractLogData pbstrLogData, pvbHadData
```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pbstrLogData</td>
<td>String. Returns a log of the most recent call to CreateStarSchemaExtDim.</td>
</tr>
<tr>
<td>pbvHadData</td>
<td>Boolean. Returns a flag that indicates whether GetExtractLogData could find the log information. Returns TRUE if the log information is found, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**GetPersistedSettings**

*Deprecated* - use GetPersistedSettingsExtDim.

**GetPersistedSettingsExtDim**

*For internal use.*

**QuitAsynchronousTask**

Terminates the thread launched by CreateStarSchema.

**setDefaultTablePrefix**

*For internal use.*

**SetPersistedSettings**

*Deprecated* - use SetPersistedSettingsExtDim.

**SetPersistedSettingsExtDim**

*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**
---|---
`plHsvSessionUnk` | 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:

```vbscript
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 121, and are described in detail in the following topics.

**DeleteTemplate**

Deletes a Extended Analytics template.

**Syntax**

```vbscript
<IHsvStarSchemaTemplates>.DeleteTemplate bstrTmpName
```

**Argument** | **Description**
---|---
`bstrTmpName` | String (ByVal). The name of the template to delete.

**EnumTemplates**

Returns the names of the application’s Extended Analytics templates.

**Syntax**

```vbscript
<IHsvStarSchemaTemplates>.EnumTemplates()
```
Return Value
Variant. Returns an array of strings that contain the templates’ names.

**GetTemplate**
Returns an XML string that contains a Extended Analytics template definition.

Syntax

```xml
<IHsvStarSchemaTemplates>.GetTemplate(bstrTmplName)
```

**Argument** | **Description**
--- | ---
`bstrTmplName` | 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 733.

**SetTemplate**
Creates an Extended Analytics template, using an XML string that contains the template definition.

Syntax

```xml
<IHsvStarSchemaTemplates>.SetTemplate bstrTmplName, bstrTmplContents, vbOverwrite
```

**Argument** | **Description**
--- | ---
`bstrTmplName` | String (ByVal). The name of the new template.
`bstrTmplContents` | String (ByVal). The XML string containing the template definition. The string’s schema is described below.
`vbOverwrite` | 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 `bstrTmplContents` argument and returned by GetTemplate.

```xml
<povTemplate>
  <povEA/>
  <options>
    <tablePrefix/>
    <exportOption/>
    <selectedDSN/>
    <excludeDynAccts/>
  </options>
</povTemplate>
```
The following table describes the XML string’s schema:

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>povTemplate</td>
<td>The root element.</td>
</tr>
<tr>
<td>povEA</td>
<td>Specifies the dimension members for the template. Members for each dimension are preceded by the following characters:</td>
</tr>
<tr>
<td></td>
<td>● A# = Account</td>
</tr>
<tr>
<td></td>
<td>● I# = Intercompany Partner</td>
</tr>
<tr>
<td></td>
<td>● C1# = Custom1</td>
</tr>
<tr>
<td></td>
<td>● C2# = Custom2</td>
</tr>
<tr>
<td></td>
<td>● C3# = Custom3</td>
</tr>
<tr>
<td></td>
<td>● C4# = Custom4</td>
</tr>
<tr>
<td></td>
<td>● S# = Scenario</td>
</tr>
<tr>
<td></td>
<td>● Y# = Year</td>
</tr>
<tr>
<td></td>
<td>● P# = Period</td>
</tr>
<tr>
<td></td>
<td>● W# = View</td>
</tr>
<tr>
<td></td>
<td>● E# = Entity</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> The Entity dimension must include both the parent and child members, delimited by a period.</td>
</tr>
<tr>
<td></td>
<td>● V# = Value</td>
</tr>
<tr>
<td></td>
<td>Dimensions are delimited with periods. Multiple members for a dimension are delimited by semicolons. Member list labels are enclosed with braces ({}).</td>
</tr>
<tr>
<td>options</td>
<td>Contains the elements that define the template’s extract options.</td>
</tr>
<tr>
<td>tablePrefix</td>
<td>The Relational Table Prefix, which is the prefix that identifies the star schema’s tables.</td>
</tr>
<tr>
<td>exportOption</td>
<td>The aggregation to use. Valid values are listed below:</td>
</tr>
<tr>
<td></td>
<td>● 0 = Standard</td>
</tr>
<tr>
<td></td>
<td>● 1 = Metadata Only</td>
</tr>
<tr>
<td></td>
<td>● 2 = Selected Metadata Only</td>
</tr>
<tr>
<td></td>
<td>● 3 = For internal use</td>
</tr>
<tr>
<td></td>
<td>● 4 = Essbase</td>
</tr>
<tr>
<td></td>
<td>● 5 = Data Warehouse</td>
</tr>
<tr>
<td>selectedDSN</td>
<td>The Extended Analytics Data Source Name that points to the database for the star schema.</td>
</tr>
<tr>
<td>excludeDynAccts</td>
<td>A flag that specifies whether to exclude dynamic accounts. Zero (0) indicates to include dynamic accounts, a non-zero value indicates to exclude them.</td>
</tr>
</tbody>
</table>

**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#2012.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
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 737. The IHsvAdminICM interface exposes the administration of intercompany transactions and is described in “IHsvAdminICM Interface Methods” on page 755.

**HsvICM Object Methods**

The HsvICM object enables you to work with intercompany transactions. For example, you can use this object to create, match, and unmatch transactions.

To set an HsvICM object reference, use the HsvSession object’s ICM property as shown in the following example:

```vbscript
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 121, 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. See “HsvICTransactionsData Object Methods” on page 663.

**AllMVTransPosted**

Indicates whether all transactions for a scenario, year, period, and entity that are in a matched state are 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**

- **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 are posted, FALSE otherwise.

**CheckReportSecurity**

*For internal use.*

**CreateICTransaction**

*Deprecated* - use **CreateICTransactionExtDim**.

**CreateICTransactionExtDim**

Creates an intercompany transaction for a cell. Supersedes CreateICTransaction.

**Tip:** To test whether a cell is valid for intercompany transactions, use

- **DoesCellSupportICTransactionDetail.** To update an existing transaction, use **SaveICTransactionExtDim**.

**Syntax**

```<HsvICM>.CreateICTransactionExtDim vbOverwrite, pIUnkHfmPovCOM, ITRCur, IReason, dTRAmt, dTRLAmt, dTRRate, dTRDate, bstrID, bstrSubID, bstrRefID, bstrComment1, bstrComment2```

**Argument**

- **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.
### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIUNKHfmPovCOM</td>
<td>HfmPovCOM object representing the POV</td>
</tr>
<tr>
<td>lTRCur</td>
<td>Long (ByVal). The ID of the transaction’s currency.</td>
</tr>
<tr>
<td>Tip:</td>
<td>You can obtain the ID of an Entity dimension member’s currency with \textit{GetEntityCurrencyID}. You also can obtain a currency ID with \textit{GetTransCurrencyID}.</td>
</tr>
<tr>
<td>lReason</td>
<td>Long (ByVal). The ID of the transaction’s reason code.</td>
</tr>
<tr>
<td>Tip:</td>
<td>You can get a reason code ID with \textit{GetICReasonCodeID}.</td>
</tr>
<tr>
<td>dTRAmnt</td>
<td>Double (ByVal). The transaction amount.</td>
</tr>
<tr>
<td>dTRLAmnt</td>
<td>Double (ByVal). The entity currency amount.</td>
</tr>
<tr>
<td>dTRRate</td>
<td>Double (ByVal). The conversion rate.</td>
</tr>
<tr>
<td>dTRDate</td>
<td>Double (ByVal). The transaction date formatted as a Double.</td>
</tr>
<tr>
<td>bstrID</td>
<td>String (ByVal). The Transaction ID.</td>
</tr>
<tr>
<td>bstrSubID</td>
<td>String (ByVal). The transaction’s Sub ID.</td>
</tr>
<tr>
<td>bstrRefID</td>
<td>String (ByVal). The transaction’s Reference ID.</td>
</tr>
<tr>
<td>bstrComment1</td>
<td>String (ByVal). The first comment for the transaction.</td>
</tr>
<tr>
<td>Note:</td>
<td>You can specify a maximum of 256 characters.</td>
</tr>
<tr>
<td>bstrComment2</td>
<td>String (ByVal). The second comment for the transaction.</td>
</tr>
<tr>
<td>Note:</td>
<td>You can specify a maximum of 256 characters.</td>
</tr>
</tbody>
</table>

### Return Value
None.

### Example
See HfmPOVCom documentation for how to set the POV.

### DeleteICReasonCode

Deletes a reason code.

#### Syntax
```plaintext
<HsvICM>.DeleteICReasonCode bstrReasonCode
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrReasonCode</td>
<td>String (ByVal). The reason code to be deleted.</td>
</tr>
</tbody>
</table>
DoesCellSupportICTransactionDetail

Deprecated - use DoesCellSupportICTransactionDetailExtDim.

DoesCellSupportICTransactionDetailExtDim

Indicates whether a cell supports intercompany transactions. Supersedes DoesCellSupportICTransactionDetail.

Syntax

<HsvICM>.DoesCellSupportICTransactionDetailExtDim plUnkHfmPovCOM

Argument Description
plUnkHfmPovCOM HfmPovCOM. HfmPovCOM object representing the POV.

Return Value

Boolean. Returns TRUE if the cell supports intercompany transactions, FALSE otherwise.

Example

See HfmPovCOM documentation on how to set the POV.

GetColumnFilter

For internal use.

GetCurrencyInfo

Returns a currency’s label, translation operator, and scale.

Syntax

<HsvICM>.GetCurrencyInfo lCurID, pbstrCurrencyLabel, pbstrCurrencyOp, psCurrencyScale

Argument Description
lCurID Long (ByVal). The currency ID of the currency.
pbstrCurrencyLabel String. Returns the currency’s label.
pbstrCurrencyOp String. Returns the currency’s translation operator.
psCurrencyScale Integer. Returns the currency’s scale.
**GetEntitiesContacts**

Returns the usernames and security identifiers of the users to be alerted for a scenario and intercompany transaction-related event for the specified entities.

Syntax

```c
<HsvICM>.GetEntitiesContacts lEventType, lScenario, varalEntities, pvara2DUsersInfo
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lEventType</td>
<td>Long (ByVal). The ID of the intercompany transaction-related event. Valid values are represented by the HFMConstants enumeration tagICEVENTTYPE, which is described in “Event Constants” on page 902.</td>
</tr>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>varalEntities</td>
<td>Long array (ByVal). The member IDs of the Entity dimension members.</td>
</tr>
</tbody>
</table>
| pvara2DUsersInfo | 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 varalEntities argument. The second dimension contains the user information and is indexed from 0 to 1:  
  - 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

```c
<HsvICM>.GetEntityCurrencyID(lEntityID)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lEntityID</td>
<td>Long (ByVal). The member ID of the Entity dimension member.</td>
</tr>
</tbody>
</table>

**Return Value**

Long. Returns the currency’s ID.

**Example**

GetEntityCurrencyID is used in the example for CreateICTransactionExtDim.

**GetICReasonCodeID**

Gets the ID of a reason code.
Syntax

```
<HsvICM>.GetICReasonCodeID(bstrReasonCodeLabel)
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrReasonCodeLabel</td>
</tr>
</tbody>
</table>

**Return Value**

Long. Returns the reason code’s ID.

**Example**

GetICReasonCodeID is used in the example for **CreateICTransactionExtDim**.

---

**GetICReasonCodeLabel**

Returns a reason code from the code’s internal ID.

**Syntax**

```
<HsvICM>.GetICReasonCodeLabel(lReasonCodeID)
```

**Argument**

<table>
<thead>
<tr>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lReasonCodeID</td>
</tr>
</tbody>
</table>

**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:

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Long. Returns the IDs of the reason codes.</td>
</tr>
</tbody>
</table>
### GetICReasonCodes

<table>
<thead>
<tr>
<th>Index</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>String. Returns the reason codes.</td>
</tr>
<tr>
<td>2</td>
<td>String. Returns the descriptions of the reason codes.</td>
</tr>
</tbody>
</table>

**Example**

The following example prints an application’s reason codes and descriptions to Visual Basic’s Immediate window.

```vbnet
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**

- `pIUnkICTransactionsData` Object (ByVal). An initialized HsvICTransactionsData object reference. `GetICTransactions` populates the object with the intercompany transactions for the specified scenario, year, and period.

- `vbSkipDimSecurity` Boolean (ByVal). A flag that specifies whether to filter transactions containing dimension members for which the connected user does not have security rights. Pass TRUE to return transactions regardless of the user’s security rights, FALSE 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.
GetMonitorICDetails

Syntax

```<HsvICM>.GetMonitorICDetails lScenario, lYear, lPeriod, lEntity, plUnpostedUnMatched, plUnpostedMisMatched, plUnpostedMatched, plUnpostedReasonMisMatched, plPostedUnMatched, plPostedMisMatched, plPostedMatched, plPostedReasonMisMatched```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code></td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td><code>lPeriod</code></td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td><code>lEntity</code></td>
<td>Long (ByVal). The member ID of the Entity dimension member.</td>
</tr>
<tr>
<td><code>plUnpostedUnMatched</code></td>
<td>Long. Returns the number of intercompany transactions that are unposted and unmatched.</td>
</tr>
<tr>
<td><code>plUnpostedMisMatched</code></td>
<td>Long. Returns the number of intercompany transactions that are unposted and mismatched without a reason code.</td>
</tr>
<tr>
<td><code>plUnpostedMatched</code></td>
<td>Long. Returns the number of intercompany transactions that are unposted and matched.</td>
</tr>
<tr>
<td><code>plUnpostedReasonMisMatched</code></td>
<td>Long. Returns the number of intercompany transactions that are unposted and mismatched with a reason code.</td>
</tr>
<tr>
<td><code>plPostedUnMatched</code></td>
<td>Long. Returns the number of intercompany transactions that are posted and unmatched.</td>
</tr>
<tr>
<td><code>plPostedMisMatched</code></td>
<td>Long. Returns the number of intercompany transactions that are posted and mismatched without a reason code.</td>
</tr>
<tr>
<td><code>plPostedMatched</code></td>
<td>Long. Returns the number of intercompany transactions that are posted and matched.</td>
</tr>
<tr>
<td><code>plPostedReasonMisMatched</code></td>
<td>Long. Returns the number of intercompany transactions that are posted and mismatched with a reason code.</td>
</tr>
</tbody>
</table>

GetMonitorICSummary

Returns the number of Entity dimension members that have various combinations of locking and process statuses for the Scenario, Year, Period, and Entity dimension members.

Syntax

```<HsvICM>.GetMonitorICSummary lScenario, lYear, lPeriod, varalEntitiesIDs, plUnlockedNotStarted, plLockedNotStarted, plUnlockedStarted, plLockedStarted```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code></td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td><code>lPeriod</code></td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td><code>varalEntitiesIDs</code></td>
<td>Long array (ByVal). The member IDs of the Entity dimension members.</td>
</tr>
<tr>
<td><code>plUnlockedNotStarted</code></td>
<td></td>
</tr>
<tr>
<td><code>plLockedNotStarted</code></td>
<td></td>
</tr>
<tr>
<td><code>plUnlockedStarted</code></td>
<td></td>
</tr>
<tr>
<td><code>plLockedStarted</code></td>
<td></td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>plUnlockedNotStarted</td>
<td>Long. Returns the number of Entity dimension members that have an Unlocked locking status and a Not Started process status.</td>
</tr>
<tr>
<td>plLockedNotStarted</td>
<td>Long. Returns the number of Entity dimension members that have a Locked locking status and a Not Started process status.</td>
</tr>
<tr>
<td>plUnlockedStarted</td>
<td>Long. Returns the number of Entity dimension members that have an Unlocked locking status and a Started process status.</td>
</tr>
<tr>
<td>plLockedStarted</td>
<td>Long. Returns the number of Entity dimension members that have a Locked locking status and a Started process status.</td>
</tr>
</tbody>
</table>

### GetMonitorICTransactions

Indicates whether one or more Entity dimension members have intercompany transactions for 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**

```plaintext
<HsvICM>.GetMonitorICTransactions lScenario, lYear, lPeriod, varalEntities, lSortFilterOptions, lFromRec, lMaxRows, plTotalEntities, pvara2DMonitor
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>varalEntities</td>
<td>Long array (ByVal). The member IDs of the Entity dimension members.</td>
</tr>
<tr>
<td>lSortFilterOptions</td>
<td>Long (ByVal). Specifies filtering and sorting options for the information returned by the pvara2DMonitor argument. Valid values are represented by the HFMConstants enumeration tagICM_MONITOR_FILTER_SORT_FLAGS, which is described in &quot;Filtering and Sorting Options&quot; on page 901. 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: ICM_MONITOR_FILTER_STARTED + ICM_MONITOR_FILTER_NOTSTARTED</td>
</tr>
<tr>
<td>lFromRec</td>
<td>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 pvara2DMonitor argument.</td>
</tr>
<tr>
<td>lMaxRows</td>
<td>Long (ByVal). The maximum number of transactions to include in the array returned by the pvara2DMonitor argument.</td>
</tr>
<tr>
<td>plTotalEntities</td>
<td>Long. Returns the number of Entity dimension members that match the criteria.</td>
</tr>
</tbody>
</table>
**GetRowFilter**

*For internal use.*

**GetTransCurrencyID**

Returns the currency ID of a specified currency.

**Syntax**

\[\text{<HsvICM>.GetTransCurrencyID(bstrTRCurr)}\]

**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**

\[\text{<HsvICM>.GetUnMatchTransactions pIUnkICTransactionsData}\]
**IsOneSideOfTransactionGroupWriteable**

Indicates whether the user has write access to either the entity or Intercompany Partner cell of all transactions in a range of an HsvICTransactionsData object reference’s transactions.

For example, suppose that you are unmatching the transactions listed in the following table:

<table>
<thead>
<tr>
<th>Entity</th>
<th>Intercompany Partner (Entity)</th>
<th>Account</th>
<th>Amount</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>Receivable</td>
<td>100</td>
</tr>
<tr>
<td>A</td>
<td>B</td>
<td>Sales</td>
<td>100</td>
</tr>
<tr>
<td>B</td>
<td>A</td>
<td>Payable</td>
<td>200</td>
</tr>
</tbody>
</table>

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**

```csharp
<HsvICM>.IsOneSideOfTransactionGroupWriteable(pIUnkICTransactionsData, lScenario, lYear, lPeriod, lFromIndex, lToIndex)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code></td>
<td>Long (ByVal). The member ID of the Scenario dimension member for the transactions.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the Year dimension member for the transactions.</td>
</tr>
<tr>
<td><code>lPeriod</code></td>
<td>Long (ByVal). The member ID of the Period dimension member for the transactions.</td>
</tr>
<tr>
<td><code>lFromIndex</code></td>
<td>Long (ByVal). The index of the first transaction in the range of the HsvICTransactionsData object reference’s transactions to be tested.</td>
</tr>
<tr>
<td><code>lToIndex</code></td>
<td>Long (ByVal). The index of the last transaction in the range of the HsvICTransactionsData object reference’s transactions to be tested.</td>
</tr>
</tbody>
</table>
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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member for which to match intercompany transactions.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member for which to match intercompany transactions.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member for which to match intercompany transactions.</td>
</tr>
<tr>
<td>varalEntities</td>
<td>Long array (ByVal). The member IDs of the Entity dimension members for which to match intercompany transactions.</td>
</tr>
<tr>
<td>varalICPs</td>
<td>Long array (ByVal). The member IDs of the Intercompany Partner dimension members for which to match intercompany transactions.</td>
</tr>
<tr>
<td>varabstrAcctsPOV</td>
<td>String array (ByVal). The strings that specify the Account and Custom dimension members for which to match intercompany transactions.</td>
</tr>
<tr>
<td></td>
<td>- You must specify an Account dimension member.</td>
</tr>
<tr>
<td></td>
<td>- You optionally can specify members for any of the Custom dimensions.</td>
</tr>
<tr>
<td></td>
<td>- Dimensions are delimited with periods.</td>
</tr>
<tr>
<td></td>
<td>- Members for each dimension are preceded by the following characters:</td>
</tr>
<tr>
<td></td>
<td>- A# = Account</td>
</tr>
<tr>
<td></td>
<td>- C1# = Custom1</td>
</tr>
<tr>
<td></td>
<td>- C2# = Custom2</td>
</tr>
<tr>
<td></td>
<td>- C3# = Custom3</td>
</tr>
<tr>
<td></td>
<td>- C4# = Custom4</td>
</tr>
<tr>
<td>varabstrMAcctsPOV</td>
<td>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 <code>varabstrAcctsPOV</code> argument.</td>
</tr>
<tr>
<td>lMatchOption</td>
<td>Long (ByVal). This argument is reserved for future use. However, you must pass a valid Long, such as zero (0).</td>
</tr>
<tr>
<td>dMatchTolerance</td>
<td>Double (ByVal). The matching tolerance.</td>
</tr>
<tr>
<td>bstrMatchCurr</td>
<td>String (ByVal). The transaction currency by which to match. To match for all transaction currencies, pass a blank string.</td>
</tr>
</tbody>
</table>
**MatchAutoIDs**

Matches intercompany transactions by Transaction ID or Reference ID.

**Syntax**

```<HsvICM>.MatchAutoIDs lScenario, lYear, lPeriod, varalEntities, varalICPs, lMatchOption, bstrMatchCurr, bstrMatchIDs```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member for which to match intercompany transactions.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member for which to match intercompany transactions.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member for which to match intercompany transactions.</td>
</tr>
<tr>
<td>varalEntities</td>
<td>Long array (ByVal). The member IDs of the Entity dimension members for which to match intercompany transactions.</td>
</tr>
<tr>
<td>varalICPs</td>
<td>Long array (ByVal). The member IDs of the Intercompany Partner dimension members for which to match intercompany transactions.</td>
</tr>
<tr>
<td>lMatchOption</td>
<td>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:</td>
</tr>
<tr>
<td></td>
<td>● ICT_MATCH_REFERENCE</td>
</tr>
<tr>
<td></td>
<td>● ICT_MATCH_TRANSACTIONID</td>
</tr>
<tr>
<td></td>
<td>For descriptions of these constants, see “Match Option Constants” on page 900.</td>
</tr>
<tr>
<td>bstrMatchCurr</td>
<td>String (ByVal). The transaction currency for which to match. To match for all transaction currencies, pass a blank string.</td>
</tr>
<tr>
<td>bstrMatchIDs</td>
<td>String (ByVal). A comma-delimited list of the Transaction IDs or the Reference IDs that identify the transactions to match. 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”: &quot;ref%,acct%&quot;</td>
</tr>
</tbody>
</table>

**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 scenario, year, and period.
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 scenario, year, and period.

Tip: To process only specified transactions for a scenario, year, and period, use ProcessICTransactions.

Syntax

<HsvICM>.ProcessAllICTrans(lICAAction, pUnkICTransactionsData)

Argument Description

lICAAction  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_DELETETRANS_ALL
- ICM_POSTTRANS_ALL
- ICM_UNMATCHTRANS_ALL
- ICM_UNPOSTTRANS_ALL

For descriptions of these constants, see "Processing Action Constants" on page 901.


Note: The HsvICTransactionsData method Initialize specifies the scenario, year, and period of the transactions to process.

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 an non-zero error number otherwise:

<pre>
<processtransactionsresponse status="HRESULT">
</processtransactionsresponse>
</pre>
Example

The following function deletes all transactions for the specified scenario, year, and period.

```vbnet
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

*Deprecated* - use ProcessICTransactionsExtDim.

ProcessICTransactionsExtDim

Deletes, posts, unposts, or unmatches the specified intercompany transactions for a 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. Supersedes ProcessICTransactions.

**Tip:** To process all transactions for a scenario, year, and period, use ProcessAllICTrans.

Syntax

```
<HsvICM>.ProcessICTransactionsExtDim IICaction, plUnkHfmSliceCOM, varaISeqIDs
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>IICaction</strong></td>
<td>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:</td>
</tr>
</tbody>
</table>
|                 | - ICM_POSTTRANS
|                 | - ICM_UNPOSTTRANS
|                 | - ICM_DELETETRANS
|                 | - ICM_MANUALMATCHTRANS
|                 | - ICM_UNMATCHTRANS

For descriptions of these constants, see "Processing Action Constants" on page 901.

<table>
<thead>
<tr>
<th><strong>plUnkHfmSliceCOM</strong></th>
<th>HfmSliceCOM. HfmSliceCOM object representing the POVs to process.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Nota:</strong></td>
<td>Scenario, Year, and Period are fixed (one member).</td>
</tr>
</tbody>
</table>

| **varaISeqIDs**      | Long Array. 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. |

HsvICM Object Methods  751
Return Value

Data Type and description:
String. Returns an XML string, with the format varying depending on 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`.

Example

See HfmSliceCOM documentation on how to specify multiple rows.

**SaveColumnFilter**

For internal use.

**SaveICReasonCode**

Creates a reason code.

**Syntax**

```
<HsvICM>.SaveICReasonCode bstrReasonCode, bstrReasonDescription
```

**Argument** | **Description**
--- | ---
`bstrReasonCode` | String (ByVal). The reason code.
`bstrReasonDescription` | String (ByVal). The description of the reason code.

**SaveICTransaction**

Deprecated - use `SaveICTransactionExtDim`.
SaveICTransactionExtDim

Updates an existing transaction. Supersedes SaveICTransaction.

Syntax

```
<HsvICM>.SaveICTransactionExtDim vbAmountChanged, vbOverwrite, lSeqID, plUnkHfmPovCOM, lTRCur, lReason, dTRAmt, dTRLAmt, dTRRate, dTRDate, bstrID, bstrSubID, bstrRefID, bstrComment1, bstrComment2
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vbAmountChanged</td>
<td>Boolean (ByVal). A flag that specifies whether the transaction amount is being changed. Pass TRUE to change the amount, FALSE otherwise.</td>
</tr>
<tr>
<td>vbOverwrite</td>
<td>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.</td>
</tr>
<tr>
<td>lSeqID</td>
<td>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.</td>
</tr>
<tr>
<td>plUnkHfmPovCOM</td>
<td>HfmPovCOM. HfmPovCOM object representing the POV.</td>
</tr>
<tr>
<td>lTRCur</td>
<td>Long (ByVal). The ID of the transaction’s currency.</td>
</tr>
<tr>
<td>Tip:</td>
<td>You can obtain the ID of an Entity dimension member’s currency with GetEntityCurrencyID. You also can obtain a currency ID with GetTransCurrencyID.</td>
</tr>
<tr>
<td>lReason</td>
<td>Long (ByVal). The ID of the transaction’s reason code.</td>
</tr>
<tr>
<td>Tip:</td>
<td>You can get a reason code ID with GetICReasonCodeID.</td>
</tr>
<tr>
<td>dTRAmt</td>
<td>Double (ByVal). The transaction amount.</td>
</tr>
<tr>
<td>dTRLAmt</td>
<td>Double (ByVal). The entity currency amount.</td>
</tr>
<tr>
<td>dTRRate</td>
<td>Double (ByVal). The conversion rate.</td>
</tr>
<tr>
<td>dTRDate</td>
<td>Double (ByVal). The transaction date formatted as a Double.</td>
</tr>
<tr>
<td>bstrID</td>
<td>String (ByVal). The Transaction ID.</td>
</tr>
<tr>
<td>bstrSubID</td>
<td>String (ByVal). The transaction’s Sub ID.</td>
</tr>
<tr>
<td>bstrRefID</td>
<td>String (ByVal). The transaction’s Reference ID.</td>
</tr>
<tr>
<td>bstrComment1</td>
<td>String (ByVal). The first comment for the transaction.</td>
</tr>
<tr>
<td>Note:</td>
<td>You can specify a maximum of 256 characters.</td>
</tr>
<tr>
<td>bstrComment2</td>
<td>String (ByVal). The second comment for the transaction.</td>
</tr>
<tr>
<td>Note:</td>
<td>You can specify a maximum of 256 characters.</td>
</tr>
</tbody>
</table>

Return Value

None.
Example
See HfmPovCOM documentation on how to set the POV.

### 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**

```xml
<HsvICM>.SetReasonCodeToICTransactions(lScenario, lYear, lPeriod, lReasonCodeID, varalSeqIDs, varalEntities, varalICPs)
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lScenario</code></td>
<td>Long (ByVal). The member ID of the transactions’ Scenario dimension member.</td>
</tr>
<tr>
<td><code>lYear</code></td>
<td>Long (ByVal). The member ID of the transactions’ Year dimension member.</td>
</tr>
<tr>
<td><code>lPeriod</code></td>
<td>Long (ByVal). The member ID of the transactions’ Period dimension member.</td>
</tr>
<tr>
<td><code>lReasonCodeID</code></td>
<td>Long (ByVal). The ID of the reason code to set.</td>
</tr>
<tr>
<td><code>varalSeqIDs</code></td>
<td>Long array (ByVal). The transactions’ sequence IDs.</td>
</tr>
<tr>
<td><code>varalEntities</code></td>
<td>Long array (ByVal). The member IDs of the transactions’ Entity dimension members.</td>
</tr>
<tr>
<td><code>varalICPs</code></td>
<td>Long array (ByVal). The member IDs of the transactions’ Intercompany Partner dimension members.</td>
</tr>
</tbody>
</table>

**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 76** SetReasonCodeToICTransactions Return Value XML String

<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ictransactions</td>
<td>Root element, and contains one <code>&lt;transaction&gt;</code> per intercompany transaction.</td>
</tr>
</tbody>
</table>

754  HsvICM Type Library
<table>
<thead>
<tr>
<th>Element</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>transaction</td>
<td>Represents an intercompany transaction, and contains the dimension member, ID, and HRESULT elements described in the following rows.</td>
</tr>
<tr>
<td>entity</td>
<td>The label of the transaction’s Entity dimension member.</td>
</tr>
<tr>
<td>partner</td>
<td>The label of the transaction’s Intercompany Partner dimension member.</td>
</tr>
<tr>
<td>account</td>
<td>The label of the transaction’s Account dimension member.</td>
</tr>
<tr>
<td>transid</td>
<td>The transaction’s Transaction ID.</td>
</tr>
<tr>
<td>transsubid</td>
<td>The transaction’s Sub ID.</td>
</tr>
<tr>
<td>hresult</td>
<td>The HRESULT associated with updating the transaction’s reason code.</td>
</tr>
</tbody>
</table>

**UnMatchICTransactions**

Unmatches transactions to which the specified match codes are assigned.

**Syntax**

```xml
<HsvICM>.UnMatchICTransactions(lScenario, lYear, lPeriod, varabstrMatchCodes)
```

**Argument**

- **lScenario**  
  Long (ByVal). The Scenario dimension member for which to unmatch transactions.
- **lYear**  
  Long (ByVal). The Year dimension member for which to unmatch transactions.
- **lPeriod**  
  Long (ByVal). The Period dimension member for which to unmatch transactions.
- **varabstrMatchCodes**  
  String array (ByVal). The match codes for which to unmatch 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 an non-zero error number otherwise:

```xml
<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 123, 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 occurs as an HRESULT. If the user has security rights, zero occurs as an HRESULT.

**Tip:** In Visual Basic, check for the return value with `Err.Number`.

**Syntax**

```vbnet
<HsvAdminICM>.CheckSecurityForICExtract
```

**CloseICPeriod**

Closes a period for intercompany transactions.

**Syntax**

```vbnet
<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`.

```vbnet
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)
```

756  HsvICM Type Library
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 scenario, year, and period.

Note: To return an Integer that indicates whether an 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 899.

GetLockStatusICEntities

Deprecated - use GetICEntitiesLockStatus.

GetLockStatusICEntity
Indicates whether an entity is locked or unlocked for a 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 HFMConstants type library constants listed in “Lock Status Constants” on page 899.

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

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.
Argument | Description
--- | ---
pSStatus | Integer. Returns the period’s status. Valid values are represented by the HFMConstants type library constants listed in “Period Status Constants” on page 899.
pSMVBP | Integer. 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 899.
pDMatchTolerance | 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.

```vba
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**
--- | ---
lScenario | Long (ByVal). The member ID of the Scenario dimension member.
lYear | Long (ByVal). The member ID of the Year dimension member.
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:

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>psStatus</td>
<td>Integer. Returns the period’s status. Valid values are represented by the HFMConstants type library constants listed in “Period Status Constants” on page 899.</td>
</tr>
<tr>
<td>psMVBP</td>
<td>Integer. 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 899.</td>
</tr>
<tr>
<td>pdMatchTolerance</td>
<td>Double. Returns the matching tolerance specified for the period.</td>
</tr>
<tr>
<td>Index</td>
<td>Description</td>
</tr>
<tr>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td>0</td>
<td>Long. Returns the member ID of the Period dimension member to which the settings apply.</td>
</tr>
<tr>
<td>1</td>
<td>Byte. Returns the period’s status. Valid values are represented by the HFMConstants type library constants listed in “Period Status Constants” on page 899.</td>
</tr>
<tr>
<td>2</td>
<td>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 899.</td>
</tr>
<tr>
<td>3</td>
<td>Double. Returns the matching tolerance specified for the period.</td>
</tr>
</tbody>
</table>

**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 were 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

Deprecated - use LoadICTransactionsExtDim.

LoadICTransactionsExtDim

Loads or scans intercompany transactions from arrays that specify the transactions’ dimension members and data.

Syntax

<IHsvAdminICM>.LoadICTransactionsExtDim pIUnkICTransactionsData, lLoadOptions, dStartLoadTime

Argument       Description

pIUnkICTransactionsData        HsvICTransactionData. HsvICTransactionData object contains an array of intercompany transactions for a specified scenario, year, and period.

lLoadOptions        Long. 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.

dStartLoadTime        Double. The timestamp indicating when to start the loading task. The timestamp must be a Date cast to a Double.

Example

The following example loads an intercompany transaction.

Dim cAdminICM As IHsvAdminICM
Set cAdminICM = g_cSession.ICM
Dim cICTransData As HsvICTransactionsData
Set cICTransData = New HsvIcTransactionsData
Dim lScen As Long, lYear As Long, lPer As Long
Dim lIndex As Long, lTrCur As Long
lScen = GetMemberID(DIMENSIONSCENARIO, "ActMon")
lYear = GetMemberID(DIMENSIONYEAR, "2012")
lPer = GetMemberID(DIMENSIONPERIOD, "February")
lTrCur = cICM.GetTransCurrencyID "EUR"
cICTransData.Initialize lScen, lYear, lPer
cICTransData.AddICTransactionDataExtDim -1, pIUnkHfmPovCOM, _
lTrCur, 0, 0, -1, -1, Now, CDb1(CDate("2/13/2012")), "100", "100", "1.0", _
NULL, "Tran001", "Sub001", "Ref001", NULL, "comment1", "comment2", lIndex
cAdminICM.LoadICTransactionsExtDim cICTransData, ICT_LOAD_REPLACE, Now

LockICEntity

Locks an entity for a scenario, year, and period.

Syntax

<IHsvAdminICM>.LockICEntity lScenario, lYear, lPeriod, lEntity, vbCheckMVBP
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member.</td>
</tr>
</tbody>
</table>

vbCheckMVBP  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 occurs.

**OpenICPeriod**

Opens a period for a scenario and year, and specifies the period’s matching tolerance and Match/Validate Before Post settings.

Syntax

```csharp
<IHsvAdminICM>.OpenICPeriod lScenario, lYear, lPeriod, lMVBP, dMatchTolerance
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>lPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>lMVBP</td>
<td>Integer (ByVal). The Match/Validate Before Post setting for the period. Valid values are represented by the HFMConstants type library constants listed in &quot;Match/Validate Before Post Constants&quot; on page 899.</td>
</tr>
<tr>
<td>dMatchTolerance</td>
<td>Double (ByVal). The matching tolerance setting for the period.</td>
</tr>
</tbody>
</table>

**OpenICPeriod2**

Opens a period for a scenario and year, and specifies the period’s Match/Validate Before Post settings and absolute or percentage matching tolerance.

Syntax

```csharp
<IHsvAdminICM>.OpenICPeriod2 lScenario, lYear, lPeriod, lMVBP, dAcctTol, dTIDTol, dManTol, dAcctPct, dTIDPct, dManPct
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>lYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>IPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>IMVBP</td>
<td>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 899</td>
</tr>
<tr>
<td>dAcctTol</td>
<td>Double (ByVal). Absolute matching tolerance for match by accounts.</td>
</tr>
<tr>
<td>dTIDTol</td>
<td>Double (ByVal). Absolute matching tolerance for match by ID.</td>
</tr>
<tr>
<td>dManTol</td>
<td>Double (ByVal). Absolute matching tolerance for manual match.</td>
</tr>
<tr>
<td>dAcctPct</td>
<td>Double (ByVal). Percentage matching tolerance for match by accounts.</td>
</tr>
<tr>
<td>dTIDPct</td>
<td>Double (ByVal). Percentage matching tolerance for match by IDs.</td>
</tr>
</tbody>
</table>

**SavePeriodsSettings**

*For internal use.*

**UnlockICEntity**

Unlocks an entity for a scenario, year, and period.

**Syntax**

```
<IHsvAdminICM>.UnlockICEntity lScenario, lYear, lPeriod, lEntity
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>IScenario</td>
<td>Long (ByVal). The member ID of the Scenario dimension member.</td>
</tr>
<tr>
<td>IYear</td>
<td>Long (ByVal). The member ID of the Year dimension member.</td>
</tr>
<tr>
<td>IPeriod</td>
<td>Long (ByVal). The member ID of the Period dimension member.</td>
</tr>
<tr>
<td>IEntity</td>
<td>Long (ByVal). The member ID of the Entity dimension member.</td>
</tr>
</tbody>
</table>

**UpdatePeriodSettings**

Updates an open period’s matching tolerance and Match/Validate Before Post settings for a 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**
--- | ---
*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.
*sMVBP* | 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 899.

**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 scenario and year.

Syntax

<IHsvAdminICM>.UpdatePeriodSettings2 lScenario, lYear, lPeriod, sMVBP, dAcctTol, dTIDTol, dManTol, dAcctPct, dTIDPct, dManPct

**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.
*sMVBP* | 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 899.

d*AcctTol* | Double (ByVal). Absolute matching tolerance for match by accounts.

d*TIDTol* | Double (ByVal). Absolute matching tolerance for match by ID.

d*ManTol* | Double (ByVal). Absolute matching tolerance for manual match.

d*AcctPct* | Double (ByVal). Percentage matching tolerance for match by accounts.

d*TIDPct* | Double (ByVal). Percentage matching tolerance for match by IDs.

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 programatically 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 programatically loading and extracting security information, metadata, data, and journals. Use the following objects to set load and extract options:

- For a 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 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 766 and “IHsvLoadExtractOption Interface Properties” on page 767.

**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. 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. 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 these topics:
- “Security Load Options” on page 777
- “Security Extract Options” on page 778
- “Metadata Load Options” on page 783
- “Metadata Extract Options” on page 786
- “Data Load Options” on page 800
- “Data Extract Options” on page 802
- “Journal Load Options” on page 810
- “Journal Extract Options (Unfiltered)” on page 811

Data Type

IHsvLoadExtractOption interface.

Example

Item is used in several examples, including the following:
- “Example for Loading Security” on page 775
- “Example for Extracting Metadata” on page 783
- “Example for Extracting Data” on page 799
- “Example for Loading Journals” on page 807

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 45 on page 126, and are described in the following topics.

To assign IHsvLoadExtractOption object references, use the IHsvLoadExtractOptions interface’s Item property. See “Item” on page 766.

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 these topics:

- “Security Load Options” on page 777
- “Security Extract Options” on page 778
- “Metadata Load Options” on page 783
- “Metadata Extract Options” on page 786
- “Data Load Options” on page 800
- “Data Extract Options” on page 802
- “Journal Load Options” on page 810
- “Journal Extract Options (Unfiltered)” on page 811

Example

`CurrentValue` is used in several examples, including the following examples:

- “Example for Loading Security” on page 775
- “Example for Loading Metadata” on page 782
- “Example for Extracting Metadata” on page 783
- “Example for Extracting Data” on page 799
- “Example for Extracting Journals Without Filters” on page 808

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.
`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.
`MinValue` 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 `IHsvLoadExtractOptions.Item`; see “Item” on page 766.

**OptionID**

Returns the numeric ID of a load or extract option. You can pass this ID to `IHsvLoadExtractOptions.Item`; see “Item” on page 766.

**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.

**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
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 777
- “Example for Extracting Metadata” on page 783
- “Example for Extracting Data” on page 799
- “Example for Extracting Journals Without Filters” on page 808

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 \texttt{LoadOptions} to access load options with the \texttt{Item} property.

**Applicable Objects**

- \texttt{HsvcDataLoad}
- \texttt{HsvJournalLoadACV}
- \texttt{HsvMetadataLoadACV}
- \texttt{HsvSecurityLoadACV}

**Data Type**

\texttt{IHsvLoadExtractOptions} interface.

**Example**

\texttt{LoadOptions} is used in the following examples:

- “Example for Loading Security” on page 775
- “Example for Loading Metadata” on page 782
- “Example for Loading Data” on page 798
- “Example for Loading Journals” on page 807

**Common Load and Extract Methods**

The \texttt{HsvSecurityLoadACV}, \texttt{HsvMetadataLoadACV}, \texttt{HsvcDataLoad}, and \texttt{HsvJournalLoadACV} objects all implement the \texttt{Extract} and \texttt{Load} methods, which load and extract metadata, data, and journals. In addition, these objects all implement the \texttt{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 \texttt{SetSession} and set the \texttt{ExtractOptions} property before calling \texttt{Extract}. The various extract operations may have other prerequisites for calling \texttt{Extract}. For information on how \texttt{Extract} is used with its applicable objects, see the following topics:

- “Extracting Security Information” on page 776
- “Extracting Metadata” on page 782
- “Extracting Data” on page 798
- “Extracting Journals” on page 807
Applicable Objects

- HsvcDataLoad
- HsvJournalLoadACV
- HsvMetadataLoadACV
- HsvSecurityLoadACV

Syntax

```<objectRef>.Extract bstrClientFilename, bstrClientLogFileName```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClientFilename</td>
<td>String (ByVal). The name and path of the file into which the security information, metadata, data, or journals extracted. The path must exist on the client computer. For details on the extracted file’s format, see the Oracle Hyperion Financial Management Administrator’s Guide.</td>
</tr>
<tr>
<td>bstrClientLogFileName</td>
<td>String (ByVal). The name and path of the log file for the extraction operation.</td>
</tr>
</tbody>
</table>

Example

Extract is used in the following examples:

- “Example for Extracting Security” on page 777
- “Example for Extracting Metadata” on page 783
- “Example for Loading Data” on page 798
- “Example for Extracting Journals Without Filters” on page 808

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, the load process fails._

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 781
- “Loading Data” on page 797
- “Loading Journals” on page 807

Applicable Objects

- HsvcDataLoad
- HsvJournalLoadACV
- HsvMetadataLoadACV

**Note:** The HsvSecurityLoadACV object uses a slightly different `Load` method; see “Load” on page 779. In addition, the HsvcDataLoad object offers an alternate method for loading data. This method returns a flag that indicates whether errors were logged; see “Load2” on page 805.

**Syntax**

```csharp
<objectRef>.Load bstrClientFilename, bstrClientLogFileName
```

**Argument** | **Description**
--- | ---
`bstrClientFilename` | String (ByVal). The name and path of the file to be loaded. The path must be valid for the client computer. For details on the file’s required format, see the *Oracle Hyperion Financial Management Administrator’s Guide*. **Tip:** If the metadata load file is in an obsolete file format, an error occurs.

`bstrClientLogFileName` | 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 782
- “Example for Loading Data” on page 798
- “Example for Loading Journals” on page 807

**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 occurs. This rule applies to the common properties; for example, you must call `SetSession` before setting the `ExtractOptions` property.

**Syntax**

```csharp
<objectRef>.SetSession pIHsvSessionUnk
```
**Argument**  
**Description**

`plHsvSessionUnk`  
HsvSession object (ByVal). The HsvSession object that was returned by `HsxClient.OpenApplication` or `HsxClientUI.OpenApplication` when the application was opened.

For information on `HsxClient.OpenApplication`, see “OpenApplication” on page 176; for information on `HsxClientUI.OpenApplication`, see “OpenApplication” on page 183.

**Example**

`SetSession` is used in several examples, including the following examples:

- “Example for Extracting Security” on page 777
- “Example for Loading Metadata” on page 782
- “Example for Extracting Data” on page 799
- “Example for Loading Journals” on page 807

**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 774
- “Loading Security Information” on page 775
- “Extracting Security Information” on page 776
- “Security Load Options” on page 777
- “Security Extract Options” on page 778
- “HsvSecurityLoadACV Object Properties” on page 779
- “HsvSecurityLoadACV Object Methods” on page 779

**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 765.
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 777, and the options available for extracting data are described in “Security Extract Options” on page 778. 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, the load process fails.

Tip: For an example, see “Example for Loading Security” on page 775.

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 777.

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 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.

```vbscript
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 777.

1. 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 778.
   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.

```vba
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 766. For information on CurrentValue, see “CurrentValue” on page 767.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Load Option Information</th>
</tr>
</thead>
</table>
| HSV_SECURITYLOAD_OPT_AUTHENTICATE_USERS | Name property: ValidateUsers  
**Usage:** Determines whether the usernames and user groups specified in a load file authenticated as valid Windows users.  
**Pass to CurrentValue:** Boolean — TRUE to validate users, FALSE to allow users to be loaded without validation.  
**Tip:** If the ValidateUsers option is set to TRUE and the load file contains invalid usernames or user groups, Load’s `pvbWarnings` argument returns TRUE and the invalid user details are noted in the log file. See “Load” on page 779. |
| HSV_SECURITYLOAD_OPT_CLEAR_ALL | Name property: ClearAll  
**Usage:** Determines whether existing security information cleared before the file is loaded.  
**Pass to CurrentValue:** Boolean — TRUE to clear existing information, FALSE otherwise. |
<table>
<thead>
<tr>
<th>Constant</th>
<th>Load Option Information</th>
</tr>
</thead>
</table>
| HSV_SECURITYLOAD_OPT_DELIMITER_CHAR          | Name property: Delimiter  
Usage: Specifies a load file's delimiter.  
Pass to CurrentValue: String — a valid delimiter character. The ValidationList property returns the valid delimiters. |
| HSV_SECURITYLOAD_OPT_ROLE_ACCESS             | Name property: RoleAccess  
Usage: Determines whether role access definitions loaded.  
Pass to CurrentValue: Boolean — TRUE to load role access definitions, FALSE otherwise. |
| HSV_SECURITYLOAD_OPT_SECURITY_CLASS_ACCESS   | Name property: SecurityClassAccess  
Usage: Determines whether security class access definitions loaded.  
Pass to CurrentValue: Boolean — TRUE to load security access definitions, FALSE otherwise. |
| HSV_SECURITYLOAD_OPT_SECURITY_CLASSES        | Name property: SecurityClasses  
Usage: Determines whether security classes loaded. This option is supported only for Classic applications.  
Pass to CurrentValue: Boolean — TRUE to load security classes, FALSE otherwise.  
Caution! If you attempt to load security classes into a Performance Management Architect application, the load process fails. |
| HSV_SECURITYLOAD_OPT_USERS                   | Name property: Users  
Usage: Determines whether usernames and user groups loaded.  
Pass to CurrentValue: Boolean — TRUE to load usernames and user groups, FALSE otherwise. |

## 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 766. For information on CurrentValue, see “CurrentValue” on page 767.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Extract Option Information</th>
</tr>
</thead>
</table>
| HSV_SECURITYEXTRACT_OPT_DELIMITER_CHAR       | Name property: Delimiter  
Usage: Specifies an extract file's delimiter.  
Pass to CurrentValue: String — a valid delimiter character. The ValidationList property returns the valid delimiters. |
### HSV_SECURITYEXTRACT_OPT_ROLE_ACCESS

**Name property:** RoleAccess  
**Usage:** Determines whether role access definitions extracted.  
**Pass to CurrentValue:** Boolean — TRUE to extract role access definitions, FALSE otherwise.

### HSV_SECURITYEXTRACT_OPT_SECURITY_CLASS_ACCESS

**Name property:** SecurityClassAccess  
**Usage:** Determines whether security class access definitions extracted.  
**Pass to CurrentValue:** Boolean — TRUE to extract security access definitions, FALSE otherwise.

### HSV_SECURITYEXTRACT_OPT_SECURITY_CLASSES

**Name property:** SecurityClasses  
**Usage:** Determines whether security classes extracted.  
**Pass to CurrentValue:** Boolean — TRUE to extract security classes, FALSE otherwise.

### HSV_SECURITYEXTRACT_OPT USERS

**Name property:** Users  
**Usage:** Determines whether usernames and user groups 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 770.
- **LoadOptions.** See “LoadOptions” on page 770.

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 771.
- **Load.** See the following topic.
- **SetSession.** See “SetSession” on page 773.

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, the load process fails.

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 775.

Syntax

```<HsvSecurityLoadACV>.Load bstrClientFilename, bstrClientLogFileName, pvbWarnings```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrClientFilename</code></td>
<td>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 Oracle Hyperion Financial Management Administrator’s Guide.</td>
</tr>
<tr>
<td><code>bstrClientLogFileName</code></td>
<td>String (ByVal). The name and path of the log file for the load operation.</td>
</tr>
<tr>
<td><code>pvbWarnings</code></td>
<td>Boolean. Indicates whether warning messages were generated in the log file for a successful load. Returns TRUE if warnings were generated, FALSE otherwise. If this returns TRUE, Load does not generate a trappable error. <strong>Tip:</strong> Warning messages are generated if the ValidateUsers load option is set to TRUE and the load file contains invalid usernames or user groups.</td>
</tr>
</tbody>
</table>

Example

Load is used in the “Example for Loading Security” on page 775.

**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 781
- “Loading Metadata” on page 781
- “Extracting Metadata” on page 782
- “Metadata Load Options” on page 783
- “Metadata Extract Options” on page 786
- “HsvMetadataLoadACV Object Properties” on page 789
- “HsvMetadataLoadACV Object Methods” on page 789
Tip: To load metadata from and extract data to application servers, use the HsvMetadata type library; see “Load” on page 225 and “Extract” on page 203. 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 765.

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 783, and the options available for extracting metadata are described in “Metadata Extract Options” on page 786. 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, the load process fails.

For an example, see “Example for Loading Metadata” on page 782.

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 783.

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.

```vbscript
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_METALOADEX_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 783.

- 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.
Set an object reference to the IHsvLoadExtractOptions interface with the HsvMetadataLoadACV.ExtractOptions property.

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.

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 786.

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. Application settings are not extracted.

```vba
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_METALOADEX_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 766 and “CurrentValue” on page 767.)

Tip: The options for loading dimension members default to TRUE, and the options for clearing previously loaded members default to FALSE.
<table>
<thead>
<tr>
<th>Constant</th>
<th>Load Option Information</th>
</tr>
</thead>
</table>
| HSV_METALOAD_OPT_ACCOUNTS | **Name property:** Accounts  
**Usage:** Specifies whether Account dimension members are loaded.  
**Pass to CurrentValue:** Boolean — TRUE to load accounts, otherwise FALSE. |
| HSV_METALOAD_OPT_ACCOUNTS_SYSTEM | **Name property:** SystemAccounts  
**Usage:** Specifies whether system accounts are loaded.  
**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. By default, this option is set to FALSE. |
| HSV_METALOAD_OPT_APP_SETTINGS | **Name property:** AppSettings  
**Usage:** Specifies whether application settings are loaded.  
**Pass to CurrentValue:** Boolean — TRUE to load application settings, otherwise FALSE. |
| HSV_METALOAD_OPT_CELLTEXT_LABELS | **Name property:** CellTextLabels  
**Usage:** Determines whether cell text labels are loaded.  
**Pass to CurrentValue:** Boolean — TRUE to load cell text labels; otherwise FALSE. |
| HSV_METALOAD_OPT_CHECK_INTEGRITY | **Name property:** CheckIntegrity  
**Usage:** Specifies whether to validate the integrity of the metadata file against the metadata in the current application.  
**Note:** If integrity errors occur, they are noted in the log file and no portion of the file is loaded into the application.  
**Pass to CurrentValue:** Boolean — TRUE to check integrity, otherwise FALSE. |
| HSV_METALOAD_OPT_CLEAR_ACCOUNTS | **Name property:** ClearAll  
**Usage:** Specifies whether previously loaded dimension members are deleted before metadata is loaded.  
**Pass to CurrentValue:** Boolean — TRUE to delete previously loaded dimension members, otherwise FALSE. |
<p>| HSV_METALOAD_OPT_CLEARALL | <strong>Deprecated - use HSV_METALOAD_OPT_CLEARALL</strong> |
| HSV_METALOAD_OPT_CLEAR_CONSOL_METHODS | <strong>Deprecated - use HSV_METALOAD_OPT_CLEARALL</strong> |
| HSV_METALOAD_OPT_CLEAR_CURRENCIES | <strong>Deprecated - use HSV_METALOAD_OPT_CLEARALL</strong> |
| HSV_METALOAD_OPT_CLEAR_CUSTOM1 | <strong>Deprecated - use HSV_METALOAD_OPT_CLEARALL</strong> |
| HSV_METALOAD_OPT_CLEAR_CUSTOM2 | <strong>Deprecated - use HSV_METALOAD_OPT_CLEARALL</strong> |
| HSV_METALOAD_OPT_CLEAR_CUSTOM3 | <strong>Deprecated - use HSV_METALOAD_OPT_CLEARALL</strong> |</p>
<table>
<thead>
<tr>
<th>Constant</th>
<th>Load Option Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSV_METALOAD_OPT_CLEAR_CUSTOM4</td>
<td>Deprecated - use HSV_METALOAD_OPT_CLEARALL</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_CLEAR_ENTITIES</td>
<td>Deprecated - use HSV_METALOAD_OPT_CLEARALL</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_CLEAR_SCENARIOS</td>
<td>Deprecated - use HSV_METALOAD_OPT_CLEARALL</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_CONSOL_METHODS</td>
<td>Name property: ConsolMethods&lt;br&gt;Usage: Specifies whether consolidation methods are loaded.&lt;br&gt;&lt;br&gt;&lt;b&gt;Pass to CurrentValue&lt;/b&gt;: Boolean — TRUE to load consolidation methods, otherwise FALSE.</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_CURRENCIES</td>
<td>Name property: Currencies&lt;br&gt;Usage: Specifies whether currencies are loaded.&lt;br&gt;&lt;br&gt;&lt;b&gt;Pass to CurrentValue&lt;/b&gt;: Boolean — TRUE to load currencies, otherwise FALSE.</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_CUSTOM1</td>
<td>Deprecated - use HSV_METALOAD_OPT_CUSTOMX</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_CUSTOM2</td>
<td>Deprecated - use HSV_METALOAD_OPT_CUSTOMX</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_CUSTOM3</td>
<td>Deprecated - use HSV_METALOAD_OPT_CUSTOMX</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_CUSTOM4</td>
<td>Deprecated - use HSV_METALOAD_OPT_CUSTOMX</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_CUSTOMX</td>
<td>Name property: CustomDims&lt;br&gt;Usage: Specifies whether Custom dimension members are loaded.&lt;br&gt;&lt;br&gt;&lt;b&gt;Pass to CurrentValue&lt;/b&gt;: Array of Boolean — TRUE to load the Custom dimension, otherwise FALSE.</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_DELIMITER_CHAR</td>
<td>Name property: Delimiter&lt;br&gt;Usage: Specifies a load file’s delimiter.&lt;br&gt;&lt;br&gt;&lt;b&gt;Pass to CurrentValue&lt;/b&gt;: String — a valid delimiter character. The ValidationList property returns the valid delimiters.</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_ENTITIES</td>
<td>Name property: Entities&lt;br&gt;Usage: Specifies whether Entity dimension members are loaded.&lt;br&gt;&lt;br&gt;&lt;b&gt;Pass to CurrentValue&lt;/b&gt;: Boolean — TRUE to load entities, otherwise FALSE.</td>
</tr>
<tr>
<td>HSV_METALOAD_OPT_FILE_FORMAT</td>
<td>Name property: FileFormat&lt;br&gt;Usage: Specifies whether the metadata load file is in an ASCII text or XML format.&lt;br&gt;&lt;br&gt;&lt;b&gt;Pass to CurrentValue&lt;/b&gt;: Specify one of the constants in Table 81 on page 789. By default, ASCII text files are loaded.</td>
</tr>
<tr>
<td>Constant</td>
<td>Load Option Information</td>
</tr>
<tr>
<td>----------</td>
<td>-------------------------</td>
</tr>
</tbody>
</table>
| 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 | For internal use. |
| HSV_METALOAD_OPT_MAX | For internal use. |
| HSV_METALOAD_OPT_MIN | For internal use. |
| HSV_METALOAD_OPT_PRESSCAN | **Name property:** Prescan  
**Usage:** Specifies whether a load file is loaded or is merely scanned for syntax accuracy when Load is called.  
**Pass to CurrentValue:** Boolean — TRUE to scan without loading, FALSE to load the metadata. By default, this option is set to FALSE. |
| HSV_METALOAD_OPT_SCENARIOS | **Name property:** Scenarios  
**Usage:** Specifies whether Scenario dimension members are loaded.  
**Pass to CurrentValue:** Boolean — TRUE to load scenarios, otherwise FALSE. |
| HSV_METALOAD_OPT_USE_REPLACE_MODE | **Name property:** UseReplaceMode  
**Usage:** Specifies whether the metadata replaces or is merged with existing metadata.  
**Pass to CurrentValue:** Boolean — TRUE to replace existing metadata, FALSE to merge with existing metadata. This option is set to FALSE by default. |
| HSV_METALOAD_OPT_VALUE | **Name property:** Values  
**Usage:** Specifies whether Value dimension members are loaded.  
**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 LoadSystemMembers option to TRUE. |

**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 766 and “CurrentValue” on page 767.
Tip: The options for extracting dimension members default to TRUE. An exception is the option for system-generated accounts, which defaults to FALSE.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Extract Option Information</th>
</tr>
</thead>
</table>
| HSV_METAEXTRACT_OPT_ACCOUNTS | Name property: Accounts  
Usage: Specifies whether Account dimension members are extracted.  
Pass to `CurrentValue`: Boolean — TRUE to extract accounts, otherwise FALSE. |
| HSV_METAEXTRACT_OPT_ACCOUNTS_SYSTEM | Name property: SystemAccounts  
Usage: Specifies whether system-generated Account dimension members are extracted.  
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. |
| HSV_METAEXTRACT_OPT_APP_SETTINGS | Name property: AppSettings  
Usage: Specifies whether application settings are extracted.  
Pass to `CurrentValue`: Boolean — TRUE to extract application settings, otherwise FALSE. |
| HSV_METAEXTRACT_OPTCELLTEXT_LABELS | Name property: CellTextLabels  
Usage: Determines whether cell text labels are extracted.  
Pass to `CurrentValue`: Boolean — TRUE to extract cell text labels; otherwise FALSE. |
| HSV_METAEXTRACT_OPTCONSOL_METHODS | Name property: ConсолMethods  
Usage: Specifies whether consolidation methods are extracted.  
Pass to `CurrentValue`: Boolean — TRUE to extract consolidation methods, otherwise FALSE. |
| HSV_METAEXTRACT_OPTCURRENCIES | Name property: Currencies  
Usage: Specifies whether currencies are extracted.  
Pass to `CurrentValue`: Boolean — TRUE to extract currencies, otherwise FALSE. |
| HSV_METAEXTRACT_OPTCUSTOM1 | Deprecated - use HSV_METAEXTRACT_OPT_CUSTOMX |
| HSV_METAEXTRACT_OPTCUSTOM2 | Deprecated - use HSV_METAEXTRACT_OPT_CUSTOMX |
| HSV_METAEXTRACT_OPTCUSTOM3 | Deprecated - use HSV_METAEXTRACT_OPT_CUSTOMX |
| HSV_METAEXTRACT_OPTCUSTOM4 | Deprecated - use HSV_METAEXTRACT_OPT_CUSTOMX |
| HSV_METAEXTRACT_OPTCUSTOMX | Name property: CustomDims  
Usage: Specifies whether Custom dimension members are extracted  
Pass to `CurrentValue`: Array of Boolean — TRUE to extract the Custom dimension, otherwise FALSE. |
<table>
<thead>
<tr>
<th>Constant</th>
<th>Extract Option Information</th>
</tr>
</thead>
</table>
| HSV_METAEXTRACT_OPT_DELIMITER_CHAR | **Name property:** Delimiter  
**Usage:** Specifies an extract file's delimiter.  
**Pass to CurrentValue:** String — a valid delimiter character. The ValidationList property returns the valid delimiters. |
| HSV_METAEXTRACT_OPT_ENTITIES | **Name property:** Entities  
**Usage:** Specifies whether Entity dimension members are extracted.  
**Pass to CurrentValue:** Boolean — TRUE to extract entities, otherwise FALSE. |
| HSV_METAEXTRACT_OPT_EXTRACT_SYSTEM_MEMBERS | **Name property:** ExtractSystemMembers  
**Usage:** Specifies whether system members are extracted.  
**Pass to CurrentValue:** Boolean — TRUE to extract 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 extracted. |
| HSV_METAEXTRACT_OPT_FILE_FORMAT | **Name property:** FileFormat  
**Usage:** Specifies whether the metadata extract file is in an ASCII text or XML format.  
**Pass to CurrentValue:** Specify one of the constants in Table 81 on page 789. By default, metadata is extracted to ASCII text files. |
| HSV_METAEXTRACT_OPT_ICP | **Name property:** ICPS  
**Usage:** Specifies whether Intercompany Partner dimension members are extracted.  
**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. |
| HSV_METAEXTRACT_OPT_MAX | For internal use. |
| HSV_METAEXTRACT_OPT_MIN | For internal use. |
| HSV_METAEXTRACT_OPT_PERIODS | **Name property:** Periods  
For internal use. |
| HSV_METAEXTRACT_OPT_SCENARIOS | **Name property:** Scenarios  
**Usage:** Specifies whether Scenario dimension members are extracted.  
**Pass to CurrentValue:** Boolean — TRUE to extract scenarios, otherwise FALSE. |
| HSV_METAEXTRACT_OPT_VALUE | **Name property:** Values  
**Usage:** Specifies whether Value dimension members are extracted.  
**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. |
| HSV_METAEXTRACT_OPT_VIEWS | **Name property:** Views  
For internal use. |
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>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSV_METALOADEX_FORMAT_NATIVE</td>
<td>Metadata is to be loaded or extracted in an ASCII text file.</td>
</tr>
<tr>
<td>HSV_METALOADEX_FORMAT_XML</td>
<td>Metadata is to be loaded or extracted in an XML text file.</td>
</tr>
<tr>
<td>HSV_METALOADEX_FORMAT_XML_COMMONSCHEMA</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

For examples that use the `HSV_METALOADEX_FORMAT_XML` constant, see “Example for Loading Metadata” on page 782 and “Example for Extracting Metadata” on page 783.

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, the load process fails.

- `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 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 50 on page 128, 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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClientFilename</td>
<td>String (ByVal). The name and path of the file into which the rules are extracted. The path must exist on the client computer. For details on the extracted file's format, see the Oracle Hyperion Financial Management Administrator's Guide.</td>
</tr>
<tr>
<td>bstrLogFilename</td>
<td>String (ByVal). The name and path of the log file for the extraction operation.</td>
</tr>
</tbody>
</table>

**Example**

The following example extracts rules from an application.

```vbnet
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
```
**ExtractCalcRulesEx**

Extracts rules from an application into a file on the client computer. For the rules deployed from Oracle Hyperion Calculation Manager, rules can be extracted either in Calculation Manager rules format or Financial Management rules format.

**Note:** You must call `SetSession` before calling `ExtractCalcRulesEx`. `SetSession` points to the application from which rules are being extracted.

**Syntax**

```
<HsvRulesLoadACV>.ExtractCalcRulesEx bstrClientFilename, bstrLogFilename, IExtractType
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrClientFilename</code></td>
<td>String (ByVal). The name and path of the file into which the rules are extracted. The path must exist on the client computer. For details on the extracted file's format, see the Oracle Hyperion Financial Management Administrator's Guide.</td>
</tr>
<tr>
<td><code>bstrLogFilename</code></td>
<td>String (ByVal). The name and path of the log file for the extraction operation.</td>
</tr>
<tr>
<td><code>IExtractType</code></td>
<td>Long (ByVal). Extract Type in which rules have to be extracted. 0 - Financial Management Rules Format (RLE) 1 - Oracle Hyperion Calculation Manager Rules Format (XML)</td>
</tr>
</tbody>
</table>

**Example**

The following example extracts rules from an application in Financial Management rules format.

```vbnet
Dim cHsvRulesLoadACV As HsvRulesLoadACV, bSession As Boolean
Dim IExtractType As Long
Set cHsvRulesLoadACV = New HsvRulesLoadACV
bSession = cHsvRulesLoadACV.SetSession(m_HsvSession, _
HFM_LANGUAGE_INSTALLED)
IExtractType = 0
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

- **bstrClientFilename**: String (ByVal). The name and path of the file into which the member lists extracted. The path must exist on the client computer.
  
  For details on the extracted file’s format, see the *Oracle Hyperion Financial Management Administrator’s Guide*.

- **bstrLogFilename**: String (ByVal). The name and path of the log file for the extraction operation.

### Example

The following example extracts member lists from an application.

```vba
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
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
</table>
| **bstrClientFilename** | 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 *Oracle Hyperion Financial Management Administrator’s Guide*. |
<p>| <strong>bstrLogFilename</strong> | String (ByVal). The name and path of the log file for the load or scan operation. |</p>
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vbScanOnly</td>
<td>Boolean (ByVal). A flag that specifies whether to load or scan the rules file. Pass TRUE to scan, FALSE to load.</td>
</tr>
<tr>
<td>pvbErrorsWereFound</td>
<td>Boolean. Indicates whether LoadCalcRules included any validation errors in the log file. Returns TRUE if errors were found, FALSE otherwise.</td>
</tr>
<tr>
<td>pvbWarningsWereFound</td>
<td>Boolean. Indicates whether LoadCalcRules included any validation warnings in the log file. Returns TRUE if warnings were found, FALSE otherwise.</td>
</tr>
<tr>
<td>pvbInfoWasProvided</td>
<td>Boolean. Indicates whether LoadCalcRules 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.</td>
</tr>
</tbody>
</table>

**Example**

The following example loads rules into an application.

```vbnet
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**

```vbnet
<HsvRulesLoadACV>.LoadCalcRules2 bstrClientFilename, bstrLogFilename, vbScanOnly, vbCheckRefInteg, pbErrorsWereFound, pbWarningsWereFound, pvbInfoWasProvided
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrClientFilename</td>
<td>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 Oracle Hyperion Financial Management Administrator's Guide.</td>
</tr>
<tr>
<td>bstrLogFilename</td>
<td>String (ByVal). The name and path of the log file for the load or scan operation.</td>
</tr>
<tr>
<td>vbScanOnly</td>
<td>Boolean (ByVal). A flag that specifies whether to load or scan the rules file. Pass TRUE to scan, FALSE to load.</td>
</tr>
</tbody>
</table>
**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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>bstrClientFilename</code></td>
<td>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 Oracle Hyperion Financial Management Administrator's Guide.</td>
</tr>
<tr>
<td><code>bstrLogFilename</code></td>
<td>String (ByVal). The name and path of the log file for the load or validate operation.</td>
</tr>
<tr>
<td><code>vbScanOnly</code></td>
<td>Boolean (ByVal). Determines whether <code>LoadMemberListRules</code> loads after validating. Pass TRUE to load after validation, FALSE to validate without loading.</td>
</tr>
<tr>
<td><code>pvbErrorsWereFound</code></td>
<td>Boolean. Indicates whether <code>LoadMemberListRules</code> included any validation errors in the log file. Returns TRUE if errors were found, FALSE otherwise.</td>
</tr>
<tr>
<td><code>pvbWarningsWereFound</code></td>
<td>Boolean. Indicates whether <code>LoadMemberListRules</code> included any validation warnings in the log file. Returns TRUE if warnings were found, FALSE otherwise.</td>
</tr>
<tr>
<td><code>pvbInfoWasProvided</code></td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 occurs.

#### Syntax

```vbnet
<HsvRulesLoadACV>.SetSession(pIHsvSessionUnk, lLanguageID)
```

#### Argument Description

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIHsvSessionUnk</code></td>
<td>HsvSession object (ByVal). The HsvSession object that was returned by <code>HsxClient.OpenApplication</code> or <code>HsxClientUI.OpenApplication</code> when the application was opened.</td>
</tr>
<tr>
<td></td>
<td>For information on <code>HsxClient.UI.OpenApplication</code>, see “OpenApplication” on page 176; for information on <code>HsxClientUI.OpenApplication</code>, see “OpenApplication” on page 183.</td>
</tr>
</tbody>
</table>

| `lLanguageID`     | 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 856. |
|                   | **Tip:** To have error messages display in the installed language, pass the `HFM_LANGUAGE_INSTALLED` 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**

```c
<HsvRulesLoadACV>.SetSessionAndResource(pIHsvSessionUnk, pIUnkResourceManager, lLanguageID)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>pIHsvSessionUnk</td>
<td>HsvSession object (ByVal). The HsvSession object that was returned by HsxClient.OpenApplication or HsxClientUI.OpenApplication when the application was opened. For information on HsxClient.OpenApplication, see “OpenApplication” on page 176; for information on HsxClientUI.OpenApplication, see “OpenApplication” on page 183.</td>
</tr>
<tr>
<td>lLanguageID</td>
<td>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 856. Tip: To display error messages in the installed language, pass the HFM_LANGUAGE_INSTALLED constant.</td>
</tr>
</tbody>
</table>

**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 797
- “Loading Data” on page 797
- “Extracting Data” on page 798
- “Data Load Options” on page 800
- “Data Extract Options” on page 802
- “HsvcDataLoad Object Properties” on page 804
- “HsvcDataLoad Object Methods” on page 805
Tip: To load data from and extract data to application servers, use the HsvData type library; see “Load” on page 376 and “Extract” on page 335. 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 765.

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 800, and the options available for extracting data are described in “Data Extract Options” on page 802. 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 798.

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 82 on page 800.

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.

```vbnet
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 799.

➢ 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.
Set an object reference to the IHsvLoadExtractOptions interface with the HsvcDataLoad.ExtractOptions property.

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.

Optional. To override the defaults for other extract options, specify the values for these options. The available extract options are listed in Table 83 on page 802.

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 = 2012
- 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, "2012")
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 82  Data Load Options

<table>
<thead>
<tr>
<th>Constant</th>
<th>Load Option Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSV_DATALOAD_OPT_ACCUMULATE_WITHIN_FILE</td>
<td><strong>Name property:</strong> Accumulate within file</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether multiple values for the same cell within the load file should be accumulated or allowed to overwrite each other.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> Data for system accounts never gets accumulated.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to CurrentValue:</strong> TRUE to accumulate, FALSE to overwrite.</td>
</tr>
<tr>
<td>HSV_DATALOAD_OPT_DELIMITER_CHAR</td>
<td><strong>Name property:</strong> Delimiter</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies a load file’s delimiter.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to CurrentValue:</strong> String — a valid delimiter character. The ValidationList property returns the valid delimiters.</td>
</tr>
<tr>
<td>Constant</td>
<td>Load Option Information</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HSV_DATALOAD_OPT_DUPLICATES</td>
<td><strong>Name property:</strong> Duplicates</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Sets the data load mode to replace, merge, or accumulate.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> This option corresponds to the Load Mode options Financial Management. See the Oracle Hyperion Financial Management Administrator's Guide.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to</strong> <strong>CurrentValue:</strong> One of the following constants:</td>
</tr>
<tr>
<td></td>
<td>- HSV_DATALOAD_MERGE. Merge mode.</td>
</tr>
<tr>
<td></td>
<td>- HSV_DATALOAD_REPLACE. Replace mode.</td>
</tr>
<tr>
<td></td>
<td>- HSV_DATALOAD_ACCUMULATE. Accumulate mode.</td>
</tr>
<tr>
<td></td>
<td>- HSV_DATALOAD_REPLACEWITHSECURITY. Replace with Security mode.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> For descriptions of these modes, see the description of the HsvData type library's corresponding constants in “Update Mode Constants” on page 320. You should be especially aware of the security-related differences between the Replace and Replace with Security modes.</td>
</tr>
<tr>
<td>HSV_DATALOAD_OPT_FILE_CONTAINS_SHARES</td>
<td><strong>Name property:</strong> Does the file contain shares data</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether the load file contains shares data such as “shares outstanding” or “voting outstanding” or “owned.”</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to</strong> <strong>CurrentValue:</strong> TRUE if the file contains shares data, FALSE otherwise.</td>
</tr>
<tr>
<td>HSV_DATALOAD_OPT_FILE_FORMAT</td>
<td><strong>Name property:</strong> FileFormat</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies the load file format. In this release, only the native file format is supported.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to</strong> <strong>CurrentValue:</strong> The following constant, which represents the only supported option:</td>
</tr>
<tr>
<td></td>
<td>- HSV_DATALOAD_FILE_FORMAT_NATIVE</td>
</tr>
<tr>
<td>HSV_DATALOAD_OPT_LOAD_CALC</td>
<td><strong>For internal use.</strong></td>
</tr>
<tr>
<td>HSV_DATALOAD_OPT_LOG_FILE_APPEND</td>
<td><strong>Name property:</strong> Append to Log File</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether log data is appended to or overwrites the existing log file.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to</strong> <strong>CurrentValue:</strong> Boolean — TRUE to append, FALSE to overwrite.</td>
</tr>
<tr>
<td>HSV_DATALOAD_OPT_FILE_CONTAINS_SUBMISSIONPHASE</td>
<td><strong>Name property:</strong> Does the file contain submission phase data</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies whether the load file contains data for phased submissions.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to</strong> <strong>CurrentValue:</strong> Boolean — TRUE if phased submissions data is being loaded.</td>
</tr>
<tr>
<td>10</td>
<td><strong>Name property:</strong> DecimalChar</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This option is not represented by a constant; 10 is the option's index in the array of data load options.</td>
</tr>
<tr>
<td></td>
<td><strong>Usage:</strong> Specifies the decimal character used in the load file. By default this option is set to an empty string, which indicates the load operation uses the decimal character set as the user's preference.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> To get the user preferences for the decimal and thousands separator characters, use the HsvSystemInfo method GetNumberFormattingUserParameters.</td>
</tr>
<tr>
<td></td>
<td><strong>Pass to</strong> <strong>CurrentValue:</strong> String — if the load file’s decimal character differs from that specified as the user’s preference, specify the load file’s decimal character.</td>
</tr>
</tbody>
</table>
### Constant Load Option Information

<table>
<thead>
<tr>
<th>Constant</th>
<th>Name property: ThousandsChar</th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>This option is not represented by a constant; 11 is the option's index in the array of data load options.</td>
</tr>
<tr>
<td>Note:</td>
<td>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 uses the thousands separator character set as the user's preference.</td>
</tr>
<tr>
<td>Usage:</td>
<td>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.</td>
</tr>
</tbody>
</table>

| HSV_DATALOAD_OPT_MAX | Usage: Points to the object that represents the load option with the highest index in the IHsvLoadExtractOptions collection. |
|----------------------| Pass to CurrentValue: A valid value for the option with the highest index. |
| 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. |

| HSV_DATALOAD_OPT_MIN | Usage: Points to the object that represents the load option with the lowest index in the IHsvLoadExtractOptions collection. |
|----------------------| Pass to CurrentValue: A valid value for the option with the lowest index. |
| 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. |

<table>
<thead>
<tr>
<th>HSV_DATALOAD_OPT_MODE</th>
<th>Name property: Mode</th>
</tr>
</thead>
<tbody>
<tr>
<td>Usage:</td>
<td>Pass to CurrentValue: One of the following constants:</td>
</tr>
<tr>
<td></td>
<td>HSV_DATALOAD_LOAD. Data loaded.</td>
</tr>
<tr>
<td></td>
<td>HSV_DATALOAD_SCAN. Data scanned but not loaded.</td>
</tr>
</tbody>
</table>

The Mode option defaults to scanning, so you must set the option to HSV_DATALOAD_LOAD 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>
<thead>
<tr>
<th>Table 83 Data Extract Options</th>
<th>Constant</th>
<th>Extract Option Information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>HSV_DATAEXTRACT_OPT_ACCOUNT_SUBSET</td>
<td>Name property: Account Subset</td>
</tr>
<tr>
<td></td>
<td>Usage: Specifies the Account dimension members for which data is being extracted.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Pass to CurrentValue: A Long array containing Account dimension member IDs.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>HSV_DATAEXTRACT_OPT_CUSTOM1_SUBSET</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

---

802 Type Libraries for Loading and Extracting Information
<table>
<thead>
<tr>
<th>Constant</th>
<th>Extract Option Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSV_DATAEXTRACT_OPT_CUSTOM2_SUBSET</td>
<td>For internal use.</td>
</tr>
<tr>
<td>HSV_DATAEXTRACT_OPT_CUSTOM3_SUBSET</td>
<td>For internal use.</td>
</tr>
<tr>
<td>HSV_DATAEXTRACT_OPT_CUSTOM4_SUBSET</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>
| HSV_DATAEXTRACT_OPT_DELIMITER_CHAR | Name property: Delimiter  
Usage: Specifies an extract file's delimiter.  
*Pass to CurrentValue:* String — a valid delimiter character. The ValidationList property returns the valid delimiters. |
| HSV_DATAEXTRACT_OPT_ENTITY_SUBSET | Name property: Entity Subset  
Usage: Specifies the child Entity dimension members for which data is being extracted.  
*Pass to CurrentValue:* A Long array containing Entity dimension member IDs. |
| HSV_DATAEXTRACT_OPT_EXTRACT_ALL_DATA | For internal use. |
| HSV_DATAEXTRACT_OPT_EXTRACT_CALC | Name property: Extract Calculated  
Usage: Specifies whether to extract calculated data.  
*Pass to CurrentValue:* TRUE to extract calculated data, FALSE otherwise. |
| HSV_DATAEXTRACT_OPT_EXTRACT_FILE_APPEND | For internal use. |
| HSV_DATAEXTRACT_OPT_ICP_SUBSET | For internal use. |
| HSV_DATAEXTRACT_OPT_LOG_FILE_APPEND | Name property: Append to log file  
Usage: Specifies whether log data is appended to or overwrites the existing log file.  
*Pass to CurrentValue:* TRUE to append, FALSE to overwrite. |
| HSV_DATAEXTRACT_OPT_LOG_FILE_NAME | For internal use. |
| HSV_DATAEXTRACT_OPT_MAX | Usage: Points to the object that represents the extract option with the highest index in the IHsvLoadExtractOptions collection.  
*Pass to CurrentValue:* A valid value for the option with the highest index.  
*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. |
### HsvcDataLoad Object Properties

The HsvcDataLoad object provides the following properties:

- **ExtractOptions.** See “ExtractOptions” on page 770.
- **LoadOptions.** See “LoadOptions” on page 770.
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 for internal use only.

**DMELoad**

*For internal use.*

**LoadAsync**

*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**

**bstrClientFileName**

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 *Oracle Hyperion Financial Management Administrator’s Guide*.

**bstrClientLogFileName**

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.
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 806
- “Loading Journals” on page 807
- “Extracting Journals” on page 807
- “Journal Load Options” on page 810
- “Journal Extract Options (Unfiltered)” on page 811
- “Journal Extract Options (Filtered)” on page 812
- “HsvJournalLoadACV Object Properties” on page 814
- “HsvJournalLoadACV Object Methods” on page 815

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 765.

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 84 on page 811, and the options available for extracting journals are described in Table 85 on page 811. 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 807.

1. 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. See Table 84 on page 811. 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, they are also loaded.

### Example for Loading Journals

The following example loads the journals and templates in a load file.

```vbscript
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 Scenario, Year, and Period dimension members. See "Extracting Journals without Filters" on page 808.

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 types and statuses. See "Extracting Journals with Filters" on page 809.

**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 808.

1. 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 85 on page 811.

5. Optional. To override the defaults for other extract options, specify the values for these options. If you do not override any defaults, journals extracted, and standard and recurring templates are not extracted.


**Example for Extracting Journals Without Filters**

The following example extracts journals for the Actual scenario in the year 2012.

**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.

```vbnet
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 "2012"
Set cOpt = cOptions.Item(HSV_JOURNALEXTRACT_OPT_YEAR)
lYear = GetMemberID(DIMENSIONYEAR, "2012")
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 810.

➢ 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 86 on page 813.

6 Optional. To override the defaults for other extract options, specify the values for these options. If you do not override any defaults, journals extracted, standard and recurring templates are not extracted, and no filtering criteria is 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.
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 2012 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`.

```vba
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 "2012"
Set cOpt = cOptions.Item(HSV_JOURNALEXTRACT_EX_OPT_YEAR)
lYear = GetMemberID(DIMENSIONYEAR, "2012")
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 `IHsvLoadExtractOptionCurrentValue`.
Table 84  Journal Load Options

<table>
<thead>
<tr>
<th>Constant</th>
<th>Load Option Information</th>
</tr>
</thead>
</table>
| HSV_JOURNALLOAD_OPT_DELIMITER_CHAR            | Name property: Delimiter
|                                               | Usage: Specifies a load file’s delimiter.                                               |
|                                               | Pass to CurrentValue: String — a valid delimiter character. The ValidationList property returns the valid delimiters. |
| HSV_JOURNALLOAD_OPT_MAX                      | Usage: Points to the object that represents the extract option with the highest index in the IHsvLoadExtractOptions collection. |
|                                               | Pass to CurrentValue: A valid value for the option with the highest index.               |
|                                               | 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. |
| HSV_JOURNALLOAD_OPT_MIN                      | Usage: Points to the object that represents the extract option with the lowest index in the IHsvLoadExtractOptions collection. |
|                                               | Pass to CurrentValue: A valid value for the option with the lowest index.               |
|                                               | 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. |

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 85  Journal Extract Options (Unfiltered)

<table>
<thead>
<tr>
<th>Constant</th>
<th>Extract Option Information</th>
</tr>
</thead>
</table>
| HSV_JOURNALEXTRACT_OPT_DELIMITER_CHAR         | Name property: Delimiter
|                                               | Usage: Specifies an extract file’s delimiter.                                           |
|                                               | Pass to CurrentValue: String — a valid delimiter character. The ValidationList property returns the valid delimiters. |
| HSV_JOURNALEXTRACT_OPT_SCENARIO              | Name property: Scenario
|                                               | Usage: Specifies the Scenario dimension member for which journals are being extracted.   |
|                                               | Pass to CurrentValue: A Long containing the Scenario dimension member ID.                |
| HSV_JOURNALEXTRACT_OPT_PERIOD                | Name property: Period
|                                               | Usage: Specifies the Period dimension member for which journals are being extracted.    |
|                                               | Pass to CurrentValue: A Long containing the Period dimension member ID.                  |
### 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>
<thead>
<tr>
<th>Constant</th>
<th>Extract Option Information</th>
</tr>
</thead>
</table>
| HSV_JOURNAL_EXTRACT_EX_OPT_DELIMITER_CHAR | **Name property:** Delimiter  
**Usage:** Specifies an extract file’s delimiter.  
**Pass to CurrentValue:** String — a valid delimiter character. The ValidationList property returns the valid delimiters. |
| HSV_JOURNAL_EXTRACT_EX_OPT_SCENARIO | **Name property:** Scenario  
**Usage:** Specifies the Scenario dimension member for which journals are being extracted.  
**Pass to CurrentValue:** A Long containing the Scenario dimension member ID. |
| HSV_JOURNAL_EXTRACT_EX_OPT_PERIOD | **Name property:** Period  
**Usage:** Specifies the Period dimension members for which journals are being extracted.  
**Pass to CurrentValue:** A Long array that contains the IDs of the Period dimension members. |
| HSV_JOURNAL_EXTRACT_EX_OPT_YEAR | **Name property:** Year  
**Usage:** Specifies the Year dimension member for which journals are being extracted.  
**Pass to CurrentValue:** A Long containing the Year dimension member ID. |
| HSV_JOURNAL_EXTRACT_EX_OPT_ENTITIES | **Name property:** Entity  
**Usage:** Specifies the child Entity dimension members for which journals are being extracted.  
**Pass to CurrentValue:** A Long array that contains the IDs of the child Entity dimension members. |
| HSV_JOURNAL_EXTRACT_EX_OPT_PARENTS | **Name property:** Parent  
**Usage:** Specifies the parent Entity dimension members for which journals are being extracted.  
**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. |
| HSV_JOURNAL_EXTRACT_EX_OPT_VALUES | **Name property:** Value  
**Usage:** Specifies the Value dimension members for which journals are being extracted.  
**Pass to CurrentValue:** A Long array that contains the IDs of the Value dimension members. |
| HSV_JOURNAL_EXTRACT_EX_OPT_REGULAR | **Name property:** Regular  
**Usage:** Specifies whether journals are being extracted.  
**Pass to CurrentValue:** Boolean - TRUE to extract journals, otherwise FALSE. This option defaults to TRUE. |
| HSV_JOURNAL_EXTRACT_EX_OPT_STANDARD | **Name property:** Standard  
**Usage:** Specifies whether standard templates are being extracted.  
**Pass to CurrentValue:** Boolean - TRUE to extract standard templates, otherwise FALSE. This option defaults to FALSE. |
| HSV_JOURNAL_EXTRACT_EX_OPT_RECURRING | **Name property:** Recurring  
**Usage:** Specifies whether recurring templates are being extracted.  
**Pass to CurrentValue:** Boolean - TRUE to extract recurring templates, otherwise FALSE. This option defaults to FALSE. |
<table>
<thead>
<tr>
<th>Constant</th>
<th>Extract Option Information</th>
</tr>
</thead>
</table>
| HSV_JOURNALEXTRACT_EX_OPT_FILTER_BALANCE     | **Name property:** Balance Type  
**Usage:** Specifies the balance type filter for the extraction.  
**Pass to CurrentValue:** A Long array containing the balance types to use as filtering criteria. Valid values are represented by the HFMConstants type library constants listed in “Balance Type Constants” on page 873. |
| HSV_JOURNALEXTRACT_EX_OPT_FILTER_GROUP       | **Name property:** Group  
**Usage:** Specifies the journal group filter for the extraction.  
**Pass to CurrentValue:** A String array containing the journal group filtering criteria. You can use the pound character ( # ) as a wildcard. |
| HSV_JOURNALEXTRACT_EX_OPT_FILTER_LABEL       | **Name property:** Label  
**Usage:** Specifies the journal label filter for the extraction.  
**Pass to CurrentValue:** A String array containing the journal label filtering criteria. You can use the pound character ( # ) as a wildcard. |
| HSV_JOURNALEXTRACT_EX_OPT_FILTER_STATUS      | **Name property:** Status  
**Usage:** Specifies the journal status filter for the extraction.  
**Pass to CurrentValue:** A Long array containing the status filtering criteria. Valid values are represented by the HFMConstants type library constants listed in “Journal Status Constants” on page 875. |
| HSV_JOURNALEXTRACT_EX_OPT_FILTER_TYPE        | **Name property:** Type  
**Usage:** Specifies the journal type filter for the extraction.  
**Pass to CurrentValue:** A Long array containing the journal type filtering criteria. Valid values are represented by the HFMConstants type library constants listed in “Journal Type Constants” on page 876. |
| HSV_JOURNALEXTRACT_EX_OPT_MAX                | **Usage:** Points to the object that represents the extract option with the highest index in the IHsvLoadExtractOptions collection.  
**Pass to CurrentValue:** A valid value for the option with the highest index.  
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. |
| HSV_JOURNALEXTRACT_EX_OPT_MIN                | **Usage:** Points to the object that represents the extract option with the lowest index in the IHsvLoadExtractOptions collection.  
**Pass to CurrentValue:** A valid value for the option with the lowest index.  
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. |

**HsvJournalLoadACV Object Properties**

The HsvJournalLoadACV object provides the following properties:

- **ExtractOptions.** See “ExtractOptions” on page 770.
- **ExtractOptionsEx.**
- **LoadOptions.** See “LoadOptions” on page 770.
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 809.

**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 809.

**Note:** To extract journals without applying filtering criteria, use `Extract`.

**Syntax**

```
<HsvJournalLoadACV>.ExtractEx bstrServerFilename, bstrServerLogFilename
```

**Argument**

**Description**

- `bstrServerFilename`
  
  String. The name and path of the file into which the journals extracted. The path must exist on the client computer. For details on the extracted file's format, see the *Oracle Hyperion Financial Management Administrator's Guide*. 

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>bstrServerLogFilename</td>
<td>String, The name and path of the log file for the extraction.</td>
</tr>
</tbody>
</table>
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 dimension’s tab. 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 55 on page 131, 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 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**

- **lDim**: Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 857.
- **bCheck**: 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.

```vbnet
' cFormPOV represents an initialized HsvPOVSelection control
cFormPOV.CheckAllItemsUsingSelectedList DIMENSIONENTITY, False
```

---

**EnableDimension**

Displays or hides the tab for a 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**

- **lDim**: Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 857.
- **bEnable**: 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.

```vbnet
' 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
End For
```
End If
Next
' cSession is a previously set HsvSession instance
cFormPOV.Initialize cSession

GetCheckedItems

Returns the member IDs of the selected members on a 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

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lDim</td>
<td>Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMCollections type library constants listed in &quot;Dimension ID Constants&quot; on page 857.</td>
</tr>
<tr>
<td>pvaralItemIDs</td>
<td>Variant. Returns an array of member IDs for the selected members. The array items are Longs.</td>
</tr>
<tr>
<td>pvaralParentIDs</td>
<td>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 pvaralItemIDs argument’s array. The array items are Longs.</td>
</tr>
</tbody>
</table>

Note: If an Entity member has no parent, the HFMCollections type library constant MEMBERNOTUSED is returned. For information on this constant, see “Dimension Member Constants” on page 858. 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 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 GetOBPIInfo.

Syntax

<HSVPOVSelection>.GetListInfo lDim, plListID, plListTopMemberID
### Argument | Description
--- | ---
`lDim` | Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 857.

`plListID` | Long. Returns the ID of the member list.

**Tip:** To get a member list name from its ID, use the IHsvTreeInfo method `GetMemberListName`.

`plListTopMemberID` | Long. Returns the member ID of the specified top member.

If no top member is specified, the HFMConstants type library constant `MEMBERNOTUSED` is returned. For information on this constant, see "Dimension Member Constants" on page 858.

### Example

The following function returns the name of the member list that is selected on a dimension’s tab. The name is obtained by passing the list ID to IHsvTreeInfo.GetMemberListName.

```vbs
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 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

```vbs
<HsvPOVSelection>.GetMember lDim, plItemID, plParentID
```

### Argument | Description
--- | ---
`lDim` | Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 857.

`plItemID` | 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

plParentID  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 858.

Example

The following function returns the name of the member that is selected on a 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 dimension’s tab.

Tip: To get the member IDs of members for which check boxes are selected, use GetCheckedItems.

Syntax

<HsvPOVSelection>.GetNumCheckedItems lDim, plNumCheckedItems

Argument Description

lDim  Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 857.

plNumCheckedItems  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```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pvarbEnabled</code></td>
<td>Boolean. Indicates whether only active Entity members are displayed. Returns TRUE if only active entities are displayed, FALSE otherwise.</td>
</tr>
<tr>
<td><code>plScenario</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>plYear</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>plPeriod</code></td>
<td>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.</td>
</tr>
</tbody>
</table>

**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 pIHsvSessionUnk```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pIHsvSessionUnk</code></td>
<td>HsvSession object (ByVal). The HsvSession instance for the application.</td>
</tr>
</tbody>
</table>

**Tip:** HsvSession object references are returned by the `OpenApplication` methods of the HsxClient and HsxClientUI 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 dimension.

Syntax

```<HsvPOVSelection>.SelectDimension lDim```

**Argument Description**

- `lDim` Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 857.

**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```
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lDim</code></td>
<td>Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 857.</td>
</tr>
<tr>
<td><code>varalItemIDs</code></td>
<td>Long array (ByVal). The member IDs of the members for which check boxes are being selected or cleared.</td>
</tr>
<tr>
<td><code>varalParentIDs</code></td>
<td>Variant (ByVal). The member IDs for the parents of the <code>varalItemIDs</code> argument’s entities, if the <code>lDim</code> 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 <code>varalItemIDs</code> argument’s array.</td>
</tr>
<tr>
<td><code>bCheck</code></td>
<td>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.</td>
</tr>
<tr>
<td><code>bClearAllExistingCheckedItems</code></td>
<td>Boolean (ByVal). A flag that specifies whether to clear check boxes that had been selected before the call to <code>SetCheckedItems</code>. Pass TRUE to clear previously selected check boxes, FALSE otherwise.</td>
</tr>
</tbody>
</table>

**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.

```vbscript
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 dimension, as well as the top member of the list to display.

**Syntax**

```<HsvPOVSelection>.SetListInfo lDim, lListID, lListTopMemberID```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lDim</td>
<td>Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMC+Constants type library constants listed in “Dimension ID Constants” on page 857.</td>
</tr>
<tr>
<td>lListID</td>
<td>Long (ByVal). The ID of the member list to display.</td>
</tr>
<tr>
<td>lListTopMemberID</td>
<td>Long (ByVal). The member ID of the member to display. If you do not want to specify a top member, pass the HFMC+Constants type library constant MEMBERNOTUSED. For information on this constant, see “Dimension Member Constants” on page 858.</td>
</tr>
</tbody>
</table>

**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`.

```vbs
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 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

IDim Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in "Dimension ID Constants" on page 857.

lItemID Long (ByVal). The member ID of the member to be selected.

IParentID Long (ByVal). For the Entity dimension, the member ID of the lItemID 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

varbEnabled Boolean (ByVal). Specifies whether the control displays only active Entity members. Pass TRUE to display only active entities, FALSE otherwise.

When TRUE 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.

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.

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 specified language.

You can use only languages into which the Financial Management release was localized. If a release is not 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**

```vbscript
<HsvPOVSelection>.SetResourceAndLanguage pIUnkHsvResourceManager, lHFMLanguage
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lHFMLanguage</code></td>
<td>Long (ByVal). The language ID. To obtain IDs of the languages into which the release was localized, use the HsvResourceManager method <code>GetAvailableLanguages</code>.</td>
</tr>
</tbody>
</table>

**ShowCheckBoxes**

Configures a 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**

```vbscript
<HsvPOVSelection>.ShowCheckboxes lDim, bShow
```
Argument | Description
--- | ---
`lDim` | Long (ByVal). The ID of the dimension. Dimension IDs are represented by the HFMConstants type library constants listed in “Dimension ID Constants” on page 857.
`bShow` | 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 56 on page 132, and are described in detail in the following topics.

**CheckBoxChanged**

Launches 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, and a flag indicating whether the check box was selected or cleared.

**Note:** This event happens only when the user selects one check box. If the user clicks the button for selecting or clearing all check boxes, the MultiCheckBoxChanged event occurs.

**Syntax**

```
<HsvPOVSelection>_CheckBoxChanged(lDim As Long, lItemID As Long, lParentID As Long, bNewCheckState As Boolean)
```

**Argument | Description**
--- | ---
`lDim` | 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 857.
`lItemID` | Long (ByVal). The member ID of the member.
`lParentID` | Long (ByVal). The parent of the `lItemID` argument’s member.
For dimensions other than Entity, or for Entity members without parents, the system passes the HFMConstants type library constant MEMBERNOTUSED to this argument. For information on this constant, see “Dimension Member Constants” on page 858.
`bNewCheckState` | 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**

Launches 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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lDim</td>
<td>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 857.</td>
</tr>
</tbody>
</table>

**MultiCheckBoxChanged**

Launches 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**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lDim</td>
<td>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 857.</td>
</tr>
<tr>
<td>bNewCheckState</td>
<td>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.</td>
</tr>
</tbody>
</table>
**SelectionChanged**

Launches 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 occurs the first time that a dimension’s tab is selected.

**Syntax**

```vbscript
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lDim</code></td>
<td>Long (ByVal). The ID of the member’s dimension. Dimension IDs are represented by the HFMConstants type library constants listed in &quot;Dimension ID Constants&quot; on page 857.</td>
</tr>
<tr>
<td><code>lPrevItemID</code></td>
<td>Long (ByVal). The member ID of the previously selected member.</td>
</tr>
<tr>
<td><code>lPrevParentID</code></td>
<td>Long (ByVal). The parent of the <code>lPrevItemID</code> 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 &quot;Dimension Member Constants&quot; on page 858.</td>
</tr>
<tr>
<td><code>lNewItemID</code></td>
<td>Long (ByVal). The member ID of the currently selected member.</td>
</tr>
<tr>
<td><code>lNewParentID</code></td>
<td>Long (ByVal). The parent of the <code>lNewItemID</code> 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.</td>
</tr>
</tbody>
</table>

**Example**

The following event procedure prints the labels of the previously and currently selected members to Visual Basic’s Immediate window. The I HSVTreeInfo method `GetItemID` is used to get the member labels from the member IDs passed to the event procedure.

```vbscript
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
```
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 these benefits:

- **Error message localization** – Error messages are translated into the same languages as Financial Management.

- **Centralized location** – Previous 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 such as “invalid account”, a dynamic error message specifies the name of the invalid account. For example, if the invalid account name was Sales1, the message might read:

  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.

- **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. If a release is localized, error messages are localized.

To use the HsvResourceManager type library, you must reference HsvResourceManager.dll in your project. The HsvResourceManager type library contains only the HsvResourceManager object.

To obtain an HsvResourceManager object reference in Visual Basic or VBScript, use the New keyword as shown in the following example:

```vbnet
Dim cResourceManager As HsvResourceManager
Set cResourceManager = New HsvResourceManager
```

**Caution!** In Visual Basic, do not use CreateObject, or an error occurs.

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.

1. **To handle errors:**
   - **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 `GetResourceStringFromHR`.

   All of these methods return error messages that you can display to users. However, `GetFormattedError` and `GetFormattedErrorWithLineFeed` also return more technical information that may help debug an issue.

   **Note:** These methods take language IDs. If the release is localized, error messages are localized, otherwise error messages are returned in Financial Management’s default language. You can enumerate the languages for which localized error message strings are available with `GetAvailableLanguages`.

---

834  Error Handling and the HsvResourceManager Type Library
**System Message Detail Strings**

Some methods return strings of technical information for system messages such as errors. These strings contain a uniquely identifying Error Reference Number, followed by various fields of information. These fields are delimited by semicolons, and each has a label followed by a colon, as in the following example:

```
Error Reference Number: (219EB33B-BF50-11D6-A43E-0000863DCCF1)
Num: 0x800415c6; Type: 1; DTime: 9/3/12 12:20:10 PM; Svr: GSZABO1; File:
C\hsxServerImpl.cpp; Line: 1842; Ver: 3.0.0.196;
```

The following table describes the fields:

<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Num</td>
<td>Error number in hexadecimal form.</td>
</tr>
<tr>
<td>Type</td>
<td>For internal use.</td>
</tr>
<tr>
<td>DTime</td>
<td>TimeStamp of the error.</td>
</tr>
<tr>
<td>Svr</td>
<td>Machine name of the computer on which the error occurred.</td>
</tr>
<tr>
<td>File</td>
<td>Name of the source code file to which the error applies.</td>
</tr>
<tr>
<td>Line</td>
<td>Line number in the source code file to which the error applies.</td>
</tr>
<tr>
<td>Ver</td>
<td>Version number of the DLL to which the error applies.</td>
</tr>
</tbody>
</table>

**GetAvailableLanguages**

Returns arrays that identify the languages for which resources are available. Resources are available only for those languages into which Financial Management is localized.

GetAvailableLanguages returns an array of language IDs and an array of language names. These arrays have a one-to-one correspondence.

**Syntax**

```
<HsvResourceManager>.GetAvailableLanguages pvarastrLanguagesIDs, pvarastrLanguages
```

**Argument**

`pvarastrLanguagesIDs` Variant. Returns an array of language IDs that identify the languages for which resources are available.

**Tip:** The HFMConstants enumeration tagHFM_LANGUAGES represents language IDs that are valid for all releases; see "Supported Language Constants" on page 856.

`pvarastrLanguages` 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**

```csharp
<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**

```csharp
<HsvResourceManager>.GetCurrentVersionInUserDisplayFormat()
```

**Return Value**

Variant. Returns the version number as a String subtype.

**GetFormattedDateTime**

Converts a 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**

```csharp
<HsvResourceManager>.GetFormattedDateTime(lFormatType, dDateTime)
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>lFormatType</code></td>
<td>Long (ByVal). The format of the string to return. Valid values are represented by the HFMConstants type library constants listed in &quot;Date and Time Format Constants&quot; on page 903.</td>
</tr>
</tbody>
</table>
**DateTime**

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 timestamp.

```vba
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**

```vba
<HsvResourceManager>.GetFormattedDateTimeForLanguage(lLanguageId, dDateTime, bDateValueOnly, bTimeValueOnly)
```

**Argument**

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lLanguageId</td>
<td>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; see “Supported Language Constants” on page 856.</td>
</tr>
<tr>
<td>dDateTime</td>
<td>Double (ByVal). The timestamp.</td>
</tr>
<tr>
<td>bDateValueOnly</td>
<td>Long (ByVal). TRUE to return only the date value.</td>
</tr>
<tr>
<td>bTimeValueOnly</td>
<td>Long (ByVal). TRUE to return only the time value.</td>
</tr>
</tbody>
</table>

**Return Value**

String

**GetFormattedError**

Returns two strings in a specified 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 line feed character to the technical details string, use `GetFormattedErrorWithLineFeed`.

### Syntax

```vbscript
<HsvResourceManager>.GetFormattedError lLanguageId, hr, bstrXMLError, bstrDefaultError, pvarbstrFormattedError, pvarbstrTechnicalError
```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lLanguageId</td>
<td>Long (ByVal). The language ID. You can obtain the IDs of a Financial Management release's valid languages with <code>GetAvailableLanguages</code>.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> The HFMConstants enumeration <code>tagHFM_LANGUAGES</code> represents language IDs that are valid for all releases; see “Supported Language Constants” on page 856.</td>
</tr>
<tr>
<td>hr</td>
<td>Long (ByVal). The HRESULT that identifies the error.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> In Visual Basic and VBScript, you can get the HRESULT with the <code>Number</code> property of the <code>Err</code> object.</td>
</tr>
<tr>
<td>bstrXMLError</td>
<td>String (ByVal). The XML string that describes the error.</td>
</tr>
<tr>
<td></td>
<td><strong>Tip:</strong> In Visual Basic and VBScript, you can get the XML string with the <code>Description</code> property of the <code>Err</code> object.</td>
</tr>
<tr>
<td>bstrDefaultError</td>
<td>String (ByVal). A default error message string. If the error is not generated by Financial Management, the specified default string is returned by the <code>pvarbstrFormattedError</code> argument.</td>
</tr>
<tr>
<td></td>
<td>The way this works is that <code>GetFormattedError</code> first looks for a resource string that corresponds to the XML string passed in the <code>bstrXMLError</code> 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.</td>
</tr>
<tr>
<td>pvarbstrFormattedError</td>
<td>Variant. Returns a simple description of the error. This is text that can be displayed to the user.</td>
</tr>
<tr>
<td>pvarbstrTechnicalError</td>
<td>Variant. Returns detailed technical information regarding the error. See “System Message Detail Strings” on page 835.</td>
</tr>
</tbody>
</table>

### 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`.

```vbscript
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
    For i = LBound(vaNames) To UBound(vaNames)
        If lLanguageID = vaNames(i) Then
            vUserError = vaIDs(i)
            Exit For
        End If
    Next i
    lLanguageID = HFM_LANGUAGE_DEFAULT
    GetFormattedError lLanguageID, Err.Number, Err.Description, , , , , ,
End Function
```

838  Error Handling and the HsvResourceManager Type Library
End If
Next i
cResourceManager.GetFormattedError lLanguageID, Err.Number, _
    Err.Description, "Unknown Error", vUserError, vTechError
getHFMEError = vUserError
End Function

GetFormattedErrorWithLineFeed

Returns two strings in a specified 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

lLanguageId Long (ByVal). Identifies the language for which the strings returned. Pass one of the HFMConstants type library constants listed in “Supported Language Constants” on page 856.

hr Long (ByVal). The HRESULT that identifies the error.
In Visual Basic and VBScript, you can get the HRESULT with the Number property of the Err object.

bstrXMLError String (ByVal). The XML string that describes the error.
In Visual Basic and VBScript, you can get the XML string with the Description property of the Err object.

bstrDefaultError String (ByVal). A default error message string. If the error is not generated by Financial Management, the specified default string is returned by the pvarbstrFormattedError argument.
The way this works is that GetFormattedError first looks for a resource string that corresponds to the XML string passed in the bstrXMLError argument. If no matching resource string is found, GetFormattedError looks for a resource string that corresponds to the passed HRESULT. If no matching resource string is found, then the default string is returned.

bstrLineFeed String (ByVal). The line feed character to apply to the pvarbstrTechnicalError argument’s string.

pvarbstrFormattedError Variant. Returns a simple description of the error. This is text that can be displayed to the user.

pvarbstrTechnicalError Variant. Returns detailed technical information regarding the error. See “System Message Detail Strings” on page 835.

GetFormattedResourceString

For internal use.
GetHelpDirectoryForLanguageID

Returns the name of the subdirectory that contains the Financial Management help system for a specified language.

Syntax

\(<\text{HsvResourceManager}>\).\text{GetHelpDirectoryForLanguageID}(\text{lLanguageId})

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{lLanguageId}</td>
<td>Long (ByVal). Identifies the language for which the subdirectory returned. Pass one of the HFMConstants type library constants listed in &quot;Supported Language Constants&quot; on page 856.</td>
</tr>
</tbody>
</table>

Return Value

Variant. Returns the subdirectory name.

GetHFMLanguageIdFromUserLanguages

Returns the Financial Management ID for the language identified by a language code. The method returns IDs only of languages for which resources were translated.

Syntax

\(<\text{HsvResourceManager}>\).\text{GetHFMLanguageIdFromUserLanguages}(\text{bstrLanguages})

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{bstrLanguages}</td>
<td>String (ByVal). The two-letter language code. To obtain language codes, use \text{GetLanguageCountryCodeFromLanguageId}.</td>
</tr>
</tbody>
</table>

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 \text{GetAvailableLanguages}.

GetLanguageCountryCodeFromLanguageId

Returns the two-letter language code for a language ID.

Syntax

\(<\text{HsvResourceManager}>\).\text{GetLanguageCountryCodeFromLanguageId}(\text{lLanguageId})
<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lLanguageId</td>
<td>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; see “Supported Language Constants” on page 856.</td>
</tr>
</tbody>
</table>

Return Value

String. Returns the language code.

**GetLocaleIdFromLanguageId**

Returns a Windows locale ID for the Financial Management language ID.

Syntax

```<HsvResourceManager>.GetLocaleIdFromLanguageId(lLanguageId)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lLanguageId</td>
<td>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; see “Supported Language Constants” on page 856.</td>
</tr>
</tbody>
</table>

Return Value

Long

**GetResourceString**

*For internal use.*

**GetResourceStringFromHR**

Returns the resource string for an HRESULT in the specified language.

Syntax

```<HsvResourceManager>.GetResourceStringFromHR(lLanguageId, hr)```

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>lLanguageId</td>
<td>Long (ByVal). Identifies the language for which the error message string returned. Pass one of the HFMConstants type library constants listed in “Supported Language Constants” on page 856.</td>
</tr>
<tr>
<td>hr</td>
<td>Long (ByVal). The HRESULT that identifies the error.</td>
</tr>
</tbody>
</table>

**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.

**GetUserDisplayDataTimeFormats**

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 903.

Tip: To format a timestamp, use `GetFormattedDateTime`.

Syntax

```
<HsvResourceManager>.GetUserDisplayDataTimeFormats pvaravarDateTimeFormats
```

Argument | Description
--- | ---
`pvaravarDateTimeFormats` | Variant. Returns an array of strings that describe the formats.

**GetUserLanguageFromHFMLanguageId**

Returns the two-letter language code for the language represented by a Financial Management language ID.

Syntax

```
<HsvResourceManager>.GetUserLanguageFromHFMLanguageId lLanguageId, bstrUserLanguage
```

Argument | Description
--- | ---
`lLanguageId` | 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; see “Supported Language Constants” on page 856.

`bstrUserLanguage` | String (ByRef). The user language.

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
```

---

842 Error Handling and the HsvResourceManager Type Library
Argument Description

pvarbstrDateSeparator  Variant. Returns a string containing the date separator character.

pvarbstrDateOrder  Variant. Returns a string containing the short date format.

GetWindowsDateFormatForLocale

Returns the Windows date format for the locale.

Syntax

```
<HsvResourceManager>.GetWindowsDateFormatForLocale lLocaleId, pvarbstrDateSeparator, pvarbstrDateOrder
```

Argument Description

lLocaleId  Long (ByVal). Windows locale ID. Can be obtained by calling GetLocaleId from LanguageId.

pvarbstrDateSeparator  Variant array (ByVal). Returns a string containing the date separator character.

pvarbstrDateOrder  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 880.

Example

Initialize is used in the example for GetFormattedError.

Error Message Lookup Utility

You can use a utility to look up error messages associated with Financial Management’s HRESULTs. The utility is installed in Financial Management’s Consultant Utilities directory, and has a file name of HFMErrorLookup.exe.

After launching the utility, take the following steps to look up an error number.
To look up an error number:

1. Perform one of the following steps:
   a. To look up a severity HRESULT, select the **SEVERITY ERROR** check box. Severity HRESULTs are prefixed with 0x8004.
   b. To look up a success HRESULT, clear the **SEVERITY ERROR** check box. Success HRESULTs are prefixed with 0x0004.

   **Note:** This check box determines the HRESULT prefix that displays in the read-only HRESULT text box to the left of the editable text box.

2. In the editable **HRESULT** text box, enter the portion of the HRESULT that is not displayed in the read-only text box.

3. From the **Language** drop-down list, select the language in which you want to view the error message.

4. Click **OK** to view the message for the specified HRESULT.
This appendix lists constants for Financial Management. The DLL for the HFMConstants type library is HFMConstants.dll.

Note: The HFMConstants type library also contains enumerations that are not documented in this appendix. These are undocumented because they are used internally, and not by the methods and properties of the exposed object model.

Metadata-Related Constants

The HFMConstants type library includes numerous metadata-related constants. The following categories of metadata constants are provided:

- “Account Dimension Constants” on page 846
- “Custom Dimension Constants” on page 850
- “Consolidation Method Constants” on page 852
- “Currency Attribute Constants” on page 853
Account Dimension Constants

The HFMConstants type library provides the following types of constants for the Account dimension:

- “Account Attribute Constants” on page 846
- “System Account Constants” on page 848
- “Account Member List Constants” on page 848
- “Account Type Constants” on page 849

Account Attribute Constants

Note: All attribute ID constants (beginning with “ATTRIB_”) are now deprecated. The new constants begin with “ATTRIBEX_”. For example, the tagACCOUNT_ATTRIBS enum is replaced by tagACCOUNT_ATTRIBS_EX enum. Each attribute id constant in the enum will begin with ATTRIBEX_.

Note: Also note that for Account dimension and Application Setting attributes, special consideration must be given for attributes that are related to custom dimensions. To use the new constant, you must OR in the custom dimension number. Examples:

VB, VBScript

(ATRIBEX_ACCOUNT_CUSTOM_TOP_MEMBER OR 1)

C++

(ATRIBEX_ACCOUNT_CUSTOM_TOP_MEMBER | 1).

The following constants represent attributes of accounts.
<table>
<thead>
<tr>
<th>Attribute Constant</th>
<th>New Attribute Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_ACCOUNT_CALC_ATTRIBUTE</td>
<td>ATTRIBEX_ACCOUNT_CALC_ATTRIBUTE</td>
<td>CalcAttribute attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_CUSTOM1_TOP_MEMBER</td>
<td>ATTRIBEX_ACCOUNT_CUSTOM1_TOP_MEMBER</td>
<td>Custom TopMember attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_CUSTOM2_TOP_MEMBER</td>
<td>ATTRIBEX_ACCOUNT_CUSTOM2_TOP_MEMBER</td>
<td>Custom TopMember attribute. (VB, VB Script): ATTRIBEX_ACCOUNT_CUSTOM2_TOP_MEMBER OR 1 (C++ Script): ATTRIBEX_ACCOUNT_CUSTOM2_TOP_MEMBER</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_CUSTOM3_TOP_MEMBER</td>
<td>ATTRIBEX_ACCOUNT_CUSTOM3_TOP_MEMBER</td>
<td>Custom TopMember attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_CUSTOM4_TOP_MEMBER</td>
<td>ATTRIBEX_ACCOUNT_CUSTOM4_TOP_MEMBER</td>
<td>Custom TopMember attribute. (VB, VB Script): ATTRIBEX_ACCOUNT_CUSTOM4_TOP_MEMBER OR 1 (C++): ATTRIBEX_ACCOUNT_CUSTOM4_TOP_MEMBER</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_ENABLE_CUSTOM1_AGGR</td>
<td>ATTRIBEX_ACCOUNT_ENABLE_CUSTOM1_AGGR</td>
<td>EnableCustomAggr attribute. (VB, VB Script): ATTRIBEX_ACCOUNT_ENABLE_CUSTOM1_AGGR OR 1 (C++): ATTRIBEX_ACCOUNT_ENABLE_CUSTOM1_AGGR</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_ENABLE_CUSTOM2_AGGR</td>
<td>ATTRIBEX_ACCOUNT_ENABLE_CUSTOM2_AGGR</td>
<td>EnableCustomAggr attribute. (VB, VB Script): ATTRIBEX_ACCOUNT_ENABLE_CUSTOM2_AGGR OR 1 (C++): ATTRIBEX_ACCOUNT_ENABLE_CUSTOM2_AGGR</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_ENABLE_CUSTOM3_AGGR</td>
<td>ATTRIBEX_ACCOUNT_ENABLE_CUSTOM3_AGGR</td>
<td>EnableCustomAggr attribute. (VB, VB Script): ATTRIBEX_ACCOUNT_ENABLE_CUSTOM3_AGGR OR 1 (C++): ATTRIBEX_ACCOUNT_ENABLE_CUSTOM3_AGGR</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_ENABLE_CUSTOM4_AGGR</td>
<td>ATTRIBEX_ACCOUNT_ENABLE_CUSTOM4_AGGR</td>
<td>EnableCustomAggr attribute. (VB, VB Script): ATTRIBEX_ACCOUNT_ENABLE_CUSTOM4_AGGR OR 1 (C++): ATTRIBEX_ACCOUNT_ENABLE_CUSTOM4_AGGR</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_ENABLE_DATA_AUDIT</td>
<td>ATTRIBEX_ACCOUNT_ENABLE_DATA_AUDIT</td>
<td>EnableDataAudit attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_ICP_TOP_MEMBER</td>
<td>ATTRIBEX_ACCOUNT_ICP_TOP_MEMBER</td>
<td>ICP TopMember attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_IS_CALCULATED</td>
<td>ATTRIBEX_ACCOUNT_IS_CALCULATED</td>
<td>IsCalculated attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_IS_CONsolidated</td>
<td>ATTRIBEX_ACCOUNT_IS_CONsolidated</td>
<td>IsConsolidated attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_IS_ICP</td>
<td>ATTRIBEX_ACCOUNT_IS_ICP</td>
<td>IsICP attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_MAX</td>
<td>ATTRIBEX_ACCOUNT_MAX</td>
<td>Represents the last account attribute; use this to loop through the account attributes.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_MIN</td>
<td>ATTRIBEX_ACCOUNT_MIN</td>
<td>Represents the first account attribute; use this to loop through the account attributes.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_NUM_ATTRIBS</td>
<td>ATTRIBEX_ACCOUNT_NUM_ATTRIBS</td>
<td>Represents the total number of account attributes.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_NUM_DECIMAL_PLACES</td>
<td>ATTRIBEX_ACCOUNT_NUM_DECIMAL_PLACES</td>
<td>NumDecimalPlaces attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_PLUG_ACCOUNT</td>
<td>ATTRIBEX_ACCOUNT_PLUG_ACCOUNT</td>
<td>PlugAcct attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_SECURITY_CLASS</td>
<td>ATTRIBEX_ACCOUNT_SECURITY_CLASS</td>
<td>SecurityClass attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_TYPE</td>
<td>ATTRIBEX_ACCOUNT_TYPE</td>
<td>AccountType attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_USERDEF1</td>
<td>ATTRIBEX_ACCOUNT_USERDEF1</td>
<td>UserDefined1 attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_USERDEF2</td>
<td>ATTRIBEX_ACCOUNT_USERDEF2</td>
<td>UserDefined2 attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_USERDEF3</td>
<td>ATTRIBEX_ACCOUNT_USERDEF3</td>
<td>UserDefined3 attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_USES_LINE_ITEMS</td>
<td>ATTRIBEX_ACCOUNT_USES_LINE_ITEMS</td>
<td>UsesLineItems attribute.</td>
</tr>
<tr>
<td>ATTRIB_ACCOUNT_XBRL_TAGS</td>
<td>ATTRIBEX_ACCOUNT_XBRL_TAGS</td>
<td>XBRLTags attribute.</td>
</tr>
</tbody>
</table>
**System Account Constants**

The following constants represent system accounts.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBERACTIVESTATUS</td>
<td>Active account.</td>
</tr>
<tr>
<td>MEMBERCONSOL1</td>
<td>Consol1 account.</td>
</tr>
<tr>
<td>MEMBERCONSOL2</td>
<td>Consol2 account.</td>
</tr>
<tr>
<td>MEMBERCONSOL3</td>
<td>Consol3 account.</td>
</tr>
<tr>
<td>MEMBERCONSOLIDATIONMETHOD</td>
<td>Method account.</td>
</tr>
<tr>
<td>MEMBERDIRECTPERCENTOWNERSHIP</td>
<td>DOWN account.</td>
</tr>
<tr>
<td>MEMBERFIRSTSYSTEMACCOUNT</td>
<td>Represents the first system account; use this to loop through the system accounts.</td>
</tr>
<tr>
<td>MEMBERLASTSYSTEMACCOUNT</td>
<td>Represents the last system account; use this to loop through the system accounts.</td>
</tr>
<tr>
<td>MEMBERPERCENTCONSOLIDATION</td>
<td>PCON account.</td>
</tr>
<tr>
<td>MEMBERPERCENTCONTROL</td>
<td>PCTRL account.</td>
</tr>
<tr>
<td>MEMBERPERCENTOWNERSHIP</td>
<td>POWN account.</td>
</tr>
<tr>
<td>MEMBERSHARESOUTSTANDING</td>
<td>SharesOutstanding account.</td>
</tr>
<tr>
<td>MEMBERSHARESOWNED</td>
<td>SharesOwned account.</td>
</tr>
<tr>
<td>MEMBERSHARESPERCENTOWNED</td>
<td>Shares%Owned account.</td>
</tr>
<tr>
<td>MEMERVOTINGOUTSTANDING</td>
<td>VotingOutstanding account.</td>
</tr>
<tr>
<td>MEMERVOTINGOWNED</td>
<td>VotingOwned account.</td>
</tr>
<tr>
<td>MEMERVOTINGPERCENTOWNED</td>
<td>Voting%Owned account.</td>
</tr>
</tbody>
</table>

**Account Member List Constants**

The following constants represent system-defined member lists for accounts.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNT_MEMBER_LIST_ALL_HIERARCHY</td>
<td>Hierarchy member list.</td>
</tr>
<tr>
<td>ACCOUNT_MEMBER_LIST_ANCESTORS</td>
<td>Ancestors member list.</td>
</tr>
<tr>
<td>ACCOUNT_MEMBER_LIST_BASE</td>
<td>Base member list.</td>
</tr>
<tr>
<td>ACCOUNT_MEMBER_LIST_CHILDREN</td>
<td>Children member list.</td>
</tr>
</tbody>
</table>
### Account Type Constants

The following constants represent the valid account types.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ACCOUNTTYPE_ASSET</td>
<td>Asset account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_BALANCE</td>
<td>Balance account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_BALANCERECURRING</td>
<td>BalanceRecurring account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_CURRENCYRATE</td>
<td>CurrencyRate account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_DYNAMIC</td>
<td>Dynamic account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_EXPENSE</td>
<td>Expense account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_FLOW</td>
<td>Flow account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_GROUPLABEL</td>
<td>Grouplabel account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_INCOME</td>
<td>This constant was replaced by the ACCOUNTTYPE_REVENUE constant.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_LIABILITY</td>
<td>Liability account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_MAX</td>
<td>Represents the last account type; use this to loop through the account types.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_MIN</td>
<td>Represents the first account type; use this to loop through the account types.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_NONFINANCIAL</td>
<td>Nonfinancial account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_REVENUE</td>
<td>Revenue account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_SYSTEM</td>
<td>System account.</td>
</tr>
<tr>
<td>ACCOUNTTYPE_TEXT</td>
<td>Text account.</td>
</tr>
</tbody>
</table>

**Note:** This account type is no longer used, but the constant exists for backwards compatibility purposes.

**Note:** This account type is no longer used, but the constant exists for backwards compatibility purposes.

---

**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.
**Custom Dimension Constants**

The HFMConstants type library provides the following types of constants for the Custom dimensions:

- “Custom 1 and 2 Dimension Member Lists” on page 850
- “Custom Dimension Attributes” on page 851

**Custom 1 and 2 Dimension Member Lists**

The following constants represent system-defined member lists for the Custom 1 and Custom 2 dimensions.

**Table 92**  
tagC1ANDC2MEMBERLISTS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOM1_MEMBER_LIST_CONSOL_METHODS</td>
<td>ConsolMethods member list.</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> This applies only to the Custom 1 dimension.</td>
</tr>
<tr>
<td>CUSTOM1AND2_MEMBER_LIST_ALL_HIERARCHY</td>
<td>Hierarchy member list.</td>
</tr>
<tr>
<td>CUSTOM1AND2_MEMBER_LIST ANCESTORS</td>
<td>Ancestors member list.</td>
</tr>
<tr>
<td>CUSTOM1AND2_MEMBER_LIST_BASE</td>
<td>Base member list.</td>
</tr>
<tr>
<td>CUSTOM1AND2_MEMBER_LIST_CHILDREN</td>
<td>Children member list.</td>
</tr>
<tr>
<td>CUSTOM1AND2_MEMBER_LIST_CURRENCIES</td>
<td>Currencies member list.</td>
</tr>
<tr>
<td>CUSTOM1AND2_MEMBER_LIST_DESCENDANTS</td>
<td>Descendants member list.</td>
</tr>
<tr>
<td>CUSTOM1AND2_MEMBER_LIST_PARENTS</td>
<td>Parents member list.</td>
</tr>
<tr>
<td>NUM_PREDEFINED_CUSTOM1_MEMBER_LISTS</td>
<td>Represents the number of system-defined member lists for the Custom 1 dimension.</td>
</tr>
<tr>
<td>NUM_PREDEFINED_CUSTOM2_MEMBER_LISTS</td>
<td>Represents the number of system-defined member lists for the Custom 2 dimension.</td>
</tr>
</tbody>
</table>

**Custom 3 and 4 Dimension Member Lists**

The tagC3ANDC4MEMBERLISTS enum is now deprecated, and is replaced by the tagCUSTOMMEMBERLISTS enum. The tagC1ANDC2MEMBERLISTS enum is still valid, since it contains member list IDs that are specific to Custom1 and Custom2 dimensions.

**Table 93**  
tagC3ANDC4MEMBERLISTS Enumeration

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOM3AND4_MEMBER_LIST_ALL_HIERARCHY</td>
<td>CUSTOMMEMBERLISTS_ALL_HIERARCHY</td>
<td>Hierarchy member list.</td>
</tr>
<tr>
<td>CUSTOM3AND4_MEMBER_LIST_ANCESTORS</td>
<td>CUSTOMMEMBERLISTS_ANCESTORS</td>
<td>Ancestors member list.</td>
</tr>
</tbody>
</table>
### Deprecated Constant

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CUSTOM3AND4_MEMBER_LIST_BASE</td>
<td>CUSTOMMEMBERLISTS_BASE</td>
<td>Base member list.</td>
</tr>
<tr>
<td>CUSTOM3AND4_MEMBER_LIST_CHILDREN</td>
<td>CUSTOMMEMBERLISTS_CHILDREN</td>
<td>Children member list.</td>
</tr>
<tr>
<td>CUSTOM3AND4_MEMBER_LIST_DESCENDANTS</td>
<td>CUSTOMMEMBERLISTS_DESCENDANTS</td>
<td>Descendants member list.</td>
</tr>
<tr>
<td>CUSTOM3AND4_MEMBER_LIST_PARENTS</td>
<td>CUSTOMMEMBERLISTS_PARENTS</td>
<td>Parents member list.</td>
</tr>
<tr>
<td>NUM_PREDEFINED_CUSTOM3AND4_MEMBER_LISTS</td>
<td>NUM_PREDEFINED_CUSTOMMEMBERLISTS</td>
<td>Represents the number of system-defined member lists for the Custom 3 and 4 dimensions.</td>
</tr>
</tbody>
</table>

### Custom Dimension Attributes

The following constants represent attributes of the Custom dimensions.

**Table 94  tagCUSTOM_ATTRIBS Enumeration**

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_CUSTOM_IS_CALCULATED</td>
<td>ATTRIBEX_CUSTOM_IS_CALCULATED</td>
<td>IsCalculated attribute.</td>
</tr>
<tr>
<td>ATTRIB_CUSTOM_MAX</td>
<td>ATTRIBEX_CUSTOM_MAX</td>
<td>Represents the last Custom dimension attribute; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_CUSTOM_MIN</td>
<td>ATTRIBEX_CUSTOM_MIN</td>
<td>Represents the first Custom dimension attribute; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_CUSTOM_NUM_ATTRIBS</td>
<td>ATTRIBEX_CUSTOM_NUM_ATTRIBS</td>
<td>Represents the total number of Custom dimension attributes.</td>
</tr>
<tr>
<td>ATTRIB_CUSTOM_SECURITY_CLASS</td>
<td>ATTRIBEX_CUSTOM_SECURITY_CLASS</td>
<td>SecurityClass attribute.</td>
</tr>
<tr>
<td>ATTRIB_CUSTOM_SWITCH_SIGN</td>
<td>ATTRIBEX_CUSTOM_SWITCH_SIGN</td>
<td>SwitchSignForFlow attribute.</td>
</tr>
<tr>
<td>ATTRIB_CUSTOM_SWITCH_TYPE</td>
<td>ATTRIBEX_CUSTOM_SWITCH_TYPE</td>
<td>SwitchTypeForFlow attribute.</td>
</tr>
<tr>
<td>ATTRIB_CUSTOM_USERDEF1</td>
<td>ATTRIBEX_CUSTOM_USERDEF1</td>
<td>UserDefined1 attribute</td>
</tr>
<tr>
<td>ATTRIB_CUSTOM_USERDEF2</td>
<td>ATTRIBEX_CUSTOM_USERDEF2</td>
<td>UserDefined2 attribute.</td>
</tr>
<tr>
<td>ATTRIB_CUSTOM_USERDEF3</td>
<td>ATTRIBEX_CUSTOM_USERDEF3</td>
<td>UserDefined3 attribute.</td>
</tr>
</tbody>
</table>
Consolidation Method Constants

The HFMConstants type library provides the following types of constants for consolidation methods:

- “Consolidation Method Attribute Constants” on page 852
- “Consolidation Methods: Control Attribute Constants” on page 852
- “Consolidation Methods: ToPercentControlComp Attribute Constants” on page 853
- “Consolidation Methods: PercentConsol Attribute Constants” on page 853

Consolidation Method Attribute Constants

The following constants represent the attributes of consolidation methods.

Table 95  tagCONSOLMETHOD_ATTRIBS Enumeration

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_CONSOLMETH_CONTROL</td>
<td>ATTRIBEX_CONSOLMETH_CONTROL</td>
<td>Control attribute.</td>
</tr>
<tr>
<td>ATTRIB_CONSOLMETH_ISHOLDINGMETHOD</td>
<td>ATTRIBEX_CONSOLMETH_ISHOLDINGMETHOD</td>
<td>IsHoldingMethod attribute.</td>
</tr>
<tr>
<td>ATTRIB_CONSOLMETH_MAX</td>
<td>ATTRIBEX_CONSOLMETH_MAX</td>
<td>Represents the last attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_CONSOLMETH_MIN</td>
<td>ATTRIBEX_CONSOLMETH_MIN</td>
<td>Represents the first attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_CONSOLMETH_NUM_ATTRIBS</td>
<td>ATTRIBEX_CONSOLMETH_NUM_ATTRIBS</td>
<td>Represents the total number of attributes in this enumeration.</td>
</tr>
<tr>
<td>ATTRIB_CONSOLMETH_PCTCONSOL</td>
<td>ATTRIBEX_CONSOLMETH_PCTCONSOL</td>
<td>PercentConsol attribute.</td>
</tr>
<tr>
<td>ATTRIB_CONSOLMETH_TOPCTCTRL</td>
<td>ATTRIBEX_CONSOLMETH_TOPCTCTRL</td>
<td>ToPercentControl attribute.</td>
</tr>
<tr>
<td>ATTRIB_CONSOLMETH_TOPCTCTRLPOP</td>
<td>ATTRIBEX_CONSOLMETH_TOPCTCTRLPOP</td>
<td>ToPercentControlComp attribute.</td>
</tr>
<tr>
<td>ATTRIB_CONSOLMETH_USEDBYCALCROUTINE</td>
<td>ATTRIBEX_CONSOLMETH_USEDBYCALCROUTINE</td>
<td>UsedByCalcRoutine attribute.</td>
</tr>
</tbody>
</table>

Consolidation Methods: Control Attribute Constants

The following constants represent the valid values for the Control attribute of consolidation methods.
Table 96  tagCONSOLMETHOD_CONTROL Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSOLMETHOD_CONTROL_EMPTY</td>
<td>The attribute does not have a value.</td>
</tr>
<tr>
<td>CONSOLMETHOD_CONTROL_FULL</td>
<td>Full.</td>
</tr>
<tr>
<td>CONSOLMETHOD_CONTROL_LIMITED</td>
<td>Limited.</td>
</tr>
<tr>
<td>CONSOLMETHOD_CONTROL_NO</td>
<td>No.</td>
</tr>
</tbody>
</table>

Consolidation Methods: ToPercentControlComp Attribute Constants

The following constants represent the valid values for the ToPercentControlComp attribute of consolidation methods.

Table 97  tagCONSOLMETHOD_TOPCTCTRLOPS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSOLMETHOD_TOPCTCTRLOP_LESSTHAN</td>
<td>Less-than (&lt;) operator.</td>
</tr>
<tr>
<td>CONSOLMETHOD_TOPCTCTRLOP_LESSTHANOREQUAL</td>
<td>Less-than-or-equal-to (&lt;=) operator.</td>
</tr>
<tr>
<td>CONSOLMETHOD_TOPCTCTRLOP_NONE</td>
<td>The attribute does not have a value.</td>
</tr>
</tbody>
</table>

Consolidation Methods: PercentConsol Attribute Constants

The following constants represent system-defined values for the PercentConsol attribute.

Table 98  tagCONSOLPCTCONSTANTS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSOLPCT_POWN</td>
<td>POWN system account.</td>
</tr>
<tr>
<td>CONSOLPCT_POWNMIN</td>
<td>POWNMIN system account.</td>
</tr>
<tr>
<td>CONSOLPCT_UNSPECIFIED</td>
<td>No value is specified for this attribute.</td>
</tr>
</tbody>
</table>

Currency Attribute Constants

The following constants represent currency attributes.

Table 99  tagCURRENCY_ATTRIBS Enumeration

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_CURRENCY_ICT_DISPLAY</td>
<td>ATTRIBEX_CURRENCY_ICT_DISPLAY</td>
<td>DisplayInICT attribute.</td>
</tr>
<tr>
<td>Deprecated Constant</td>
<td>New Constant</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ATTRIB_CURRENCY_MAX</td>
<td>ATTRIBEX_CURRENCY_MAX</td>
<td>Represents the last attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_CURRENCY_MIN</td>
<td>ATTRIBEX_CURRENCY_MIN</td>
<td>Represents the first attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_CURRENCY_NUM_ATTRIBS</td>
<td>ATTRIBEX_CURRENCY_NUM_ATTRIBS</td>
<td>Represents the total number of attributes in this enumeration.</td>
</tr>
<tr>
<td>ATTRIB_CURRENCY_SCALE</td>
<td>ATTRIBEX_CURRENCY_SCALE</td>
<td>Scale attribute.</td>
</tr>
<tr>
<td>ATTRIB_CURRENCY_TRANSOP</td>
<td>ATTRIBEX_CURRENCY_TRANSOP</td>
<td>TranslationOperator attribute.</td>
</tr>
</tbody>
</table>

**Entity Dimension Constants**

The HFMConstants type library provides the following types of constants for entities:

- “Entity Attribute Constants” on page 854
- “Entity Member List Constants” on page 855

**Entity Attribute Constants**

The following constants represent attributes of Entity dimension members.

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_EAP_SECURITY_CLASS</td>
<td>ATTRIBEX_EAP_SECURITY_CLASS</td>
<td>SecurityAsPartner attribute.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_ALLOW_ADJS_FROM_CHILDREN</td>
<td>ATTRIBEX_ENTITY_ALLOW_ADJS_FROM_CHILDREN</td>
<td>AllowAdjFromChildren attribute.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_ALLOW_ADJUSTMENTS</td>
<td>ATTRIBEX_ENTITY_ALLOW_ADJUSTMENTS</td>
<td>AllowAdjs attribute.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_DEFAULT_VALUE_ID</td>
<td>ATTRIBEX_ENTITY_DEFAULT_VALUE_ID</td>
<td>DefCurrency attribute.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_HOLDING_COMPANY</td>
<td>ATTRIBEX_ENTITY_HOLDING_COMPANY</td>
<td>HoldingCompany attribute.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_IS_ICP</td>
<td>ATTRIBEX_ENTITY_IS_ICP</td>
<td>IsICP attribute.</td>
</tr>
</tbody>
</table>
## Deprecated Constant

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_ENTITY_MAX</td>
<td>ATTRIBEX_ENTITY_MAX</td>
<td>Represents the last attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_MIN</td>
<td>ATTRIBEX_ENTITY_MIN</td>
<td>Represents the first attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_NUM_ATTRIBS</td>
<td>ATTRIBEX_ENTITY_NUM_ATTRIBS</td>
<td>Represents the total number of attributes in this enumeration.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_SECURITY_CLASS</td>
<td>ATTRIBEX_ENTITY_SECURITY_CLASS</td>
<td>SecurityClass attribute.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_USERDEF1</td>
<td>ATTRIBEX_ENTITY_USERDEF1</td>
<td>UserDefined1 attribute.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_USERDEF2</td>
<td>ATTRIBEX_ENTITY_USERDEF2</td>
<td>UserDefined2 attribute.</td>
</tr>
<tr>
<td>ATTRIB_ENTITY_USERDEF3</td>
<td>ATTRIBEX_ENTITY_USERDEF3</td>
<td>UserDefined3 attribute.</td>
</tr>
</tbody>
</table>

## Entity Member List Constants

The following constants represent member lists that are system-defined for the Entity dimension.

### Table 101  tagENTITYMEMBERLISTS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENTITY_MEMBER_LIST_ADJUSTMENT_ENTITIES</td>
<td>Adjustment Entities member list.</td>
</tr>
<tr>
<td>ENTITY_MEMBER_LIST_ALL_HIERARCHY</td>
<td>Hierarchy member list.</td>
</tr>
<tr>
<td>ENTITY_MEMBER_LIST_ANCESTORS</td>
<td>Ancestors member list.</td>
</tr>
<tr>
<td>ENTITY_MEMBER_LIST_BASE</td>
<td>Base member list.</td>
</tr>
<tr>
<td>ENTITY_MEMBER_LIST_CHILDREN</td>
<td>Children member list.</td>
</tr>
<tr>
<td>ENTITY_MEMBER_LIST_DESCENDANTS</td>
<td>Descendants member list.</td>
</tr>
<tr>
<td>ENTITY_MEMBER_LIST_PARENT_ADJUSTMENT_ENTITIES</td>
<td>Parent Adjustment Entities member list.</td>
</tr>
<tr>
<td>ENTITY_MEMBER_LIST_PARENTS</td>
<td>Parents member list.</td>
</tr>
<tr>
<td>NUM_PREDEFINED_ENTITY_MEMBER_LISTS</td>
<td>Represents the number of system-defined member lists for the Entity dimension.</td>
</tr>
</tbody>
</table>

## View Dimension Constants

The HFMConstants type library provides these types of constants for the View dimension:

- “Frequencies Constants” on page 856
Frequencies Constants

The following constants represent attributes of frequencies.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBERLOWESTFREQAVAILABLE</td>
<td>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.</td>
</tr>
<tr>
<td>MEMBERYEARLY</td>
<td>Yearly frequency.</td>
</tr>
</tbody>
</table>

View Dimension Member Constants

The following constants represent members of the View dimension.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBERLOWESTVIEWAVAILABLE</td>
<td>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.</td>
</tr>
<tr>
<td>MEMBERPERIODIC</td>
<td>Periodic member.</td>
</tr>
<tr>
<td>MEMBERSCENARIOVIEW</td>
<td>&lt;Scenario View&gt; member.</td>
</tr>
<tr>
<td>MEMBERYTD</td>
<td>YTD member.</td>
</tr>
</tbody>
</table>

Supported Language Constants

The following constants represent languages supported by Financial Management.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFM_LANGUAGE_DEFAULT</td>
<td>Represents the default language.</td>
</tr>
<tr>
<td>HFM_LANGUAGE_ENGLISH</td>
<td>English.</td>
</tr>
<tr>
<td>HFM_LANGUAGE_INSTALLED</td>
<td>Represents the installed language.</td>
</tr>
<tr>
<td>HFM_LANGUAGE_USER_DEFAULT</td>
<td>Represents the user's default language.</td>
</tr>
<tr>
<td>HFM_NO_LANGUAGE</td>
<td>Used to get non-translated resource strings. This constant can also be used in cases where a language is not applicable, such as with FindMatchingMembersFromHierarchyWildCard.</td>
</tr>
</tbody>
</table>
Dimension-Related Constants

The HFMConstants type library provides the following types of dimension-related constants:

- “Dimension ID Constants” on page 857
- “Constant Representing All Dimensions” on page 858
- “Dimension Member Constants” on page 858

Dimension ID Constants

The following constants represent internal IDs of Financial Management’s dimensions.

The old tagHFMDIMENSIONS constants are now deprecated. They are replaced by the tagHFMDIMENSIONS2 constants. Note that there are no longer separate dimension ID constants for each custom dimension. Instead, you can use DIMID_CUSTOMBASE as the starting point for custom dimension IDs. For example, DIMID_CUSTOMBASE would be Custom1, DIMID_CUSTOMBASE+1 would be Custom2, etc.

Table 105  tagHFMDIMENSIONS Enumeration

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DIMENSIONACCOUNT</td>
<td>DIMID_ACCOUNT</td>
<td>Account dimension.</td>
</tr>
<tr>
<td>DIMENSIONCUSTOM1</td>
<td>DIMID_CUSTOMBASE</td>
<td>Custom 1 dimension.</td>
</tr>
<tr>
<td>DIMENSIONCUSTOM2</td>
<td>DIMID_CUSTOMBASE+1</td>
<td>Custom 2 dimension.</td>
</tr>
<tr>
<td>DIMENSIONCUSTOM3</td>
<td>DIMID_CUSTOMBASE+2</td>
<td>Custom 3 dimension.</td>
</tr>
<tr>
<td>DIMENSIONCUSTOM4</td>
<td>DIMID_CUSTOMBASE+3</td>
<td>Custom 4 dimension.</td>
</tr>
<tr>
<td>DIMENSIONENTITY</td>
<td>DIMID_ENTITY</td>
<td>Entity dimension.</td>
</tr>
<tr>
<td>DIMENSIONICP</td>
<td>DIMID_ICP</td>
<td>Intercompany Partner dimension.</td>
</tr>
<tr>
<td>DIMENSIONPARENT</td>
<td>DIMID_PARENT</td>
<td>Parent entity, Entity dimension.</td>
</tr>
<tr>
<td>DIMENSIONPERIOD</td>
<td>DIMID_PERIOD</td>
<td>Period dimension.</td>
</tr>
<tr>
<td>DIMENSIONSCENARIO</td>
<td>DIMID_SCENARIO</td>
<td>Scenario dimension.</td>
</tr>
<tr>
<td>DIMENSIONVALUE</td>
<td>DIMID_VALUE</td>
<td>Value dimension.</td>
</tr>
<tr>
<td>DIMENSIONVIEW</td>
<td>DIMID_VIEW</td>
<td>View dimension.</td>
</tr>
<tr>
<td>DIMENSIONYEAR</td>
<td>DIMID_YEAR</td>
<td>Year dimension.</td>
</tr>
<tr>
<td>NUMDIMENSIONS</td>
<td>NUMDIMENSIONS</td>
<td>Represents the number of dimensions.</td>
</tr>
<tr>
<td>NUMDIMENSIONSCLUDINGPARENT</td>
<td>NUMDIMENSIONSINCLUDINGPARENT</td>
<td>Represents the number of dimensions, counting the parent entity as a dimension.</td>
</tr>
<tr>
<td>DIMENSION_LBOUND</td>
<td>DIMID_LBOUND</td>
<td>Represents the first dimension in the set of dimensions; use this to loop through the dimensions.</td>
</tr>
<tr>
<td>Deprecated Constant</td>
<td>New Constant</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>DIMENSION_UBOUND</td>
<td>DIMID_UBOUND</td>
<td>Represents the last dimension in the set of dimensions; use this to loop through the dimensions.</td>
</tr>
</tbody>
</table>

**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 857, which represent specific dimensions).

**Dimension Member Constants**

The following constants can be useful when working with dimension members.

**Table 106  `tagPOVDEFAULTS` Enumeration**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBERALL</td>
<td>Specifies that you want to use all members of a dimension.</td>
</tr>
<tr>
<td>MEMBERALLEXCEPTNONE</td>
<td>Used for the Intercompany Partner dimension to ignore the [ICP None] member.</td>
</tr>
<tr>
<td>MEMBERANYONE</td>
<td>Specifies that you want to use any one member of a dimension.</td>
</tr>
<tr>
<td>MEMBERDEFAULT</td>
<td>Represents the default member of a dimension.</td>
</tr>
<tr>
<td>MEMBERNONE</td>
<td>Some dimensions, such as the Account dimension, have a member named [None].</td>
</tr>
<tr>
<td>MEMBERNOTUSED</td>
<td>Specifies that no specific member applies.</td>
</tr>
</tbody>
</table>

**Intercompany Partner Constants**

The HFMConstants type library provides the following types of constants for the Intercompany Partner dimension:

- “Intercompany Partner Member Constants” on page 858
- “Intercompany Partner Member List Constants” on page 859
- “Intercompany Partner Attribute Constants” on page 859

**Intercompany Partner Member Constants**

The following constants represent members of the Intercompany Partner dimension.
Table 107  tagICPCONSTANTS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBERFIRSTSYSTEMICP</td>
<td>Represents the first member of the Intercompany Partner dimension; use this to loop through the dimension’s members.</td>
</tr>
<tr>
<td>MEMBERICPALL</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>MEMBERICPDEFAULT</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>MEMBERICPNONE</td>
<td>Represents the [ICP None] member.</td>
</tr>
<tr>
<td>MEMBERICPTOP</td>
<td>Represents the [ICP Top] member.</td>
</tr>
<tr>
<td>MEMBERINTERCOMPANYENTITIES</td>
<td>Represents the [ICP Entities] member.</td>
</tr>
<tr>
<td>MEMBERLASTSYSTEMICP</td>
<td>Represents the last member of the Intercompany Partner dimension; use this to loop through the dimension’s members</td>
</tr>
<tr>
<td>MEMBERLOWESTICPAVAILABLE</td>
<td>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.</td>
</tr>
</tbody>
</table>

**Intercompany Partner Member List Constants**

The following constants represent system-defined member lists for the Intercompany Partner dimension.

Table 108  tagICPMEMBERLISTS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICP_MEMBER_LIST_ALL_HIERARCHY</td>
<td>Hierarchy member list.</td>
</tr>
<tr>
<td>ICP_MEMBER_LIST_ANCESTORS</td>
<td>Ancestors member list.</td>
</tr>
<tr>
<td>ICP_MEMBER_LIST_BASE</td>
<td>Base member list.</td>
</tr>
<tr>
<td>ICP_MEMBER_LIST_CHILDREN</td>
<td>Children member list.</td>
</tr>
<tr>
<td>ICP_MEMBER_LIST_DESCENDANTS</td>
<td>Descendants member list.</td>
</tr>
<tr>
<td>ICP_MEMBER_LIST_PARENTS</td>
<td>Parents member list.</td>
</tr>
<tr>
<td>ICP_MEMBER_LIST_SYSTEM</td>
<td>System member list.</td>
</tr>
<tr>
<td>NUM_PREDEFINED_ICP_MEMBER_LISTS</td>
<td>The number of system-defined member lists.</td>
</tr>
</tbody>
</table>

**Intercompany Partner Attribute Constants**

The following table lists constants for Intercompany Partner dimension attributes.
Table 109  tagICP_ATTRIBS Enumeration

<table>
<thead>
<tr>
<th>Old Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_ICP_MAX</td>
<td>ATTRIBEX_ICP_MAX</td>
<td>Represents the last attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_ICP_MIN</td>
<td>ATTRIBEX_ICP_MIN</td>
<td>Represents the first attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_ICP_NUM_ATTRIBS</td>
<td>ATTRIBEX_ICP_NUM_ATTRIBS</td>
<td>Represents the total number of attributes in this enumeration.</td>
</tr>
<tr>
<td>ATTRIB_ICP_SECURITY_CLASS</td>
<td>ATTRIBEX_ICP_SECURITY_CLASS</td>
<td>SecurityClass attribute.</td>
</tr>
</tbody>
</table>

Period Dimension Constants

The HFMConstants type library provides the following types of constants for the Period dimension:

- “Period Member Constant” on page 860
- “Period Member List Constants” on page 860

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 110  tagPERIODMEMBERLISTS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM_PREDEFINED_PERIOD_MEMBER_LISTS</td>
<td>The number of system-defined member lists for the Period dimension.</td>
</tr>
<tr>
<td>PERIOD_MEMBER_LIST_ALL_HIERARCHY</td>
<td>Hierarchy member list.</td>
</tr>
<tr>
<td>PERIOD_MEMBER_LIST_ANCESTORS</td>
<td>Ancestors member list.</td>
</tr>
<tr>
<td>PERIOD_MEMBER_LIST_CHILDREN</td>
<td>Children member list.</td>
</tr>
<tr>
<td>PERIOD_MEMBER_LIST_DESCENDANTS</td>
<td>Descendants member list.</td>
</tr>
<tr>
<td>PERIOD_MEMBER_LIST_FIFTH_GENERATION</td>
<td>Fifth Generation member list.</td>
</tr>
<tr>
<td>PERIOD_MEMBER_LIST_FIRST_GENERATION</td>
<td>First Generation member list.</td>
</tr>
</tbody>
</table>
### Scenario Attribute Constants

The following constants represent attributes of the Scenario dimension.

**Table 111  tagSCENARIO_ATTRIBUTES Enumeration**

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_SCENARIO_CONSOLIDATE_YTD</td>
<td>ATTRIBEX_SCENARIO_CONSOLIDATE_YTD</td>
<td>ConsolidateYTD attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_DEFAULT_FREQ_FOR_IC_TRANSACTIONS</td>
<td>ATTRIBEX_SCENARIO_DEFAULT_FREQ_FOR_IC_TRANSACTIONS</td>
<td>DefFreqForICTrans attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_DEFAULT_FREQUENCY</td>
<td>ATTRIBEX_SCENARIO_DEFAULT_FREQUENCY</td>
<td>DefaultFreq attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_DEFAULT_VIEW</td>
<td>ATTRIBEX_SCENARIO_DEFAULT_VIEW</td>
<td>DefaultView attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_ENABLE_DATA_AUDIT</td>
<td>ATTRIBEX_SCENARIO_ENABLE_DATA_AUDIT</td>
<td>EnableDataAudit attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_MAX</td>
<td>ATTRIBEX_SCENARIO_MAX</td>
<td>Represents the last attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_MAX_REVIEW_LEVEL</td>
<td>ATTRIBEX_SCENARIO_MAX_REVIEW_LEVEL</td>
<td>MaximumReviewLevel attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_MIN</td>
<td>ATTRIBEX_SCENARIO_MIN</td>
<td>Represents the first attribute in this enumeration; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_MISSING_DATA_ADJ</td>
<td>ATTRIBEX_SCENARIO_MISSING_DATA_ADJ</td>
<td>ZeroViewForAdj attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_MISSING_DATA_NON_ADJ</td>
<td>ATTRIBEX_SCENARIO_MISSING_DATA_NON_ADJ</td>
<td>ZeroViewForNonadj attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_NUM_ATTRIBS</td>
<td>ATTRIBEX_SCENARIO_NUM_ATTRIBS</td>
<td>Represents the total number of attributes in this enumeration.</td>
</tr>
</tbody>
</table>
### Deprecated Constant

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_SCENARIO_SECURITY_CLASS</td>
<td>ATTRIBEX_SCENARIO_SECURITY_CLASS</td>
<td>SecurityClass attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_SUPPORTS_PROCESS_FLOW</td>
<td>ATTRIBEX_SCENARIO_SUPPORTS_PROCESS_FLOW</td>
<td>SupportsProcessManagement attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_USERDEF1</td>
<td>ATTRIBEX_SCENARIO_USERDEF1</td>
<td>UserDefined1 attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_USERDEF2</td>
<td>ATTRIBEX_SCENARIO_USERDEF2</td>
<td>UserDefined2 attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_USERDEF3</td>
<td>ATTRIBEX_SCENARIO_USERDEF3</td>
<td>UserDefined3 attribute.</td>
</tr>
<tr>
<td>ATTRIB_SCENARIO_USES_LINE_ITEMS</td>
<td>ATTRIBEX_SCENARIO_USES_LINE_ITEMS</td>
<td>UsesLineItems attribute</td>
</tr>
</tbody>
</table>

### 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 112  tagSYSTEMLISTS Enumeration**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBER_LIST_ALL_HIERARCHY</td>
<td>[Hierarchy] member list.</td>
</tr>
<tr>
<td>MEMBER_LIST_DESCENDANTS</td>
<td>[Descendants] member list.</td>
</tr>
<tr>
<td>MEMBER_LIST_DML_START</td>
<td>Dynamic member list.</td>
</tr>
</tbody>
</table>

### Value Dimension Constants

The HFMConstants type library provides the following types of constants for the Value dimension:

- “Value Dimension Member Constants” on page 862
- “Value Dimension Member List Constants” on page 863

### Value Dimension Member Constants

The following constants represent Value dimension members.
<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBERALLENTITYVALUES</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>MEMBERALLNODEVALUES</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>MEMBERCONTRIBUTION</td>
<td><em>[Contribution]</em></td>
</tr>
<tr>
<td>MEMBERCONTRIBUTIONADJS</td>
<td><em>[Contribution Adjs]</em></td>
</tr>
<tr>
<td>MEMBERCONTRIBUTIONTOTAL</td>
<td><em>[Contribution Total]</em></td>
</tr>
<tr>
<td>MEMBERDERIVEPROPORTION</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>MEMBERELIMINATION</td>
<td><em>[Elimination]</em></td>
</tr>
<tr>
<td>MEMBERENTITYCURRENCY</td>
<td><em>[Entity Currency]</em></td>
</tr>
<tr>
<td>MEMBERENTITYCURRENCYADJS</td>
<td><em>[Entity Curr Adjs]</em></td>
</tr>
<tr>
<td>MEMBERENTITYCURRENCYTOTAL</td>
<td><em>[Entity Curr Total]</em></td>
</tr>
<tr>
<td>MEMBERFIRSTSPECIALVALUE</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>MEMBERLASTSPECIALVALUE</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>MEMBERLOWESTVALUEAVAILABLE</td>
<td>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.</td>
</tr>
<tr>
<td>MEMBERPARENT</td>
<td><em>[Parent]</em></td>
</tr>
<tr>
<td>MEMBERPARENTADJS</td>
<td><em>[Parent Adjs]</em></td>
</tr>
<tr>
<td>MEMBERPARENTCURRENCY</td>
<td><em>[Parent Currency]</em></td>
</tr>
<tr>
<td>MEMBERPARENTCURRENCYADJS</td>
<td><em>[Parent Curr Adjs]</em></td>
</tr>
<tr>
<td>MEMBERPARENTCURRENCYTOTAL</td>
<td><em>[Parent Curr Total]</em></td>
</tr>
<tr>
<td>MEMBERPARENTTOTAL</td>
<td><em>[Parent Total]</em></td>
</tr>
<tr>
<td>MEMBERPROPORTION</td>
<td><em>[Proportion]</em></td>
</tr>
<tr>
<td>MEMBERSPECIALCURRENCY</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>MEMBERSPECIALCURRENCYADJ</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>MEMBERSPECIALCURRENCYTOTAL</td>
<td><em>For internal use.</em></td>
</tr>
</tbody>
</table>

**Value Dimension Member List Constants**

The following constants represent system-defined member lists for the Value dimension.
### Table 114  tagVALUEMEMBERLISTS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM_PREDEFINED_VALUE_MEMBER_Lists</td>
<td>The number of system-defined member lists for the Value dimension.</td>
</tr>
<tr>
<td>VALUE_MEMBER_LIST_ADJS</td>
<td>Adjustments member list.</td>
</tr>
<tr>
<td>VALUE_MEMBER_LIST_ALL_HIERARCHY</td>
<td>Hierarchy member list.</td>
</tr>
<tr>
<td>VALUE_MEMBER_LIST_DEFAULT_CURRENCIES</td>
<td>Default Currencies member list.</td>
</tr>
<tr>
<td>VALUE_MEMBER_LIST_DESCENDANTS</td>
<td>Descendants member list.</td>
</tr>
<tr>
<td>VALUE_MEMBER_LIST_INPUTS</td>
<td>Inputs member list.</td>
</tr>
<tr>
<td>VALUE_MEMBER_LIST_TOTALS</td>
<td>Totals member list.</td>
</tr>
</tbody>
</table>

### IHsvTreeInfo Interface Constants

The following constants can be used with the IHsvTreeInfo interface.

### Table 115  tagIHSVTREECONSTANTS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_PARENT_UNDEFINED</td>
<td>Represents a member with no defined default parent.</td>
</tr>
<tr>
<td>NUM_STANDARD_ITEM_COLUMNS</td>
<td>For internal use.</td>
</tr>
<tr>
<td>SIBLING_NONE</td>
<td>Represents no siblings.</td>
</tr>
<tr>
<td>TREE_POSITION_FIRST_SIBLING</td>
<td>Represents the first sibling.</td>
</tr>
<tr>
<td>TREE_POSITION_LAST_SIBLING</td>
<td>Represents the last sibling.</td>
</tr>
<tr>
<td>TREE_ROOT</td>
<td>Represents the root of the dimension hierarchy.</td>
</tr>
</tbody>
</table>

### Application Setting Attribute ID Constants

The following constants represent IDs of application setting attributes.

### Table 116  tagAPPSETTING_ATTRIBS Enumeration

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_APPSETTING_CONsolidationRULES</td>
<td>ATTRIBEX_APPSETTING_CONsolidationRULES</td>
<td>ConsolidationRules attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_DEFAULTCURRENCY</td>
<td>ATTRIBEX_APPSETTING_DEFAULTCURRENCY</td>
<td>DefaultCurrency attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_DEFAULTRATEFORBALANCEACCOUNTS</td>
<td>ATTRIBEX_APPSETTING_DEFAULTRATEFORBALANCEACCOUNTS</td>
<td>DefaultRateForBalanceAccounts attribute.</td>
</tr>
<tr>
<td>Deprecated Constant</td>
<td>New Constant</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_DEFAULTRATEFORFLOWACCOUNTS</td>
<td>ATTRIBEX_APPSETTING_DEFAULTRATEFORFLOWACCOUNTS</td>
<td>DefaultRateForFlowAccounts attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_DEFAULTVALUEFORACTIVE</td>
<td>ATTRIBEX_APPSETTING_DEFAULTVALUEFORACTIVE</td>
<td>DefaultValueForActive attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_ENABLEMETADATASECURITYFILTERING</td>
<td>ATTRIBEX_APPSETTING_ENABLEMETADATASECURITYFILTERING</td>
<td>EnableMetadataSecurityFiltering attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_ICPENTITIESAGGREGATIONWEIGHT</td>
<td>ATTRIBEX_APPSETTING_ICPENTITIESAGGREGATIONWEIGHT</td>
<td>ICPEntitiesAggregationWeight attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_MAX_CELLTEXT_SIZE</td>
<td>ATTRIBEX_APPSETTING_MAX_CELLTEXT_SIZE</td>
<td>MaxCellTextSize attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_MAX_DOC_ATTACHMENT_SIZE</td>
<td>ATTRIBEX_APPSETTING_MAX_DOC_ATTACHMENT_SIZE</td>
<td>MaxDocAttachmentSize attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_MAX_NUM_DOC_ATTACHMENTS</td>
<td>ATTRIBEX_APPSETTING_MAX_NUM_DOC_ATTACHMENTS</td>
<td>MaxNumDocAttachments attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_NODESECURITY</td>
<td>ATTRIBEX_APPSETTING_NODESECURITY</td>
<td>NodeSecurity attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_OROGByPERIODAPPLICATION</td>
<td>ATTRIBEX_APPSETTING_OROGByPERIODAPPLICATION</td>
<td>OrgByPeriodApplication attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_SUPPORTSSUBMISSIONPHASEFORCUSTOM</td>
<td>ATTRIBEX_APPSETTING_SUPPORTSSUBMISSIONPHASEFORCUSTOM</td>
<td>SupportsSubmissionPhaseForCustom attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USEPVAFORBANCEACCOUNTS</td>
<td>ATTRIBEX_APPSETTING_USEPVAFORBANCEACCOUNTS</td>
<td>UsePVAForBalanceAccounts attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USEPVAFLOWACCOUNTS</td>
<td>ATTRIBEX_APPSETTING_USEPVAFLOWACCOUNTS</td>
<td>UsePVAForFlowAccounts attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USESECURITYFORACCOUNTS</td>
<td>ATTRIBEX_APPSETTING_USESECURITYFORACCOUNTS</td>
<td>UseSecurityForAccounts attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USESECURITYFORCUSTOM1</td>
<td>VB, VBScript: ATTRIBEX_APPSETTING_USESECURITYFORCUSTOM OR 1 C++: ATTRIBEX_APPSETTING_USESECURITYFORCUSTOM</td>
<td>UseSecurityForCustom attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USESECURITYFORCUSTOM2</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USESECURITYFORCUSTOM3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USESECURITYFORCUSTOM4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USESECURITYFORENTITIES</td>
<td>ATTRIBEX_APPSETTING_USESECURITYFORENTITIES</td>
<td>UseSecurityForEntities attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USESECURITYFORICP</td>
<td>ATTRIBEX_APPSETTING_USESECURITYFORICP</td>
<td>UseSecurityForICP attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_USESECURITYFORSCENARIOS</td>
<td>ATTRIBEX_APPSETTING_USESECURITYFORSCENARIOS</td>
<td>UseSecurityForScenarios attribute.</td>
</tr>
</tbody>
</table>
### Deprecated Constant

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ATTRIB_APPSETTING_VALIDATIONACCOUNT</td>
<td>ATTRIBEX_APPSETTING_VALIDATIONACCOUNT</td>
<td>ValidationAccount attribute.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_MIN</td>
<td>ATTRIBEX_APPSETTING_MIN</td>
<td>Represents the first application setting attribute; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_MAX</td>
<td>ATTRIBEX_APPSETTING_MAX</td>
<td>Represents the last application setting attribute; use this to loop through the attributes.</td>
</tr>
<tr>
<td>ATTRIB_APPSETTING_NUM_ATTRIBS</td>
<td>ATTRIBEX_APPSETTING_NUM_ATTRIBS</td>
<td>Represents the total number of attributes in this enumeration.</td>
</tr>
</tbody>
</table>

### Cell Status Constants

The HFMConstants type library provides the following types of status-related constants:

- “Cell Calculation Status Constants” on page 866
- “Cell Status Constants” on page 867
- “Cell Metadata Status Constants” on page 867
- “Subcube Period Calculation Status Constants” on page 868
- “Transaction Dimension Constants” on page 869
- “Cell Transaction Type Constants” on page 871
- “Calculation Status Types” on page 871
- “Additional Status Information Constant” on page 872

### Cell Calculation Status Constants

The following constants represent calculation statuses of cells.

#### Table 117  tagCALCSTATUSHIGHBITS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELLSTATUS_ERROR</td>
<td>An error applies to the cell.</td>
</tr>
<tr>
<td>CELLSTATUS_INUSE</td>
<td>No data was entered or processed for the cell’s subcube.</td>
</tr>
<tr>
<td>CELLSTATUS_LOCKED</td>
<td>The cell is locked and users are not allowed to modify the cell’s data.</td>
</tr>
<tr>
<td>CELLSTATUS_NEEDSCHARTLOGIC</td>
<td>A calculation should be run for the cell.</td>
</tr>
<tr>
<td>CELLSTATUS_NEEDSCONSOLIDATION</td>
<td>A consolidation should be run for the cell.</td>
</tr>
<tr>
<td>CELLSTATUS_NEEDSTRANSLATION</td>
<td>A translation should be run for the cell.</td>
</tr>
</tbody>
</table>
### Cell Status Constants

The following constants represent statuses of cells.

**Table 118**  
<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELLSTATUS_DERIVED</td>
<td>The cell’s data is derived.</td>
</tr>
<tr>
<td>CELLSTATUS_HASICDETALTRANS</td>
<td>The cell contains intercompany transactions.</td>
</tr>
<tr>
<td>CELLSTATUS_HASTRANSACTIONS</td>
<td>The cell contains line items.</td>
</tr>
<tr>
<td>CELLSTATUS_NODATA</td>
<td>The cell has no data.</td>
</tr>
<tr>
<td>CELLSTATUS_PARENT_LEVEL_INPUT</td>
<td>The cell contains data that was manually entered at the parent entity level.</td>
</tr>
</tbody>
</table>

### Cell Metadata Status Constants

The following constants represent metadata statuses for cells.

**Table 119**  
<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELLSTATUS_DRILLABLE</td>
<td>The cell allows drill through to the source data.</td>
</tr>
<tr>
<td>CELLSTATUS_CANWRITE</td>
<td>The cell cannot be written to.</td>
</tr>
<tr>
<td>CELLSTATUS_HASTEXT</td>
<td>The cell contains a cell text description.</td>
</tr>
<tr>
<td>CELLSTATUS_INVALID</td>
<td>Invalid metadata combination.</td>
</tr>
<tr>
<td>CELLSTATUS_ISADJUSTMENTMEMBER</td>
<td>The cell’s Value dimension member is an ADJS member.</td>
</tr>
</tbody>
</table>
### Subcube Period Calculation Status Constants

The following constants represent calculation statuses for periods in subcubes.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CELLSTATUS_ISINPUTFREQUENCY</td>
<td>The cell is an input-level time period.</td>
</tr>
<tr>
<td>CELLSTATUS_NOREADACCESS</td>
<td>The user does not have read access to the cell.</td>
</tr>
<tr>
<td>CELLSTATUS_NOWRITEACCESS</td>
<td>The user does not have write access to the cell.</td>
</tr>
<tr>
<td>CELLSTATUS_SUPPORTSCHARTLOGIC</td>
<td>The cell supports calculations.</td>
</tr>
<tr>
<td>CELLSTATUS_SUPPORTSCONSOLIDATION</td>
<td>The cell supports consolidations.</td>
</tr>
<tr>
<td>CELLSTATUS_SUPPORTSLINEITEMS</td>
<td>The cell supports line items.</td>
</tr>
<tr>
<td>CELLSTATUS_SUPPORTSPERIODALLOCATIONS</td>
<td>The cell is a parent-level time period whose children are input cells.</td>
</tr>
<tr>
<td>CELLSTATUS_SUPPORTSTRANSLATION</td>
<td>The cell supports translations.</td>
</tr>
<tr>
<td>CALCSTATUS_ADJ_IS_NODATA</td>
<td>The cells that intersect the period and the “Adjs” member of the subcube’s Value dimension triplet contain no data.</td>
</tr>
<tr>
<td>CALCSTATUS_ADJ_NEEDS_CALC</td>
<td>The cells that intersect the period and the “Adjs” member of the subcube’s Value dimension triplet require a calculation.</td>
</tr>
<tr>
<td>CALCSTATUS_CONSOLIDATION_TRANSACTIONS_ARE_INVALID</td>
<td>The subcube cells have invalid consolidation transactions.</td>
</tr>
<tr>
<td>CALCSTATUS_CONTRIBUTIONADJ_IS_NODATA</td>
<td>The subcube cells that intersect the period and the [Contribution Adjs] Value dimension member contain no data.</td>
</tr>
<tr>
<td>CALCSTATUS_CONTRIBUTIONADJ_NEEDS_CALC</td>
<td>The subcube cells that intersect the period and the [Contribution Adjs] Value dimension member require a calculation.</td>
</tr>
<tr>
<td>CALCSTATUS_ELIMINATION_IS_NODATA</td>
<td>The subcube cells that intersect the period and the [Elimination] Value dimension member contain no data.</td>
</tr>
<tr>
<td>CALCSTATUS_ELIMINATION_NEEDS_CALC</td>
<td>The subcube cells that intersect the period and the [Elimination] Value dimension member require a calculation.</td>
</tr>
<tr>
<td>CALCSTATUS_INPUT_IS_NODATA</td>
<td>The cells that intersect the period and the input member of the subcube’s Value dimension triplet contain no data.</td>
</tr>
<tr>
<td>CALCSTATUS_INPUT_NEEDS_CALC</td>
<td>The cells that intersect the period and the input member of the subcube’s Value dimension triplet require a calculation.</td>
</tr>
<tr>
<td>CALCSTATUS_INUSE</td>
<td>The period of the subcube is in use; data was entered or a calculation was performed.</td>
</tr>
<tr>
<td>CALCSTATUS_LOCKED</td>
<td>No one is allowed to modify data for the subcube cells that intersect the period.</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>CALCSTATUS_NEEDSCHARTLOGIC</td>
<td>A calculation should be run for the subcube cells that intersect the period.</td>
</tr>
<tr>
<td>CALCSTATUS_NEEDSCONSOLIDATION</td>
<td>A consolidation should be run for the subcube cells that intersect the period.</td>
</tr>
<tr>
<td>CALCSTATUS_NEEDSTRANSLATION</td>
<td>A translation should be run for the subcube cells that intersect the period.</td>
</tr>
<tr>
<td>CALCSTATUS_NODATA</td>
<td>No data exists for the subcube cells that intersect the period.</td>
</tr>
<tr>
<td>CALCSTATUS_OK_BUT_SYSTEM_CHANGED</td>
<td>The system has changed because someone has loaded rules or metadata. A recalculation via Force Calculate is recommended but is not required.</td>
</tr>
<tr>
<td>CALCSTATUS_PARENTADJ_IS_NODATA</td>
<td>The subcube cells that intersect the period and the [Parent Adjs] Value dimension member contain no data.</td>
</tr>
<tr>
<td>CALCSTATUS_PARENTADJ_NEEDS_CALC</td>
<td>The subcube cells that intersect the period and the [Parent Adjs] Value dimension member require a calculation.</td>
</tr>
<tr>
<td>CALCSTATUS_PROCESS_FLOW_BIT1</td>
<td>For internal use.</td>
</tr>
<tr>
<td>CALCSTATUS_PROCESS_FLOW_BIT2</td>
<td>For internal use.</td>
</tr>
<tr>
<td>CALCSTATUS_PROCESS_FLOW_BIT3</td>
<td>For internal use.</td>
</tr>
<tr>
<td>CALCSTATUS_PROCESS_FLOW_BIT4</td>
<td>For internal use.</td>
</tr>
<tr>
<td>CALCSTATUS_PROPORTION_IS_NODATA</td>
<td>The subcube cells that intersect the period and the [Proportion] Value dimension member contain no data.</td>
</tr>
<tr>
<td>CALCSTATUS_PROPORTION_NEEDS_CALC</td>
<td>The subcube cells that intersect the period and the [Proportion] Value dimension member require a calculation.</td>
</tr>
</tbody>
</table>

### Transaction Dimension Constants

The following constants represent the transaction dimensions for statutory consolidations.

**Note:** For TRANSACTION_DIMENSIONS_DES_CUSTOM_EX and TRANSACTION_DIMENSIONS_SRC_CUSTOM_EX, you must OR in the custom dimension # (1-n).

#### Table 121  tagTRANSACTION_DIMENSIONS Enumeration

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSACTION_DIMENSIONS_CUR_ENTITY</td>
<td>TRANSACTION_DIMENSIONS_CUR_ENTITY_EX</td>
<td>The current entity.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_CUR_PARENT</td>
<td>TRANSACTION_DIMENSIONS_CUR_PARENT_EX</td>
<td>The parent of the current entity.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_CUR_PERIOD</td>
<td>TRANSACTION_DIMENSIONS_CUR_PERIOD_EX</td>
<td>The current period.</td>
</tr>
<tr>
<td>Deprecated Constant</td>
<td>New Constant</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_DES_ACCOUNT</td>
<td>TRANSACTION_DIMENSIONS_DES_ACCOUNT_EX</td>
<td>The destination account.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_DES_CUSTOM1</td>
<td>VB: TRANSACTION_DIMENSIONS_DES_CUSTOM_EX OR 1</td>
<td>The destination member of the Custom 1 dimension.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_DES_CUSTOM2</td>
<td>C++: TRANSACTION_DIMENSIONS_DES_CUSTOM_EX</td>
<td></td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_DES_CUSTOM3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_DES_CUSTOM4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_DES_ENTITY</td>
<td>TRANSACTION_DIMENSIONS_DES_ENTITY_EX</td>
<td>The destination entity.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_DES_ICP</td>
<td>TRANSACTION_DIMENSIONS_DES_ICP_EX</td>
<td>The destination member of the Intercompany Partner dimension.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_DES_VALUE</td>
<td>TRANSACTION_DIMENSIONS_DES_VALUE_EX</td>
<td>The destination member of the Value dimension.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_ACCOUNT</td>
<td>TRANSACTION_DIMENSIONS_SRC_ACCOUNT_EX</td>
<td>The source account.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_CUSTOM1</td>
<td>VB: TRANSACTION_DIMENSIONS_SRC_CUSTOM_EX</td>
<td>The source member of the Custom 1 dimension.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_CUSTOM2</td>
<td>C++: TRANSACTION_DIMENSIONS_SRC_CUSTOM_EX</td>
<td></td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_CUSTOM3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_CUSTOM4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_ENTITY</td>
<td>TRANSACTION_DIMENSIONS_SRC_ENTITY_EX</td>
<td>The source entity.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_ICP</td>
<td>TRANSACTION_DIMENSIONS_SRC_ICP_EX</td>
<td>The source member of the Intercompany Partner dimension.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_PARENT</td>
<td>TRANSACTION_DIMENSIONS_SRC_PARENT_EX</td>
<td>The parent of the source entity.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_PERIOD</td>
<td>TRANSACTION_DIMENSIONS_SRC_PERIOD_EX</td>
<td>The source period.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_SCENARIO</td>
<td>TRANSACTION_DIMENSIONS_SRC_SCENARIO_EX</td>
<td>The source scenario.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_VALUE</td>
<td>TRANSACTION_DIMENSIONS_SRC_VALUE_EX</td>
<td>The source member of the Value dimension.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_VIEW</td>
<td>TRANSACTION_DIMENSIONS_SRC_VIEW_EX</td>
<td>The source member of the View dimension.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_SRC_YEAR</td>
<td>TRANSACTION_DIMENSIONS_SRC_YEAR_EX</td>
<td>The source year.</td>
</tr>
<tr>
<td>TRANSACTION_DIMENSIONS_TOTAL</td>
<td>TRANSACTION_DIMENSIONS_TOTAL_EX</td>
<td>Represents the total number of transaction dimensions.</td>
</tr>
</tbody>
</table>
Cell Transaction Type Constants

The following constants represent transaction types for cells.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TRANSACTIONTYPE_AGGREGATED</td>
<td>The cell’s data was calculated by aggregation.</td>
</tr>
<tr>
<td>TRANSACTIONTYPE_CALCULATED</td>
<td>The cell’s data was calculated by a rule.</td>
</tr>
<tr>
<td>TRANSACTIONTYPE_DERIVED</td>
<td>The cell’s data was calculated by time intelligence.</td>
</tr>
<tr>
<td>TRANSACTIONTYPE_HASTRANSACTIONS</td>
<td>The cell has intercompany transactions.</td>
</tr>
<tr>
<td>TRANSACTIONTYPE_INPUT</td>
<td>The cell’s data was input.</td>
</tr>
<tr>
<td>TRANSACTIONTYPE_NODATA</td>
<td>The cell contains no data.</td>
</tr>
<tr>
<td>TRANSACTIONTYPE_PARENT_LEVEL_INPUT</td>
<td>The cell’s data was manually entered at the parent entity level.</td>
</tr>
</tbody>
</table>

Calculation Status Types

The following constants represent calculation statuses. See the Oracle Hyperion Financial Management Administrator’s Guide.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALCSTATUS_STATSCOL_ALL</td>
<td>All statuses.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_CH</td>
<td>CH status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_CHND</td>
<td>CH ND status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_CN</td>
<td>CN status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_CNND</td>
<td>CN ND status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_LOCKED</td>
<td>Locked status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_NODATA</td>
<td>NoData status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_OK</td>
<td>OK ND status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_OKND</td>
<td>OK status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_OKSC</td>
<td>OK SC status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_TR</td>
<td>TR status.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_TRND</td>
<td>TR ND status.</td>
</tr>
</tbody>
</table>

Cell Status Constants
<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CALCSTATUS_STATSCOL_MIN</td>
<td>Represents the lowest value that one of the status constants represents; use this and CALCSTATUS_STATSCOL_MAX to loop through the statuses.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_MAX</td>
<td>Represents the highest value that one of the status constants represents.</td>
</tr>
<tr>
<td>CALCSTATUS_STATSCOL_UNKNOWN</td>
<td>Unknown calculation status.</td>
</tr>
</tbody>
</table>

**Additional Status Information Constant**

Additional cell status information, such as that returned by `HsvData.GetStatusEx`, is represented by the `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 872
- “Balance Type Constants” on page 873
- “Journal Column Display Constants” on page 873
- “Debit/Credit Constants” on page 874
- “Period Status Constants” on page 874
- “Journal Report Display Option Constants” on page 874
- “Journal Report Sort Option Constants” on page 875
- “Journal Report Total Flag Constants” on page 875
- “Journal Status Constants” on page 875
- “Journal Type Constants” on page 876
- “Journal Web Session Parameter Constants” on page 876
- “Template Column Display Constants” on page 876
- “Template Type Constants” on page 877
- “Journal and Template Type Constants” on page 877

**Journal Action Constants**

The following constants represent journal actions.
Table 124  tagJOURNALACTION Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBA_APPROVE</td>
<td>Approve journal.</td>
</tr>
<tr>
<td>JBA_DELETE</td>
<td>Delete journal.</td>
</tr>
<tr>
<td>JBA_POST</td>
<td>Post journal.</td>
</tr>
<tr>
<td>JBA_REJECT</td>
<td>Reject journal.</td>
</tr>
<tr>
<td>JBA_SUBMIT</td>
<td>Submit journal.</td>
</tr>
<tr>
<td>JBA_UNPOST</td>
<td>Unpost journal.</td>
</tr>
<tr>
<td>JBA_UNSUBMIT</td>
<td>Unsubmit journal.</td>
</tr>
</tbody>
</table>

Balance Type Constants
The following constants represent balance types of journals.

Table 125  tagJOURNALBALANCETYPEFLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JBTF_BALANCED</td>
<td>Balanced journal.</td>
</tr>
<tr>
<td>JBTF_BALANCED_BYENTITY</td>
<td>Balanced-by-entity journal.</td>
</tr>
<tr>
<td>JBTF_UNBALANCED</td>
<td>Unbalanced journal.</td>
</tr>
</tbody>
</table>

Journal Column Display Constants
The following constants represent journals' display columns.

Table 126  tagJOURNALCOLUMNSFORVIEWANDFILTER Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_JOURNALAPPROVEDBY</td>
<td>Approved By column.</td>
</tr>
<tr>
<td>COLUMN_JOURNALAPPROVEDON</td>
<td>Approved On column.</td>
</tr>
<tr>
<td>COLUMN_JOURNALBALANCEATTRIBUTE</td>
<td>Balance Type column.</td>
</tr>
<tr>
<td>COLUMN_JOURNALCREATEDBY</td>
<td>Created By column.</td>
</tr>
<tr>
<td>COLUMN_JOURNALCREATEDON</td>
<td>Date Created column.</td>
</tr>
<tr>
<td>COLUMN_JOURNALDESCRIPTION</td>
<td>Description column.</td>
</tr>
<tr>
<td>COLUMN_JOURNALGROUP</td>
<td>Group column.</td>
</tr>
</tbody>
</table>
### Debit/Credit Constants

The following constants represent debits and credits.

**Table 127  tagDEBITCREDITUNIT Enumeration**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JE_TYPE_CREDIT</td>
<td>Credit</td>
</tr>
<tr>
<td>JE_TYPE_DEBIT</td>
<td>Debit</td>
</tr>
<tr>
<td>JE_TYPE_UNIT</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

### Period Status Constants

The following constants represent period statuses of journals.

**Table 128  tagJOURNALPERIODSTATUS Enumeration**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JPS_CLOSED</td>
<td>Closed period.</td>
</tr>
<tr>
<td>JPS_OPENED</td>
<td>Open period.</td>
</tr>
<tr>
<td>JPS_UNOPENED</td>
<td>Unopened period.</td>
</tr>
</tbody>
</table>

### Journal Report Display Option Constants

The following constants represent display options for journal reports.
Table 129  tagJOURNALREPORTDISPLAYFLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOURNALREPORT_DISPLAY_NOREPEAT</td>
<td>NoRepeat.</td>
</tr>
<tr>
<td>JOURNALREPORT_DISPLAY_REPEAT</td>
<td>Repeat.</td>
</tr>
<tr>
<td>NUM_JOURNALREPORT_DISPLAY_FLAGS</td>
<td>Represents the number of available display options.</td>
</tr>
</tbody>
</table>

Journal Report Sort Option Constants

The following constants represent sorting options for journal reports.

Table 130  tagJOURNALREPORTSORTFLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOURNALREPORT_SORT_ASCENDING</td>
<td>Ascending sort order.</td>
</tr>
<tr>
<td>JOURNALREPORT_SORT_DESCENDING</td>
<td>Descending sort order.</td>
</tr>
<tr>
<td>NUM_JOURNALREPORT_SORT_FLAGS</td>
<td>Represents the number of available sorting options.</td>
</tr>
</tbody>
</table>

Journal Report Total Flag Constants

The following constants represent options for journal report totalling.

Table 131  tagJOURNALREPORTTOTALFLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOURNALREPORT_TOTAL_NO</td>
<td>No totals.</td>
</tr>
<tr>
<td>JOURNALREPORT_TOTAL_YES</td>
<td>Yes — display totals.</td>
</tr>
<tr>
<td>NUM_JOURNALREPORT_TOTAL_FLAGS</td>
<td>Represents the number of available totalling options.</td>
</tr>
</tbody>
</table>

Journal Status Constants

The following constants represent journal statuses.

Table 132  tagJOURNALSTATUSFLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JSF_APPROVED</td>
<td>Approved status.</td>
</tr>
<tr>
<td>JSF_POSTED</td>
<td>Posted status.</td>
</tr>
<tr>
<td>JSF_REJECTED</td>
<td>Rejected status.</td>
</tr>
</tbody>
</table>
The following constants represent journal types.

Table 133  tagJOURNALTYPEFLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JTF_AUTOREVERSAL</td>
<td>System-generated autoreversing journal. When an autoreversing journal is posted, the system automatically generates a journal in the next period.</td>
</tr>
<tr>
<td>JTF_AUTOREVERSING</td>
<td>Autoreversing journal.</td>
</tr>
<tr>
<td>JTF_REGULAR</td>
<td>Regular journal.</td>
</tr>
<tr>
<td>JTF_UNIT</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

The following constants represent default settings for Web sessions.

Table 134  tagJOURNALWEBSESSIONPARAMETERS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JWSP_BUFFERSIZE</td>
<td>For internal use.</td>
</tr>
<tr>
<td>JWSP_NUM_ENTRIES_DISPLAYED</td>
<td>Represents the default number of journals that are displayed on a page.</td>
</tr>
</tbody>
</table>

The following constants represent display columns of journal templates.

Table 135  tagTEMPLATECOLUMNSFORVIEWANDFILTER Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_TEMPLATEBALANCEATTRIBUTE</td>
<td>Balance Type column.</td>
</tr>
<tr>
<td>COLUMN_TEMPLATEDESCRIPTION</td>
<td>Description column.</td>
</tr>
<tr>
<td>COLUMN_TEMPLATEGROUP</td>
<td>Group column.</td>
</tr>
<tr>
<td>COLUMN_TEMPLATELABEL</td>
<td>Label column.</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------</td>
</tr>
<tr>
<td>COLUMN_TEMPLATELINEITEMENTITY</td>
<td>Entity column.</td>
</tr>
<tr>
<td>COLUMN_TEMPLATELINEITEMPARENT</td>
<td>Parent column.</td>
</tr>
<tr>
<td>COLUMN_TEMPLATETYPE</td>
<td>Security Class column.</td>
</tr>
<tr>
<td>COLUMN_TEMPLATETRUNCATEDDESCRIPTION</td>
<td>Short Description column.</td>
</tr>
<tr>
<td>COLUMN_TEMPLATEVALUEDIMENSION</td>
<td>Type column.</td>
</tr>
<tr>
<td>COLUMN_TEMPLATEVALUEDIMENSION</td>
<td>Value column.</td>
</tr>
<tr>
<td>NUMTEMPLATECOLUMNS</td>
<td>Represents the number of available display columns.</td>
</tr>
</tbody>
</table>

### Template Type Constants

The following constants represent types of journal templates.

**Table 136**  tagTEMPLATETYPEFLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>TTF_RECURRING</td>
<td>Recurring template.</td>
</tr>
<tr>
<td>TTF_STANDARD</td>
<td>Standard template.</td>
</tr>
</tbody>
</table>

### 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.JounalIDsForExtractFilter.

**Table 137**  tagJOURNALEXTRACTJOURNALTYPE Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOURNALEXTRACT_JOURNALTYPE_REGULAR</td>
<td>Journals.</td>
</tr>
<tr>
<td>JOURNALEXTRACT_JOURNALTYPE_STANDARD</td>
<td>Standard journal templates.</td>
</tr>
<tr>
<td>JOURNALEXTRACT_JOURNALTYPE_RECURRING</td>
<td>Recurring journal templates.</td>
</tr>
</tbody>
</table>

### Process Management Constants

The HFMConstants type library provides process management-related constants for actions and review levels.
Process Management Action Constants

The following constants represent process management actions.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS_FLOW_ACTION_APPROVE</td>
<td>Approve action.</td>
</tr>
<tr>
<td>PROCESS_FLOW_ACTION_PROMOTE</td>
<td>Promote action.</td>
</tr>
<tr>
<td>PROCESS_FLOW_ACTION_PUBLISH</td>
<td>Publish action.</td>
</tr>
<tr>
<td>PROCESS_FLOW_ACTION_REJECT</td>
<td>Reject action.</td>
</tr>
<tr>
<td>PROCESS_FLOW_ACTION_SIGN_OFF</td>
<td>Sign off action.</td>
</tr>
<tr>
<td>PROCESS_FLOW_ACTION_START</td>
<td>Start action.</td>
</tr>
<tr>
<td>PROCESS_FLOW_ACTION_SUBMIT</td>
<td>Submit action.</td>
</tr>
<tr>
<td>NUM_PROCESS_FLOW_ACTIONS</td>
<td>Represents the total number of process management actions.</td>
</tr>
</tbody>
</table>

Process Management Review Level Constants

The following constants represent process management review levels.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROCESS_FLOW_STATE_APPROVED</td>
<td>Approved.</td>
</tr>
<tr>
<td>PROCESS_FLOW_STATE_FIRST_PASS</td>
<td>First pass.</td>
</tr>
<tr>
<td>PROCESS_FLOW_STATE_NOT_STARTED</td>
<td>Not started.</td>
</tr>
</tbody>
</table>
| PROCESS_FLOW_STATE_NOT_SUPPORTED       | Process management is not supported. This constant represents either of the following conditions:  
- Process management is not enabled for the process unit’s Scenario dimension member.  
- The connected user does not have the required security access for the process management operation. |
| PROCESS_FLOW_STATE_PUBLISHED           | Published.                                            |
| PROCESS_FLOW_STATE_REVIEW1             | Review Level 1.                                       |
| PROCESS_FLOW_STATE_REVIEW10            | Review Level 10.                                      |
| PROCESS_FLOW_STATE_REVIEW2             | Review Level 2.                                       |
| PROCESS_FLOW_STATE_REVIEW3             | Review Level 3.                                       |
| PROCESS_FLOW_STATE_REVIEW4             | Review Level 4.                                       |
Report Column Constants

The following constants represent columns in reports.

The tag REPORTECOLUMNSFORVIEWANDFILTER is now deprecated, and is replaced by the tag REPORTECOLUMNSFORVIEWANDFILTEREXTDIM enum.

Note: When using COL_REPORTLINEITEMCUSTOMBASE, OR in the custom dimension # (1-n).

<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_REPORTAPPROVEDBY</td>
<td>COL_REPORTAPPROVEDBY</td>
<td>Approved By column.</td>
</tr>
<tr>
<td>COLUMN_REPORTAPPROVEDON</td>
<td>COL_REPORTAPPROVEDON</td>
<td>Date Approved column.</td>
</tr>
<tr>
<td>COLUMN_REPORTBALANCEATTRIBUTE</td>
<td>COL_REPORTBALANCEATTRIBUTE</td>
<td>Balance Type column.</td>
</tr>
<tr>
<td>COLUMN_REPORTCREATEDBY</td>
<td>REPORTCREATEDBY</td>
<td>Created By column.</td>
</tr>
<tr>
<td>COLUMN_REPORTCREATEDON</td>
<td>COL_REPORTCREATEDON</td>
<td>Date Created column.</td>
</tr>
<tr>
<td>COLUMN_REPORTDESCRIPTION</td>
<td>COL_REPORTDESCRIPTION</td>
<td>Description column.</td>
</tr>
<tr>
<td>COLUMN_REPORTGROUP</td>
<td>COL_REPORTGROUP</td>
<td>Group column.</td>
</tr>
<tr>
<td>COLUMN_REPORTLABEL</td>
<td>COL_REPORTLABEL</td>
<td>Label column.</td>
</tr>
<tr>
<td>COLUMN_REPORTLASTACTEDBY</td>
<td>COL_REPORTLASTACTEDBY</td>
<td>Posted By column.</td>
</tr>
<tr>
<td>COLUMN_REPORTLASTACTIONON</td>
<td>COL_REPORTLASTACTIONON</td>
<td>Date Posted column.</td>
</tr>
<tr>
<td>COLUMN_REPORTLINEITEMACCOUNT</td>
<td>COL_REPORTLINEITEMACCOUNT</td>
<td>Account column.</td>
</tr>
</tbody>
</table>

Table 140  tag REPORTECOLUMNSFORVIEWANDFILTER Enumeration
<table>
<thead>
<tr>
<th>Deprecated Constant</th>
<th>New Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>COLUMN_REPORTLINEITEMCUSTOM1</td>
<td>VB, VBScript: COL_REPORTLINEITEMCUSTOMBASE OR 1</td>
<td>Custom column.</td>
</tr>
<tr>
<td>COLUMN_REPORTLINEITEMCUSTOM2</td>
<td>C++: COL_REPORTLINEITEMCUSTOMBASE</td>
<td>1</td>
</tr>
<tr>
<td>COLUMN_REPORTLINEITEMCUSTOM3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_REPORTLINEITEMCUSTOM4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>COLUMN_REPORTLINEITEMDESCRIPTION</td>
<td>COL_REPORTLINEITEMDESCRIPTION</td>
<td>Line item description constant.</td>
</tr>
<tr>
<td>COLUMN_REPORTLINEITEMENTITY</td>
<td>COL_REPORTLINEITEMENTITY</td>
<td>Entity column.</td>
</tr>
<tr>
<td>COLUMN_REPORTLINEITEMICP</td>
<td>COL_REPORTLINEITEMICP</td>
<td>Intercompany Partner column.</td>
</tr>
<tr>
<td>COLUMN_REPORTLINEITEMTEMPARENT</td>
<td>COL_REPORTLINEITEMTEMPARENT</td>
<td>Parent column.</td>
</tr>
<tr>
<td>COLUMN_REPORTSECURITYCLASS</td>
<td>COL_REPORTSECURITYCLASS</td>
<td>Security Class column.</td>
</tr>
<tr>
<td>COLUMN_REPORTSTATUS</td>
<td>COL_REPORTSTATUS</td>
<td>Status column.</td>
</tr>
<tr>
<td>COLUMN_REPORTTRUNCATEDDESCRIPTION</td>
<td>COL_REPORTTRUNCATEDDESCRIPTION</td>
<td>Short Description column.</td>
</tr>
<tr>
<td>COLUMNREPORTTYPE</td>
<td>COL_REPORTTYPE</td>
<td>Type column.</td>
</tr>
<tr>
<td>NUMREPORTCOLUMNS</td>
<td></td>
<td>Represents the number of available report columns.</td>
</tr>
</tbody>
</table>

### Consolidation Type Constants

The following constants represent types of consolidations.

**Table 141  tagCONSOLIDATIONTYPE Enumeration**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CONSOLIDATE_ALL</td>
<td>Consolidate All.</td>
</tr>
<tr>
<td>CONSOLIDATE_ALLWITHDATA</td>
<td>Consolidate All With Data.</td>
</tr>
<tr>
<td>CONSOLIDATE_ENTITYONLY</td>
<td>Consolidate Entity Only.</td>
</tr>
<tr>
<td>CONSOLIDATE_FORCEENTITYONLY</td>
<td>Consolidate Force Entity Only.</td>
</tr>
<tr>
<td>CONSOLIDATE_IMPACTED</td>
<td>Consolidate Impacted.</td>
</tr>
</tbody>
</table>

### Tier Constants

The following constants represent tiers of Financial Management’s multi-tier architecture.
Table 142 tagHFM_TIERS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFM_TIER1</td>
<td>Client tier.</td>
</tr>
<tr>
<td>HFM_TIER2</td>
<td>Application server tier.</td>
</tr>
<tr>
<td>HFM_WEB_TIER</td>
<td>Web server tier.</td>
</tr>
</tbody>
</table>

Security Constants

The following categories of constants represent security access rights, roles, and tasks:

- “Access Rights Constants” on page 881
- “Role Constants” on page 881
- “Task Constants” on page 884

Access Rights Constants

The following constants represent access rights.

Table 143 tagHFM_NUM_ACCESS_TYPES Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFM_ACCESS_RIGHTS_ALL</td>
<td>All</td>
</tr>
<tr>
<td>HFM_ACCESS_RIGHTS_NONE</td>
<td>None</td>
</tr>
<tr>
<td>HFM_ACCESS_RIGHTS_READANDPROMOTE</td>
<td>Promote</td>
</tr>
<tr>
<td>HFM_ACCESS_RIGHTS_READONLY</td>
<td>Read</td>
</tr>
<tr>
<td>HFM_ACCESS_RIGHTS_UNSPECIFIED</td>
<td>For internal use.</td>
</tr>
<tr>
<td>HFM_ACCESS_RIGHTS_VIEW</td>
<td>Metadata.</td>
</tr>
</tbody>
</table>

Role Constants

The following constants represent roles.

Table 144 tagHFM_ROLE_ENUM Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFM_NUM_ROLES</td>
<td>Represents the number of roles for an application.</td>
</tr>
<tr>
<td>HFM_ROLE_ADVANCED_USER</td>
<td>Advanced User</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>HFM_ROLE_APPLICATION_ADMINISTRATOR</td>
<td>Application Administrator</td>
</tr>
<tr>
<td>HFM_ROLE_APPROVE_JOURNALS</td>
<td>Approve Journals</td>
</tr>
<tr>
<td>HFM_ROLE_CONSOLIDATE_ALL</td>
<td>Consolidate All</td>
</tr>
<tr>
<td>HFM_ROLE_CONSOLIDATE_ALL_DATA</td>
<td>Consolidate</td>
</tr>
<tr>
<td>HFM_ROLE_CREATE_INTEGRATIONS</td>
<td>Create Integrations</td>
</tr>
<tr>
<td>HFM_ROLE_CREATE_JOURNALS</td>
<td>Create Journals</td>
</tr>
<tr>
<td>HFM_ROLE_CREATE_UNBALANCED_JOURNALS</td>
<td>Create Unbalanced Journals</td>
</tr>
<tr>
<td>HFM_ROLE_DATA_FORM_WRITEBACK_EXCEL</td>
<td>Enable write back in Web Grid</td>
</tr>
<tr>
<td>HFM_ROLE_DB_MANAGEMENT</td>
<td>Database Management.</td>
</tr>
<tr>
<td>HFM_ROLE_DEFAULT</td>
<td>Default Role</td>
</tr>
<tr>
<td>HFM_ROLE_EXTENDED_ANALYTICS</td>
<td>Extended Analytics</td>
</tr>
<tr>
<td>HFM_ROLE_GENERATE_RECURRING</td>
<td>Generate Recurring</td>
</tr>
<tr>
<td>HFM_ROLE_IC_AUTO_MATCH_BY_ACCOUNT</td>
<td>Intercompany Transaction Auto Match by Account</td>
</tr>
<tr>
<td>HFM_ROLE_IC_AUTO_MATCH_BY_ID</td>
<td>Intercompany Transaction Auto Match by ID</td>
</tr>
<tr>
<td>HFM_ROLE_IC_MANUAL_MATCH</td>
<td>Intercompany Transaction Manual Match</td>
</tr>
<tr>
<td>HFM_ROLE_IC_MANUAL_MATCH_TOL</td>
<td>Intercompany Transaction Manual Match with Tolerance</td>
</tr>
<tr>
<td>HFM_ROLE_IC_MATCH_TEMPLATE</td>
<td>Intercompany Transaction Match Template</td>
</tr>
<tr>
<td>HFM_ROLE_IC_POST_UNPOST</td>
<td>Intercompany Transaction Post/Unpost</td>
</tr>
<tr>
<td>HFM_ROLE_IC_TRANSACTION_ADMIN</td>
<td>Intercompany Transaction Admin</td>
</tr>
<tr>
<td>HFM_ROLE_IC_TRANSACTION_USER</td>
<td>Intercompany Transaction User</td>
</tr>
<tr>
<td>HFM_ROLE_IC_UNMATCH</td>
<td>Intercompany Transaction Unmatch</td>
</tr>
<tr>
<td>HFM_ROLE_INTERCOMPANY_EMAIL_ALERTING</td>
<td>Receive Email Alerts for IC Transactions</td>
</tr>
<tr>
<td>HFM_ROLE_JOURNALS_ADMINISTRATOR</td>
<td>Journals Administrator</td>
</tr>
<tr>
<td>HFM_ROLE_LOAD_EXCEL_DATA</td>
<td>Load Excel Data</td>
</tr>
<tr>
<td>HFM_ROLE_LOAD_SYSTEM</td>
<td>Load System</td>
</tr>
<tr>
<td>HFM_ROLE_LOCK_DATA</td>
<td>Lock Data</td>
</tr>
<tr>
<td>HFM_ROLE_MANAGE_CUSTOM_DOCUMENTS</td>
<td>Manage Custom Documents</td>
</tr>
<tr>
<td>HFM_ROLE_MANAGE_DATA_ENTRY_FORMS</td>
<td>Manage Data Forms</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------</td>
<td>--------------------------------------------------------------------</td>
</tr>
<tr>
<td>HFM_ROLE_MANAGE_MODELS</td>
<td>Manage Models</td>
</tr>
<tr>
<td>HFM_ROLE_MANAGE_OWNERSHIP</td>
<td>Manage Ownership</td>
</tr>
<tr>
<td>HFM_ROLE_MANAGE_TEMPLATES</td>
<td>Manage Templates</td>
</tr>
<tr>
<td>HFM_ROLE_POST_JOURNALS</td>
<td>Post Journals</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER1</td>
<td>Reviewer 1</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER10</td>
<td>Reviewer 10</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER2</td>
<td>Reviewer 2</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER3</td>
<td>Reviewer 3</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER4</td>
<td>Reviewer 4</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER5</td>
<td>Reviewer 5</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER6</td>
<td>Reviewer 6</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER7</td>
<td>Reviewer 7</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER8</td>
<td>Reviewer 8</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_REVIEWER9</td>
<td>Reviewer 9</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_SUBMITTER</td>
<td>Submitter</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_SUPERVISOR</td>
<td>Review Supervisor</td>
</tr>
<tr>
<td>HFM_ROLE_PROCESS_FLOW_MGMT_EMAIL_ALERTING</td>
<td>Receive Email Alerts for Process Management</td>
</tr>
<tr>
<td>HFM_ROLE_READ_JOURNALS</td>
<td>Read Journals</td>
</tr>
<tr>
<td>HFM_ROLE_RUN_ALLOCATION</td>
<td>Run Allocation</td>
</tr>
<tr>
<td>HFM_ROLE_RUN_CONsolidation</td>
<td>Consolidate</td>
</tr>
<tr>
<td>HFM_ROLE_RUN_INTEGRATIONS</td>
<td>Run Integrations</td>
</tr>
<tr>
<td>HFM_ROLE_SAVE_SYSTEM_REPORT_ON_SERVER</td>
<td>Save System Report On Server</td>
</tr>
<tr>
<td>HFM_ROLE_PROVISIONING_MANAGER</td>
<td>Provisioning Manager</td>
</tr>
<tr>
<td>HFM_ROLE_SECURITY_ADMINISTRATOR</td>
<td>Provisioning Manager (deprecated). This constant is deprecated; instead, use HFM_ROLE_PROVISIONING_MANAGER.</td>
</tr>
<tr>
<td>HFM_ROLE_TASK_AUTOMATION</td>
<td>Task Automation</td>
</tr>
<tr>
<td>HFM_ROLE_UNLOCK_DATA</td>
<td>Unlock Data</td>
</tr>
<tr>
<td>HFM_ROLE_WEBGRID_WRITE_BACK</td>
<td>Enable Write Back in Web Grid</td>
</tr>
</tbody>
</table>
## Task Constants

The following constants represent tasks.

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFM_NUM_TASKS</td>
<td>Represents the number of tasks for an application.</td>
</tr>
<tr>
<td>HFM_TASK_ADVANCED_USER_UI</td>
<td>Advanced User for UI</td>
</tr>
<tr>
<td>HFM_TASK_APPLICATION_CREATE_APPLICATION</td>
<td>Create Application</td>
</tr>
<tr>
<td>HFM_TASK_APPLICATION_DEFINE_APPLICATION_PROFILE</td>
<td>Define Application Profile</td>
</tr>
<tr>
<td>HFM_TASK_APPLICATION_DELETE_APPLICATION</td>
<td>Delete Application</td>
</tr>
<tr>
<td>HFM_TASK_CLOSE_APPLICATION</td>
<td>Close Application</td>
</tr>
<tr>
<td>HFM_TASK_CREATE_INTERCOMPANY_MATCHING_REPORT</td>
<td>Create Intercompany Matching Report</td>
</tr>
<tr>
<td>HFM_TASK_DATA_AUDIT</td>
<td>Data Audit</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_ALLOCATE</td>
<td>Allocate</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CALCULATE</td>
<td>Calculate</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CALCULATE_CONTRIBUTION</td>
<td>Calculate Contribution</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CALCULATE_OWNERSHIP</td>
<td>Calculate Ownership</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CELL_ADJUSTMENTS</td>
<td>Cell Adjustments</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CELL_INFORMATION</td>
<td>Cell Information</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CELL_LINE_ITEM_DETAIL</td>
<td>Cell Line Item Detail</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CELL_TEXT</td>
<td>Cell Text</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_COMMIT_DATA</td>
<td>Commit Data</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CONsolidate</td>
<td>Consolidate</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CONsolidate_all</td>
<td>Consolidate All</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CONsolidate_all_with_data</td>
<td>Consolidate All with Data</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_CONsolidate_transaction</td>
<td>Consolidation Transaction</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_FORCE_CALCULATE</td>
<td>Force Calculate</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_FORCE_CALCULATE_CONTRIBUTION</td>
<td>Force Calculate Contribution</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_FORCE_TRANSLATE</td>
<td>Force Translate</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_GRID_SYSTEM_REPORT</td>
<td>Data Explorer Grid - System Report</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_LOCK</td>
<td>Lock Data</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_MANAGE_PROCESS</td>
<td>Manage Process</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_OPEN_GRID</td>
<td>Open Grid</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_REFRESH_DATA</td>
<td>Refresh Data</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_SAVE_SETTINGS</td>
<td>Save Data Explorer Settings</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_SETTINGS</td>
<td>Data Explorer Settings</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_TRANSLATE</td>
<td>Translate</td>
</tr>
<tr>
<td>HFM_TASK_DATA_EXPLORER_UNLOCK</td>
<td>Unlock Data</td>
</tr>
<tr>
<td>HFM_TASK_DATABASE_MANAGEMENT</td>
<td>Database Management</td>
</tr>
<tr>
<td>HFM_TASK_DB_MANAGEMENT</td>
<td>Database Management</td>
</tr>
<tr>
<td>HFM_TASK_ENTER_DATA</td>
<td>Enter Data</td>
</tr>
<tr>
<td>HFM_TASK_ENTER_SHARES_DATA</td>
<td>Enter Shares Data</td>
</tr>
<tr>
<td>HFM_TASK_ESSBASE</td>
<td>Essbase</td>
</tr>
<tr>
<td>HFM_TASK_EXPLORE_DATA</td>
<td>Explore Data</td>
</tr>
<tr>
<td>HFM_TASK_EXTENDED_ANALYTICS</td>
<td>Extended Analytics</td>
</tr>
<tr>
<td>HFM_TASK_EXTRACT_DATA</td>
<td>Extract Data</td>
</tr>
<tr>
<td>HFM_TASK_EXTRACT_JOURNALS</td>
<td>Extract Journals</td>
</tr>
<tr>
<td>HFM_TASK_EXTRACT_MEMBER_LISTS</td>
<td>Extract Member Lists</td>
</tr>
<tr>
<td>HFM_TASK_EXTRACT_METADATA</td>
<td>Extract Metadata</td>
</tr>
<tr>
<td>HFM_TASK_EXTRACT_RULES</td>
<td>Extract Rules</td>
</tr>
<tr>
<td>HFM_TASK_EXTRACT_SECURITY</td>
<td>Extract Security</td>
</tr>
<tr>
<td>HFM_TASK_ICT_AUTO_MATCH_BY_ACCOUNT</td>
<td>Intercompany Transaction Auto Match by Account</td>
</tr>
<tr>
<td>HFM_TASK_ICT_AUTO_MATCH_BY_ID</td>
<td>Intercompany Transaction Auto Match by ID</td>
</tr>
<tr>
<td>HFM_TASK_ICT_CREATE_TRANSACTIONS</td>
<td>Create Intercompany Transactions</td>
</tr>
<tr>
<td>HFM_TASK_ICT_DELETE_TRANSACTIONS</td>
<td>Delete Intercompany Transactions</td>
</tr>
<tr>
<td>HFM_TASK_ICT_DRILL_THROUGH</td>
<td>Drill Through Intercompany Transactions</td>
</tr>
<tr>
<td>HFM_TASK_ICT_EDIT_TRANSACTIONS</td>
<td>Edit Intercompany Transactions</td>
</tr>
<tr>
<td>HFM_TASK_ICT_EXTRACT_TRANSACTIONS</td>
<td>Extract Intercompany Transactions</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-------------------------------------------------------</td>
</tr>
<tr>
<td>HFM_TASK_ICT_LOAD_TRANSACTIONS</td>
<td>Load Intercompany Transactions</td>
</tr>
<tr>
<td>HFM_TASK_ICT_LOCK_UNLOCK_ENTITY</td>
<td>Intercompany Transactions - Lock/Unlock Entities</td>
</tr>
<tr>
<td>HFM_TASK_ICT_MANAGE_MATCHING TEMPLATE</td>
<td>Intercompany Transaction - Manage Match Template</td>
</tr>
<tr>
<td>HFM_TASK_ICT_MANAGE_PERIODS</td>
<td>Intercompany Transaction - Manage Periods</td>
</tr>
<tr>
<td>HFM_TASK_ICT_MANAGE_REASON_CODE</td>
<td>Intercompany Transaction - Manage Reason Codes</td>
</tr>
<tr>
<td>HFM_TASK_ICT_MANUAL_MATCH</td>
<td>Intercompany Transaction Manual Match</td>
</tr>
<tr>
<td>HFM_TASK_ICT_MANUAL_MATCH_TOL</td>
<td>Intercompany Transaction Manual Match with Tolerance</td>
</tr>
<tr>
<td>HFM_TASK_ICT_MATCH_REPORT_BY_ACCOUNT</td>
<td>Intercompany Transaction Match Report by Account</td>
</tr>
<tr>
<td>HFM_TASK_ICT_MATCH_REPORT_BY_ID</td>
<td>Intercompany Transaction Match Report by ID</td>
</tr>
<tr>
<td>HFM_TASK_ICT_POST_TRANSACTIONS</td>
<td>Post Intercompany Transactions</td>
</tr>
<tr>
<td>HFM_TASK_ICT_PROCESS_IC_TRANSACTIONS</td>
<td>Process Intercompany Transactions</td>
</tr>
<tr>
<td>HFM_TASK_ICT_TRANSACTION_REPORT</td>
<td>Intercompany Transaction Report</td>
</tr>
<tr>
<td>HFM_TASK_ICT_UNMATCH_TRANSACTIONS</td>
<td>Intercompany Transaction Unmatch</td>
</tr>
<tr>
<td>HFM_TASK_ICT_UNPOST_TRANSACTIONS</td>
<td>Intercompany Transaction Unpost</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_ENUM_REC_TEMPLATES</td>
<td>Enumerate recurring journal templates.</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_ENUM_STD_TEMPLATES</td>
<td>Enumerate standard journal templates.</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_OPEN_REC_TEMPLATES</td>
<td>Open recurring journal templates.</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_OPEN_STD_TEMPLATES</td>
<td>Open standard journal templates.</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS</td>
<td>Process Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_ALLOW_UNBALANCED</td>
<td>Allow Unbalanced Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_APPROVE</td>
<td>Approve Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_DELETE</td>
<td>Delete Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_NEW</td>
<td>New Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_OPEN</td>
<td>Open Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_POST</td>
<td>Post Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_REJECT</td>
<td>Reject Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_SUBMIT</td>
<td>Submit Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_SYSTEM_REPORT</td>
<td>Journals System Report</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_UNPOST</td>
<td>Unpost Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_PROCESS_JOURNALS_UNSUBMIT</td>
<td>Unsubmit Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_SETUP_JOURNALS</td>
<td>Setup Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_SETUP_JOURNALS_GENERATE_RECURRING</td>
<td>Generate Recurring Journals</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_SETUP_JOURNALS_MANAGE_GROUPS</td>
<td>Manage Journal Groups</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_SETUP_JOURNALS_MANAGE_PERIODS</td>
<td>Manage Periods</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_SETUP_JOURNALS_MANAGE_TEMPLATES</td>
<td>Manage Templates</td>
</tr>
<tr>
<td>HFM_TASK_JOURNALS_SETUP_JOURNALSTEMPLATES</td>
<td>Journal Templates</td>
</tr>
<tr>
<td>HFM_TASK_LOAD_DATA</td>
<td>Load Data</td>
</tr>
<tr>
<td>HFM_TASK_LOAD_DATA_REPLACE_BY_SECURITY</td>
<td>Load Data - Replace by Security mode</td>
</tr>
<tr>
<td>HFM_TASK_LOAD_JOURNALS</td>
<td>Load Journals</td>
</tr>
<tr>
<td>HFM_TASK_LOAD_MEMBER_LISTS</td>
<td>Load Member Lists</td>
</tr>
<tr>
<td>HFM_TASK_LOAD_METADATA</td>
<td>Load Metadata</td>
</tr>
<tr>
<td>HFM_TASK_LOAD_RULES</td>
<td>Load Rules</td>
</tr>
<tr>
<td>HFM_TASK_LOAD_SECURITY</td>
<td>Load Security</td>
</tr>
<tr>
<td>HFM_TASK_LOGOFF</td>
<td>Log Off</td>
</tr>
<tr>
<td>HFM_TASK_MANAGE_CUSTOM_DOCUMENTS</td>
<td>Manage Custom Documents</td>
</tr>
<tr>
<td>HFM_TASK_MANAGE_DATA_ENTRY_FORMS</td>
<td>Manage Data Forms</td>
</tr>
<tr>
<td>HFM_TASK_MANAGE_METADATA</td>
<td>Manage Metadata</td>
</tr>
<tr>
<td>HFM_TASK_MANAGE_OWNERSHIP</td>
<td>Manage Ownership</td>
</tr>
<tr>
<td>HFM_TASK_MANAGE_RULES</td>
<td>Manage Rules</td>
</tr>
<tr>
<td>HFM_TASK_NONE</td>
<td>For internal use.</td>
</tr>
<tr>
<td>HFM_TASK_OPEN_APPLICATION</td>
<td>Open Application</td>
</tr>
<tr>
<td>HFM_TASK_PIVOT_DATA</td>
<td>Pivot Data</td>
</tr>
<tr>
<td>HFM_TASK_RETRIEVE_DATA</td>
<td>Retrieve Data</td>
</tr>
<tr>
<td>HFM_TASK_RUN_DYNAMIC_INTERCOMPANY_MATCHING_REPORT</td>
<td>Run Dynamic Intercompany Matching Report</td>
</tr>
<tr>
<td>HFM_TASK_SECURITY.Configure_ACCESS</td>
<td>Configure Access</td>
</tr>
<tr>
<td>HFM_TASK_SECURITY_DEFINE_SECURITY_CLASSES</td>
<td>Define Security Classes</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------------------------</td>
<td>-------------------------------------------------</td>
</tr>
<tr>
<td>HFM_TASK_SECURITY_DEFINE_USERS_AND_GROUPS</td>
<td>Define Users and Groups</td>
</tr>
<tr>
<td>HFM_TASK_SYSTEM_REPORTS</td>
<td>System Reports</td>
</tr>
<tr>
<td>HFM_TASK_SYSTEM_REPORTS_SAVE_REMOTELY</td>
<td>Save File Remotely</td>
</tr>
<tr>
<td>HFM_TASK_TASK_AUDIT</td>
<td>Task Audit</td>
</tr>
<tr>
<td>HFM_TASK_TASK_AUTOMATION</td>
<td>Task Automation</td>
</tr>
<tr>
<td>HFM_TASK_USER_PREFERENCES</td>
<td>User Preferences</td>
</tr>
<tr>
<td>HFM_TASK_USERS_ON_SYSTEM</td>
<td>Users on System</td>
</tr>
<tr>
<td>HFM_TASK_WEBGRID_WRITE_BACK</td>
<td>Enable write back in Web Grid</td>
</tr>
</tbody>
</table>

**Web Constants**

The HFMConstants type library includes the following categories of constants for the Web components.

- “Data Explorer Task Constants” on page 888
- “Data Explorer Process Management Constants” on page 889
- “Data Grid Definition Constants” on page 890
- “Data Grid Member Expansion Mode Constants” on page 890
- “Data Grid Dimension Expansion Mode Constants” on page 891
- “Data Grid Transaction Information Constants” on page 891
- “Data Information Display Constants” on page 891
- “Data Display Page Constant” on page 892
- “Member Display Constants” on page 892
- “Metadata Information Constants” on page 892
- “Document Type Constants” on page 893
- “Document File Type Constants” on page 894

**Note:** There is also an enum named WEBOM_POV_SELECTTYPES, but this is for internal use only.

**Data Explorer Task Constants**

The following constants represent Data Explorer tasks.
### Table 146  
**tagWEBOM_DATAGRID_TASKMASK_ENUM Enumeration**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DATAGRID_TASK_ALLOCATE</td>
<td>Allocate.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_CALCULATE</td>
<td>Calculate.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_CALCULATECONTRIBUTION</td>
<td>Calculate contribution.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_CELLADJUSTMENTS</td>
<td>Cell adjustments.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_CELLINFORMATION</td>
<td>Cell information.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_CELLLINEITEMDETAIL</td>
<td>Line item detail.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_CELTEXT</td>
<td>Cell text.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_CONsolidate</td>
<td>Consolidate.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_CONsolidateALL</td>
<td>Consolidate all.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_CONsolidateALLWITHDATA</td>
<td>Consolidate all with data.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_DESTINATIONTRANSACTIONS</td>
<td>Destination transactions.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_ENTITYDETAILREPORT</td>
<td>Entity detail report.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_FORCECALCULATE</td>
<td>Force calculate.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_FORCECALCULATECONTRIBUTION</td>
<td>Force calculate contribution.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_FORCETRANSLATE</td>
<td>Force translate.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_ICTRANSACTIONREPORT</td>
<td>Intercompany transaction report.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_LOCK</td>
<td>Lock subcube.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_MANAGEPROCESS</td>
<td>Manage process.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_SOURCETRANSACTIONS</td>
<td>Source transactions.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_TRANSLATE</td>
<td>Translate.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TASK_UNLOCK</td>
<td>Unlock subcube.</td>
</tr>
</tbody>
</table>

### Data Explorer Process Management Constants

The following constants represent process management actions for data grids.

### Table 147  
**tagWEBOM_DATAGRID_PROCESSFLOWACTION_ENUM Enumeration**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DATAGRID_PROCESSFLOWACTION_APPROVE</td>
<td>Approve.</td>
</tr>
</tbody>
</table>
### Data Grid Definition Constants

The following constants represent the types of grid definition information that can be returned.

<table>
<thead>
<tr>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DATAGRID_DEFINITION_INFO_ALL</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DEFINITION_INFO_COLDIMS</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DEFINITION_INFO_EXPANSIONS</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DEFINITION_INFO_POV</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DEFINITION_INFO_PROCESS</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DEFINITION_INFO_ROWdims</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DEFINITION_INFO_UISTATE</td>
</tr>
</tbody>
</table>

#### 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_ROWdims**: 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>
<thead>
<tr>
<th>Constant</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DATAGRID_MEMBER_EXPANSION_MODE_NONE</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_MEMBER_EXPANSION_MODE_NO_CHILDREN</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_MEMBER_EXPANSION_MODE_IS_EXPANDED</td>
</tr>
</tbody>
</table>

#### 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.
Data Grid Dimension Expansion Mode Constants

The following constants represent dimension expansion modes applicable to the HFMwDataGrid component.

Table 150  tagWEBOM_DATAGRID_DIMENSION_EXPANSION_MODES Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DATAGRID_DIMENSION_EXPANSION_MODE_COLLAPSED</td>
<td>The dimension is collapsed.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DIMENSION_EXPANSION_MODE_EXPANDED</td>
<td>The dimension is expanded.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DIMENSION_EXPANSION_MODE_INHERITED</td>
<td>The dimension’s state is inherited from the previous row or column.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DIMENSION_EXPANSION_MODE_NONE</td>
<td>No expansion mode applies, as the dimension is the rightmost dimension in the row or column.</td>
</tr>
</tbody>
</table>

Data Grid Transaction Information Constants

The following constants represent transaction information applicable to the HFMwDataGrid component.

Table 151  tagWEBOM_DATAGRID_TRANSACTION_INFO_FLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DATAGRID_TRANSACTION_INFO_ALL</td>
<td>Return all transaction information.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TRANSACTION_INFO_DESTINATION</td>
<td>Return information for destination transactions.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TRANSACTION_INFO_HEADER</td>
<td>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.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_TRANSACTION_INFO_SOURCE</td>
<td>Return information for source transactions.</td>
</tr>
</tbody>
</table>

Data Information Display Constants

The following constants represent the types of information that can be displayed in the data grid.
Table 152  tagWEBOM_DATAGRID_DATA_DISPLAY_FLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DATAGRID_DATA_DISPLAY_CALCSTATUS</td>
<td>Display calculation statuses.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DATA_DISPLAY_CONTROLPANEL_STATUS</td>
<td>Display process control status.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DATA_DISPLAY_DATA</td>
<td>Display data.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DATA_DISPLAY_LASTUSED</td>
<td>Display the last previously displayed type of information.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_DATA_DISPLAY_PROCESSFLOWSTATE</td>
<td>Display process flow states.</td>
</tr>
</tbody>
</table>

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 153  tagWEBOM_DATAGRID_METADATA_DISPLAY_FLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DATAGRID_METADATA_DISPLAY_LABELS</td>
<td>Display labels only.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_METADATA_DISPLAY_DESCRIPTIONS</td>
<td>Display descriptions only.</td>
</tr>
<tr>
<td>WEBOM_DATAGRID_METADATA_DISPLAY_BOTH</td>
<td>Display both labels and descriptions.</td>
</tr>
</tbody>
</table>

Metadata Information Constants

The following constants represent types of metadata information and ways of sorting dimension members.

Table 154  tagWEBOM_METADATA_INFO_FLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_METADATA_INFO_ALL</td>
<td>Represents all of the metadata information represented by the other constants in this enum.</td>
</tr>
<tr>
<td>WEBOM_METADATA_INFO_DESCRIPTION</td>
<td>The description of the specified item.</td>
</tr>
<tr>
<td>WEBOM_METADATA_INFO_ID</td>
<td>The internal ID of the specified item.</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>WEBOM_METADATA_INFO_LABEL</td>
<td>The label of the specified item.</td>
</tr>
<tr>
<td>WEBOM_METADATA_INFO_NUMCHILDREN</td>
<td>The number of children of the specified dimension member.</td>
</tr>
<tr>
<td>WEBOM_METADATA_INFO_SORT_DESCENDING</td>
<td>Sort members in descending order.</td>
</tr>
<tr>
<td>WEBOM_METADATA_INFO_SORTBY_DESC</td>
<td>Sort members by description.</td>
</tr>
<tr>
<td>WEBOM_METADATA_INFO_SORTBY_ID</td>
<td>Sort members by member ID.</td>
</tr>
<tr>
<td>WEBOM_METADATA_INFO_SORTBY_LABEL</td>
<td>Sort members by label.</td>
</tr>
</tbody>
</table>

**Document Type Constants**

The following constants represent document types.

**Table 155  tagDOCUMENTTYPES Enumeration**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DOCTYPE_ALL</td>
<td>All document types represented by this enum.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_CUSTOM</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_FOLDER</td>
<td>Folder.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_INVALID</td>
<td>Invalid document.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_LB</td>
<td>The lower bound of the constants in this enum.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_LINK</td>
<td>Link.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RELATEDCONTENT</td>
<td>Related content.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RPTDATAEXPLORER</td>
<td>Data explorer report.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RPTHYP</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RPTICMATCHBYACCOUNT</td>
<td>Intercompany matching by account report.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RPTICMATCHBYTRANSID</td>
<td>Intercompany matching by ID report.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RPTICMATCHINGTEMPLATE</td>
<td>Intercompany matching template report.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RPTICMONITOR</td>
<td><em>For internal use.</em></td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RPTICTRANSACTION</td>
<td>Intercompany matching by account report.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RPTINTERCOMPANY</td>
<td>Intercompany report.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_RPTJOURNAL</td>
<td>Journal report.</td>
</tr>
<tr>
<td>WEBOM_DOCTYPE_TASK</td>
<td>Task in a workspace.</td>
</tr>
</tbody>
</table>
### Document File Type Constants

The following constants represent the file types of documents.

**Table 156** `tagDOCUMENTFILETYPES Enumeration`

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>WEBOM_DOCFILETYPE_RPTDEF</td>
<td>Report definition file, RPT format.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_RPTXML</td>
<td>Report definition file, XML format.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_RPTHTML</td>
<td>Report definition file, HTML format.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_FORMDEF</td>
<td>Web form.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_FOLDER</td>
<td>Folder.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_XML</td>
<td>XML file.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_CUSTOM</td>
<td>For internal use.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_TASK</td>
<td>Task in a workspace.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_ALL</td>
<td>All file types represented by this enum.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_LB</td>
<td>The lower bound of the constants in this enum.</td>
</tr>
<tr>
<td>WEBOM_DOCFILETYPE_UB</td>
<td>The count of file types that are represented by constants in this enum.</td>
</tr>
</tbody>
</table>

### Extracted File Encoding Constants

The following constants represent encoding types of extracted files.

**Table 157** `tagEXTRACTFILEENCODING Enumeration`

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>EXTRACT_FILE_ENCODING_ANSI</td>
<td>ANSI encoding.</td>
</tr>
<tr>
<td>EXTRACT_FILE_ENCODING_STREAM</td>
<td>Binary data encoding.</td>
</tr>
<tr>
<td>EXTRACT_FILE_ENCODING_UTF8</td>
<td>UTF-8 encoding.</td>
</tr>
</tbody>
</table>
### User Activity Constants

The following constants represent user activities.

#### Table 158  tagUSERACTIVITYCODE Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Activity</th>
</tr>
</thead>
<tbody>
<tr>
<td>NUM_USERACTIVITYCODE</td>
<td>Represents the number of constants in this enumeration.</td>
</tr>
<tr>
<td>USERACTIVITYCODE__LBOUND</td>
<td>Represents the lower bounds of this enumeration.</td>
</tr>
<tr>
<td>USERACTIVITYCODE__UBOUND</td>
<td>Represents the upper bounds of this enumeration.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_ALLOCATE</td>
<td>Allocate.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_APPLICATION_DELETION</td>
<td>Delete Application.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_ATTACH_DOCUMENT</td>
<td>Attach Document.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_CHART_LOGIC</td>
<td>Chart Logic.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_CONsolidATION</td>
<td>Consolidation.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_CUSTOM_LOGIC</td>
<td>Custom Logic.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_AUDIT_PURGED</td>
<td>Purge Data Audit Records.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_CLEAR</td>
<td>Data Clear.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_COPY</td>
<td>Data Copy.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_DELETE_INVALID_RECORDS</td>
<td>Delete Invalid Records.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_ENTRY</td>
<td>Data Entry.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_EXTRACT</td>
<td>Data Extract.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_EXTRACT_HAL</td>
<td>Data Extract via HAL.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_LOAD</td>
<td>Data Load.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_RETRIEVAL</td>
<td>Data Retrieval.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DATA_SCAN</td>
<td>Data Scan.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_DETACH_DOCUMENT</td>
<td>Detach Document.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_EA_DELETE</td>
<td>Extended Analytics - delete.</td>
</tr>
<tr>
<td>Constant</td>
<td>Activity</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------</td>
</tr>
<tr>
<td>USERACTIVITYCODE_EA_EXPORT</td>
<td>Extended Analytics - extract.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_EXTERNAL</td>
<td>Custom activity.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_AUTOMATCHBYACCT</td>
<td>Automatch Intercompany Transactions by Account.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_AUTOMATCHBYID</td>
<td>Automatch Intercompany Transactions by ID.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_CREATE_TRANSACTIONS</td>
<td>Create Intercompany Transaction.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_DELETE_TRANSACTIONS</td>
<td>Delete Intercompany Transaction.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_DELETEALL</td>
<td>Delete All Intercompany Transactions.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_EDIT_TRANSACTIONS</td>
<td>Edit Intercompany Transaction.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_LOCKUNLOCK_ENTITIES</td>
<td>Lock and Unlock Entities.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_MANAGE_PERIODS</td>
<td>Manage Periods.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_MANAGE_REASONCODES</td>
<td>Manage Reason Codes.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_MANUALMATCH_TRANSACTIONS</td>
<td>Manual Match Intercompany Transactions.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_MATCHINGRPTBYACCT</td>
<td>Intercompany Transaction Matching Report by Account.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_MATCHINGRPTBYID</td>
<td>Intercompany Transaction Matching Report by ID.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_POST_TRANSACTIONS</td>
<td>Post Intercompany Transaction.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC_POSTALL</td>
<td>Post All Intercompany Transactions.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC.TRANSACTIONRPT</td>
<td>Intercompany Transaction Report.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC.TRANSACTIONS_EXTRACT</td>
<td>Extract Intercompany Transaction.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC.TRANSACTIONS_LOAD</td>
<td>Load Intercompany Transaction.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC.UNMATCH_TRANSACTIONS</td>
<td>Unmatch Intercompany Transactions.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC.UNMATCHALL</td>
<td>Unmatch All Intercompany Transactions.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC.UNPOST_TRANSACTIONS</td>
<td>Unpost Intercompany Transaction.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IC.UNPOSTALL</td>
<td>Unpost All Intercompany Transactions.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_IDLE</td>
<td>Idle - no current activity.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_JOURNAL_ENTRY</td>
<td>Journal Entry.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_JOURNAL_POSTING</td>
<td>Journal Posting.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_JOURNAL_REPORT</td>
<td>Journal Reports.</td>
</tr>
<tr>
<td>USERACTIVITYCODE_JOURNAL_RETRIEVAL</td>
<td>Journal Retrieval.</td>
</tr>
</tbody>
</table>
### Task Status Constants

The following constants represent task statuses.

**Table 159**  tagUSERACTIVITYSTATUS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>USERACTIVITYSTATUS_ABORTED</td>
<td>Task is aborted.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_COMPLETED</td>
<td>Task is completed.</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_NOT_RESPONDING</td>
<td>Task is not responding.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_PAUSED</td>
<td>Task is paused.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_RUNNING</td>
<td>Task is running.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_SCHEDULED_START</td>
<td>Task is scheduled to start.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_SCHEDULED_STOP</td>
<td>Task is scheduled to stop.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_STARTING</td>
<td>Task is starting.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_STOPPED</td>
<td>Task is stopped.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_STOPPING</td>
<td>Task is stopping.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS_UNDEFINED</td>
<td>Task status is unknown.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS__LBOUND</td>
<td>Represents the lower bounds of this enumeration.</td>
</tr>
<tr>
<td>USERACTIVITYSTATUS__UBOUND</td>
<td>Represents the upper bounds of this enumeration.</td>
</tr>
<tr>
<td>NUM_USERACTIVITYSTATUS</td>
<td>Represents the number of statuses represented by this enumeration.</td>
</tr>
</tbody>
</table>

**Log Severity Constants**

The following constants represent severity levels of logs.

| Table 160  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 899
Lock Status Constants
The following constants represent entity lock statuses for intercompany transactions.

Table 161  tagICMENTITYSTATUS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM_LOCKED</td>
<td>The entity is locked.</td>
</tr>
<tr>
<td>ICM_UNLOCKED</td>
<td>The entity is unlocked.</td>
</tr>
<tr>
<td>ICM_LOCKABLE</td>
<td>The entity can be locked.</td>
</tr>
<tr>
<td>ICM_UNLOCKABLE</td>
<td>The entity can be unlocked.</td>
</tr>
</tbody>
</table>

Period Status Constants
The following constants represent period statuses for intercompany transactions.

Table 162  tagICMPERIODSTATUS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM_CLOSED</td>
<td>The period is closed for intercompany transactions.</td>
</tr>
<tr>
<td>ICM.OPENED</td>
<td>The period is open for intercompany transactions.</td>
</tr>
<tr>
<td>ICM.UNOPENED</td>
<td>The period is has not been opened for intercompany transactions.</td>
</tr>
</tbody>
</table>

Match/Validate Before Post Constants
The following constants represent the valid Match/Validate Before Post settings for a period:
### Table 163  tagICMVBPSETTING Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM_MVBP_OFF</td>
<td>Match/Validate Before Post is on.</td>
</tr>
<tr>
<td>ICM_MVBP_ON</td>
<td>Match/Validate Before Post is off.</td>
</tr>
<tr>
<td>Unmatch All Intercompany Transactions</td>
<td>Restrict.</td>
</tr>
</tbody>
</table>

### Match Option Constants
The following constants represent intercompany matching options:

#### Table 164  tagICMMATCHOPTIONSENUM Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICT_MATCH_REFERENCE</td>
<td>Match by Reference ID.</td>
</tr>
<tr>
<td>ICT_MATCH_TRANSACTIONID</td>
<td>Match by Transaction ID.</td>
</tr>
<tr>
<td>ICT_SUPPRESS_INTER</td>
<td>For internal use.</td>
</tr>
<tr>
<td>ICT_SUPPRESS_INTRA</td>
<td>For internal use.</td>
</tr>
</tbody>
</table>

### Posting Status Constants
The following constants represent posting statuses of intercompany transactions:

#### Table 165  tagICMPOSTSTATUS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM_POSTED</td>
<td>The transaction is posted.</td>
</tr>
<tr>
<td>ICM_UNPOSTED</td>
<td>The transaction is not posted.</td>
</tr>
</tbody>
</table>

### Matching Status Constants
The following constants represent matching statuses of intercompany transactions:

#### Table 166  tagICMMATCHSTATUS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM_MATCH</td>
<td>The transaction is matched.</td>
</tr>
<tr>
<td>ICM_MISMATCH</td>
<td>The transaction is mismatched.</td>
</tr>
<tr>
<td>ICM_UNMATCH</td>
<td>The transaction is unmatched.</td>
</tr>
</tbody>
</table>
Processing Action Constants

The following constants represent types of intercompany processing actions:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM_AUTOMATCHTRANS</td>
<td>Auto-match the specified transactions.</td>
</tr>
<tr>
<td>ICM_DELETETRANS</td>
<td>Delete the specified transactions.</td>
</tr>
<tr>
<td>ICM_DELETETRANS_ALL</td>
<td>Delete all transactions for the specified scenario, year, and period.</td>
</tr>
<tr>
<td>ICM_EDITTRANS</td>
<td>Edit the specified transaction.</td>
</tr>
<tr>
<td>ICM_MANUALMATCHTRANS</td>
<td>Manually match the specified transactions.</td>
</tr>
<tr>
<td>ICM_NEWTRANS</td>
<td>Create a new transaction.</td>
</tr>
<tr>
<td>ICM_POSTTRANS</td>
<td>Post the specified transactions.</td>
</tr>
<tr>
<td>ICM_POSTTRANS_ALL</td>
<td>Post all transactions for the specified scenario, year, and period.</td>
</tr>
<tr>
<td>ICM_SETREASONCODE</td>
<td>Set reason code.</td>
</tr>
<tr>
<td>ICM_UNMATCHTRANS</td>
<td>Unmatch the specified transactions.</td>
</tr>
<tr>
<td>ICM_UNMATCHTRANS_ALL</td>
<td>Unmatch all transactions for the specified scenario, year, and period.</td>
</tr>
<tr>
<td>ICM_UNPOSTTRANS</td>
<td>Unpost the specified transactions.</td>
</tr>
<tr>
<td>ICM_UNPOSTTRANS_ALL</td>
<td>Unpost all transactions for the specified scenario, year, and period.</td>
</tr>
</tbody>
</table>

Transaction Load Mode Constants

The following constants represent modes for loading intercompany transactions:

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICT_LOAD_MERGE</td>
<td>Load the transactions, merging data for existing transactions.</td>
</tr>
<tr>
<td>ICT_LOAD_REPLACE</td>
<td>Load the transactions, replacing data for existing transactions.</td>
</tr>
<tr>
<td>ICT_SCAN_MERGE</td>
<td>Scan the transactions, merging data for existing transactions.</td>
</tr>
<tr>
<td>ICT_SCAN_REPLACE</td>
<td>Scan the transactions, replacing data for existing transactions.</td>
</tr>
</tbody>
</table>

Filtering and Sorting Options

The following constants represent filtering and sorting options for intercompany transactions:
Table 169  tagICM_MONITOR_FILTER_SORT_FLAGS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM_MONITOR_FILTER_LOCKABLE</td>
<td>Lockable lock status filter.</td>
</tr>
<tr>
<td>ICM_MONITOR_FILTER_LOCKED</td>
<td>Locked lock status filter.</td>
</tr>
<tr>
<td>ICM_MONITOR_FILTER_NOTSTARTED</td>
<td>Not Started process status filter.</td>
</tr>
<tr>
<td>ICM_MONITOR_FILTER_STARTED</td>
<td>Started process status filter.</td>
</tr>
<tr>
<td>ICM_MONITOR_FILTER_UNLCKABL</td>
<td>Unlockable lock status filter.</td>
</tr>
<tr>
<td>ICM_MONITOR_NO_SORT</td>
<td>Do not sort.</td>
</tr>
<tr>
<td>ICM_MONITOR_PROC_SORT</td>
<td>Sort by process status.</td>
</tr>
<tr>
<td>ICM_MONITOR_SORT</td>
<td>Sort by locking status.</td>
</tr>
<tr>
<td>ICM_MONITOR_SORT_DESC</td>
<td>Sort in descending order.</td>
</tr>
</tbody>
</table>

**Process Status Constants**

The following constants represent process statuses.

Table 170  tagICMENTITYPROCESSSTATUS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM_NOTSTARTED</td>
<td>Not Started status.</td>
</tr>
<tr>
<td>ICM_STARTED</td>
<td>Started status.</td>
</tr>
</tbody>
</table>

**Event Constants**

The following constants represent intercompany transaction-related events.

Table 171  tagICEVENTTYPE Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ICM_EVENT_MATCH_RPT_BY_ID</td>
<td>Intercompany Transactions - Matching Report By Transaction ID.</td>
</tr>
<tr>
<td>ICM_EVENT_MONITOR</td>
<td>Monitor Intercompany Transactions.</td>
</tr>
<tr>
<td>ICM_EVENT_PROCESS_TRANS</td>
<td>Process Intercompany Transactions.</td>
</tr>
<tr>
<td>ICM_EVENT_SYSTEM_MATCH_RPT</td>
<td>Intercompany Transaction Report.</td>
</tr>
</tbody>
</table>
### 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>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ENUMSHOWPRIVATE_DONTSHOW</td>
<td>Show only public documents.</td>
</tr>
<tr>
<td>ENUMSHOWPRIVATE_SHOW</td>
<td>Show only private documents.</td>
</tr>
<tr>
<td>ENUMSHOWPRIVATE_SHOWALL</td>
<td>Show both public &amp; private documents.</td>
</tr>
<tr>
<td>ENUMSHOWPRIVATE_ID_LBOUND</td>
<td>Represents the lower bounds of this enumeration.</td>
</tr>
</tbody>
</table>

### Date and Time Format Constants

The following constants represent date and time formats. For example, these are used with `HsvResourceManager.FormattedDateTime`.

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>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFM_DATE_TIME_FORMAT_DD_MM_YY</td>
<td>Represents the format <strong>DD/MM/YYYY</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_DD_MM_YY_HH_MI_SS</td>
<td>Represents the format <strong>DD/MM/YYYY hh:MI:SS TT</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_DD_MM_YY_HHHH_MI_SS</td>
<td>Represents the format <strong>DD/MM/YYYY HH:MI:SS</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_HH_MI_SS</td>
<td>Represents the format <strong>hh:MI:SS TT</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_HHHH_MI_SS</td>
<td>Represents the format <strong>HH:MI:SS</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_MM_DD_YY</td>
<td>Represents the format <strong>MM/DD/YYYY</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_MM_DD_YY_HH_MI_SS</td>
<td>Represents the format <strong>MM/DD/YYYY hh:MI:SS TT</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_MM_DD_YY_HHHH_MI_SS</td>
<td>Represents the format <strong>MM/DD/YYYY HH:MI:SS</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_YY_MM_DD</td>
<td>Represents the format <strong>YYYY/MM/DD</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_YY_MM_DD_HH_MI_SS</td>
<td>Represents the format <strong>YYYY/MM/DD hh:MI:SS TT</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_YY_MM_DD_HHHH_MI_SS</td>
<td>Represents the format <strong>YYYY/MM/DD HH:MI:SS</strong>.</td>
</tr>
<tr>
<td>HFM_DATE_TIME_FORMAT_TOTAL</td>
<td>Returns a count of the constants in this enumeration.</td>
</tr>
</tbody>
</table>

### Invalid ID Constant

The `tagHFMINVALIDS` class contains the `INVALID_ID` constant. This constant indicates 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>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAXLEN_ANNOTATION</td>
<td>Maximum length of comments in process units.</td>
</tr>
<tr>
<td>MAXLEN_APPDESC</td>
<td>Maximum length of application descriptions.</td>
</tr>
<tr>
<td>MAXLEN_APPLABEL</td>
<td>Maximum length of application labels.</td>
</tr>
<tr>
<td>Constant</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------</td>
<td>------------------------------------------------------------------</td>
</tr>
<tr>
<td>MAXLEN_CALC_ATTRIBUTE</td>
<td>Maximum length of calculation attributes.</td>
</tr>
<tr>
<td>MAXLEN_CLUSTER_NAME</td>
<td>Maximum length of cluster names.</td>
</tr>
<tr>
<td>MAXLEN_CUSTOMALIAS</td>
<td>Max length of custom dimension alias</td>
</tr>
<tr>
<td>MAXLEN_CUSTOMNAME</td>
<td>Max length of custom dimension short name</td>
</tr>
<tr>
<td>MAXLEN_DECIMAL_DISPLAY</td>
<td>Maximum number of characters for a number-based string.</td>
</tr>
<tr>
<td>MAXLEN_DECIMAL_PRECISION</td>
<td>Maximum number of characters to the right of the decimal character.</td>
</tr>
<tr>
<td>MAXLEN_DESCRIPTION_CELL</td>
<td>Maximum length of cell text descriptions.</td>
</tr>
<tr>
<td>MAXLEN_DESCRIPTION_LINEITEM</td>
<td>Maximum length of line item descriptions.</td>
</tr>
<tr>
<td>MAXLEN_ELEMENTDESC</td>
<td>Maximum length of metadata descriptions.</td>
</tr>
<tr>
<td>MAXLEN_ELEMENTDESC_PER</td>
<td>Maximum length of Period dimension member descriptions.</td>
</tr>
<tr>
<td>MAXLEN_ELEMENTLABEL</td>
<td>Maximum length of metadata labels.</td>
</tr>
<tr>
<td>MAXLEN_FDMAPPNAME</td>
<td>Maximum length of FDM application name application setting.</td>
</tr>
<tr>
<td>MAXLEN_ITEMID_STRING</td>
<td>For internal use.</td>
</tr>
<tr>
<td>MAXLEN_SERVERNAME</td>
<td>Maximum length of registered application server names.</td>
</tr>
<tr>
<td>MAXLEN_SQLTABLENAME</td>
<td>Maximum length of table names in databases.</td>
</tr>
<tr>
<td>MAXLEN_USERDEF_ATTRIB</td>
<td>Maximum length of user-defined attributes of dimension members.</td>
</tr>
<tr>
<td>MAXLEN_VALUE_LINEITEM</td>
<td>Maximum length of line item values.</td>
</tr>
<tr>
<td>MAXLEN_XBRL_TAGS</td>
<td>Maximum length of XBRL tags in Account dimension members.</td>
</tr>
<tr>
<td>MAXLEN_YEAR</td>
<td>Maximum length of years.</td>
</tr>
</tbody>
</table>

### Aggregation Type Constants

The following table represents the types of aggregation.

**Table 175  tagHSV_AGGREGATION_TYPE Enumeration**

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSV_AGGR_TYPE_ADD</td>
<td>Aggregation by addition.</td>
</tr>
<tr>
<td>HSV_AGGR_TYPE_DIVIDE</td>
<td>Aggregation by division.</td>
</tr>
<tr>
<td>HSV_AGGR_TYPE_INVALID</td>
<td>Invalid aggregation type.</td>
</tr>
<tr>
<td>HSV_AGGR_TYPE_MULTIPLY</td>
<td>Aggregation by multiplication.</td>
</tr>
</tbody>
</table>
### 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 176 tagMEMBERIDLIMITS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>MEMBERHIGHESTAVAILABLE</td>
<td>The upper limit</td>
</tr>
<tr>
<td>MEMBERLOWESTAVAILABLE</td>
<td>The lower limit</td>
</tr>
</tbody>
</table>

### Number Defaults Constants

The following constants represent Financial Management’s defaults for certain properties of numbers.

#### Table 177 tagPRECISIONCONSTANTS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DEFAULT_NUM_DECIMALS</td>
<td>The default number of digits to the right of decimal characters.</td>
</tr>
<tr>
<td>DEFAULT_SCALE</td>
<td>The default scale.</td>
</tr>
</tbody>
</table>

### 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.
### 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`.

### Validation Type Constants

The following constants represent various types of information that can be validated.
<table>
<thead>
<tr>
<th>Constant</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>HFM_VALIDATIONTYPE_CUSTOMDIM_ALIAS</td>
<td>Validate whether the string contains a valid custom dimension alias</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_CUSTOMDIM_SHORTNAME</td>
<td>Validate whether the string contains a valid custom dimension short name</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_DOCUMENTNAME</td>
<td>Validate whether the string represents a valid document name.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_JOURNAL_GROUP_LABEL</td>
<td>Validate whether the string represents a valid journal group label.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_JOURNAL_LINEITEM_DESCRIPTION</td>
<td>Validate whether the string represents a valid description for a journal line item.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_JOURNALDESCRIPTION</td>
<td>Validate whether the string represents a valid journal description.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_JOURNALNAME</td>
<td>Validate whether the string represents a valid journal label.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_LBOUND</td>
<td>Represents the lower bounds of the total number of available validation types.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_LINEITEM_DESCRIPTION</td>
<td>Validate whether the string represents a valid description for a cell line item.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_LOADEXTRACT_DELIMITER</td>
<td>Validate whether the string represents a valid delimiter for a load or extract file.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_MEMBERDESCRIPTION</td>
<td>Validate whether the string represents a valid description of a dimension member.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_MEMBERLABEL</td>
<td>Validate whether the string represents a valid dimension member label.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_PROFILE_FILE</td>
<td>For internal use.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_REPORTDESCRIPTION</td>
<td>Validate whether the string represents a valid description of a report.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_REPORTNAME</td>
<td>Validate whether the string represents a valid report label.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_SECURITYCLASS</td>
<td>Validate whether the string represents a valid security class label.</td>
</tr>
<tr>
<td>HFM_VALIDATIONTYPE_USERNAME</td>
<td>Validate whether the string represents a valid username.</td>
</tr>
</tbody>
</table>

**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 181  tagMODULEIDS Enumeration

<table>
<thead>
<tr>
<th>Constant</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODULEID_ADMINISTRATION</td>
<td>Administration</td>
</tr>
<tr>
<td>MODULEID_AUTO_MATCH</td>
<td>Intercompany Transactions - AutoMatch</td>
</tr>
</tbody>
</table>

908  Constants: The HFMConstants Type Library
<table>
<thead>
<tr>
<th>Constant</th>
<th>Module</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODULEID_CELL_HISTORY</td>
<td>Cell History</td>
</tr>
<tr>
<td>MODULEID_CLOSE_APPLICATION</td>
<td>Close Application</td>
</tr>
<tr>
<td>MODULEID_CONsolidation</td>
<td>Consolidation</td>
</tr>
<tr>
<td>MODULEID_CREATE_APPLICATION</td>
<td>Create Application</td>
</tr>
<tr>
<td>MODULEID_DATA_AUDIT</td>
<td>Data Audit</td>
</tr>
<tr>
<td>MODULEID_DATA_ENTRY_FORMS</td>
<td>Data Forms</td>
</tr>
<tr>
<td>MODULEID_DATABASE_MANAGEMENT</td>
<td>Database Management</td>
</tr>
<tr>
<td>MODULEID_DELETE_APPLICATION</td>
<td>Delete Application</td>
</tr>
<tr>
<td>MODULEID_DOCUMENT_EXTRACT</td>
<td>Document Extract</td>
</tr>
<tr>
<td>MODULEID_DOCUMENTS</td>
<td>Documents</td>
</tr>
<tr>
<td>MODULEID_EDIT_INTERCOMPANY_TRANSACTION</td>
<td>Edit Intercompany Transaction</td>
</tr>
<tr>
<td>MODULEID_EDIT_JOURNAL_PAGE</td>
<td>Edit Journals</td>
</tr>
<tr>
<td>MODULE_EDIT_TEMPLATE_PAGE</td>
<td>Edit Journal Templates</td>
</tr>
<tr>
<td>MODULEID_EMPTY_WORKSPACE</td>
<td>Empty Workspace</td>
</tr>
<tr>
<td>MODULEID_ERROR</td>
<td>Error</td>
</tr>
<tr>
<td>MODULEID_EXPLORE_DATA</td>
<td>Explore Data</td>
</tr>
<tr>
<td>MODULEID_EXTENDED_ANALYTICS</td>
<td>Extended Analytics</td>
</tr>
<tr>
<td>MODULEID_EXTRACT_APPLICATION_ELEMENTS</td>
<td>Extract Application Elements</td>
</tr>
<tr>
<td>MODULEID_EXTRACT_DATA</td>
<td>Extract Data</td>
</tr>
<tr>
<td>MODULEID_EXTRACT_JOURNALS</td>
<td>Extract Journals</td>
</tr>
<tr>
<td>MODULEID_EXTRACT_MEMBERLISTS</td>
<td>Extract Member Lists</td>
</tr>
<tr>
<td>MODULEID_EXTRACT_METADATA</td>
<td>Extract Metadata</td>
</tr>
<tr>
<td>MODULEID_EXTRACT_REGIONS</td>
<td>Extract Regions</td>
</tr>
<tr>
<td>MODULEID_EXTRACT_RULES</td>
<td>Extract Rules</td>
</tr>
<tr>
<td>MODULEID_EXTRACT_SECURITY</td>
<td>Extract Security</td>
</tr>
<tr>
<td>MODULEID_EXTRACT_TRANSACTIONS</td>
<td>Extract Intercompany Transactions</td>
</tr>
<tr>
<td>MODULEID_FAVORITES</td>
<td>Favorites</td>
</tr>
<tr>
<td>MODULEID_HAL</td>
<td>Oracle Hyperion Application Link</td>
</tr>
<tr>
<td>Constant</td>
<td>Module</td>
</tr>
<tr>
<td>---------------------------------------------------------</td>
<td>---------------------------------------------</td>
</tr>
<tr>
<td>MODULEID_HFM</td>
<td>Financial Management</td>
</tr>
<tr>
<td>MODULEID_HOME</td>
<td>Home</td>
</tr>
<tr>
<td>MODULEID_IC_TRANSACTIONS</td>
<td>Intercompany Transactions</td>
</tr>
<tr>
<td>MODULEID_ICT_FILTER</td>
<td>Intercompany Transactions Filter</td>
</tr>
<tr>
<td>MODULEID_ICT_REPORT</td>
<td>Intercompany Transactions Report</td>
</tr>
<tr>
<td>MODULEID_ICTRANS_MATCHING_REPORT_BY_ACCOUNT</td>
<td>Intercompany Transactions - Matching Report By Account</td>
</tr>
<tr>
<td>MODULEID_ICTRANS_MATCHING_REPORT_BY_TRANSACTION_ID</td>
<td>Intercompany Transactions - Matching Report By Transaction ID</td>
</tr>
<tr>
<td>MODULEID_IMPORT_FROM_EXCEL</td>
<td>Import From Excel</td>
</tr>
<tr>
<td>MODULEID_JOURNALS</td>
<td>Journals</td>
</tr>
<tr>
<td>MODULEID_JOURNALS_CREATE_NEW_JOURNAL</td>
<td>Create Journal</td>
</tr>
<tr>
<td>MODULEID_JOURNALS_FILTER</td>
<td>Journal Filters</td>
</tr>
<tr>
<td>MODULEID_JOURNALS_REPORT</td>
<td>Journal Report</td>
</tr>
<tr>
<td>MODULEID_LCM</td>
<td>Lifecycle Management</td>
</tr>
<tr>
<td>MODULEID_LINK</td>
<td>Link</td>
</tr>
<tr>
<td>MODULEID_LOAD_APPLICATION_ELEMENTS</td>
<td>Load Application Elements</td>
</tr>
<tr>
<td>MODULEID_LOAD_DATA</td>
<td>Load Data</td>
</tr>
<tr>
<td>MODULEID_LOAD_DOCUMENTS</td>
<td>Load Documents</td>
</tr>
<tr>
<td>MODULEID_LOAD_JOURNALS</td>
<td>Load Journals</td>
</tr>
<tr>
<td>MODULEID_LOAD_LOCAL_REPORT</td>
<td>Load Reports</td>
</tr>
<tr>
<td>MODULEID_LOAD_MEMBERLISTS</td>
<td>Load Member Lists</td>
</tr>
<tr>
<td>MODULEID_LOAD_METADATA</td>
<td>Load Metadata</td>
</tr>
<tr>
<td>MODULEID_LOAD_REGIONS</td>
<td>Load Regions</td>
</tr>
<tr>
<td>MODULEID_LOAD_RULES</td>
<td>Load Rules</td>
</tr>
<tr>
<td>MODULEID_LOAD_SECURITY</td>
<td>Load Security</td>
</tr>
<tr>
<td>MODULEID_LOAD_TRANSACTIONS</td>
<td>Load Intercompany Transactions</td>
</tr>
<tr>
<td>MODULEID_LOCK_AND_UNLOCK_ENTITIES</td>
<td>Lock and Unlock Entities</td>
</tr>
<tr>
<td>MODULEID_LOGOFF</td>
<td>Logoff</td>
</tr>
<tr>
<td>MODULEID_LOGON</td>
<td>Logon</td>
</tr>
<tr>
<td>Constant</td>
<td>Module</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>-------------------------------------------------------------</td>
</tr>
<tr>
<td>MODULEID_MANAGE_EPU</td>
<td>Manage Equity Pickup</td>
</tr>
<tr>
<td>MODULEID_MANAGE_GROUPS</td>
<td>Manage Groups</td>
</tr>
<tr>
<td>MODULEID_MANAGE_IC_PERIODS</td>
<td>Manage Intercompany Periods</td>
</tr>
<tr>
<td>MODULEID_MANAGE_IC_REASON_CODES</td>
<td>Manage Intercompany Transaction Reason Codes</td>
</tr>
<tr>
<td>MODULEID_MANAGE_JOURNALTEMPLATES</td>
<td>Manage Journal Templates</td>
</tr>
<tr>
<td>MODULEID_MANAGE_PERIODS</td>
<td>Manage Periods</td>
</tr>
<tr>
<td>MODULEID_MANAGE_SERVERS_AND_APPLICATIONS</td>
<td>Manage Servers and Applications</td>
</tr>
<tr>
<td>MODULEID_MANAGE_TASKFLOWS</td>
<td>Manage Taskflows</td>
</tr>
<tr>
<td>MODULEID_MEMBER_SELECTOR</td>
<td>Member Selector</td>
</tr>
<tr>
<td>MODULEID_MONITOR_INTERCOMPANY_TRANSACTIONS</td>
<td>Monitor Intercompany Transactions</td>
</tr>
<tr>
<td>MODULEID_NEW_FOLDER</td>
<td>New Folder</td>
</tr>
<tr>
<td>MODULEID_NEW_INTERCOMPANY_TRANSACTION</td>
<td>New Intercompany Transaction</td>
</tr>
<tr>
<td>MODULEID_OFFICE_ADDIN</td>
<td>Office Add-in</td>
</tr>
<tr>
<td>MODULEID_OWNERSHIP_MANAGEMENT</td>
<td>Manage Ownership</td>
</tr>
<tr>
<td>MODULEID_PREFERENCES</td>
<td>Preferences</td>
</tr>
<tr>
<td>MODULEID_PROCESS_CONTROL_PANEL</td>
<td>Process Control Panel</td>
</tr>
<tr>
<td>MODULEID_PROCESS_INTERCOMPANY_TRANSACTIONS</td>
<td>Process Intercompany Transactions</td>
</tr>
<tr>
<td>MODULEID_PROCESS_JOURNALS</td>
<td>Process Journals</td>
</tr>
<tr>
<td>MODULEID_PROCESS_MANAGEMENT</td>
<td>Process Management</td>
</tr>
<tr>
<td>MODULEID_REGISTER_APPLICATION</td>
<td>Register Application with Oracle Hyperion Shared Services</td>
</tr>
<tr>
<td>MODULEID_REGISTER_SMARTVIEW_PROVIDER</td>
<td>Register Smart View Provider</td>
</tr>
<tr>
<td>MODULEID_RELATED_CONTENT</td>
<td>Related Content</td>
</tr>
<tr>
<td>MODULEID_REPORTS</td>
<td>Reports</td>
</tr>
<tr>
<td>MODULEID_RUNNING_TASKS</td>
<td>Running Tasks</td>
</tr>
<tr>
<td>MODULEID_SAVE_DOCUMENTS</td>
<td>Save Documents</td>
</tr>
<tr>
<td>MODULEID_SAVE_REPORT_LOCAL</td>
<td>Reports - Save Local</td>
</tr>
<tr>
<td>MODULEID_SCAN_ICTRANSACTIONS</td>
<td>Scan Intercompany Transactions</td>
</tr>
<tr>
<td>MODULEID_SELECT_APPLICATION</td>
<td>Select Application</td>
</tr>
<tr>
<td>Constant</td>
<td>Module</td>
</tr>
<tr>
<td>----------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>MODULEID_SELECT_CLUSTER</td>
<td>Select Cluster</td>
</tr>
<tr>
<td>MODULEID_SUBMISSION_PHASE</td>
<td>Process Management Submission Phase</td>
</tr>
<tr>
<td>MODULEID_SYSTEM_MESSAGE_DETAILS</td>
<td>System Message Details</td>
</tr>
<tr>
<td>MODULEID_SYSTEM_MESSAGES</td>
<td>System Messages</td>
</tr>
<tr>
<td>MODULEID_TASK_AUDIT</td>
<td>Task Audit</td>
</tr>
<tr>
<td>MODULEID_TASK_AUTOMATION</td>
<td>Task Automation</td>
</tr>
<tr>
<td>MODULEID_UNKNOWN</td>
<td>Unknown Module</td>
</tr>
<tr>
<td>MODULEID_UNMATCH_TRANSACTION</td>
<td>Intercompany Transaction - Unmatch</td>
</tr>
<tr>
<td>MODULEID_USERS_ON_SYSTEM</td>
<td>Users on System</td>
</tr>
<tr>
<td>MODULEID_VIEW_TASKFLOW_STATUS</td>
<td>View Taskflow Status</td>
</tr>
<tr>
<td>MODULEID_WEB_DATA_ENTRY_FORM_BUILDER</td>
<td>Data Form Builder</td>
</tr>
<tr>
<td>MODULEID_WORKSPACE</td>
<td>Workspaces</td>
</tr>
<tr>
<td>MODULEID_WORKSPACE_PREFERENCES</td>
<td>Workspace Preferences</td>
</tr>
</tbody>
</table>
In This Appendix

Communicating with the HTTP Listener ................................................................. 913
Format of Input Parameters .................................................................................. 914
Application Tag .................................................................................................... 915
Input Parameter Headers ....................................................................................... 915
ExtractType .......................................................................................................... 915
PushOption ........................................................................................................... 915
Element Selections for all Dimension ................................................................. 916
Input Parameter File Example .............................................................................. 916
Response Message ............................................................................................... 917

One of the methods for exporting data from Financial Management for use in external applications is through the Extended Analytics component. Extended Analytics allows users to export selected data to an external Star Schema database for use by Oracle Essbase and other products to perform a variety of functions. Previously, the Extended Analytics extract could only be performed using Financial Management or as a COM-based API call. Now, the HTTP listener allows Extended Analytics tasks to be launched through an HTTP request.

**Communicating with the HTTP Listener**

The HTTP listener is located here:

http://<web server>/hfm/EIE/EIEListener.asp?
app_name=HFMAppName&action=launch_EA_extract

The caller needs to specify the following in the header section of the HTTP request:

L"awb-sso-token"

The caller also needs to obtain the SSO token by logging in Financial Management or other services and then pass in the SSO token with the following HTTP request:

L"CONTENT_TYPE"

This should be set to “text/xml”
Format of Input Parameters

The input parameters should be sent in the HTTP request stream using the format described below:

```xml
<applicatoin.name="AppName" action="launch_EA_extract">  
<parameters>  
!ExtractType=[see below for options]  
!PushOption=[see below for options]  
!DSN=[name of DSN to use]  
!Prefix=[prefix used for output database tables]  
!ExcludeDynamicAccounts=[Y/N]  
!Scenario  
[Labels for selected scenarios, one selection per line]  
!Year  
[Labels for selected years, one selection per line]  
!Period  
[Labels for selected periods, one selection per line]  
!View  
[Labels for selected views, one selection per line]  
!Entity  
[Labels for selected parent and entity pairs, one selection per line]  
!Value  
[Labels for selected values, one selection per line]  
!Account  
[Labels for selected accounts, one selection per line]  
!ICP  
[Labels for selected ICPs, one selection per line]  
!Custom1  
[Labels for selected Custom1, one selection per line]  
!Custom2  
[Labels for selected Custom2, one selection per line]  
!Custom3  
[Labels for selected Custom3, one selection per line]  
!Custom4
```
Application Tag

The attribute “name” is used to indicate the name of Financial Management application. The attribute “action” should always be set to “launch_EA_extract” for using this API.

Input Parameter Headers

The lines start with a “!” are section headers. The order of the headers can be swapped. The headers are also not case sensitive.

ExtractType

The options are:

```
enum EA_EXTRACT_TYPE_FLAGS
{
    EA_EXTRACT_TYPE_STANDARD = 0,
    EA_EXTRACT_TYPE_METADATA_ALL = 1,
    EA_EXTRACT_TYPE_METADATA_SELECTED = 2,
    EA_EXTRACT_TYPE_SQL_AGG = 3,
    EA_EXTRACT_TYPE_ESSBASE = 4,
    EA_EXTRACT_TYPE_WAREHOUSE = 5,
    EA_EXTRACT_TYPE_FLATFILE = 6,
}
```

Exactly one integer associated with the desired option should be used.

PushOption

The options are:

```
enum SS_PUSH_OPTIONS
{
    ssCREATE = 0,
    ssUPDATE = 1
```
Exactly one integer associated with the desired option should be used.

**Element Selections for all Dimension**

Labels for the selected elements should be used. Financial Management default and custom member lists are supported. If a member list, either custom or predefined list, are used, `{ }` brackets should be used to enclose the entire line. There should be exactly one section per line. The selections should be XML encoded. For example:

```xml
!Account
{GroMar.[Decendants]}
{Asstot.[Base]}
{TOP.[Decendants]}

Shares

For element selections for Entity dimension, the user may want to provide the parent information for some entities. For example:

```xml
!Entity
TOP.A1
GROUPA.A1
{GROUPB.[Base]}
{GROUPS.[Hierarchy]}
```

**Input Parameter File Example**

```xml
!ExtractType=0
!PushOption=0
!DSN=StarSchema_SQL
!Prefix=AllSales
!ExcludeDynamicAccounts=Y
!Scenario
Actual
!Year
2011
2012
!Periods
```

HTTP Listener for Extended Analytics
January
February
March

Views
<br>&lt;Scenario View&gt;

Entity
United States.[Base]
Europe.[Base]
ASIA.[Base]

Value
<br>&lt;Entity Currency&gt;

Account
TotalRevenues.[Hierarchy]

ICP
[ICP None]

Custom1
[Base]

Custom2
[Base]

Custom3
[Base]

Custom4
[Base]

**Response Message**

An XML similar to the one below returned from Financial Management after the Extended Analytics task is launched. Since Extended Analytics task is asynchronous, the task is most likely still in progress when the caller receive the response message. The message contains `<taskID>`, which can be used to track the progress through Running Task in the HsvSystemInfo module.

```xml
<status>
<code>-3</code>
<taskID>-1</taskID>
<message>Extended Analytics Extract failed for reason: %s</message>
```
The `<code>` tag contains 0 if the task was successfully launched. It has a negative number if the task could not be launched.

Current valid values include:

- 0 Successfully launched
- 1 User does not have right to start this task
- 2 Failed to open HFM application
- 3 Invalid input parameter file
- 4 Invalid action name
- 5 Another Extended Analytics extract by this user is still in progress
- 6 Errors in input parameter file at line X
- 7 Missing parameter section X in the input parameter file
- 8 Other errors. Refer to the Extended Analytics and HFM logs for more information.

The `<taskID>` tag -1 if the `<code>` tag contains an error code. Otherwise, `<taskID>` contains a positive integer for the caller to track the progress of the Extended Analytic task.
This Appendix lists changes to the Financial Management object model since release 1.0.

**Changes for Release 11.1.2.3**

These topics list changes since release 11.1.2.2. These Type Libraries were updated:

“HsvCalculate Type Library Overview” on page 85
“HsvData Type Library Overview” on page 78
“HsvDQI Type Library Overview” on page 117
HsvCalculate Type Library
The method “EnumOnDemandRules” on page 397 was added.
The method “ExecuteOnDemandRules” on page 397 was added.
The method “GetEPUInfoEx” on page 400 was added.

HsvDataType Library
The method “ClearDataAuditItemsEx” on page 324 was added.

HsvDQI Type Library
The method “SelectMemberSourceUserDefinedMemberList2” on page 714 was added.

HsvJournals Type Library
The method “EnumJournalGroupsEx” on page 418 was added.

HsvMetadata Type Library
The method “ExtractModuleConfigurations” on page 206 was added.
The method “GetDefaultScaleAndNumDecimal” on page 215 was added.
The method “GetICPEntitiesAggregationWeightEx” on page 218 was added.
The method “LoadModuleConfigurations” on page 229 was added.

HsvSecurityAccess Type Library
The method “DeleteSecurityClasses” on page 470 was added.

HsvSession Type Library
The method “GetECID” on page 193 was added.
**HsvSystemInfo Type Library**

The method “AddTaskToAuditWithAttachment” on page 502 was added.
The method “EnumAuditTasksEx” on page 511 was added.
The method “GetTaskAuditAttachment” on page 529 was added.

**Changes for Release 11.1.2.2.300**

These topics list changes since release 11.1.2.2. These Type Libraries were updated:

“HsxClient Type Library Overview” on page 59
“HsvMetadata Type Library Overview” on page 65
“HsvSecurityAccess Type Library Overview” on page 90
“HsxServer Type Library Overview” on page 63
“HsvSystemInfo Type Library Overview” on page 96

**Changes for Release 11.1.2.2**

These topics list changes since release 11.1.2.1. Many new methods were added to support Extended Dimensionality. See the new methods ending in “ExtDim”. These Type Libraries were updated:

“HsvCalculate Type Library Overview” on page 85
“HsxClient Type Library Overview” on page 59
“HsvData Type Library Overview” on page 78
“HsvDataCubes Type Library Overview” on page 115
“HsvICM Type Library” on page 121
“HsvJournals Type Library Overview” on page 86
“HsvMDArrays Type Library Overview” on page 104
“HsvMetadata Type Library Overview” on page 65
“HsvProcessFlow Type Library Overview” on page 99
“HsvReports Type Library Overview” on page 103
“HsvSecurityAccess Type Library Overview” on page 90
“HsxServer Type Library Overview” on page 63
“HsvSession Type Library Overview” on page 64
“HsvStarSchemaACM Type Library Overview” on page 120
“HsvSystemInfo Type Library Overview” on page 96
“Type Libraries for Loading and Extracting Information” on page 124

In addition, two new Type Libraries were added:

“HfmSliceCOM Type Library” on page 134
“HsvDQI Type Library Overview” on page 117

**Changes for Release 11.1.2.1**

The following topics list changes since release 11.1.2:

**HsvJournals Type Library**

**Hsv SystemInfo Type Library**

**HsvJournals Type Library**

The method “GetEntityJournalsExtDim” on page 421 was added.

**Hsv SystemInfo Type Library**

The method “ExtractTaskAudit2” on page 521 was added.

**Changes for Release 11.1.2**

The following topics list changes since release 11.1.1.3:

**HsvData Type Library**
**HsvJournals Type Library**
**HsvReports Type Library**
**HsvStarSchemaACM Type Library**
**HsvSystemInfo Type Library**
**HsxClient Type Library**

**HsvData Type Library**

The following new methods are for internal use:

- “GetDynamicAccountRulesCache” on page 363
- “GetMultiServerMaxSyncDelayForChanges” on page 364
**HsvJournals Type Library**

The method “GetEntityJournals2” on page 422 was added.

**HsvReports Type Library**

The new method “ValidateMembersAgainstSlice” on page 609 is for internal use.

**HsvStarSchemaACM Type Library**

The following new methods are for internal use:
- “CreateStarSchemaAndReturnTaskID” on page 727
- “CreateStarSchemaFromHTTP” on page 728

**HsvSystemInfo Type Library**

The following new methods are for internal use:
- “EnumActivityUsersForRunningTasks” on page 507
- “GetVBScriptCalcRulesEx” on page 531

**HsxClient Type Library**

The new method “EnumRegisteredClustersOrServers” on page 167 is for internal use.

**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
- HsxServer Type Library
HsvData Type Library

The following methods were added:

- “ExtractDrillableRegions” on page 340
- “ExtractDrillableRegionsByURLNames” on page 340
- “GetAllURLNames” on page 350
- “GetURLByName” on page 373
- “GetURLsForCell” on page 374
- “LoadDrillableRegions” on page 379

HsvJournals Type Library

The method “GetJournal2” on page 425 was added.

HsvMetadata Type Library

The following methods were added:

- “EnumMembersWithAttribValue” on page 245.
- “GetFrequencyID” on page 217

HsvResourceManager Type Library

The following methods were added:

- “GetWindowsDateFormatForLocale” on page 843
- “GetLocaleIdFromLanguageId” on page 841
- “GetFormattedDateTimeForLanguage” on page 837

HsvSecurityAccess Type Library

The following new methods are for internal use:

- “GenerateSecurityReportForBiPub” on page 478
- “SetRulesMode” on page 492
- “GetRulesMode” on page 482

HsvSystemInfo Type Library

The new method “GetFormattedDateTime” on page 525 was added.

The new method “AddTaskToRunningTasksAndUpdatePOV” on page 502 is for internal use.
HsxClient Type Library

The new method “EnumUsersOnSystemEx2” on page 170 was added.

HsxServer Type Library

The new method “GetFileTransfer” on page 188 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 were added:

- “CalcEPU” on page 393
- “GetEPUInfo” on page 400
- “IsEntityAnEPUOwner” on page 402
- The following new methods are for internal use only:
  - LoadCalcManagerRules
  - LoadCalcManagerRules2

HsvDataCubes Type Library

The following methods were changed:

- “BeginEnumerationOfStoredData” on page 676
- “EndEnumerationOfStoredData” on page 676
HsvICM Type Library
The following methods were added:
- “OpenICPeriod2” on page 762
- “UpdatePeriodSettings2” on page 764

HsvMDArrays Type Library
The new method CreateDataIndexListEx is for internal use only.

HsvMetadata Type Library
The following changes were added to the IHsvTreeInfo interface:
- “EnumIDsOfIChildren” on page 241
- “EnumSortedIDsOfIChildren” on page 247
- “EnumPhasedSubmissionStartYears” on page 303
- “GetPhasedSubmissionStartYear” on page 303
- “IsPhasedSubmissionEnabled” on page 309

HsvResourceManager Type Library
The new method GetCurrentHSSRegistrationVersion is for internal use only.

HsvRulesLoadACV Type Library
The new method GetCalcRulesType is for internal use only.

HsvSecurityAccess Type Library
The new method GetCalcRulesType is for internal use only.

HsvSession Type Library
The following new methods are for internal use only:
- “LockMetadataLoadWithSystemChangeCheck” on page 195
- “UnlockMetadataLoad” on page 195
**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 927
- “HsvData Type Library” on page 927
- “HsvSecurityAccess Type Library” on page 927
- “HsvcDataLoad Type Library” on page 928

**HsvSession Type Library**

GetLicenseExpirationStatus was deprecated.

**HsvData Type Library**

The following methods were added:

- GetUnassignedGroups
- SetMinMaxPeriod
- GetStatusUsingPhaseID

Data load options for phased submissions, decimal character, and thousands separator character were added. See Table 64, “Data Load Options,” on page 377.

**HsvSecurityAccess Type Library**

The following methods were added to the HsvSecurityAccess object:

- RenameSecurityClass
- GetUserInfoFromUniqueID2
- EnumUsers3
- EnumUsersWithFilter2
- EnumUsersInSecurityClass3

The following HsvSecurityAccess methods were 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 were added. See “Data Load Options” on page 800.

**Changes for Release 9.3.0.1**

The following topics list changes since release 4.1:

- “Classic Applications” on page 928
- “HsxClient Type Library” on page 928
- “HsxClientUI Type Library” on page 929
- “HsvMetadata Type Library” on page 929
- “HsvData Type Library” on page 930
- “HsvProcessFlow Type Library” on page 931
- “HsvSystemInfo Type Library” on page 931
- “HsvSecurityAccess Type Library” on page 930
- “HsvMDArrays Type Library” on page 931
- “HsvICM Type Library” on page 932
- “HsvMetadataLoadACV Type Library” on page 932
- “HsvcDataLoad Type Library” on page 932
- “HsvSecurityLoadACV Type Library” on page 933

**Classic Applications**

Some methods are supported for only Classic applications. You can determine whether an application is a Classic application using the **HsvSecurityAccess method IsClassicHFMApplication**.

**HsxClient Type Library**

The following methods apply only to Classic applications. For information on similar functionality for Performance Management Architect applications, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide*.

- **CreateApplicationCAS**
- `DeleteApplication`
- `RegisterApplicationCAS`

The following methods were added:

- `IsValidApplication`
- The following new methods are for internal use only:
  - `EnumUserAppPreferences`
  - `UpdateUserAppPreferences`
  - `CreateApplicationCASWithAccessCode`
  - `DeleteApplicationWithAccessCode`
  - `RegisterApplicationCASWithAccessCode`

### HsxClientUI Type Library

`DeleteApplication` now applies only to Classic applications.

### HsvMetadata Type Library

The following changes were made to the HsvMetadata type library:

- The following HsvMetadata object methods apply only to Classic applications. For similar functionality for Performance Management Architect applications, see the *Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide*.
  - `Load`
  - `EnumLoadOptions`
- The following methods were added to the HsvMetadata object:
  - `GetByIndexValidationAccount`
  - `GetPdmAppName`
  - `GetSupportSubmissionPhaseForAccountFlag`
  - `GetSupportSubmissionPhaseForCustom1Flag`
  - `GetSupportSubmissionPhaseForCustom2Flag`
  - `GetSupportSubmissionPhaseForCustom3Flag`
  - `GetSupportSubmissionPhaseForCustom4Flag`
  - `GetSupportSubmissionPhaseForICPFlag`
  - `GetUseSubmissionPhaseFlag`
  - `LoadWithAccessCode` was added to the HsvMetadata object, but are for internal use only:
- `GetSubmissionGroup` was added to the HsvAccounts object.
- `GetSubmissionGroup` was added to the HsvCustom object.
- `GetSubmissionGroup` was added to the HsvICPs object.

## HsvData Type Library

The following methods were added to the HsvData object:

- `GetCalcStatusEx`
- `GetPhaseSubmissionGridForGivenScenarioPeriod`
- `SetPhaseSubmissionGridForGivenScenarioPeriod`
- `DMELoad`

## HsvSecurityAccess Type Library

The following changes were made to the HsvSecurityAccess object:

- The following HsvSecurityAccess object methods apply only to Classic applications. For similar functionality for Performance Management Architect applications, see the Oracle Hyperion Enterprise Performance Management Architect Administrator's Guide.
  - `AddSecurityClass`
  - `DeleteSecurityClass`
  - `InsertDefaultSecurityClass`
  - `SetSecurityClassLabel`
- The following methods were added:
  - `EnumRolesForPrincipal`
  - `EnumRolesForUser`
  - `EnumSecurityClassRightsForPrincipal`
  - `EnumUsersWithFilter`
  - `GetAllSecurityClassRightsForConnectedUser`
  - `IsClassicHFMApplication`

The following methods were added to the IHsvDataSecurity interface:

- `GetProcessUnitAccessRightsAndStateEx`
- `GetProcessUnitAccessRightsEx`
HsvProcessFlow Type Library

The following methods were added to the HsvProcessFlow object:

- GetGroupPhaseFromCell
- GetPhasedSubmissionHistory
- GetPhasedSubmissionState
- GetPhasedSubmissionStateUsingPhaseIDExtDim
- 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 was added.

HsvMDArrays Type Library

The following methods were added to the HsvMDDDataBuffer object:
The following methods were added to the HsvMDDataBufferLite object:

- EraseRecordFromPMBuffer
- GetPMErrorRecordCount
- GetPMRecordCount
- GetRecordFromPMBuffer
- GetRecordFromPMErrorBuffer
- InsertRecordIntoPMBuffer
- InsertRecordIntoPMErrorBuffer

The following methods were added to the HsvICTransactionsData object:

- GetPartnerQueryDimensionMemberIDs
- SetPartnerQueryDimensionMemberIDs

**HsvICM Type Library**

`SavePeriodsSettings` was 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 *Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide*.

- Load method
- LoadOptions property

**Note:** `LoadWithAccessCode` was added, but is for internal use only.

**HsvcDataLoad Type Library**

The `DMELoad` method was added, but is for internal use only.
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 applications, see the Oracle Hyperion Enterprise Performance Management Architect Administrator’s Guide.

Note: LoadWithAccessCode was added, but is for internal use only.

Changes for Release 4.1

The following topics list changes for release 4.1:
- “HsxClient Type Library” on page 933
- “HsxServer Type Library” on page 933
- “HsvSession Type Library” on page 934
- “HsvMetadata Type Library” on page 934
- “HsvData Type Library” on page 934
- “HsvCalculate Type Library” on page 934
- “HsvSecurityAccess Type Library” on page 934
- “HsvSystemInfo Type Library” on page 936
- “HsvICM Type Library” on page 936
- “HsvMDArrays Type Library” on page 937
- “HsvResourceManager Type Library” on page 938

HsxClient Type Library

The following changes were made to the HsxClient object:
- The following methods were added:
  - CreateApplicationCAS
  - EnumProvisioningProjects
  - EnumUsersOnSystemEx
  - RegisterApplicationCAS
- CreateApplication was deprecated - use CreateApplicationCAS.

HsxServer Type Library

The following changes were made to the HsxServer object:
- The GetClusterInfo method was removed.
- The `GetClustersAndServers` method was added.

**HsvSession Type Library**

The `GetLicenseExpirationStatus` method was added to the HsvSession object.

**HsvMetadata Type Library**

The following changes were made to the HsvMetadata type library:

- The `GetCalcAttribute` method was added to the HsvAccounts object.
- The `EnumCurrencies2` method was added to the HsvCurrencies object.
- The following methods were added to the HsvMetadata object:
  - `GetAttributeValue`
  - `TranslateAttributeValueForDisplay`
- The `SupportsEmailAlerting` method was added to the HsvScenarios object.
- The following methods were added to the IHsvTreeInfo interface:
  - `FindByDesc`
  - `FindMatchingMembersFromHierarchyByDesc`
  - `FindMatchingMembersFromHierarchyWildCard`
  - `GetDefaultMemberID`

**HsvData Type Library**

The following methods were added to the HsvData object:

- `GetCalcStatusStatistics`
- `GetMaxCellTextSize`
- `IsValidCellText`

**HsvCalculate Type Library**

The `FindOverlappingConsolidation` method was added to the HsvCalculate object.

**HsvSecurityAccess Type Library**

The following changes were made to the HsvSecurityAccess object:

- Several methods were deprecated. The following table lists these methods and their replacements. If you call a deprecated method, it returns an error:
### Deprecated Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>New method</th>
</tr>
</thead>
<tbody>
<tr>
<td>AddApplicationAdministrator</td>
<td>AddApplicationAdministrator2</td>
</tr>
<tr>
<td>AddOrRemoveApplicationAdministrators</td>
<td>AddOrRemoveApplicationAdministrators2</td>
</tr>
<tr>
<td>AddOrRemoveRolesFromUser</td>
<td>AddOrRemoveRolesFromUser2</td>
</tr>
<tr>
<td>AddOrRemoveUsersFromRole</td>
<td>AddOrRemoveUsersFromRole2</td>
</tr>
<tr>
<td>AddUser</td>
<td>AddUser2</td>
</tr>
<tr>
<td>AddUserToRole</td>
<td>AddUserToRole2</td>
</tr>
<tr>
<td>EnumApplicationAdministrators</td>
<td>EnumApplicationAdministrators2</td>
</tr>
<tr>
<td>EnumUsers</td>
<td>EnumUsers2 and EnumUsersOrGroups</td>
</tr>
<tr>
<td>EnumUsersInRole</td>
<td>EnumUsersInRole2 and EnumUsersInRole3</td>
</tr>
<tr>
<td>EnumUsersInSecurityClass</td>
<td>EnumUsersInSecurityClass2</td>
</tr>
<tr>
<td>GetApplicationAdministratorAccessForAllUsers</td>
<td>GetApplicationAdministratorAccessForAllUsers2</td>
</tr>
<tr>
<td>GetConnectedUser</td>
<td>GetConnectedUser2</td>
</tr>
<tr>
<td>GetRoleAccessForAllUsers</td>
<td>GetRoleAccessForAllUsers2</td>
</tr>
<tr>
<td>GetSecurityClassAccessForAllUsers</td>
<td>GetSecurityClassAccessForAllUsers2</td>
</tr>
<tr>
<td>GetUserAccessForAllRoles</td>
<td>GetUserAccessForAllRoles2</td>
</tr>
<tr>
<td>GetUserAccessForAllSecurityClasses</td>
<td>GetUserAccessForAllSecurityClasses2</td>
</tr>
<tr>
<td>GetUserID</td>
<td>GetUserSID</td>
</tr>
<tr>
<td>GetUserName</td>
<td>GetUserName2</td>
</tr>
<tr>
<td>RemoveApplicationAdministrator</td>
<td>RemoveApplicationAdministrator2</td>
</tr>
<tr>
<td>RemoveUser</td>
<td>RemoveUser2</td>
</tr>
<tr>
<td>RemoveUserFromRole</td>
<td>RemoveUserFromRole2</td>
</tr>
<tr>
<td>SetManySecurityClassRightsForUser</td>
<td>SetManySecurityClassRightsForUser2</td>
</tr>
<tr>
<td>SetSecurityClassRightsForManyUsers</td>
<td>SetSecurityClassRightsForManyUsers2</td>
</tr>
<tr>
<td>SetSecurityClassRightsForUser</td>
<td>SetSecurityClassRightsForUser2</td>
</tr>
</tbody>
</table>

- In addition, these methods were added to the HsvSecurityAccess object:
  - EnumUserClassAccess
&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&n...
The following changes were made to the HsvAdminICM object:

- The following methods were added:
  - CheckSecurityForICExtract
  - GetICEntitiesLockStatus

- The HsvAdminICM method GetLockStatusICEntities was deprecated - use GetICEntitiesLockStatus.

**HsvMDArrays Type Library**

The following changes were made to the HsvMDArrays type library:

- These changes were made to the HsvICTransactionsData object:
  - These methods were added:
    - GetAccessRights
    - GetPartnerAsEntityList
    - GetTransGroupType
    - IsEntityInPartnerAsEntityList
    - SetAccessRights
    - SetPartnerAsEntityList

  - SetAccessRight was removed. To specify a user’s access rights to a transaction, use SetAccessRights.

- These methods were added to the HsvMDDataBuffer object and are for internal use only:
  - GetSortedNature
  - SetSortedNature
  - Sort

- These methods were added to the HsvMDDataBufferLite object and are for internal use only:
  - GetSortedNature
  - SetSortedNature
  - Sort
HsvResourceManager Type Library

The `GetLanguageCountryCodeFromLanguageId` method was added to the HsvResourceManager object.

Changes for Release 4.0

The following topics list changes for release 4.0:

- “HsvSession Type Library” on page 938
- “HsvMetadata Type Library” on page 938
- “HsvData Type Library” on page 939
- “HsvCalculate Type Library” on page 939
- “HsvJournals Type Library” on page 940
- “HsvSystemInfo Type Library” on page 940
- “HsvProcessFlow Type Library” on page 941
- “HsvReports Type Library” on page 941
- “HsvStarSchemaACM Type Library” on page 941
- “HsvICM Type Library” on page 942
- “HsvMDArrays Type Library” on page 942
- “HsvRulesLoadACV Type Library” on page 942
- “HsvcDataLoad Type Library” on page 942
- “HsvMetadataLoadACV Type Library” on page 943
- “HsvResourceManager Type Library” on page 943
- “HsxClient Type Library” on page 943

HsvSession Type Library

The following items were added to the HsvSession type library:

- `ICM` property
- `IsBusy` method
- `IsRunningTasks` method

HsvMetadata Type Library

- The HsvCurrencies object was added. This object represents an application’s currencies. See “HsvCurrencies Object Methods” on page 313.
An option for extracting system accounts was added, and the index numbers for the Values and ICPs options changed to 18 and 19. Extract options are described in Table 60 on page 204.

These items were added to the HsvMetadata object:
- `GetApplicationCurrency` method
- `Currencies` property

The `IsSecurityAsPartnerEnabled` method was added to the HsvEntities object.

The following methods were added to the HsvValues object:
- `GetCurrencyIDFromValueID`
- `GetValueIDFromCurrencyID`

The following changes were made to the IHsvTreeInfo interface:
- The `IEntity` argument was added to `EnumMembers2`.
- The following methods was added:
  - `EnumSortedIDsOfChildren`
  - `EnumSortedMembers`

### HsvData Type Library

The following changes were made to the HsvData object:
- The `pvbDataExistedPriorToClear` argument was added to `ClearInputData`.
- The `vbEnableDetailedLogging` argument was added to `CopyInputDataForMultipleEntities`.

These methods were added:
- `AttachDocumentToCell`
- `DetachDocumentFromCell`
- `GetAttachedDocumentsToCell`
- `GetCountOfAttachedDocumentsToCell`
- `ExtractDataAuditItems`
- `GetStatusEx`
- `SetCells2`
- `SetFileForLoad`
- `StartLoad`

### HsvCalculate Type Library

The following methods were added to the HsvCalculate object:
- `GetDefaultExchangeRate`
- GetConsolidationProgress
- SetVBScriptRules2
- StopConsolidation

**HsvJournals Type Library**

The following methods were added to the HsvJournals type library:

- GenerateRecurring was added to the IHsvJournalsEx interface.
- GetReportData2 was added to the IHsvJournalsReport interface.

**HsvSystemInfo Type Library**

The following changes were made to the HsvSystemInfo object:

- The `pvarbstrModuleNameNames` argument was added to `EnumAuditTasks`.
- The `pvarbstrModuleNameNames` argument was added to `EnumAuditTasks2`.
- These methods were 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 were 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 was added to the HsvReports object.

**HsvStarSchemaACM Type Library**

The following changes were made to the HsvStarSchemaACM type library:

- The **IHsvStarSchemaTemplates** interface was added. See “**IHsvStarSchemaTemplates Interface**” on page 732.
- The following changes were made to the HsvStarSchemaACM object:
  - The following changes were made to the **CreateStarSchema** method:
    - The data types for the arguments that specify the dimension members was changed from Long to Variant. In addition, you can 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 were added.
    - The **vbRunAsynchronous**, **varalssSelectionMethods**, and **plNumDataRecords** arguments were removed.
  - The data type of the **GetAsynchronousTaskStatus** method **plCurrentTask** argument was changed.
  - The **GetExtractLogData** method was added.
For the SS_PUSH_OPTIONS enumeration, the ssCLEAR_ALL_DATA and ssCLEAR_DATA_SYPE constants were removed, and the ssCREATE and ssUPDATE constants were added. See CreateStarSchema ssPushType argument.

- The EA_EXTRACT_TYPE_FLAGS enumeration was added. See CreateStarSchema method’s eaExtractType argument.
- The EA_TASK_STATUS_FLAGS enumeration was added. See the GetAsynchronousTaskStatus method’s plCurrentTask argument.
- The SS_SELECTION_METHOD enumeration was removed.
- The HFMPOVS type was removed.

**HsvICM Type Library**

The HsvICM type library was added. This library exposes Financial Management’s intercompany transaction features. See Chapter 21, “HsvICM Type Library.”

**HsvMDArrays Type Library**

The following changes were made to the HsvMDArrays type library:

- The following methods were added to the HsvMDDataBuffer object:
  - InsertDataAtBeginning
  - InsertDescriptionAtBeginning
  - InsertLineItemsAtBeginning

- The HsvICTransactionsData object was added. This object contains methods that supplement the intercompany transaction features exposed by the HsvICM object. See “HsvICTransactionsData Object Methods” on page 663.

  **Note:** The HsvMetadataSecurityBuffer object was added; it is for internal use only.

**HsvRulesLoadACV Type Library**

The following methods were added to the HsvRulesLoadACV object:

- LoadCalcRules2
- ScriptableLoadCalcRules2

**HsvcDataLoad Type Library**

The following methods were added to the HsvcData object:

- Load2
- SetFileForLoad
HsvMetadataLoadACV Type Library
An option for extracting system accounts was added. See “Metadata Extract Options” on page 786.

HsvResourceManager Type Library
GetCurrentVersionInUserDisplayFormat was added to the HsvResourceManager object.

HsxClient Type Library
The following methods were added to the HsxClient object for internal use only:
- AuthenticateCSSToken
- AuthenticateSecurityAgentCredentials
- AuthenticateUserCredentials

Changes for Release 3.5.1
The following topics list changes for release 3.5.1:
- “HsvMetadata Type Library” on page 943
- “HsvData Type Library” on page 943
- “HsvPOVSelection Type Library” on page 944
- “HsvResourceManager Type Library” on page 944
- “Documentation Errors” on page 944

HsvMetadata Type Library
The GetDefaultHierarchyPosition method was added to the IHsvTreeInfo interface.

HsvData Type Library
The GetDataForAllMetadataCombinations method was added to the HsvData object.
HsvPOVSelection Type Library

The HsvPOVSelection type library was added. This library exposes an insertable control that provides a user interface for selecting dimension members. See Chapter 23, “HsvPOVSelection Type Library.”

HsvResourceManager Type Library

The following methods were added to the HsvResourceManager object:

- GetCurrentVersion
- GetFormattedDateTime
- GetFormattedErrorWithLineFeed
- GetFormattedResourceString
- GetHelpDirectoryForLanguageID
- GetHFMLanguageIdFromUserLanguages
- GetUserDisplayDateTimeFormats
- GetUserLanguageFromHFMLanguageId
- GetWindowsDateFormat

Documentation Errors

The following documentation errors were corrected:

- In “Document File Type Constants” on page 894, WEBOM_DOCFILETYPE_UB was described as representing the upper bound of the constants in the enum. WEBOM_DOCFILETYPE_UB actually represents the count of file types that are represented by constants in the enum.
- In “Document Type Constants” on page 893, WEBOM_DOCTYPE_UB was described as representing the upper bound of the constants in the enum. 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 945
- “HsvMetadata Type Library” on page 945
- “HsvData Type Library” on page 945
- “HsvSystemInfo Type Library” on page 945
- “HsvStarSchemaACM Type Library” on page 946
- “HsvReports Type Library” on page 946
HsxClient Type Library

The following methods were added to the HsxClient object:

- DoesUserHaveCreateApplicationRights
- DoesUserHaveSystemAdminRights
- GetHFMErrorLogRecordSet
- GetSSOTokenUsingWebSecurityAgentCredentials
- GetWebSecurityAgentSettings

HsvMetadata Type Library

The following methods were added to the HsvMetadata object:

- GetApplicationSettingsTimeStamp
- GetConsolidationMethodsTimeStamp
- GetCurrencyTimeStamp

The following methods were added to the IHsvTreeInfo interface:

- GetTreeTimeStamp
- GetItemIDQL
- GetQualifiedLabel

HsvData Type Library

The following methods were added to the HsvData object:

- ClearInvalidData
- GetCellHistory2
- EnumDataAuditItems2ExtDim

HsvSystemInfo Type Library

The following methods were added to the HsvSystemInfo object:

- ClearAuditTasks2
- EnumAuditTasks2
**HsvStarSchemaACM Type Library**

The HsvStarSchemaACM type library was added to this document. HsvStarSchemaACM exposes Extended Analytics features. See Chapter 20, "HsvStarSchemaACM Type Library."

**HsvReports Type Library**

The `lShowPrivateDocs` argument was added to `EnumDocumentsEx`.

**HsvMDArrays Type Library**

The following methods were added to the HsvMDDataBuffer object:

- `GetCheckLineItemDetailsForCaseInsensitiveDuplicates`
- `SetCheckLineItemDetailsForCaseInsensitiveDuplicates`

The following methods were added to the HsvMDDataBufferLite object:

- `GetCheckLineItemDetailsForCaseInsensitiveDuplicates`
- `SetCheckLineItemDetailsForCaseInsensitiveDuplicates`

**HsvMetadataLoadACV Type Library**

Load options for loading system accounts and validating metadata integrity were added.

**Changes for Release 3.4**

The following topics list changes for release 3.4:

- “HsxClient Type Library” on page 947
- “HsxServer Type Library” on page 947
- “HsvSession Type Library” on page 947
- “HsvMetadata Type Library” on page 947
- “HsvData Type Library” on page 948
- “HsvJournals Type Library” on page 948
- “HsvSystemInfo Type Library” on page 948
- “HsvProcessFlow Type Library” on page 949
- “HsvReports Type Library” on page 949
- “HsvJournalLoadACV Type Library” on page 949
- “HsvMDArrays Type Library” on page 949
HsxClient Type Library

The following methods were added to the HsxClient object:

- **DeleteSystemErrors**
- **DisableNewConnections**
- **EnableNewConnections**
- **EnumProhibitConnections**
- **EnumUsersOnSystem**
- **KillUsers**
- **WarnUsersForShutDown**

HsxServer Type Library

The **DeleteXMLErrorsFromDatabase** method was removed from the HsxServer object.

The following members were added to the HsxServer object:

- **CSSEnabled** property
- **DeleteSystemErrors** method

HsvSession Type Library

The **HasUserStatusChanged** method was added to the HsvSession object.

HsvMetadata Type Library

The following methods were added to the HsvScenarios object:

- **GetMissingDataZeroViewForAdjValues**
- **GetMissingDataZeroViewForNonAdjValues**
- **IsConsolidateYTD**

The following methods were added to the IHsvTreeInfo interface:

- **EnumDefaultAncestors**
- **EnumDefaultAncestorsLabels**
- **EnumMembers2**
- **GetAttributeValue**
- **GetDefaultParent**
- **GetDefaultParentLabel**
- **GetItemGeneration**
HsvData Type Library

The following methods were added to the HsvData object:

- GetItemLevel
- ClearDataAuditItems
- EnumDataAuditItems
- FormatNumberToText2
- GetBaseDataForAccount
- GetCellHistory

HsvJournals Type Library

The following changes were made to IHsvJournalsEx interface methods:

- The `plSecurityClass` argument was added to `GetTextTemplate`.
- The `lSecurityClass` argument was added to `SaveTextTemplate`.
- The `plSecurityClass` argument was added to `GetTemplate`.
- The `lSecurityClass` argument was added to `SaveTemplate`.

The following methods were added to the IHsvJournalsEx interface:

- AddJournalGroup
- EnumJournalGroups
- EnumJournalGroupsForScenarioYear
- EnumJournalIDsForExtractFilter
- GetEntityJournals
- GetJournalLabelsForIDs
- GetTemplateLabelsForIDs
- RemoveAllJournalGroups
- RemoveJournalGroup
- ValidateLineItems

HsvSystemInfo Type Library

The following methods were 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 was added to the HsvProcessFlow object.

**HsvReports Type Library**

The following methods were added to the HsvReports object:

- CheckSecurityRole
- EnumDocumentsEx
- GetDocumentEx
- SaveDocumentEx

**HsvJournalLoadACV Type Library**

The `ExtractOptionsEx` property and `ExtractEx` method were added to the HsvJournalLoadACV object.

**Note:** `ExtractOptionsEx` and `ExtractEx` enable you to apply filtering criteria when extracting journals.

**HsvMDArrays Type Library**

The HsvCubeData object was added for internal use only.
Changes for Release 3.0.4

The following topics list changes for release 3.0.4:

- “HsxClient Type Library” on page 950
- “HsxServer Type Library” on page 950
- “HsvMetadata Type Library” on page 950
- “HsvData Type Library” on page 951
- “HsvSystemInfo Type Library” on page 951
- “HsvReports Type Library” on page 951
- “HsvRulesLoadACV Type Library” on page 951

HsxClient Type Library

The following changes were made to the HsxClient type library:

- The following methods were added:
  - AuthenticateUserOnClusterSSO
  - GetLogonInfoSSO
  - SetLogonInfoSSO
- The following methods were deprecated:
  - GetLogonInfo
  - SetLogonInfo

HsxServer Type Library

The following changes were made to the HsxServer type library:

- The following methods were added:
  - GetHFMErrLogRecordSet

HsvMetadata Type Library

The following changes were made to the HsvMetadata type library:

- The following methods were added to the HsvMetadata object:
  - GetCurrencyValueIDForEntityValueCombination
  - GetCurrencyValueIDsForEntityValueCombinations
- The following methods were added to the HsvEntities object:
  - IsOrgByPeriodFilteringOn
- The following methods were added to the IHsvTreeInfo interface:
- FindMatchingMembersFromHierarchy

- The following methods were added to the HsvICPs object:
  - GetSecurityClassID

**HsvData Type Library**

The following changes were made to the HsvData type library:

- EnumEntitiesWithDataForScenarioYear was added to the HsvData type object.

- The following sparse data-related methods were added to the HsvData object:
  - DoesSparseDataExist
  - FilterMembersThatHaveData
  - FilterMembersThatHaveData2
  - GetMembersThatHaveData

- The following suppression-related methods were added to the HsvData object:
  - GetCellsWithRowSuppression2
  - GetTextCellsWithRowSuppression2

- The following methods were deprecated:
  - CopyInputData

**HsvSystemInfo Type Library**

The following methods were added to the HsvSystemInfo object:

- GetExtractFileEncoding
- SetExtractFileEncoding

**HsvReports Type Library**

The following methods were added to the HsvReports object:

- DeleteDocuments
- EnumDocuments
- GetDocument
- SaveDocument

**HsvRulesLoadACV Type Library**

SetSessionAndResource was added to the HsvRulesLoadACV object.
Changes for Release 3.0

The following topics list changes for release 3.0:

- “HsxClient Type Library” on page 952
- “HsxServer Type Library” on page 953
- “HsvMetadata Type Library” on page 953
- “HsvData Type Library” on page 953
- “HsvSecurityAccess Type Library” on page 954
- “HsvSystemInfo Type Library” on page 954
- “HsvMDArrays Type Library” on page 954
- “HsvMetadataLoadACV Type Library” on page 955
- “HsvRulesLoadACV Type Library” on page 955
- “Error Numbers” on page 955

HsxClient Type Library

The following changes were made to the HsxClient type library:

- The following methods were added:
  - AuthenticateUserOnCluster
  - CreateObjectOnCluster
  - EnumRegisteredClusterNames
  - GetClusterInfo
  - GetServerOnCluster
  - RegisterCluster
  - ScriptableEnumRegisteredClusterNames
  - UnregisterAllClusters
  - UnregisterCluster

- The following methods were deprecated:
  - CreateObjectOnServer
  - EnumRegisteredServerNames
  - GetServer
  - RegisterServer
  - UnregisterServer
**HsxServer Type Library**

The following methods were added to the HsxServer type library:

- DeleteXMLErrorsFromDatabase
- GetClusterInfo
- GetXMLErrorFromDatabase
- GetXMLErrorsListFromDatabase

**HsvMetadata Type Library**

The following changes were made to the HsvMetadata type library:

- The following methods were added to the HsvAccounts object:
  - GetICPTopMember
  - GetIsICP
  - IsICPRestricted
- The following methods were added to the IHsvTreeInfo interface:
  - GetAllPathsToMember
  - TranslateAttributeValueForDisplay
- The following methods were added to the HsvCustom object:
  - GetSecurityClassID
  - GetUserDefined1
  - GetUserDefined2
  - GetUserDefined3
- The GetSecurityAsPartnerID method was added to the HsvEntities object.
- The following methods were added to the HsvScenarios object:
  - GetMaximumReviewLevel
  - UsesLineItems
- Options for loading system members, Value dimension members, and Intercompany Partner dimension members were added.
- Options for extracting system members, Value dimension members, and Intercompany Partner dimension members were added. See Table 60 on page 204.

**HsvData Type Library**

The following changes were made to the HsvData type library:

- The GetCellDescription method was added to the HsvData object.
- The *varlaParents* argument was added to the *CopyInputDataForMultipleEntities* method.

- The following line item-related methods were added to the *HsvData* object:
  - GetCellLineItems
  - SetCellLineItems
  - GetTextCellLineItems
  - SetTextCellLineItems
  - InsertLineItemDetails
  - DeleteLineItemDetails
  - SetCellsLineItems
  - SetTextCellsLineItems

- The following line item-related methods were deprecated:
  - GetLineItems – superseded by GetCellLineItems.
  - SetLineItems – superseded by SetCellLineItems.
  - GetTextLineItems – superseded by GetTextCellLineItems.
  - SetTextLineItems – superseded by SetTextCellLineItems.

- The *ClearAllLineItems* method was removed from the *HsvData* object.

**HsvSecurityAccess Type Library**

The following methods were added to the *HsvSecurityAccess* object.

- GetUserNamaFromSID
- InsertDefaultSecurityClass

**HsvSystemInfo Type Library**

The following methods were added to the *HsvSystemInfo* object:

- ExtractDMLScript
- GetUserNama
- LoadDMLScript

**HsvMDArrays Type Library**

- The *plNumElementsInDataUnit* argument was added to the *SetData* method of the *HsvMDDataBuffer* object.

- The *plNumElementsInDataUnit* argument was added to the *SetData* method of the *HsvMDDataBufferLite* object.
HsvMetadataLoadACV Type Library

The following metadata load options were added:

- HSV_METALOAD_OPT_LOAD_SYSTEM_MEMBERS
- HSV_METALOAD_OPT_VALUE
- HSV_METALOAD_OPT_ICP

The following metadata extract options were added:

- HSV_METAEXTRACT_OPT_EXTRACT_SYSTEM_MEMBERS
- HSV_METAEXTRACT_OPT_VALUE
- HSV_METAEXTRACT_OPT_ICP

HsvRulesLoadACV Type Library

The following methods were added to the HsvRulesLoadACV object:

- ExtractDMLScript
- LoadDMLScript
- ScriptableLoadDMLScript

Error Numbers

Error numbers are no longer itemized in an appendix. See Chapter 24, “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 955
- “HsvData Type Library” on page 956
- “HsvCalculate Type Library” on page 956
- “HsvMDArrays Type Library” on page 956

HsxClient, HsxServer, and HsxClientUI References

HsxClient.d11, HsxServer.exe, and HsxClientUI.d11 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 in the /Hyperion Financial Management\Client directory.
- HsxServer.exe is in the /Hyperion Financial Management\Server directory.

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 this example:

```
Available References:

- [ ] HsxClient 1.0 Type Library
- [ ] HsxClient 1.0 Type Library
```

Since the component names are identical, use the components’ directory locations to select the correct component.

**Tip:** To prevent the old components from displaying, 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 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 were added:
- Allocate2
- ChartLogic2
- Consolidate2
- CustomLogic
- Translate2

### HsvMDArrays Type Library

The following items were added:
Changes for Release 2.1

The following topics list changes for release 2.1:

- “HsvData Type Library” on page 957
- “HsvJournals Type Library” on page 957
- “HsvReports Type Library” on page 957
- “HsvMDArrays Type Library” on page 958
- “HsvDataCubes Type Library” on page 958
- “HsvRulesLoadACV Type Library” on page 959
- “HsvSecurityAccess Type Library” on page 959
- “HsxClient Type Library” on page 959

HsvData Type Library

The following changes were made to HsvData:

- These methods are new:
  - SetCalcStatusLockedForMultipleProcessUnits
  - SetCalcStatusUnlockedForMultipleProcessUnits
- The varbIncludeDerivedData argument was added to AddDataToMDDataBuffer.
- The lParent and lValue arguments was removed from ClearAllLineItems.
- The sNumDecimals and sScale arguments was added to GetTextCells.
- The sScale argument was added to SetTextCell.
- The sScale argument was added to SetTextCells.
- The sAlternateNumDecimals argument was added to FormatNumberToText.

HsvJournals Type Library

ValidateValue was added to the IHsvJournalsEx interface.

HsvReports Type Library

CheckReportSecurityClass was added to the HsvReports object.
**HsvMDArrays Type Library**

The following changes were made to the HsvMDDataBuffer object:

- The `IView` argument was removed from `GetDescription`.
- The `pIView` argument was removed from `GetDescriptionAtIndex`.
- The `IView` argument was removed from `GetLineItems`.
- The `pIView` argument was removed from `GetLineItemsAtIndex`.
- The `IView` argument was removed from `InsertDescriptionAtEnd`.
- The `IView` argument was removed from `InsertLineItemsAtEnd`.
- The `IView` argument was removed from `SetDescription`.
- The `IView` argument was removed from `SetLineItems`.

**HsvDataCubes Type Library**

The description of subcube items changed due to changes in some of the type library’s methods. The updated description is in “About Subcube Items” on page 675.

The following changes were made to the HsvCurrencyCube object:

- The `lValue` and `plHandle` arguments were added to `BeginEnumerationOfStoredData`.
- The `lHandle` argument was added to `EndEnumerationOfStoredData`.
- The following changes were made to `GetOneCellFromStoredItem`:
  - The `lHandle` and `pvbDimensionMembersAreValid` arguments were added.
  - The Value dimension argument returns a Value dimension member ID; in previous releases, the argument took a Value member ID. The argument was renamed from `lValue` to `plValue`.
- The `lHandle`, `plValue`, and `pvbDimensionMembersAreValid` arguments were added to `GetPOVFromStoredItem`.

The following changes were made to the HsvNodeCube object:

- The `lValue` and `plHandle` arguments were added to `BeginEnumerationOfStoredData`.
- The `lHandle` argument was added to `EndEnumerationOfStoredData`.
- The following changes were made to `GetOneCellFromStoredItem`:
  - The `lHandle` and `pvbDimensionMembersAreValid` arguments were added.
  - The Value dimension argument returns a Value dimension member ID; in previous releases, the argument took a Value member ID. The argument was renamed from `lValue` to `plValue`.
- The `lHandle`, `plValue`, and `pvbDimensionMembersAreValid` arguments were added to `GetPOVFromStoredItem`.
HsvRulesLoadACV Type Library

The following changes were made to SetSession:

- The `LanguageID` argument was added.
- SetSession now returns a Boolean to indicate success or failure.

HsvSecurityAccess Type Library

The following constants were added, but are for internal use only; if you attempt to use them, an error occurs:

- `HFM_ACCESS_RIGHTS_UNSPECIFIED`
- `HFM_TASK_NONE`

HsxClient Type Library

The `AuthenticateUserOnServer` method was added for internal use only.

Changes for Release 2.0

The following topics list changes for release 2.0:

- “HsxClient Type Library” on page 960
- “HsxClientUI Type Library” on page 960
- “HsxServer Type Library” on page 960
- “HsvSession Type Library” on page 961
- “HFMConstants Type Library” on page 961
- “HsvMetadata Type Library” on page 961
- “HsvData Type Library” on page 961
- “HsvCalculate Type Library” on page 962
- “HsvJournals Type Library” on page 963
- “HsvProcessFlow Type Library” on page 963
- “HsvReports Type Library” on page 963
- “HsvSystemInfo Type Library” on page 963
- “HsvMDArrays Type Library” on page 964
- “HsvRulesLoadACV Type Library” on page 964
- “HsvcDataLoad Type Library” on page 964
- “HsvJournalLoadACV Type Library” on page 965
Security: HsxSecurity and HsvSecurityAccess Type Libraries

The HsxSecurity type library is no longer supported, and was replaced by the HsvSecurityAccess type library. The HsxSecurity object’s methods moved to the HsvSecurityAccess object. In addition, these HsvSecurityAccess items 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 were removed from the HsvProcessFlow type library:
  - GetCellLevelAccessRights
  - GetProcessUnitAccessRights
  - GetProcessUnitAccessRightsAndState
  - RefreshAccessRightsCache

- The access rights constants were renamed.
- Roles are now represented by enumerated constants. In addition, new roles were added.
- Constants for tasks were 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 for internal use only:

- ScriptableEnumRegisteredServerNames
- ScriptableGetLogonInfo
- ScriptableOpenApplication

HsxClientUI Type Library

The GetServer method was added for internal use only.

HsxServer Type Library

The following methods are now for internal use only:
HsvSession Type Library

HFMConstants Type Library
The HFMConstants type library was added. This library contains constants that represent numerous Oracle Hyperion Financial Management 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 were 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 849.

- The `GetSecurityClassID` method was added to the HsvAccounts object.

- The `GetNumPeriodsInFrequency` method was added to the HsvPeriods object.

- The `GetSecurityClassID` method was added to the HsvScenarios object.

- The following methods were added to the `IHsvTreeInfo` interface:
  - `GetNumDescendants`
  - `GetNumMembers`
  - `GetNumParents`
  - `IsMemberABaseOf`
  - `IsMemberADescendantOf`

- The `GetYearRange` method was added to the HsvYears object.

HsvData Type Library
The following changes were made to HsvData:

- Enum `HSV_DATA_UPDATE_MODE` was added; see “Update Mode Constants” on page 320.
The Duplicates load option uses the constants in Enum HSV_DATA_UPDATE_MODE instead of the previously documented constants.

The following changes were made to ClearInputData:

- The varlaAccountSubset argument was replaced by the varlaAccountSubsetIncludeList, vbUseAccountSubsetIncludeList, varlaAccountSubsetExcludeList, and vbUseAccountSubsetExcludeList arguments.
- The vbClearNodeLevelValueID argument's value is ignored.

The following changes were made to CopyInputData:

- The vbClearInputValueID, vbClearNoneValueID, and vbClearNodeLevelValueID arguments were renamed to vbCopyEntityCurrencyValueID, vbCopyNoneValueID, and vbCopyNodeLevelValueID, respectively.
- The vbCopyNodeLevelValueID argument's value is ignored.
- The vbCopyDerived and IEnumUpdateMode arguments were added.

The CopyInputDataForMultipleEntities method was added.

UpdateDataUsingMDDataBuffer was changed: the bClearExistingSubCubeData argument was removed, and the IEnumUpdateMode and bAccumulateWithinBuffer arguments was added.

AddDataToMDCube, deprecated in release 1.2, was removed in release 2.0.

UpdateDataUsingMDCube, deprecated in release 1.2, was removed in release 2.0.

The load options were changed. Two new load options with index numbers 5 and 6 were added, and the index of the Mode option changed from 5 to 7. See Table 64 on page 377.

Cell statuses are now represented by constants in the HFMConstants type library; see “About Cell Statuses” on page 319.

The sNumDecimals and sScale arguments were added to GetTextCell.

The GetTransactionData method was added.

HsvCalculate Type Library

The following methods were removed from the HsvCalculate object:

- GetRules
- SetRules

The following methods were added to the HsvCalculate object:

- GetVBScriptRules
- SetVBScriptRules
- GetCOMDLLRules (for future use)
- SetCOMDLLRules (for future use)
**HsvJournals Type Library**

In the IHsvJournalsEx interface, the SaveJournal and SaveTextJournal methods produce error 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 were removed from the HsvProcessFlow type library. In addition to the methods that were moved to the IHsvDataSecurity interface (as described in “Security: HsxSecurity and HsvSecurityAccess Type Libraries” on page 960), these HsvProcessFlow methods were removed:
- 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 were made to the HsvReports object:

- The pvarabstrDescriptions argument was added to EnumReports.
- The pbstrDescription argument was added to GetReport.
- The bstrDescription argument was 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:
The HsvTransactionData object was added to the HsvMDArrays type library. Use this object to return the transaction data generated by statutory consolidations.

HsvRulesLoadACV Type Library

New arguments were added to the following methods, enabling you to validate without loading and to check whether validation messages of various severity levels are included in the log file:

- LoadCalcRules
- LoadMemberListRules

The following methods were added to HsvRulesLoadACV for internal use only:

- 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 were added.
- The HSV_DATALOAD_ACCUMULATE constant was added as a valid value for the HSV_DATALOAD_OPT_DUPLICATES data load option. This allows loading in accumulate mode.
The **HSV_JOURNAL_EXTRACT_OPT_CATEGORY** extract option was renamed to **HSV_JOURNAL_EXTRACT_OPT_SCENARIO**.

### Changes for Release 1.2

The following topics list changes for release 1.2:

- “**HsvProcessFlow Type Library**” on page 965
- “**HsvData Type Library**” on page 965
- “**HsvMetadata Type Library**” on page 966
- “**HsvReports Type Library**” on page 966
- “**HsvMetadataLoadACV Type Library**” on page 966

### HsvProcessFlow Type Library

The following list describes changes to the HsvProcessFlow type library:

- The following methods were 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, it is not returned in Visual Basic; however, the error is returned in Visual C++.

### HsvData Type Library

The following list describes changes to the HsvData type library:

- The following methods were added to the HsvData object:
  - AddDataToMDDataBuffer
  - EnumExtractOptions
  - EnumLoadOptions
  - Extract
  - Load
  - UpdateDataUsingMDDataBuffer

- The following methods were deprecated in the HsvData object:
  - AddDataToMDCube (replaced by AddDataToMDDataBuffer)
  - UpdateDataUsingMDCube (replaced by UpdateDataUsingMDDataBuffer)
- `UpdateLoadDataUsingMDCube` was removed from the HsvData object.
- The HsvMDDataBuffer object was added to the HsvMDArrays type library.
- The HsvMDCube object in the HsvMDArrays type library was deprecated. 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 for future use. Previously, **9** indicated the Text account type.
- The following methods were added to the HsvMetadata object:
  - `EnumConsolidationMethodIDs`
  - `GetConsolidationMethodDescription`
  - `GetConsolidationMethodInfo`
  - `GetDefaultValueOfActiveStatusAccount`
  - `GetICPEntitiesAggregationWeight`
  - `GetValidationAccount`
- The following methods were added to the HsvAccounts object:
  - `GetUserDefined1`
  - `GetUserDefined2`
  - `GetUserDefined3`
  - `GetXBRITags`
- The following methods were added to the HsvCustom object:
  - `GetAggregationWeight`
  - `IsSwitchSignEnabledForFlow`
  - `IsSwitchTypeEnabledForFlow`
- The `GetHoldingCompany` method was added to the HsvEntities object.
- The `RefreshCache` method was removed from the IHsvTreeInfo interface.

### HsvReports Type Library

The `DeleteReports` method was added to the HsvReports object.

### HsvMetadataLoadACV Type Library

The following metadata load and extract options were added to the HsvMetadataLoadACV type library:
Changes for Release 1.1

The following topics list changes for release 1.1:

- “HsvMetadata Type Library” on page 967
- “HsxSecurity Type Library” on page 968
- “HsvReports Type Library” on page 968
- “HsvMDArrays Type Library” on page 968
- “HsvDataCubes Type Library” on page 968
- “HsvMetadataLoadACV Type Library” on page 968
- “HsvSecurityLoadACV Type Library” on page 969

HsvMetadata Type Library

The following list describes changes to the HsvMetadata type library:

- The HsvCategories object was renamed to HsvScenarios object.
- The Categories property of the HsvMetadata object was renamed to Scenarios property.

- The bIgnoreDups argument was added to the IHsvTreeInfo interface’s EnumAncestors method.
- The bIgnoreDups argument was added to the IHsvTreeInfo interface’s EnumDescendants method.
- The following methods were added to the HsvAccounts object:
  - GetNumDecimalPlaces
  - GetPlugAccount
  - GetTopMemberOfValidCustom1Hierarchy
  - GetTopMemberOfValidCustom2Hierarchy
  - GetTopMemberOfValidCustom3Hierarchy
  - GetTopMemberOfValidCustom4Hierarchy
  - IsConsolidated
  - IsCustom1AggregationEnabled
  - IsCustom2AggregationEnabled
  - IsCustom3AggregationEnabled
- IsCustom4AggregationEnabled
- IsICP

- The following methods were added to the HsvEntities object:
  - GetAllowAdjustments
  - GetAllowAdjustmentsFromChildren
  - GetUserDefined1
  - GetUserDefined2
  - GetUserDefined3
  - IsDescendant
  - IsICP

**HsxSecurity Type Library**

The following methods were added to the HsxSecurity object:

- AddUserEx
- IsValidWindowsUser

**HsvReports Type Library**

The HsvReports type library was added to this document. See Chapter 16, “HsvReports Type Library.”

**HsvMDArrays Type Library**

The HsvMDArrays type library was added to this document. See Chapter 17, “HsvMDArrays Type Library.”

**HsvDataCubes Type Library**

The HsvDataCubes type library was added to this document. See Chapter 18, “HsvDataCubes Type Library.”

**HsvMetadataLoadACV Type Library**

Security information is no longer loaded and extracted with metadata. These security-related load and extract options were 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 was added; use this to load and extract security information. See “HsvSecurityLoadACV Type Library” on page 774.
Symbols

.LST file
  getting contents of, 531
  loading and validating, 794
.PER files, creating applications and, 160
.RLE file
  getting contents of
    GetVBScriptCalcRules, 531
    GetVBScriptRules, 401
  loading and validating
    LoadCalcRules, 792
    SetVBScriptCalcRules, 538
    SetVBScriptRules, 402

A
  access rights constants, HsvSecurityAccess, 461
accounts
  AccountType attribute, returning, 275
  Custom1TopMember attribute, returning, 280
  Custom2TopMember attribute, returning, 281
  Custom3TopMember attribute, returning, 281
  Custom4TopMember attribute, returning, 282
  EnableCustom1Aggr attribute, returning, 284
  EnableCustom2Aggr attribute, returning, 285
  EnableCustom3Aggr attribute, returning, 285
  EnableCustom4Aggr attribute, returning, 286
  IsCalculated attribute, returning, 283
  IsConsolidated attribute, returning, 284
  IsICP attribute, returning, 286
  NumDecimalPlaces attribute, returning, 278
  PlugAcct attribute, returning, 279
  testing whether Custom members valid for, 224
  UserDefined1 attribute, returning, 282
  UserDefined2 attribute, returning, 283
  UserDefined3 attribute, returning, 283
  UsesLineItems attribute, returning, 287
  validation account, 222
validation account, returning by index, 208
  XBRLTags attribute, returning, 283
Accounts (property), 230
  AccountType attribute, returning, 275
ACMs, instantiating, 193
actions, process management, constants for, 878
active status account, returning default value of, 216
activity user ID, 522
AddAccountCustomCombination, 663
AddAccountCustomCombinationExtDim, 663
AddApplicationAdministrator, 465
AddApplicationAdministrator2, 465
AddDataToMDDataBuffer, 321
AddEdgeDimension, 685
AddEdgeDimensionID, 686
AddEdgeSource, 686
AddEdgeSources, 686
AddICTransactionData, 664
AddICTransactionDataExtDim, 664
AddJournalGroup, 415
AddOrRemoveApplicationAdministrators, 465
AddOrRemoveApplicationAdministrators2, 465
AddOrRemoveRolesFromUser, 466
AddOrRemoveRolesFromUser2, 466
AddOrRemoveUsersFromRole, 467
AddOrRemoveUsersFromRole2, 467
AddQueryDimensionMemberID, 664
AddQueryField, 664
AddQueryFieldItem, 664
AddQueryOrderField, 664
AddRefToHsxServer, 501
AddSecurityClass, 468
AddSecurityClassWithAccessCode, 468
AddSourceDimension, 687
AddSourceDimensionID, 687
AddTaskToAudit, 501
AddTaskToAuditWithAttachment, 502
AddTaskToRunningTasks, 502
AddTaskToRunningTasksAndUpdatePOV, 502
AddToSourceJoinTypesForDimensions, 688
AddUser, 468
AddUser2, 468
AddUserEx, 468
AddUserToRole, 469
AddUserToRole2, 469
Administrator group, testing whether a user belongs to, 163
AllMVTransPosted, 737
Allocate, 392
Allocate2, 392
AllowAdjFromChildren attribute, returning, 291
AllowAdjs attribute, returning, 291
AllowRulesLoadForEPMAApp, 469
application creator rights
creating applications and, 159
testing whether a user has, 163
application currency, returning, 207
application profiles, creating applications and, 160
application servers
activity has occurred on, returning names of, 506
applications on, getting, 186
data link file, getting, 188
names of, getting, 167
registering
   HsxClient object, 178
   HsxClientUI object, 184
system file, getting, 188
unregistering
   HsxClient object, 180
   HsxClientUI object, 184
application settings
DefaultCurrency, 207
GetByIndexValidationAccount, 208
GetValidationAccount, 222
OrgByPeriodApplication, 225
returning when last updated, 207
values, returning
   as raw value, 206
   as string, 229
applications
creating, 159
deleting, 161
getting user information, 173
   single sign on, 173
local storage folder, getting, 172
logging onto
   example, 137
   using HsxClient, 179
   using HsxClientUI, 182
on application servers, getting, 186
opening
   using HsxClient, 176
   using HsxClientUI, 183
Approve, 544
Approve2, 545
ApproveEx, 546
ApproveJournals, 415
AttachDocumentToCell, 321
attribute, metadata, getting value of, 255
audit
data audit
   delete audit history, 323
   enumerate cells, 332
   one cell, 356
task audit
   adding task to audit history, 501
delete audit history, 503
getting audit history, 507
AuthenticateCSSToken, 158
AuthenticateSecurityAgentCredentials, 158
AuthenticateUserCredentials, 158
AuthenticateUserOnClient, 158
AuthenticateUserOnCluster, 159
AuthenticateUserOnClusterSSO, 159
AuthenticateUserOnServer, 159

B
base frequency, 300
BeginDataEnum
   HsvICTransactionsData object, 664
   HsvTransactionData object, 658
BeginEnumeration
   HsvMDDataBuffer object, 613
   HsvMDDataBufferLite object, 635
BeginEnumerationOfStoredData
   HsvCurrencyCube object, 676
   HsvNodeCube object, 680
BeginQueryEnum, 658
BeginQueryFieldEnum, 665
BeginQueryOrderEnum, 665
local storage folder, setting, 178
CalcAttribute attribute, returning, 276
CalcEPU, 393
Calculate, 191
calculation statuses
cells, 352, 353
cells, constants for, 866
subcube periods, constants for, 868
subcubes, 352
Categories. See Scenarios
cell data
eamples for getting and setting, 138
setting a cell to Null, 382
cell statuses
constants for, 866
getting with GetStatus, 365
overview, 319
CELLSTATUS_NOREADACCESS, 868
ChartLogic, 393
ChartLogic2, 394
CheckAccess, 503
CheckAllItems, 818
CheckAllItemsUsingSelectedList, 818
CheckBoxChanged event, 829
CheckReportSecurity, 738
CheckReportSecurityClass, 595
CheckReportSecurityClass2, 596
CheckSecurityForICExtract, 756
CheckSecurityRole, 596
Clear HfmPovCOM, 142
Clear HfmSliceCOM, 151
ClearAllData, 322
ClearAllDescriptionsInSubCube, 322
ClearAuditTasks, 503
ClearAuditTasks2, 504
ClearDataAuditItems, 323
ClearDataAuditItemsEx, 324
ClearInputData, 324
ClearInvalidData, 326
ClearRunningTask, 505
CloseICPeriod, 756
ClosePeriod, 411
CollapseAll, 688
CollapseGrid, 689
CollapseGridDimension, 689
CollapseGridDimensionID, 690
CollapseGridID, 690
COM data type mappings, 50
connected user, 464
Consolidate, 395
Consolidate2, 396
ConsolidateYTD attribute, Scenario members, 309
consolidation methods
returning description of, 211
returning IDs of an application’s, 199
returning information about, 212
returning when last updated, 213
consolidations, statutory, returning transaction data for, 657
constants
access rights, 461
CELLSTATUS_NOREADACCESS, 868
data extract options, 802
data load options, 800
file format constants, metadata loads and extractions, 789
journal extract options
filtered extracts, 812
unfiltered extracts, 811
journal load options, 810
MEMBER_LIST_ALL_HIERARCHY, 862
MEMBERNOTUSED, 858
security extract options, 778
security load options, 777
ConvertToHfmPovStrCOM, 142
ConvertToHfmSliceStrCOM, 151
CopyInputData, 327
CopyInputDataForMultipleEntities, 327
Count, 766
CreateApplication, 159
CreateApplicationCAS, 159
CreateApplicationCASWithAccessCode, 161
CreateDataIndexList
HsvMDDataBuffer object, 613
HsvMDDataBufferLite object, 635
CreateDataIndexListEx, 614
CreateDataIndexListExtDim
HsvMDDataBuffer object, 613
CreateEdge, 691
CreateGrid, 691
CreateICTransaction, 738
CreateICTransactionExtDim, 738
CreateObject, 193
CreateObjectOnCluster, 161
CreateObjectOnServer, 161
CreatePageControl, 692
CreateSource, 692
CreateStarSchema, 726
CreateStarSchemaAndReturnTaskID, 727
CreateStarSchemaExtDim, 726
CreateStarSchemaFromHTTP, 728
Creator group
creating applications and, 159
user, testing whether a member of, 163
CSSEnabled, 190
Currencies, 230
currencies
description, getting, 315
exchange rate, returning, 399
HsvCurrencies object, 313
IDs of, getting, 315
labels and IDs of, getting
all currencies, 314
filtering for DisplayInICT attribute, 314
labels of, getting, 316
returning when last updated, 214
scale, getting, 317
translation operator, getting, 316
TranslationOperator attribute, 316, 317
currency IDs, 313
currency subcubes
accessing with HsvCurrencyCube object, 676
described, 53
currency, application, returning, 207
CurrentValue, 767
Custom1, 231
Custom1TopMember attribute, returning, 280
Custom2, 231
Custom2TopMember attribute, returning, 281
Custom3, 231
Custom3TopMember attribute, returning, 281
Custom4, 231
Custom4TopMember attribute, returning, 282
CustomLogic, 397

D
data
examples for getting and setting, 138
extract options, 802
extracting
example, 799
overview, 798
load options, 800
loading
example, 798
overview, 797
objects and interfaces for
application data, 78
arrays of cells, 104
subcubes, access to, 115
data audit
delete audit history, 323
enumerate cells, 332
one cell, 356
data link file, getting path of, 188
Data property, 191
data type mappings, 50
decimal character
getting, 526
setting, 536
Default security class, constant for, 908
DefaultCurrency application setting, returning, 207
DefaultParent attribute, generation levels and, 262
DefaultValue, 768
DefaultValueForActive attribute, 216
DefCurrency attribute, returning, 292
DefineGrid, 693
Delete Application dialog box, displaying, 182
Delete Invalid records feature, 326
DeleteApplication
HsxClient object, 161
HsxClientUI object, 182
DeleteApplicationWithAccessCode, 162
DeleteDocument, 597
DeleteGrid, 693
DeleteICReasonCode, 739
DeleteJournals, 416
DeleteLineItemDetails, 328
DeleteReport, 597
DeleteReports, 598
DeleteSecurityClass, 469
DeleteSecurityClasses, 470
DeleteSecurityClassWithAccessCode, 470
DeleteStarSchema, 728
DeleteSystemErrors, 162, 186
DeleteSystemErrorsEx, 186
DeleteTemplate, 732
DeleteTemplates, 417
DeleteUserParameter, 505
 descriptions
   returning the default language for, 525
   setting the default language for, 535
DetachDocumentFromCell, 329
DetermineWindowsLoggedInUser, 162
Dimension, 231
dimension objects, 65
dimensions
   constants that represent, 857
   getting labels of all members, 234
   getting labels of children and parents, 235
   getting member IDs of all members, 233
   getting member IDs of children and parents, 234
   hierarchy, returning position of member in, 256
DimensionWasSelected event, 830
DisableNewConnections
   HsvSystemInfo object, 505
   HsxClient object, 162
DisplayInICT attribute, 314
DMELoad
   HsvcDataLoad object, 805
   HsvData object, 329
DoCellDescriptionsExist, 329
documents
   reports, 603
   DoesCellDescriptionExist, 330
   DoesCellSupportICTransactionDetail, 740
   DoesCellSupportICTransactionDetailExtDim, 740
   DoesDataExist, 330
   DoesDocumentExist, 598
   DoesFolderExist, 599
   DoesSparseDataExist, 331
   DoesUserHaveCreateApplicationRights, 163
   DoesUserHaveSystemAdminRights, 163
domain
   getting, 173
   setting, 179
   single sign on, getting, 173
dynamic member lists, getting members of, 243
EnableCustom1Aggr attribute, returning, 284
EnableCustom2Aggr attribute, returning, 285
EnableCustom3Aggr attribute, returning, 285
EnableCustom4Aggr attribute, returning, 286
EnableDimension, 819
EnableNewConnections
   HsvSystemInfo object, 506
   HsxClient object, 164
EndDataEnum
   HsvICTransactionsData object, 665
   HsvTransactionData object, 658
EndEnumeration
   HsvMDDataBuffer object, 614
   HsvMDDataBufferLite object, 637
EndEnumerationOfStoredData
   HsvCurrencyCube object, 676
   HsvNodeCube object, 680
EndQueryEnum, 659
EndQueryFieldEnum, 665
EndQueryOrderEnum, 665
terities
   AllowAdjFromChildren attribute, returning, 291
   AllowAdjs attribute, returning, 291
   DefCurrency attribute, returning, 292
   HoldingCompany attribute, returning, 292
   IsICP attribute, returning, 296
   IsOrgByPeriodFilteringOn attribute, returning, 297
   SecurityClass attribute, returning, 293
   UserDefined1 attribute, returning, 294
   UserDefined2 attribute, returning, 294
   UserDefined3 attribute, returning, 295
Entities (property), 232
EnumActivityServers, 506
EnumActivityUsers, 507
EnumActivityUsersForRunningTasks, 507
EnumAllMemberIDs, 233
EnumAllMemberLabels, 234
EnumAllParentAndChildIDs, 234
EnumAllParentAndChildLabels, 235
EnumAncestors, 236
EnumApplicationAdministrators, 470
EnumApplicationAdministrators2, 470
EnumApplicationStarSchemas, 728
EnumAuditTasks, 507
EnumAuditTasks2, 509
EnumAuditTasksEx, 511
EnumBaseMemberIDs, 237
EnumCellTextLabels, 198
EnumConsolidationMethodIDs, 199
EnumCurrencies, 314
EnumCurrencies2, 314
EnumCustomDimIds, 151
EnumCustomDimsForApp
  HsvMetadata object, 199
EnumCustomDimsForAppEx
  HsvMetadata object, 199
EnumDataAuditItems, 332
EnumDataAuditItems2, 333
EnumDataSources, 186
EnumDefaultAncestors, 238
EnumDefaultAncestorsLabels, 238
EnumDescendants, 239
EnumDimProperties
  HsvMetadata object, 200
EnumDocuments, 599
EnumDocumentsEx, 600
EnumEntitiesWithDataForScenarioYear, 333
EnumExtractOptions
  HsvData object, 333
  HsvMetadata object, 200
EnumIDsOfChildren, 240
EnumIDsOfChildren, 241
EnumJournalGroups, 418
EnumJournalGroupsEx, 418
EnumJournalGroupsForScenarioYear, 419
EnumJournalIDsForExtractFilter, 419
EnumLanguages, 201
EnumLoadOptions
  HsvData object, 334
  HsvMetadata object, 202
EnumMemberLists, 242
EnumMemberProperties
  HsvMetadata object, 203
EnumMembers, 242
EnumMembers2, 243
EnumMembersWithAttribValue, 245
EnumOnDemandRules, 397
EnumParents, 245
EnumPhasedSubmissionStartYears, 303
EnumProhibitConnections
  HsvSystemInfo object, 512
  HsxClient object, 165
EnumProvisioningProjects, 166
EnumRegisteredClusterNames, 166
EnumRegisteredClustersOrServers, 167
EnumRegisteredDSNs
  HsvStarSchemaACM object, 728
  HsxServer object, 187
EnumRegisteredServerNames, 167
EnumReports, 602
EnumRoles, 471
EnumRolesForPrincipal, 471
EnumRolesForUser, 472
EnumRunningTasks, 514
EnumRunningTasksEx, 516
EnumRunningTasksPOV, 518
EnumSecurityClasses, 472
EnumSecurityClassRightsForPrincipal, 473
EnumSortedIDsOfChildren, 246
EnumSortedIDsOfChildren, 247
EnumSortedMembers, 247
EnumTemplates, 732
EnumUserAppPreferences, 167
EnumUserClassAccess, 474
EnumUsers, 474
EnumUsers2, 474
EnumUsers3, 474
EnumUsersInGroup, 475
EnumUsersInRole, 475
EnumUsersInRole2, 475
EnumUsersInRole3, 476
EnumUsersInSecurityClass, 476
EnumUsersInSecurityClass2, 476
EnumUsersInSecurityClass3, 476
EnumUsersInSecurityClass4, 477
EnumUsersOnSystem, 167
EnumUsersOnSystemEx, 169
EnumUsersOnSystemEx2, 170
EnumUsersOrGroups, 477
EnumUsersWithFilter, 477
EnumUsersWithFilter2, 477
EraserRecordFromPMBuffer
  HsvMDDataBuffer object, 614
  HsvMDDataBufferLite object, 636
Err object (Visual Basic), 55
  error handling
    example, 139
    overview, 55
  error numbers, utility for, 843
  examples, getting started
    cell data, 138
    error handling, 139
    loading and extracting, 140
logging on, 137
metadata, 138
user number formatting preferences, 139
ExecuteOnDemandRules, 397
ExpandAll, 694
ExpandGrid, 694
ExpandGridColumnDimension, 695
ExpandGridColumnDimensionID, 695
ExpandGridID, 696
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, 725
external authentication
 enabled for cluster, indicating whether, 190
generating user information, 173
Extract
 HsvData object, 335
 HsvMetadata object, 203
 load- and extract-related objects, 771
ExtractCalcRules, 790
ExtractDataAuditItems, 339
ExtractDrillableRegions, 340
ExtractDrillableRegionsByURLNames, 340
ExtractEx, 815
extracting data
 example, 799
 overview, 798
extracting journals
 example
 filtered extract, 810
 unfiltered extract, 808
 overview, 807
extracting member lists, 791
extracting metadata
 example, 783
 file format constants, 789
 overview, 782
extracting rules, 790
extracting security
 example, 777
 overview, 776
ExtractMemberListRules, 791
ExtractModuleConfigurations, 206
ExtractOptions, 770
ExtractOptionsEx, 815
ExtractTaskAudit, 520
ExtractTaskAudit2, 521
F
FDM Application Name setting, 217
FilterMembersThatHaveData, 341
FilterMembersThatHaveData2, 342
Find, 248
FindByDesc, 250
FindMatchingMembersFromHierarchy, 251
FindMatchingMembersFromHierarchyByDesc, 252
FindMatchingMembersFromHierarchyWildcard, 253
FindOverlappingConsolidation, 398
FormatNumberToText, 342
FormatNumberToText2, 344
FormatStatusToText, 346
FormatStoredNumberToText, 347
FormatTextToNumber, 348
FormatTextToStoredNumber, 349
FreezeEdgeDefinition, 696
G
GenerateCartesianProduct, 152
GenerateRecurring, 420
GenerateSecurityReportForBiPub, 478
generation level, member, getting, 262
get_Dimension, 231
get_HsvDQI, 194
GetAccessRights, 665
GetAccountCustomCombination, 666
GetAccountCustomCombinationExtDim, 666
GetAccountCustomCombinationTotal, 666
GetAccountType, 275
GetActivityCodeDesc, 522
GetActivityUserID, 522
GetAggregationWeight, 287
GetAllDataForGridPageDimensionIndex, 697
GetAllDescriptionsInSubCube, 350
GetAllDims, 143
GetAllowAdjustments, 291
GetAllowAdjustmentsFromChildren, 291
GetAllPathsToMember, 254
GetAllSecurityClassRightsForConnectedUser, 478
GetAllURLNames, 350
GetAllVisibleDimensionDetails, 697
GetApplicationAdministratorAccessForAllUsers, 479
GetApplicationAdministratorAccessForAllUsers2, 479
GetApplicationAttribute, 206
GetApplicationAttributeExtDim, 206
GetApplicationCurrency, 207
GetApplicationDirectory, 522
GetApplicationFolder, 172
GetApplicationName, 523
GetApplicationSettingsTimeStamp, 207
GetAppProfileInto, 208
GetAsynchronousTaskStatus, 729
GetAttachedDocumentsToCell, 351
GetAttributeValue, 255
GetAvailableLanguages, 835
GetBaseDataForAccount, 351, 352
GetBaseFrequency, 300
GetByIndexValidationAccount, 208
GetCalcAttribute, 276
GetCalcRulesType, 524, 792
GetCalcStatus, 352
GetCalcStatusEx, 352, 353
GetCalcStatusStatistics, 353
GetCell
  HsvCurrencyCube object, 677
  HsvData object, 354
  HsvNodeCube object, 681
GetCellDescription, 355
GetCellDescriptions, 355
GetCellExtDim
  HsvCurrencyCube object, 677
  HsvNodeCube object, 681
GetCellHistory, 356
GetCellHistory2, 357
GetCellJournalEntries, 358
GetCellLevelAccessRights, 495
GetCellLevelAccessRightsExtDim, 495
GetCellLevelAccountType, 209
GetCellLevelAccountTypeExtDim, 209
GetCellLineItems, 358
GetCells, 359
GetCellsWithRowSuppression, 359, 360
GetCellsWithRowSuppression2, 361
GetCellTextIdFromLabel, 209
GetCellTextIdsFromLabels, 210
GetCellTextLabelDescription, 210
GetCellTextLabelFromId, 211
GetCheckedItems, 820
GetCheckLineItemDetailsForCaseInsensitiveDuplicates
  HsvMDDataBuffer object, 615
  HsvMDDataBufferLite object, 637
GetClusterInfo, 172
GetClustersAndServers, 187
  HsxClient object, 173
GetColumnFilter, 740
GetCOMDLLCalcRules, 524
GetCOMDLLRules, 399
GetConnectedUser, 479
GetConnectedUser2, 479
GetConsolidationMethodDescription, 211
GetConsolidationMethodInfo, 212
GetConsolidationMethodsTimeStamp, 213
GetConsolidationProgress, 399
GetCountOfAttachedDocumentsToCell, 361, 362
GetCubeIndexFromPOV
  HsvMDDataBuffer object, 615
  HsvMDDataBufferLite object, 637
GetCubePOVFromIndex
  HsvMDDataBuffer object, 615
  HsvMDDataBufferLite object, 638
GetCurrencyCube, 362
GetCurrencyDescription, 315
GetCurrencyID, 315
GetCurrencyIDFromValueID, 311
GetCurrencyInfo, 740
GetCurrencyLabel, 316
GetCurrencyTimeStamp, 214
GetCurrencyTranslationOperator, 316
GetCurrencyValueIDForEntityValueCombination, 214
GetCurrencyValueIDsForEntityValueCombinations, 215
GetCurrencyValueIDForEntityValueCombination, 214
GetCurrencyValueIDsForEntityValueCombinations, 215
GetCurrentActivity, 524
GetCurrentHSSRegistrationVersion, 836
GetCurrentVersion, 836
GetCurrentVersionInUserDisplayFormat, 836
GetCustomMemberAt, 143
GetCustoms, 144
GetCustomX, 215
GetData
  HsvMDDataBuffer object, 616
HsvMDDataBufferLite object, 638
GetDataAtIndex
HsvMDDataBuffer object, 617
HsvMDDataBufferLite object, 639
GetDataAtIndexExtDim
HsvMDDataBuffer object, 617
GetDataExtDim
HsvMDDataBuffer object, 616
GetDataForAllMetadataCombinations, 362
GetDataSource, 188
GetDefaultExchangeRate, 399
GetDefaultExchangeRateExtDim, 399
GetDefaultFrequency, 304
GetDefaultHierarchyPosition, 256
GetDefaultItemIDHierarchy, 256
GetDefaultMemberID, 257
GetDefaultParent, 258
GetDefaultParentLabel, 258
GetDefaultScaleAndNumDecimal, 215
GetDefaultValueID, 292
GetDefaultValueOfActiveStatusAccount, 216
GetDefaultView, 304
GetDescription
HsvMDDataBuffer object, 618
HsvMDDataBufferLite object, 640
IHsvTreeInfo interface, 259
GetDescriptionAtIndex
HsvMDDataBuffer object, 618
HsvMDDataBufferLite object, 641
GetDescriptionAtIndexExtDim
HsvMDDataBuffer object, 618
GetDescriptionExtDim
HsvMDDataBuffer object, 618
HsvMDDataBufferLite object, 641
GetDimension, 216
GetDimensionIdFromName, 217
GetDimensionMember, 144
GetDisplayInfo, 260
GetDisplayInfoForSeveralItems, 261
GetDocument, 603
GetDocumentEx, 604
GetDynamicAccountRulesCache, 363
GetECID, 193
GetEntitiesContacts, 741
GetEntityCurrencyID, 741
GetEntityJournals, 421
GetEntityJournals2, 422
GetEntityJournals2ExtDim, 422
GetEntityJournalsExtDim, 421
GetEntityPartnerOption, 666
GetEPUInfo, 400
GetEPUInfoExtDim, 400
GetErrorStatus, 666
GetExtractFileEncoding, 524
GetExtractLogData, 730
GetFdmAppName, 217
GetFileTransfer, 188
GetFilterOptions, 667
GetFixedDimensionMembers, 667
HsvCurrencyCube object, 677
HsvNodeCube object, 681
HsvTransactionData object, 659
GetFixedMember, 152
GetFormattedDateTime, 525, 836
GetFormattedDateTimeForLanguage, 837
GetFormattedError, 837
GetFormattedErrorWithLineFeed, 839
GetFormattedResourceString, 525, 839
GetFrequency, 300
GetFrequencyID, 217
GetGridCompleteDimensionDetails, 698
GetGridDetails, 699
GetGridDimensionDetails, 700
GetGridDimensionDetailsIDsAndLabels, 700
GetGroupPhaseFromCell, 547
GetGroupPhaseFromCellExtDim, 548
GetHelpDirectoryForLanguageID, 840
GetHFMErrorLogRecordSet
HsxClient object, 173
HsxServer object, 188
GetHFMLanguageIdFromUserLanguages, 840
GetHistory, 548
GetHoldingCompany, 292
GetICEntitiesLockStatus, 757
GetICEntitiesAggregationWeight, 218
GetICEntitiesAggregationWeightEx, 218
GetICPTopMember, 276
GetICReasonCodeID, 741
GetICReasonCodeLabel, 742
GetICReasonCodes, 742
GetICTransactionCell, 667
GetICTransactionCellExtDim, 667
GetICTransactionData, 668
GetICTransactions, 743
GetIdentityTypes, 480
GetIndexesRangeForSourcesOnEdge, 701
GetIsICP, 277
GetItem, 662
GetItemExtDim, 662
GetItemGeneration, 262
GetItemID
    HsvJournals object, 412
    IHsvTreeInfo interface, 263
GetItemIDQL, 264
GetItemLevel, 264
GetJournal, 423
GetJournal2, 425
GetJournal2ExtDim, 425
GetJournalDisplayData, 426
GetJournalExtDim, 423
GetJournalLabelsForIDs, 428
GetJournalQueryDefinitionIDs, 428
GetJournalTemplateItemID, 412
GetKillUsersStatus, 525
GetKillUserStatus, 525
GetLabel, 264
GetLanguageCountryCodeFromLanguageId, 840
GetLanguageUserParameters, 525
GetLastModifiedDateAndModifiedByForArtifact, 526
GetLastModifiedDateForArtifact, 526
GetLeftHandDims, 145
GetLicenseExpirationStatus, 194
GetLineItems
    HsvData object, 363
    HsvMDDataBuffer object, 619
    HsvMDDataBufferLite object, 642
GetLineItemsAtIndex
    HsvMDDataBuffer object, 620
    HsvMDDataBufferLite object, 643
GetLineItemsAtIndexExtDim
    HsvMDDataBuffer object, 620
    HsvMDDataBufferLite object, 643
GetLineItemsExtDim
    HsvMDDataBuffer object, 619
    HsvMDDataBufferLite object, 643
GetListInfo, 820
GetLocaleIdFromLanguageId, 841
GetLockStatusICEntities, 757
GetLockStatusICEntity, 757
GetLogonInfo, 173
GetLogonInfoSSO, 173
GetMatchCode, 669
GetMaxCellTextSize, 363
GetMaxCustomDimNumber, 218
GetMaxDimId, 218
GetMaximumReviewLevel, 305
GetMember, 821
GetMemberArrayForDim, 152
GetMemberAt, 153
GetMemberForCustomDim, 145
GetMemberID, custom function, 263
GetMemberListID, 265
GetMemberListName, 265
GetMembersThatHaveData, 363
GetMissingDataZeroViewForAdjValues, 305
GetMissingDataZeroViewForNonAdjValues, 306
GetModifyApplicationInfo
    HsxClient object, 174
GetModifyApplicationStatus
    HsxClient object, 174
GetModuleName, 526
GetMonitorICDetails, 743
GetMonitorICSummary, 744
GetMonitorICTransactions, 745
GetMultiServerMaxSyncDelayForChanges, 364
GetNodeCube, 364
GetNumBaseMembers, 266
GetNumberOfFormattingUserParameters, 526
GetNumCells, 153
GetNumCellsForData
    HsvMDDataBuffer object, 620
    HsvMDDataBufferLite object, 644
GetNumCellsForDescriptions
    HsvMDDataBuffer object, 621
    HsvMDDataBufferLite object, 644
GetNumCellsForLineItems
    HsvMDDataBuffer object, 621
    HsvMDDataBufferLite object, 645
GetNumCheckedItems, 822
GetNumChildren, 266
GetNumDecimalPlaces, 278
GetNumDescendants, 267
GetNumDimensions, 219
GetNumItems, 662
GetNumMembers, 153, 268
GetNumParents, 268
GetNumPeriodsInCubeForData
HsvMDDataBuffer object, 622
HsvMDDataBufferLite object, 645
GetNumPeriodsInCubeForDescriptions
HsvMDDataBuffer object, 622
HsvMDDataBufferLite object, 646
GetNumPeriodsInCubeForLineItems
HsvMDDataBuffer object, 623
HsvMDDataBufferLite object, 646
GetNumPeriodsInFrequency, 301
GetNumRoles, 480
GetNumTransactionsCached, 670
GetOBPInfo, 822
GetOneCellFromStoredItem
  HsvCurrencyCube object, 678
  HsvNodeCube object, 682
GetOneCellFromStoredItemExtDim
  HsvCurrencyCube object, 678
  HsvNodeCube object, 682
GetOwner, 480
GetPageControlDetails, 702
GetPagingOption, 670
GetPartnerAsEntityList, 670
GetPartnerQueryDimensionMemberIDs, 670
GetPeriodFromFrequency, 301
GetPeriodIndexFromPOVForData
  HsvMDDataBuffer object, 623
  HsvMDDataBufferLite object, 646
GetPeriodIndexFromPOVForDescriptions
  HsvMDDataBuffer object, 624
  HsvMDDataBufferLite object, 648
GetPeriodIndexFromPOVForLineItems
  HsvMDDataBuffer object, 625
  HsvMDDataBufferLite object, 649
GetPeriodPOVFromIndexForData
  HsvMDDataBuffer object, 626
  HsvMDDataBufferLite object, 650
GetPeriodPOVFromIndexForDescriptions
  HsvMDDataBuffer object, 627
  HsvMDDataBufferLite object, 650
GetPeriodPOVFromIndexForLineItems
  HsvMDDataBuffer object, 627
  HsvMDDataBufferLite object, 651
GetPeriodsInFrequency, 302
GetPeriodStatusList, 413
GetPersistedSettings, 731
GetPersistedSettingsExtDim, 731
GetPhasedSubmissionHistory, 551
GetPhasedSubmissionHistoryExtDim, 551
GetPhasedSubmissionStartYear, 303
GetPhasedSubmissionState, 552
GetPhasedSubmissionStateExtDim, 552
GetPhasedSubmissionStateUsingPhaseID, 552
GetPhasedSubmissionStateUsingPhaseIDExtDim, 553
GetPhaseSubmissionGridForGivenScenarioPeriod, 365
GetPlugAccount, 279
GetPMErrorRecordCount
  HsvMDDataBuffer object, 627
  HsvMDDataBufferLite object, 651
GetPMRecordCount
  HsvMDDataBuffer object, 628
  HsvMDDataBufferLite object, 651
GetPovAtIndex, 153
GetPOVForIndexes, 702
GetPOVFromStoredItem
  HsvCurrencyCube object, 679
  HsvNodeCube object, 683
GetPOVFromStoredItemExtDim
  HsvCurrencyCube object, 679
  HsvNodeCube object, 683
GetPOVIDsForIndexes, 703
GetProcessUnitAccessRights, 495
GetProcessUnitAccessRightsAndState, 496
GetProcessUnitAccessRightsAndStateEx, 497
GetProcessUnitAccessRightsAndStateExExtDim, 497
GetProcessUnitAccessRightsEx, 498
GetProcessUnitAccessRightsExExtDim, 498
GetQualifiedLabel, 269
GetQueryDimensionMemberIDs, 670
GetQueryFieldInformation, 670
GetQueryFieldItem, 670
GetQueryItem, 659
GetQueryItemExtDim, 659
GetQueryOrderField, 670
GetRecordFromPMBuffer
  HsvMDDataBuffer object, 628
  HsvMDDataBufferLite object, 651
GetRecordFromPMErrorBuffer
  HsvMDDataBuffer object, 628
  HsvMDDataBufferLite object, 651
GetReport, 605
GetReportData, 456
GetReportData2, 458
GetReportData3, 458
GetReportDataExtDim, 457
GetResourceLanguageUserParameters, 527
GetResourceString
  HsvResourceManager object, 841
  HsvSystemInfo object, 527
GetResourceIdFromHR, 841
  HsvSystemInfo object, 527
GetRightHandDims, 146
GetRoleAccessForAllUsers, 480
GetRoleAccessForAllUsers2, 480
GetRoleID, 481
GetRoleLabel, 481
GetRowFilter, 746
GetRulesMode, 482
GetRunningTaskLogFilePathName, 527
GetRunningTaskProgress, 528
GetRunningTasksCount, 528
GetRunningTaskStatus, 528
GetScale, 317
GetSecurityAsPartnerID, 293
GetSecurityClassAccessForAllUsers, 482
GetSecurityClassAccessForAllUsers2, 482
GetSecurityClassID, 287
  HsvAccounts object, 279
  HsvEntities object, 293
  HsvScenarios object, 306
  HsvSecurityAccess object, 482
GetSecurityClassLabel, 483
GetSecurityClassRightsForConnectedUser, 483
GetServer
  HsxClient object, 174
  HsxClientUI object, 182
GetServerName, 529
GetServerOnCluster, 174
GetSettingsICPeriod, 758
GetSettingsICPeriods, 759
GetSortedNature
  HsvMDDataBuffer object, 628
  HsvMDDataBufferLite object, 651
GetSourceIDForIndexes, 704
GetSSOTokenUsingWebSecurityAgentCredentials, 174
GetState, 553
GetStatus, 365
GetStatusEx, 365, 366
GetStatusUsingPhaseID, 366
GetSubmissionGroup
  HsvAccounts object, 280
  HsvCustom object, 288
  HsvICPs object, 299
GetSupportSubmissionPhaseForAccountFlag, 219
GetSupportSubmissionPhaseForCustom1Flag, 219
GetSupportSubmissionPhaseForCustom2Flag, 219
GetSupportSubmissionPhaseForCustom3Flag, 219
GetSupportSubmissionPhaseForCustom4Flag, 220
GetSupportSubmissionPhaseForCustomXFlag, 220
GetSupportSubmissionPhaseForICPFlag, 220
GetSystemDataLinkFile, 188
GetSystemFolder, 188
GetTaskAccessForConnectedUserFromList, 483
GetTaskAuditAttachment, 529
GetTemplate, 430, 733
GetTemplateDisplayData, 431
GetTemplateExtDim, 430
GetTemplateLabelsForIDs, 432
GetTemplateQueryDefinitionIDs, 433
GetTextCell, 367
GetTextCellLineItems, 368
GetTextCells, 368
GetTextCellsWithRowSuppression, 369
GetTextCellsWithRowSuppression2, 370
GetTextJournal, 434
GetTextJournalExtDim, 434
GetTextLineItems, 371
GetTextTemplate, 436
GetTextTemplateExtDim, 436
GetTopMemberOfValidCustom1Hierarchy, 280
GetTopMemberOfValidCustom2Hierarchy, 281
GetTopMemberOfValidCustom3Hierarchy, 281
GetTopMemberOfValidCustom4Hierarchy, 282
GetTotalTransactions, 670
GetTransactionData
  HsvData object, 372
  HsvTransactionData object, 659
GetTransCurrencyID, 746
GetTransGroupType, 670
GetTreeCapabilities, 269
GetTreeName, 269
GetTreeTimeStam, 270
GetUnassignedGroups, 372
GetUnMatchTransactions, 746
GetURLByName, 373
GetURLsForCell, 374
GetUserAccessForAllRoles, 484
GetUserAccessForAllRoles2, 484
GetUserAccessForAllSecurityClasses, 485
GetUserAccessForAllSecurityClasses2, 485
GetUserDefined1, 288
  HsvAccounts object, 282
  HsvEntities object, 294
  HsvScenarios object, 307
GetUserDefined2, 289
  HsvAccounts object, 283
  HsvEntities object, 294
  HsvScenarios object, 308
GetUserDefined3, 289
  HsvAccounts object, 283
  HsvEntities object, 295
  HsvScenarios object, 308
GetUserDisplayDateTimeFormats, 842
GetUserID, 485
GetUserIDFromSID, 485
GetUserInfoFromUniqueID, 485
GetUserInfoFromUniqueID2, 486
GetUserLanguageFromHFMLanguageId, 842
GetUserName
  HsvSecurityAccess object, 486
  HsvSystemInfo object, 530
GetUserName2, 487
GetUserNameFromSID, 487
GetUserParameter, 530
GetUserSID, 487
GetUseSubmissionPhaseFlag, 221
GetValidationAccount, 222
GetValidationAccountMembers, 223
GetValueIDFromCurrencyID, 312
GetVariance, 437
GetVBScriptCalcRules, 531
GetVBScriptCalcRulesEx, 531
GetVBScriptMemberListRules, 531
GetVBScriptRules, 401
GetWebSecurityAgentSettings, 175
GetWindowsDateFormat, 842
GetWindowsDateFormatForLocale, 843
GetWorkingDirectory, 532
GetXBRITags, 283
GetXMLErrorFromDatabase, 189
GetXMLErrorsListFromDatabase, 189
GetYearRange, 312

H
  HasChildren, 270
  HasSystemChanged, 194
  HasUserStatusChanged, 194
  HFM_ACCESS_RIGHTS_ALL, 461
  HFM_ACCESS_RIGHTS_NONE, 461
  HFM_ACCESS_RIGHTS_READANDPROMOTE, 461
  HFM_ACCESS_RIGHTS_READONLY, 461
  HFM_ACCESS_RIGHTS_VIEW, 462
  HFM_USER_GROUP_ENUM_FLAGS, 463
HfmPovCOM object
    Clear, 142
    ConvertToHfmPovStrCOM, 142
    GetAllDims, 143
    GetCustomMemberat, 143
    GetCustoms, 144
    GetDimensionMember, 144
    GetLeftHandDims, 145
    GetMemberForCustomDim, 145
    GetRightHandDims, 146
    SetAllDims, 146
    SetCustomMemberAt, 148
    SetCustoms, 148
    SetDimensionMember, 149
    SetLeftHandDims, 149
    SetRightHandDims, 150
HfmPovCOM object properties, 141
HfmSliceCOM object
    Clear, 151
    ConvertToHfmSliceStrCOM, 151
    EnumCustomDimIds, 151
    GenerateCartesianProduct, 152
    GetFixedMember, 152
    GetMemberArrayForDim, 152
    GetMemberAt, 153
    GetNumCells, 153
    GetNumMembers, 153
    GetPOVAtIndex, 153
    InitializeFromPovString, 154
    SetFixedMember, 154
    SetMemberArrayForDim, 155
    SetMemberAt, 155
HoldingCompany attribute, returning, 292
HRESULTs, utility for, 843
HSV_DATA_UPDATE_MODE, 320
HSV_DATAEXTRACT_OPTION constants, 802
HSV_DATeload_OPTION constants, 800
HSV_JOURNALEXTRACT_EX_OPTION constants, 812
HSV_JOURNALEXTRACT_OPTION constants, 811
HSV_JOURNALLOAD_OPTION constants, 810
HSV_METALOADex_FILE_FORMAT constants, 789
HSV_SECURITYEXTRACT_OPTION constants, 778
HSV_SECURITYLOAD_OPTION constants, 777
HsvAccounts object
  GetAccountType, 275
  GetCalcAttribute, 276
  GetICPTopMember, 276
  GetIsICP, 277
  GetNumDecimalPlaces, 278
  GetPlugAccount, 279
  GetSecurityClassID, 279
  GetSubmissionGroup, 280
  GetTopMemberOfValidCustom1Hierarchy, 280
  GetTopMemberOfValidCustom2Hierarchy, 281
  GetTopMemberOfValidCustom3Hierarchy, 281
  GetTopMemberOfValidCustom4Hierarchy, 282
  GetUserDefined1, 282
  GetUserDefined2, 283
  GetUserDefined3, 283
  GetXBRLTags, 283
  IsCalculated, 283
  IsConsolidated, 284
  IsCustom1AggregationEnabled, 284
  IsCustom2AggregationEnabled, 285
  IsCustom3AggregationEnabled, 285
  IsCustom4AggregationEnabled, 286
  IsICP, 286
  IsICPRestricted, 286
  methods, summarized, 70
  object references, assigning, 275
  overview, 275
  UsesLineItems, 287
HsvAdminICM object
  CheckSecurityForICExtract, 756
  GetICEntitiesLockStatus, 757
  OpenICPeriod2, 762
  UpdatePeriodSettings2, 764
HsvCalculate object
  Allocate, 392
  Allocate2, 392
  CalcEPU, 393
  ChartLogic, 393
  ChartLogic2, 394
  Consolidate, 395
  Consolidate2, 396
  CustomLogic, 397
  EnumOnDemandRules, 397
  ExecuteOnDemandRules, 397
  FindOverlappingConsolidation, 398
  GetCOMDLLRules, 399
  GetConsolidationProgress, 399
  GetDefaultExchangeRate, 399
  GetDefaultExchangeRateExtDim, 399
  GetEPUInfo, 400
  GetEPUInfoExtDim, 400
  GetVBScriptRules, 401
  IsEntityAnEPUOwner, 402
  LoadCalcManagerRules, 402
  methods, summarized, 85
  object references, assigning, 392
  overview, 392
  SetCOMDLLRules, 402
  SetVBScriptRules, 402
  SetVBScriptRules2, 404
  StopConsolidation, 406
  Translate, 406
  Translate2, 407
HsvCalculate type library
  HsvCalculate object, 392
  overview, 391
HsvCategories object. See HsvScenarios object
HsvcDataLoad object
  DMEload, 805
  Extract, 771
  extract options, 802
  ExtractOptions, 770
  Load, 772
  load options, 800
  Load2, 805
  LoadAsync, 805
  LoadOptions, 770
  methods, summarized, 130
  object references, assigning, 805
  overview, 804
  properties, summarized, 129
  SetFileForLoad, 806
  SetSession, 773
StartLoad, 806
HsvcDataLoad type library
HsvcDataLoad, 804
IHsvLoadExtractOption interface, 797
IHsvLoadExtractOptions interface, 797
overview, 796
HsvCubeData object (deprecated), 611
HsvCurrencies object, 313
EnumCurrencies, 314
EnumCurrencies2, 314
GetCurrencyDescription, 315
GetCurrencyID, 315
GetCurrencyLabel, 316
GetCurrencyTranslationOperator, 316
GetScale, 317
HsvCurrencyCube object
BeginEnumerationOfStoredData, 676
EndEnumerationOfStoredData, 676
GetCell, 677
GetCellExtDim, 677
GetFixedDimensionMembers, 677
GetOneCellFromStoredItem, 678
GetOneCellFromStoredItemExtDim, 678
GetPOVFromStoredItem, 679
GetPOVFromStoredItemExtDim, 679
methods, summarized, 115
overview, 676
HsvCustom object
GetAggregationWeight, 287
GetSecurityClassID, 287
GetSubmissionGroup, 288
GetUserDefined1, 288
GetUserDefined2, 289
GetUserDefined3, 289
IsCalculated, 289
IsSwitchSignEnabledForFlow, 290
IsSwitchTypeEnabledForFlow, 290
methods, summarized, 71
object references, assigning, 287
overview, 287
HsvData object
AddDataToMDDataBuffer, 321
AttachDocumentToCell, 321
ClearAllData, 322
ClearAllDescriptionsInSubCube, 322
ClearDataAuditItems, 323
ClearDataAuditItemsEx, 324
ClearInputData, 324
ClearInvalidData, 326
CopyInputData, 327
CopyInputDataForMultipleEntities, 327
DeleteLineItemDetails, 328
DetachDocumentFromCell, 329
DMELoad, 329
DoCellDescriptionsExist, 329
DoesCellDescriptionExist, 330
DoesDataExist, 330
DoesSparseDataExist, 331
EnumDataAuditItems, 332
EnumDataAuditItems2, 333
EnumEntitiesWithDataForScenarioYear, 333
EnumExtractOptions, 333
EnumLoadOptions, 334
Extract, 335
ExtractDataAuditItems, 339
ExtractDrillableRegions, 340
ExtractDrillableRegionsByUrlNames, 340
FilterMembersThatHaveData, 341
FilterMembersThatHaveData2, 342
FormatNumberToText, 342
FormatNumberToText2, 344
FormatStatusToText, 346
FormatStoredNumberToText, 347
FormatTextToNumber, 348
FormatTextToStoredNumber, 349
GetAllDescriptionsInSubCube, 350
GetAllURLNames, 350
GetAttachedDocumentsToCell, 351
GetBaseDataForAccount, 351, 352
GetCalcStatus, 352
GetCalcStatusEx, 352, 353
GetCalcStatusStatistics, 353
GetCell, 354
GetCellDescription, 355
GetCellDescriptions, 355
GetCellHistory, 356
GetCellHistory2, 357
GetCellJournalEntries, 358
GetCellLineItems, 358
GetCells, 359
GetCellsWithRowSuppression, 359, 360
GetCellsWithRowSuppression2, 361
GetCountOfAttachedDocumentsToCell, 361, 362
GetCurrencyCube, 362
GetDataForAllMetadataCombinations, 362
GetDynamicAccountRulesCache, 363
GetLineItems, 363
GetMaxCellTextSize, 363
GetMembersThatHaveData, 363
GetMultiServerMaxSyncDelayForChanges, 364
GetNodeCube, 364
GetPhaseSubmissionGridForGivenScenarioPeriod, 365
GetStatus, 365
GetStatusEx, 365, 366
GetStatusUsingPhaseID, 366
GetTextCell, 367
GetTextCellLineItems, 368
GetTextCells, 368
GetTextCellsWithRowSuppression, 369
GetTextCellsWithRowSuppression2, 370
GetTextLineItems, 371
GetTransactionData, 372
GetUnassignedGroups, 372
GetURLByName, 373
GetURLsForCell, 374
InsertLineItemDetails, 375
IsValidCellText, 376
Load, 376
LoadDrillableRegions, 379
methods, summarized, 78
object references, assigning, 78
overview, 320
SetCalcStatusLocked, 380
SetCalcStatusLockedForMultipleProcessUnits, 381
SetCalcStatusUnlocked, 381
SetCalcStatusUnlockedForMultipleProcessUnits, 382
SetCell, 382
SetCellDescriptions, 383
SetCellLineItems, 383, 384
SetCells, 384
SetCells2, 385
SetCellsLineItems, 385
SetFileForLoad, 386
SetLineItems, 386
SetPhaseSubmissionGridForGivenScenarioPeriod, 387
SetTextCell, 387
SetTextCellLineItems, 388
SetTextCells, 388, 389
SetTextCellsLineItems, 389
SetTextLineItems, 390
StartLoad, 390
StartLoadTask, 390
UpdateDataUsingMDDataBuffer, 390
ValidateMembersAndConvertToIDs, 390
HsvData type library
HsvData object, 320
HsvDataPipe object, 319
IHsvDataValidation interface, 319
overview, 319
HsvDataCubes type library
HsvCurrencyCube, 676
HsvNodeCube, 680
overview, 675
HsvDataPipe object, 319
HsvDQI object, 704
AddEdgeDimension, 685
AddEdgeDimensionID, 686
AddEdgeSource, 686
AddEdgeSources, 686
AddSourceDimension, 687
AddSourceDimensionID, 687
AddToSourceJoinTypesForDimensions, 688
CollapseAll, 688
CollapseGrid, 689
CollapseGridDimension, 689
CollapseGridDimensionID, 690
CreateEdge, 691
CreateGrid, 691
CreatePageControl, 692
CreateSource, 692
DefineGrid, 693
DeleteGrid, 693
ExpandAll, 694
ExpandGrid, 694
ExpandGridDimension, 695
ExpandGridDimensionID, 695
ExpandGridID, 696
FreezeEdgeDefinition, 696
GetAllDataForGridPageDimensionIndex, 697
GetAllVisibleDimensionDetails, 697
GetGridCompleteDimensionDetails, 698
GetGridDetails, 699
GetGridDimensionDetails, 700
GetGridDimensionDetailsIDsAndLabels, 700
GetIndexesRangeForSourcesOnEdge, 701
GetPageControlDetails, 702
GetPOVForIndexes, 702
GetPOVIDsForIndexes, 703
GetSourceIDForIndexes, 704
JoinSourcesInSequence, 705
methods, 685
SelectGridPageDimensionIndex, 706
SelectMemberSourceAncestors, 706
SelectMemberSourceAncestorsID, 706
SelectMemberSourceBase, 707
SelectMemberSourceBaseID, 707
SelectMemberSourceChildren, 708
SelectMemberSourceChildrenID, 708
SelectMemberSourceDescendants, 709
SelectMemberSourceHierarchy, 709
SelectMemberSourceHierarchyExpanded, 710
SelectMemberSourceHierarchyExpandedID, 710
SelectMemberSourceHierarchyID, 711
SelectMemberSourceMemberID, 711
SelectMemberSourceMemberIDs, 712
SelectMemberSourceMembers, 712
SelectMemberSourceParents, 713
SelectMemberSourceParentsID, 713
SelectMemberSourceUserDefinedMemberList, 714
SelectMemberSourceUserDefinedMemberList2, 714
SelectPageSourceMemberIDsForDim, 714
SelectPageSourceMembersForDim, 715
SelectSliceSourceMemberForDim, 716
SelectSliceSourceMemberForDimID, 716
SetColsForPageControl, 717
SetEdgeRepeatedMetaDataSuppression, 717
SetGridPageControl, 720
SetGridSuppressionType, 720
SetPageMovement, 721
SetSourceNotSuppressible, 722
SetSourceOptionFillDefaultParentForEntity, 722
SetSourceSubType, 722

HsvDQI type library

HsvDQI object
methods, 685
overview, 685

HsvEntities object
GetAllowAdjustments, 291
GetAllowAdjustmentsFromChildren, 291
GetDefaultValueID, 292
GetHoldingCompany, 292
GetSecurityAsPartnerID, 293
GetSecurityClassID, 293
GetUserDefined1, 294
GetUserDefined2, 294
GetUserDefined3, 295
IsChild, 295
IsDescendant, 296
IsICP, 296
IsOrgByPeriodFilteringOn, 297
IsSecurityAsPartnerEnabled, 297
methods, summarized, 72
object references, assigning, 291, 298
overview, 291

HsvICM interface
object references, assigning, 737

HsvICM object, 737

AllMVTransPosted, 737
CheckReportSecurity, 738
CreateICTransaction, 738
CreateICTransactionExtDim, 738
DeleteICReasonCode, 739
DoesCellSupportICTransactionDetail, 740
DoesCellSupportICTransactionDetailExtDim, 740
GetColumnFilter, 740
GetCurrencyInfo, 740
GetEntitiesContacts, 741
GetEntityCurrencyID, 741
GetICReasonCodeID, 741
GetICReasonCodeLabel, 742
GetICReasonCodes, 742
GetICTransactions, 743
GetMonitorICDetails, 743
GetMonitorICSummary, 744
GetMonitorICTransactions, 745
GetRowFilter, 746
GetTransCurrencyID, 746
GetUnMatchTransactions, 746
IsOneSideOfTransactionGroupWriteable, 747

Index  987
MatchAutoAccounts, 748
MatchAutoIDs, 749
NoTransMatchedOrPosted, 749
ProcessAllICTrans, 750
ProcessICTransactions, 751
ProcessICTransactionsExtDim, 751
SaveColumnFilter, 752
SaveCReasonCode, 752
SaveICTransaction, 752
SaveICTransactionExtDim, 753
SaveRowFilter, 754
SetReasonCodeToICTransactions, 754
UnMatchICTransactions, 754
HsvICM type library, 737
HsvICM, 737
IHsvAdminICM, 755
HsvICPs object
  GetSubmissionGroup, 299
HsvICPs object references, assigning, 232
HsvICPs property, 232
HsvICTransactionsData object, 663
  AddAccountCustomCombination, 663
  AddAccountCustomCombinationExtDim, 663
  AddICTransactionData, 664
  AddICTransactionDataExtDim, 664
  AddQueryDimensionMemberID, 664
  AddQueryField, 664
  AddQueryFieldItem, 664
  AddQueryOrderField, 664
  BeginDataEnum, 664
  BeginQueryFieldEnum, 665
  BeginQueryOrderEnum, 665
  EndDataEnum, 665
  EndQueryFieldEnum, 665
  EndQueryOrderEnum, 665
  GetAccessRights, 665
  GetAccountCustomCombination, 666
  GetAccountCustomCombinationExtDim, 666
  GetAccountCustomCombinationTotal, 666
  GetEntityPartnerOption, 666
  GetErrorStatus, 666
  GetFilterOptions, 667
  GetFixedDimensionMembers, 667
  GetICTransactionCell, 667
  GetICTransactionCellExtDim, 667
  GetICTransactionData, 668
  GetMatchCode, 669
  GetNumTransactionsCached, 670
  GetPagingOption, 670
  GetPartnerAsEntityList, 670
  GetPartnerQueryDimensionMemberIDs, 670
  GetQueryDimensionMemberIDs, 670
  GetQueryFieldInformation, 670
  GetQueryFieldItem, 670
  GetQueryOrderField, 670
  GetTotalTransactions, 670
  GetTransGroupType, 670
  Initialize, 671
  InitializeSequenceIdMap, 672
  IsEntityInPartnerAsEntityList, 672
  IsICTransactionValid, 672
  IsICTransactionValidExtDim, 672
  object references, assigning, 663
  RemoveQueryOrder, 672
  SetAccessRights, 672
  SetEntityPartnerOption, 673
  SetErrorStatus, 673
  SetFilterOptions, 673
  SetICTransactionData, 673
  SetICTransactionDataExtDim, 673
  SetPagingOption, 673
  SetPartnerAsEntityList, 673
  SetPartnerQueryDimensionMemberIDs, 673
  SetQueryDimensionMemberIDs, 674
  SetTotalTransactions, 674
  SortByCell, 674
  UnInitialize, 674
HsvJournalLoadACV object
  Extract, 771
  extract options
    filtered extracts, 812
    unfiltered extracts, 811
  ExtractEx, 815
  ExtractOptions, 770
  ExtractOptionsEx, 815
  Load, 772
  load options, 810
  LoadOptions, 770
  methods, 815
  methods, summarized, 131
  object references, assigning, 815
  properties, 814
  properties, summarized, 130
  SetSession, 773
HsvJournalLoadACV type library
HsvJournalLoadACV, 814, 815
IHsvLoadExtractOption interface, 806
IHsvLoadExtractOptions interface, 806
overview, 806
HsvJournals object
ClosePeriod, 411
GetItemID, 412
GetJournalTemplateItemID, 412
IsPeriodOpen, 414
methods, summarized, 87
object references, assigning, 411
OpenPeriod, 414
overview, 411
HsvJournals type library
HsvJournals object, 411
IHsvJournalsEx interface, 415
IHsvJournalsReport interface, 454
overview, 409
HsvJournalsEx object
GetEntityJournals2, 422
GetEntityJournals2ExtDim, 422
GetJournal2, 425
GetJournal2ExtDim, 425
HsvJournalsReport object
GetReportData3, 458
HsvMDArrays type library
HsvCubeData (deprecated), 611
HsvICTransactionsData, 663
HsvMDCube (deprecated), 611
HsvMDDataBuffer, 612
HsvMDDataBufferLite, 634
HsvMDIndexList, 661
HsvMetadataSecurityBuffer (deprecated), 611
HsvTransactionData, 657
IHsvMDDataBufferLite, 656
overview, 611
HsvMDCube object (deprecated), 611
HsvMDDataBuffer object
BeginEnumeration, 613
CreateDataIndexList, 613
CreateDataIndexListEx, 614
CreateDataIndexListExtDim, 613
EndEnumeration, 614
EraseRecordFromPMBuffer, 614
GetCheckLineItemDetailsForCaseInsensitiveDuplicates, 615
GetCubeIndexFromPOV, 615
GetCubePOVFromIndex, 615
GetData, 616
GetDataAtIndex, 617
GetDataAtIndexExtDim, 617
GetDataExtDim, 616
GetDescription, 618
GetDescriptionAtIndex, 618
GetDescriptionAtIndexExtDim, 618
GetDescriptionExtDim, 618
GetLineItems, 619
GetLineItemsAtIndex, 620
GetLineItemsAtIndexExtDim, 620
GetLineItemsExtDim, 619
GetNumCellsForData, 620
GetNumCellsForDescriptions, 621
GetNumCellsForLineItems, 621
GetNumPeriodsInCubeForData, 622
GetNumPeriodsInCubeForDescriptions, 622
GetNumPeriodsInCubeForLineItems, 622
GetPeriodIndexFromPOV, 624
GetPeriodIndexFromPOVForData, 623
GetPeriodIndexFromPOVForLineItems, 625
GetPeriodPOVFromIndex, 627
GetPeriodPOVFromIndexForData, 626
GetPeriodPOVFromIndexForLineItems, 627
GetPMErrorRecordCount, 627
GetPMRecordCount, 628
GetRecordFromPMBuffer, 628
GetRecordFromPMErrorBuffer, 628
GetSortedNature, 628
indexes for methods, obtaining, 612
InsertDataAtBeginning, 628
InsertDataAtBeginningExtDim, 628
InsertDataAtEnd, 628
InsertDataAtEndExtDim, 628
InsertDescriptionAtBeginning, 629
InsertDescriptionAtBeginningExtDim, 629
InsertDescriptionAtEnd, 629
InsertDescriptionAtEndExtDim, 630
InsertLineItemsAtBeginning, 630
InsertLineItemsAtBeginningExtDim, 631
InsertLineItemsAtEnd, 631
InsertLineItemsAtEndExtDim, 631
InsertRecordIntoPMBuffer, 631
InsertRecordIntoPMErrorBuffer, 631
methods, summarized, 105
object references, assigning, 105
overview, 612
RemoveAll, 631
SetCheckLineItemDetailsForCaseInsensitiveDuplicates, 632
SetData, 632
SetDataExtDim, 632
SetDescription, 633
SetDescriptionExtDim, 633
SetGrowByAmount, 633
SetLineItems, 633
SetLineItemsExtDim, 634
SetSortedNature, 634
Sort, 634
HsvMDDataBufferLite object
BeginEnumeration, 635
CreateDataIndexList, 635
EndEnumeration, 637
EraseRecordFromPMBuffer, 636
GetCheckLineItemDetailsForCaseInsensitiveDuplicates, 637
GetCubeIndexFromPOV, 637
GetCubePOVFromIndex, 638
GetData, 638
GetDataAtIndex, 639
GetDescription, 640
GetDescriptionAtIndex, 641
GetDescriptionExtDim, 641
GetLineItems, 642
GetLineItemsAtIndex, 643
GetLineItemsAtIndexExtDim, 643
GetLineItemsExtDim, 643
GetNumCellsForData, 644
GetNumCellsForDescriptions, 644
GetNumCellsForLineItems, 645
GetNumPeriodsInCubeForData, 645
GetNumPeriodsInCubeForDescriptions, 646
GetNumPeriodsInCubeForLineItems, 646
GetPeriodIndexFromPOVForData, 646
GetPeriodIndexFromPOVForDescriptions, 648
GetPeriodIndexFromPOVForLineItems, 649
GetPeriodPOVForIndexForData, 650
GetPeriodPOVForIndexForDescriptions, 650
GetPeriodPOVForIndexForLineItems, 651
GetPMErrorRecordCount, 651
GetPMRecordCount, 651
GetRecordFromPMBuffer, 651
GetRecordFromPMErrorBuffer, 651
GetSortedNature, 651
indexes for methods, obtaining, 612
InsertDataAtEnd, 651
InsertDescriptionAtEnd, 652
InsertDescriptionAtEndExtDim, 652
InsertLineItemsAtEnd, 653
InsertLineItemsAtEndExtDim, 653
InsertRecordIntoPMBuffer, 653
InsertRecordIntoPMErrorBuffer, 653
methods, summarized, 109
object references, assigning, 108
overview, 634
RemoveAll, 653
SetCheckLineItemDetailsForCaseInsensitiveDuplicates, 654
SetData, 654
SetDataExtDim, 654
SetDescription, 655
SetDescriptionExtDim, 655
SetGrowByAmount, 655
SetLineItems, 655
SetLineItemsExtDim, 656
SetSortedNature, 656
Sort, 656
HsvMDIndexList object, 661
GetItem, 662
GetItemExtDim, 662
GetNumItems, 662
HsvMetadata object
Accounts, 230
Currencies, 230
Custom1, 231
Custom2, 231
Custom3, 231
Custom4, 231
Dimension, 231
Entities, 232
EnumCellTextLabels, 198
EnumConsolidationMethodIDs, 199
EnumCustomDimsForApp, 199
EnumCustomDimsForAppEx, 199
EnumDimProperties, 200
EnumExtractOptions, 200
EnumLanguages, 201
EnumLoadOptions, 202
EnumMemberProperties, 203
Extract, 203
ExtractModuleConfigurations, 206
GetApplicationAttribute, 206
GetApplicationAttributeExtDim, 206
GetApplicationCurrency, 207
GetApplicationSettingsTimeStamp, 207
GetAppProfileInfo, 208
GetByIndexValidationAccount, 208
GetCellLevelAccountType, 209
GetCellLevelAccountTypeExtDim, 209
GetCellTextIdFromLabel, 209
GetCellTextIdsFromLabels, 210
GetCellTextLabelDescription, 210
GetCellTextLabelFromId, 211
GetConsolidationMethodDescription, 211
GetConsolidationMethodInfo, 212
GetConsolidationMethodsTimeStamp, 213
GetCurrencyTimeStamp, 214
GetCurrencyValueIDForEntityValueCombination, 214
GetCurrencyValueIDsForEntityValueCombinations, 215
GetCustomX, 215
GetDefaultValueOfActiveStatusAccount, 216
GetDimensionId, 216
GetDimensionIdFromName, 217
GetFdmAppName, 217
GetFrequencyID, 217
GetICPEntitiesAggregationWeight, 218
GetICPEntitiesAggregationWeightEx, 218
GetMaxCustomDimNumber, 218
GetMaxDimId, 218
GetNumDimensions, 219
GetSupportSubmissionPhaseForAccountFlag, 219
GetSupportSubmissionPhaseForCustom1Flag, 219
GetSupportSubmissionPhaseForCustom2Flag, 219
GetSupportSubmissionPhaseForCustom3Flag, 219
GetSupportSubmissionPhaseForCustom4Flag, 220
GetSupportSubmissionPhaseForCustomXFlag, 220
GetSupportSubmissionPhaseForICPFlag, 220
GetUseSubmissionPhaseFlag, 221
GetValidationAccount, 222
GetValidationAccountMembers, 223
ICPs, 232
IsCustomDimValidForApp, 223
IsCustomMemberValidForAccount, 224
IsOrgByPeriodApplication, 225
Load, 225
LoadModuleConfigurations, 229
LoadWithAccessCode, 229
methods, summarized, 67
object references, assigning, 198
overview, 198
Periods, 232
properties, summarized, 66
Scenarios, 232
TranslateApplicationAttributeForDisplay, 229
TranslateApplicationAttributeForDisplayExtDim, 229
Values, 232
Views, 233
Years, 233
HsvMetadata type library
HsvAccounts object, 275
HsvCurrencies, 313
HsvCustom object, 287
HsvEntities object, 291
HsvMetadata object
methods, 198
properties, 230
HsvPeriods object, 299
HsvScenarios object, 303
HsvValues, 311
HsvYears, 312
IHsvTreeInfo interface, 233
object hierarchy, 66
overview, 197
HsvMetadataLoadACV object
Extract, 771
ExtractOptions, 770
Load, 772
LoadOptions, 770
methods, 789
methods, summarized, 128
object references, assigning, 789
properties, 789
properties, summarized, 128
SetSession, 773
HsvMetadataLoadACV type library
    HsvMetadataLoadACV methods, 789
    HsvMetadataLoadACV properties, 789
    IHsvLoadExtractOption interface, 781
    IHsvLoadExtractOptions interface, 781
overview, 780
HsvMetadataSecurityBuffer object (deprecated), 611
HsvNodeCube object
    BeginEnumerationOfStoredData, 680
    EndEnumerationOfStoredData, 680
    GetCell, 681
    GetCellExtDim, 681
    GetFixedDimensionMembers, 681
    GetOneCellFromStoredItem, 682
    GetOneCellFromStoredItemExtDim, 682
    GetPOVFromStoredItem, 683
    GetPOVFromStoredItemExtDim, 683
methods, summarized, 116
overview, 680
HsvPeriods object
    GetBaseFrequency, 300
    GetFrequency, 300
    GetNumPeriodsInFrequency, 301
    GetPeriodFromFrequency, 301
    GetPeriodsInFrequency, 302
methods, summarized, 73
object references, assigning, 300
overview, 299
HsvPOVSelection object, 818
    CheckAllItems, 818
    CheckAllItemsUsingSelectedList, 818
    CheckBoxChanged event, 829
    DimensionWasSelected event, 830
    EnableDimension, 819
    events, 829
    GetCheckedItems, 820
    GetListInfo, 820
    GetMember, 821
    GetNumCheckedItems, 822
    GetOBPInfo, 822
    Initialize, 823
    InitializeEx, 824
    InitializeHTTP, 824
    InitializeWithConnection, 824
    multi-select mode, 828
    MultiCheckBoxChanged event, 830
SelectDimension, 824
SelectionChanged event, 831
SetCheckedItems, 824
SetDimUnk, 825
SetListInfo, 826
SetMember, 826
SetOBPInfo, 827
SetResourceAndLanguage, 828
ShowCheckBoxes, 828
single-select mode, 828
HsvPOVSelection type library, 817
HsvPOVSelection, 818
HsvProcessFlow object
    Approve, 544
    Approve2, 545
    ApproveEx, 546
    GetGroupPhaseFromCell, 547
    GetGroupPhaseFromCellExtDim, 548
    GetHistory, 548
    GetPhasedSubmissionHistory, 551
    GetPhasedSubmissionHistoryExtDim, 551
    GetPhasedSubmissionState, 552
    GetPhasedSubmissionStateExtDim, 552
    GetPhasedSubmissionStateUsingPhaseID, 552
    GetPhasedSubmissionStateUsingPhaseIDExtDim, 553
    GetState, 553
    methods, summarized, 100
    object references, assigning, 543
    overview, 543
    PhasedSubmissionApprove, 554
    PhasedSubmissionApprove2, 555
    PhasedSubmissionApproveEx, 556
    PhasedSubmissionApproveExExtDim, 556
    PhasedSubmissionGetHistory2, 557
    PhasedSubmissionGetHistory2UsingPhaseIDExtDim, 558
    PhasedSubmissionProcessManagementChangeStateForMultipleEntities2, 560
    PhasedSubmissionPromote, 561
    PhasedSubmissionPromote2, 562
    PhasedSubmissionPromote2ExtDim, 562
    PhasedSubmissionPromoteExtDim, 561
    PhasedSubmissionPublish, 563
    PhasedSubmissionPublish2, 563
    PhasedSubmissionPublishExtDim, 564
    PhasedSubmissionPublishEx, 565
PhasedSubmissionPublishExExtDim, 565
PhasedSubmissionPublishExtDim, 563
PhasedSubmissionReject, 565
PhasedSubmissionReject2, 566
PhasedSubmissionReject2ExtDim, 566
PhasedSubmissionRejectExtDim, 565
PhasedSubmissionSignOff, 567
PhasedSubmissionSignOff2, 568
PhasedSubmissionSignOff2ExtDim, 568
PhasedSubmissionSignOffExtDim, 567
PhasedSubmissionStart, 569
PhasedSubmissionStart2, 570
PhasedSubmissionStart2ExtDim, 570
PhasedSubmissionStartEx, 571
PhasedSubmissionStartExExtDim, 571
PhasedSubmissionSubmit, 571
PhasedSubmissionSubmit2, 572
PhasedSubmissionSubmit2ExtDim, 572
PhasedSubmissionSubmitExtDim, 572
ProcessManagementChangeStateForMultipleEntities, 573
ProcessManagementChangeStateForMultipleEntities2, 574
ProcessManagementChangeStateForMultipleEntities2ExtDim, 574
ProcessManagementChangeStateForMultipleEntitiesEx, 575
ProcessManagementChangeStateForMultipleEntitiesExExtDim, 575
Promote, 576
Promote2, 578
Publish, 579
Publish2, 580
PublishEx, 582
Reject, 583
SignOff, 585
SignOff2, 586
Start, 587
Start2, 589
StartEx, 590
submission phases and process units, 543
Submit, 591
Submit2, 592
HsvProcessFlow type library
HsvProcessFlow object, 543
overview, 543
HsvReports object
CheckReportSecurityClass, 595
CheckReportSecurityClass2, 596
CheckSecurityRole, 596
DeleteDocument, 597
DeleteReport, 597
DeleteReports, 598
DoesDocumentExist, 598
DoesFolderExist, 599
EnumDocuments, 599
EnumDocumentsEx, 600
EnumReports, 602
GetDocument, 603
GetDocumentEx, 604
GetReport, 605
methods, summarized, 103
object references, assigning, 103
overview, 595
Reports property and, 192
SaveDocument, 606
SaveDocumentEx, 607
SetReport, 607
ValidateMembersAgainstSlice, 609
HsvReports type library
HsvReports, 595
overview, 595
HsvResourceManager object, 834
GetAvailableLanguages, 835
GetCurrentHSSRegistrationVersion, 836
GetCurrentVersion, 836
GetCurrentVersionInUserDisplayFormat, 836
GetFormattedDateTime, 836
GetFormattedDateTimeForLanguage, 836
GetFormattedError, 837
GetFormattedErrorWithLineFeed, 839
GetFormattedResourceString, 839
GetHelpDirectoryForLanguageID, 840
GetHFMLanguageIDFromUserLanguages, 840
GetLanguageCountryCodeFromLanguageID, 840
GetLocaleIDFromLanguageID, 841
GetString, 841
GetStringFromHR, 841
GetUserDisplayDataTimeFormats, 842
GetUserLanguageFromHFMLanguageID, 842
GetWindowsDateFormats, 842
GetWindowsDateTimeFormatForLocale, 843
Initialize, 843
HsvResourceManager type library, 833
HsvResourceManager, 834
HsvRulesLoadACV object
  ExtractCalcRules, 790
  ExtractMemberListRules, 791
  GetCalcRulesType, 792
  LoadCalcRules, 792
  LoadCalcRules2, 793
  LoadMemberListRules, 794
methods, summarized, 128
object references, assigning, 790
overview, 790
ScriptableLoadCalcRules, 795
ScriptableLoadCalcRules2, 795
ScriptableLoadMemberListRules, 795
SetSession, 795
SetSessionAndResource, 796
HsvRulesLoadACV type library
HsvRulesLoadACV, 790
overview, 790
HsvScenarios object
  EnumPhasedSubmissionStartYears, 303
  GetDefaultFrequency, 304
  GetDefaultView, 304
  GetMaximumReviewLevel, 305
  GetMissingDataZeroViewForAdjValues, 305
  GetMissingDataZeroViewForNonAdjValues, 306
  GetPhasedSubmissionStartYear, 303
  GetSecurityClassID, 306
  GetUserDefined1, 307
  GetUserDefined2, 308
  GetUserDefined3, 308
  IsConsolidateYTD, 309
  IsPhasedSubmissionEnabled, 309
methods, summarized, 73, 74
object references, assigning, 303
overview, 303
SupportsAccountAllocations, 310
SupportsEmailAlerting, 310
SupportsProcessFlow, 310
SupportsTargetSetting, 311
SupportsTimePeriodAllocations, 311
UsesLineItems, 311
HsvSecurityAccess object
  AddApplicationAdministrator, 465
  AddApplicationAdministrator2, 465
  AddOrRemoveApplicationAdministrators, 465
AddOrRemoveApplicationAdministrators2, 465
AddOrRemoveRolesFromUser, 466
AddOrRemoveRolesFromUser2, 466
AddOrRemoveUsersFromRole, 467
AddOrRemoveUsersFromRole2, 467
AddSecurityClass, 468
AddSecurityClassWithAccessCode, 468
AddUser, 468
AddUser2, 468
AddUserEx, 468
AddUserToRole, 469
AddUserToRole2, 469
AllowRulesLoadForEPMAApp, 469
DeleteSecurityClass, 469
DeleteSecurityClasses, 470
DeleteSecurityClassWithAccessCode, 470
EnumApplicationAdministrators, 470
EnumApplicationAdministrators2, 470
EnumRoles, 471
EnumRolesForPrincipal, 471
EnumRolesForUser, 472
EnumSecurityClasses, 472
EnumSecurityClassRightsForPrincipal, 473
EnumUserClassAccess, 474
EnumUsers, 474
EnumUsers2, 474
EnumUsers3, 474
EnumUsersInGroup, 475
EnumUsersInRole, 475
EnumUsersInRole2, 475
EnumUsersInRole3, 476
EnumUsersInSecurityClass, 476
EnumUsersInSecurityClass2, 476
EnumUsersInSecurityClass3, 476
EnumUsersInSecurityClass4, 477
EnumUsersOrGroups, 477
EnumUsersWithFilter, 477
EnumUsersWithFilter2, 477
GenerateSecurityReportForBiPub, 478
GetAllSecurityClassRightsForConnectedUser, 478
GetApplicationAdministratorAccessForAllUsers, 479
GetApplicationAdministratorAccessForAllUsers2, 479
GetConnectedUser, 479
GetConnectedUser2, 479
GetIdentityTypes, 480
Index 995

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

GetNumRoles, 480
GetOwner, 480
GetRoleAccessForAllUsers, 480
GetRoleAccessForAllUsers2, 480
GetRoleID, 481
GetRoleLabel, 481
GetRulesMode, 482
GetSecurityClassAccessForAllUsers, 482
GetSecurityClassAccessForAllUsers2, 482
GetSecurityClassID, 482
GetSecurityClassLabel, 483
GetSecurityClassRightsForConnectedUser, 483
GetTaskAccessForConnectedUserFromList, 483
GetUserAccessForAllRoles, 484
GetUserAccessForAllRoles2, 484
GetUserAccessForAllSecurityClasses, 485
GetUserAccessForAllSecurityClasses2, 485
GetUserID, 485
GetUserIDFromSID, 485
GetUserInfoFromUniqueID, 485
GetUserInfoFromUniqueID2, 486
GetUserName, 486
GetUserName2, 487
GetUserNameFromSID, 487
GetUserID, 487
InsertDefaultSecurityClass, 487
InsertDefaultSecurityClassWithAccessCode, 488
IsApplicationAdministrator, 488
IsClassicHFMApplication, 488
IsConnectedUserAllowedToPerformTask, 488
IsConnectedUserInRole, 489
IsValidWindowsUser, 489
methods, summarized, 91
object references, assigning, 464
overview, 464
RemoveApplicationAdministrator, 489
RemoveApplicationAdministrator2, 489
RemoveUser, 490
RemoveUser2, 490
RemoveUserFromRole, 490
RemoveUserFromRole2, 490
RenameSecurityClass, 491
SetManySecurityClassRightsForUser, 491
SetManySecurityClassRightsForUser2, 491
SetRolesForUser, 492
SetRulesMode, 492
SetSecurityClassLabel, 492
SetSecurityClassLabelWithAccessCode, 493
SetSecurityClassRightsForManyUsers, 493
SetSecurityClassRightsForManyUsers2, 493
SetSecurityClassRightsForUser, 494
SetSecurityClassRightsForUser2, 494
SetUserClassAccess, 494
TakeOwnership, 495
HsvSecurityAccess type library
access rights constants, 461
connected user, 464
HsvSecurityAccess, 464
IHsvDataSecurity, 495
overview, 461
role IDs, 462
security identifiers (SIDs), 464
task IDs, 462
HsvSecurityLoadACV object
Extract, 771
extract options, 778
ExtractOptions, 770
Load, 779
load options, 777
LoadOptions, 770
methods, summarized, 127
object references, assigning, 779
overview, 779
properties, summarized, 127
SetSession, 773
HsvSecurityLoadACV type library
HsvSecurityLoadACV, 779
IHsvLoadExtractOption interface, 774
IHsvLoadExtractOptions interface, 774
overview, 774
HsvSession object
Calculate, 191
CreateObject, 193
Data property, 191
GetECID, 193
getHsvDQI, 194
GetLicenseExpirationStatus, 194
HasSystemChanged, 194
HasUserStatusChanged, 194
ICM, 192
IsBusy, 194
IsRunningTasks, 195
Journals, 192
LockMetadataLoadWithSystemChangeCheck, 195
Metadata, 192
methods, 193
object references, assigning, 191
ProcessFlow, 192
properties, 191
properties and methods, summarized, 64
Reports, 192
Security, 192
SystemInfo, 192
UnlockMetadataLoad, 195
HsvSession type library
HsvSession object
methods, 193
properties, 191
overview, 191
HsvStarSchemaACM object, 725
CreateStarSchema, 726
CreateStarSchemaAndReturnTaskID, 727
CreateStarSchemaExtDim, 726
CreateStarSchemaFromHTTP, 728
DeleteStarSchema, 728
EnumApplicationStarSchemas, 728
EnumRegisteredDSNs, 728
GetAsynchronousTaskStatus, 729
GetExtractLogData, 730
GetPersistedSettings, 731
GetPersistedSettingsExtDim, 731
object references, assigning, 726
QuitAsynchronousTask, 731
SetDefaultTablePrefix, 731
SetPersistedSettings, 731
SetPersistedSettingsExtDim, 731
SetSession, 731
TestSQLConnection, 732
HsvStarSchemaACM type library, 725
HsvStarSchemaACM, 725
IHsvStarSchemaTemplates, 732
HsvSystemInfo object
AddRefToHsxServer, 501
AddTaskToAudit, 501
AddTaskToAuditWithAttachment, 502
AddTaskToRunningTasks, 502
AddTaskToRunningTasksAndUpdatePOV, 502
CheckAccess, 503
ClearAuditTasks, 503
ClearAuditTasks2, 504
ClearRunningTask, 505
DeleteUserParameter, 505
DisableNewConnections, 505
EnableNewConnections, 506
EnumActivityServers, 506
EnumActivityUsers, 507
EnumActivityUsersForRunningTasks, 507
EnumAuditTasks, 507
EnumAuditTasks2, 509
EnumAuditTasksEx, 511
EnumProhibitConnections, 511
EnumRunningTasks, 514
EnumRunningTasksEx, 516
EnumRunningTasksPOV, 518
ExtractTaskAudit, 520
ExtractTaskAudit2, 521
GetActivityCodeDesc, 522
GetActivityUserID, 522
GetAllInfoForTask, 522
GetApplicationDirectory, 522
GetApplicationName, 523
GetCalcRulesType, 524
GetCOMDLLCalcRules, 524
GetCurrentActivity, 524
GetExtractCalcRules, 524
GetFormattedDateTime, 525
GetFormattedResourceString, 525
GetKillUsersStatus, 525
GetKillUserStatus, 525
GetLastModifiedDateAndModifiedByForArtifact, 526
GetLastModifiedDateForArtifact, 526
GetModuleName, 526
GetNumberFormattingUserParameters, 526
GetResourceLanguageUserParameters, 527
GetResourceString, 527
GetResourceStringFromHR, 527
GetRunningTaskLogFilePathName, 527
GetRunningTaskProgress, 528
GetRunningTasksCount, 528
GetRunningTaskStatus, 528
GetServerName, 529
GetTaskAuditAttachment, 529
GetUserName, 530
GetUserParameter, 530
GetVBScriptCalcRules, 531
GetVBScriptCalcRulesEx, 531
GetVBScriptMemberListRules, 531
GetWorkingDirectory, 532
IsLoggingEnabled, 532
IsScheduledTaskReadyToRun, 533
KeepRunningTaskStillAlive, 533
KillUsers, 533
methods, summarized, 96
object references, assigning, 501
OutputSystemInfo, 533
overview, 501
ReleaseHsxServer, 534
SaveLastModifiedDateAndModifiedBy, 534
SetApplicationDirectory, 534
SetCOMDLICalcRules, 534
SetCurrentActivity, 534
SetCurrentModule, 534
SetCurrentModuleEx, 535
SetExtractFileEncoding, 535
SetLanguageUserParameters, 535
SetNumberFormattingUserParameters, 536
SetResourceLanguageForCurrentSession, 536
SetResourceLanguageUserParameters, 537
SetUserParameter, 537
SetVBScriptCalcRules, 538
SetVBScriptMemberListRules, 539
StopRunningTask, 541
TraceLog, 541
UpdateRunningTaskLogFilePathName, 542
UpdateRunningTaskPOV, 542
UpdateRunningTaskProgress, 542
UpdateRunningTaskProgressDetails, 542
UpdateRunningTaskStatus, 542
WarnUsersForShutDown, 542
HsvSystemInfo type library
HsvSystemInfo object, 501
overview, 501
HsvTransactionData object
BeginDataEnum, 658
BeginQueryEnum, 658
EndDataEnum, 658
EndQueryEnum, 659
GetFixedDimensionMembers, 659
GetQueryItem, 659
GetQueryItemExtDim, 659
GetTransactionData, 659
Initialize, 660
object references, assigning, 657
overview, 657
SetAccessRight, 660
SetQueryItem, 661
SetQueryItemExtDim, 661
SetTransactionData, 661
SetTransactionDataExtDim, 661
HsvTreeInfo object
EnumMembersWithAttribValues, 245
EnumSortedIDsOfIChildren, 247
HsvValues object
GetCurrencyIDFromValueID, 311
GetValueIDFromCurrencyID, 312
methods, 311
HsvValues object, object references, assigning, 232
HsvViews object, object references, assigning, 233
HsvYears object
GetYearRange, 312
object references, assigning, 312
overview, 312
HsxClient object
AuthenticateCSSToken, 158
AuthenticateSecurityAgentCredentials, 158
AuthenticateUserCredentials, 158
AuthenticateUserOnClient, 158
AuthenticateUserOnCluster, 159
AuthenticateUserOnClusterSSO, 159
AuthenticateUserOnServer, 159
CreateApplicationCAS, 159
CreateApplicationCASWithAccessCode, 161
CreateObjectOnCluster, 161
CreateObjectOnServer, 161
DeleteApplication, 161
DeleteApplicationWithAccessCode, 162
DeleteSystemErrors, 162
DetermineWindowsLoggedOnUser, 162
DisableNewConnections, 162
DoesUserHaveCreateApplicationRights, 163
DoesUserHaveSystemAdminRights, 163
EnableNewConnections, 164
EnumProhibitConnections, 165
EnumProvisioningProjects, 166
EnumRegisteredClusterNames, 166
EnumRegisteredServerNames, 167
EnumUserAppPreferences, 167
EnumUsersOnSystem, 167
EnumUsersOnSystemEx, 169
GetApplicationFolder, 172
GetClusterInfo, 172
GetClustersAndServers, 173
GetHFMErrorLogRecordSet, 173
GetLogonInfo, 173
GetLogonInfoSSO, 173
GetModifyApplicationInfo, 174
GetModifyApplicationStatus, 174
GetServer, 174
GetServerOnCluster, 174
GetSSOTokenUsingWebSecurityAgentCredentials, 174
GetWebSecurityAgentSettings, 175
IsValidApplication, 175
KillUsers, 175
methods, summarized, 60
ModifyApplication, 176
object references, assigning, 157
OpenApplication, 176
overview, 157
RegisterApplicationCAS, 176
RegisterApplicationCASWithAccessCode, 177
RegisterCluster, 177
RegisterServer, 178
ScriptableEnumRegisteredClusterNames, 178
ScriptableEnumRegisteredServerNames, 178
ScriptableGetLogonInfo, 178
ScriptableOpenApplication, 178
SetApplicationFolder, 178
SetLogonInfo, 179
UnregisterAllClusters, 179
UnregisterCluster, 179
UnregisterServer, 180
UpdateUserAppPreferences, 180
WarnUsersForShutDown, 180
HsxClient type library
duplicate references in References dialog box, 956
HsxClient object, 157
overview, 157
HsxClientUI object
DeleteApplication, 182
GetServer, 182
Initialize, 182
Logon, 182
methods, summarized, 62
object references, assigning, 181
OpenApplication, 183
overview, 181
RegisterServer, 184
HsxClientUI type library
duplicate references in References dialog box, 956
HsxClientUI object, 181
overview, 181
HsxServer object
CSSEnabled, 190
DeleteSystemErrors, 186
DeleteSystemErrorsEx, 186
EnumDataSources, 186
EnumRegisteredDSNs, 187
GetClustersAndServers, 187
GetDataSource, 188
GetHFMErrorLogRecordSet, 188
GetSystemDataLinkFile, 188
GetSystemFolder, 188
GetXMLLErrorFromDatabase, 189
GetXMLErrorsListFromDatabase, 189
methods, summarized, 63
object references, assigning, 185
overview, 185
ScriptableEnumDataSources, 190
ScriptableEnumRegisteredDSNs, 190
HsxServer type library
duplicate references in References dialog box, 956
HsxServer object, 185
overview, 185
ICM, 192
ICPs, 232
identity types, 463
returning for users, 480
IDs, member, 197
IHsvAdminICM interface, 755
CloseICPeriod, 756
GetLockStatusICEntities, 757
GetLockStatusICEntity, 757
GetSettingsICPeriod, 758
GetSettingsICPeriods, 759
LoadICTransactions, 761
LoadICTransactionsExtDim, 761
LockICEntity, 761
object references, assigning, 755
OpenICPeriod, 762
SavePeriodsSettings, 763
UnlockICEntity, 763
UpdatePeriodSettings, 763
IHsvDataSecurity interface
GetCellLevelAccessRights, 495
GetCellLevelAccessRightsExtDim, 495
GetProcessUnitAccessRights, 495
GetProcessUnitAccessRightsAndState, 496
GetProcessUnitAccessRightsAndStateEx, 497
GetProcessUnitAccessRightsAndStateExExtDim, 497
GetProcessUnitAccessRightsEx, 498
GetProcessUnitAccessRightsExExtDim, 498
methods, summarized, 95
object references, assigning, 495
overview, 495
RefreshAccessRightsCache, 499
IHsvDataValidation interface, 319
IHsvJournalsEx interface
AddJournalGroup, 415
ApproveJournals, 415
DeleteJournals, 416
DeleteTemplates, 417
EnumJournalGroups, 418
EnumJournalGroupsEx, 418
EnumJournalGroupsForScenarioYear, 419
EnumJournalIDsForExtractFilter, 419
GenerateRecurring, 420
GetEntityJournals, 421
GetEntityJournalsExtDim, 421
GetJournal, 423
GetJournalDisplayData, 426
GetJournalExtDim, 423
GetJournalLabelsForIDs, 428
GetJournalQueryDefinitionIDs, 428
GetTemplate, 430
GetTemplateDisplayData, 431
GetTemplateExtDim, 430
GetTemplateLabelsForIDs, 432
GetTemplateQueryDefinitionIDs, 433
GetTextJournal, 434
GetTextJournalExtDim, 434
GetTextTemplate, 436
GetTextTemplateExtDim, 436
GetVariance, 437
methods, summarized, 88
object references, assigning, 415
overview, 415
PostJournals, 439
RejectJournals, 440
RemoveAllJournalGroups, 441
RemoveJournalGroup, 441
SaveJournal, 441
SaveJournalExtDim, 441
SaveTemplate, 443
SaveTemplateExtDim, 443
SaveTextJournal, 445
SaveTextTemplate, 448
SaveTextTemplateExtDim, 448
SubmitJournals, 450
UnpostJournals, 451
UnsubmitJournals, 452
ValidateLineItems, 453
ValidateLineItemsExtDim, 453
ValidateValue, 454
IHsvJournalsReport interface
GetReportData, 456
GetReportData2, 458
GetReportDataExtDim, 457
methods, summarized, 90
object references, assigning, 454
overview, 454
IHsvLoadExtractOption interface
CurrentValue, 767
DefaultValue, 768
HsvcDataLoad type library, 797
HsvJournalLoadACV type library, 806
HsvMetadataLoadACV type library, 781
HsvSecurityLoadACV type library, 774
MaxValue, 768
MinValue, 768
Name, 769
object references, assigning, 767
OptionID, 769
overview, 767
properties, summarized, 126
ValidationList, 769
IHsvLoadExtractOptions interface
Count, 766
HsvcDataLoad type library, 797
HsvJournalLoadACV type library, 806
HsvMetadataLoadACV type library, 781
HsvSecurityLoadACV type library, 774
Item, 766
object references, assigning, 766
overview, 766
properties, summarized, 125
IHsvMDDataBufferLite interface, 656
SetMinCubesInCache, 657
IHsvStarSchemaTemplates interface, 732
DeleteTemplate, 732
EnumTemplates, 732
GetTemplate, 733
object references, assigning, 732
SetTemplate, 733
IHsvTreeInfo interface
EnumAllMemberIDs, 233
EnumAllMemberLabels, 234
EnumAllParentAndChildIDs, 234
EnumAllParentAndChildLabels, 235
EnumAncestors, 236
EnumBaseMemberIDs, 237
EnumDefaultAncestors, 238
EnumDefaultAncestorsLabels, 238
EnumDescendants, 239
EnumIDsOfChildren, 240
EnumIDsOfChildren, 241
EnumMemberLists, 242
EnumMembers, 242
EnumMembers2, 243
EnumParents, 245
EnumSortedIDsOfChildren, 246
EnumSortedMembers, 247
Find, 248
FindByDesc, 250
FindMatchingMembersFromHierarchy, 251
FindMatchingMembersFromHierarchyByDesc, 252
FindMatchingMembersFromHierarchyWildCard, 253
GetAllPathsToMember, 254
GetAttributeValue, 255
GetDefaultHierarchyPosition, 256
GetDefaultItemIDHierarchy, 256
GetDefaultParent, 258
GetDefaultParentLabel, 258
GetDescription, 259
GetDisplayInfo, 260
GetDisplayInfoForSeveralItems, 261
GetItemGeneration, 262
GetItemID, 263
GetItemIDQL, 264
GetItemLevel, 264
GetLabel, 264
GetMemberListID, 265
GetMemberListName, 265
GetNumBaseMembers, 266
GetNumChildren, 266
GetNumDescendants, 267
GetNumMembers, 268
GetNumParents, 268
GetQualifiedLabel, 269
GetTreeCapabilities, 269
GetTreeNode, 269
GetTreeTimeStam, 270
HasChildren, 270
IsMemberABaseOf, 271
IsMemberAChildOf, 272
IsMemberADescendantOf, 272
methods, summarized, 75
object references, assigning, 233
overview, 233
SortMembersBasedOnList, 273
TranslateAttributeValueForDisplay, 274
IHsvTreeInfo object
GetDefaultMemberID, 257
IHsxClient object
EnumRegisteredClustersOrServers, 167
EnumUsersOnSystemEx2, 170
IHsxServer object
getFileTransfer, 188
indexes, obtaining for HsvMDDataBuffer and
HsvMDDataBufferLite objects, 612
Initialize
HsvICTransactionsData object, 671
HsvPOVSelection object, 823
HsvResourceManager object, 843
HsvTransactionData object, 660
HsxClientUI object, 182
InitializeEx, 824
InitializeFromPovString, 154
InitializeHTTP, 824
InitializeSequenceIdMap, 672
InitializeWithConnection, 824
InsertDataAtBeginning, 628
InsertDataAtBeginningExtDim, 628
InsertDataAtEnd
HsvMDDataBuffer object, 628
HsvMDDataBufferLite object, 651
InsertDataAtEndExtDim
HsvMDDataBuffer object, 628
InsertDefaultSecurityClass, 487
InsertDefaultSecurityClassWithAccessCode, 488
InsertDescriptionAtBeginning, 629
InsertDescriptionAtBeginningExtDim, 629
InsertDescriptionAtEnd
HsvMDDataBuffer object, 629
HsvMDDataBufferLite object, 652
InsertDescriptionAtEndExtDim
HsvMDDataBuffer object, 630
HsvMDDataBufferLite object, 652
InsertLineItemDetails, 375
InsertLineItemsAtBeginning, 630
InsertLineItemsAtEnd
HsvMDDataBuffer object, 631
HsvMDDataBufferLite object, 653
InsertLineItemsAtEndExtDim
HsvMDDataBuffer object, 631
HsvMDDataBufferLite object, 653
InsertRecordIntoPMBuffer
HsvMDDataBuffer object, 631
HsvMDDataBufferLite object, 653
InsertRecordIntoPMErrorBuffer
HsvMDDataBuffer object, 631
HsvMDDataBufferLite object, 653
intercompany matching
by account, 748
by ID, 749
Intercompany Partner dimension
children of member, sorting, 246
members, sorting, 247
object for, 298
intercompany transactions
managing, 755
processing, 737
sequence IDs, 668
testing if accounts used in, 286
testing if entities support, 296
invalid records, deleting, 326
IsApplicationAdministrator, 488
IsBusy, 194
IsCalculated
HsvAccounts object, 283
HsvCustom object, 289
IsCalculated attribute, returning, 283
IsChild, 295
IsClassicHFMApplication, 488
IsConnectedUserAllowedToPerformTask, 488
IsConnectedUserInRole, 489
IsConsolidated, 284
IsConsolidated attribute, returning, 284
IsConsolidateYTD, 309
IsCustom1AggregationEnabled, 284
IsCustom2AggregationEnabled, 285
IsCustom3AggregationEnabled, 285
IsCustom4AggregationEnabled, 286
IsCustomDimValidForApp, 223
IsCustomMemberValidForAccount, 224
IsDescendant, 296
IsEntityAnEPUOwner, 402
IsEntityInPartnerAsEntityList, 672
IsICP
HsvAccounts object, 286
HsvEntities object, 296
IsICPRestricted, 286
IsICTransactionValid, 672
IsICTransactionValidExtDim, 672
IsLoggingEnabled, 532
IsMemberABaseOf, 271
IsMemberAChildOf, 272
IsMemberADescendantOf, 272
IsOneSideOfTransactionGroupWriteable, 747
IsOrgByPeriodApplication, 225
IsOrgByPeriodFilteringOn
HsvEntities object, 297
IsPeriodOpen, 414
IsPhasedSubmissionEnabled, 309
IsRunningTasks, 195
IsScheduledTaskReadyToRun, 533
IsSecurityAsPartnerEnabled, 297
IsSwitchSignEnabledForFlow, 290
IsSwitchTypeEnabledForFlow, 290
IsValidApplication, 175
IsValidCellText, 376
IsValidWindowsUser, 489
Item, 766
items, subcubes, 675
J
JoinSources, 704
JoinSourcesInSequence, 705
journals
extract options
filtered extracts, 812
unfiltered extracts, 811
extracting
ever example, filtered extract, 810
ever example, unfiltered extract, 808
overview, 807
load options, 810
loading
ever example, 807
overview, 807
objects and interfaces for, 86
Journals property, 192

K
KeepRunningTaskStillAlive, 533
KillUsers
HsvSystemInfo object, 533
HsxClient object, 175

L
language
getting the ID of a language for description, 201
returning an application's languages for
description, 201
returning the default language for descriptions, 525
returning the default language for resource strings, 527
setting the default language for descriptions, 535
setting the default language for resource strings, 537
setting the default language for resource strings for
a session, 536
language code, obtaining, 840
language IDs, obtaining for supported languages, 835
level, member, getting, 264
line items, examples for getting and setting, 138
Load
HsvcDataLoad, HsvJournalLoadACV, and
HsvMetadataLoadACV objects, 772
HsvData object, 376
HsvMetadata object, 225
HsvSecurityLoadACV object, 779
Load2, 805
LoadAsync
HsvcDataLoad object, 805
LoadCalcManagerRules, 402
LoadCalcRules, 792
LoadCalcRules2, 793
LoadDrillableRegions, 379
LoadICTransactions, 761
LoadICTransactionsExtDim, 761
loading data, 797
loading journals, 807
loading member lists, 794
loading metadata
file format constants, 789
overview, 781
loading rules, 792
loading security, 775
LoadMemberListRules, 794
LoadModuleConfigurations, 229
LoadOptions, 770
LoadWithAccessCode
HsvMetadata object, 229
local storage folder
getting, 172
setting, 178
LockICEntity, 761
LockMetadataLoadWithSystemChangeCheck, 195
logging on
ever example, 137
HsxClient object, 179
HsxClientUI object, 182
Logon, 182
Logon dialog box, displaying, 182
LST file
getting contents of, 531
loading and validating, 794

M
MatchAutoAccounts, 748
MatchAutoIDs, 749
matching, intercompany
by account, 748
by ID, 749
MaximumReviewLevel attribute, 305
MaxValue, 768
member IDs, 197
member lists
dynamic, getting members of, 243
extracting, 791
getting members of, 242
loading, 794
MEMBER_LIST_ALL_HIERARCHY, 862
MEMBERNOTUSED, 858
Metadata, 192
metadata
  attribute, getting value of, 255
  examples, 138
  extract options
    file format constants, 789
    overview, 786
extracting
  example, 783
  overview, 782
load options
  file format constants, 789
  overview, 783
loading
  example, 782
  overview, 781
objects and interfaces for, 65
testing to see if reloaded, 194
metadata status
  constants for, 867
  overview, 319
methods, 193
MinValue, 768
ModifyApplication
  HsxClient object, 176
multi-select mode, 828
MultiCheckBoxChanged event, 830

N
Name, 769
node subcubes
  accessing with HsvNodeCube object, 680
described, 53
NoTransMatchedOrPosted, 749
Number property, Err object (Visual Basic), 55
NumDecimalPlaces attribute, returning, 278

O
Open Application dialog box, displaying, 183
OpenApplication
  HsxClient object, 176
  HsxClientUI object, 183
OpenICPeriod, 762
OpenICPeriod2, 762
opening application, example, 137
OpenPeriod, 414
OptionID, 769
Organization by Period, 225
organization by period
testing if entities support, 297
OrgByPeriodApplication application setting, 225
OutputSystemInfo, 533

P
parent
  getting ID of a member's default, 258
  getting label of a member's default, 258
password, setting, 179
Periods property, 232
periods, object for, 299
PhasedSubmissionApprove, 554
PhasedSubmissionApprove2, 555
PhasedSubmissionApproveEx, 556
PhasedSubmissionApproveExExtDim, 556
PhasedSubmissionGetHistory2, 557
PhasedSubmissionGetHistory2UsingPhaseIDExtDim, 558
PhasedSubmissionProcessManagementChangeStateForMultipleEntities2, 560
PhasedSubmissionPromote, 561
PhasedSubmissionPromote2, 562
PhasedSubmissionPromote2ExtDim, 562
PhasedSubmissionPromoteExtDim, 561
PhasedSubmissionPublish, 563
PhasedSubmissionPublish2, 563
PhasedSubmissionPublish2ExtDim, 564
PhasedSubmissionPublishExtDim, 565
PhasedSubmissionReject, 565
PhasedSubmissionReject2, 566
PhasedSubmissionReject2ExtDim, 566
PhasedSubmissionRejectExtDim, 565
PhasedSubmissionSignOff, 567
PhasedSubmissionSignOff2, 568
PhasedSubmissionSignOff2ExtDim, 568
PhasedSubmissionSignOffExtDim, 567
PhasedSubmissionStart, 569
PhasedSubmissionStart2, 570
PhasedSubmissionStart2ExtDim, 570
PhasedSubmissionStartEx, 571
PhasedSubmissionStartExtDim, 571
PhasedSubmissionStartExtDim, 569
PhasedSubmissionSubmit, 571
PhasedSubmissionSubmit2, 572
PhasedSubmissionSubmit2ExtDim, 572
PhasedSubmissionSubmitExtDim, 572
PlugAcct attribute, returning, 279
Point of View
control for user interface, 817
IHsvTreeInfo, methods for specifying members, 233
Point of View dialog box, 817
PostJournals, 439
preferences, user
decimal character
getting, 526
setting, 536
language
getting, 525
setting, 535
thousands separator
getting, 526
setting, 536
prerequisites, 49
process management
constants for actions, 878
constants for review levels, 878
submission phases and process units, 543
Process Management, type library for, 543
ProcessAllICTrans, 750
ProcessFlow, 192
ProcessICTransactions, 751
ProcessICTransactionsExtDim, 751
ProcessManagementChangeStateForMultipleEntities, 573
ProcessManagementChangeStateForMultipleEntities 2, 574
ProcessManagementChangeStateForMultipleEntities 2ExtDim, 574
ProcessManagementChangeStateForMultipleEntities Ex, 575
ProcessManagementChangeStateForMultipleEntities ExExtDim, 575
projects, provisioning, returning names of, 166
Promote, 576
Promote2, 578
provisioning projects, returning names of, 166
Publish, 579
Publish2, 580
PublishEx, 582
QuitAsynchronousTask, 731
records, invalid, deleting, 326
referencing type libraries, 57
RefreshAccessRightsCache, 499
Register Cluster dialog box, displaying, 184
RegisterApplicationCAS, 176
RegisterApplicationCASWithAccessCode, 177
RegisterCluster, 177
registering application servers
HsxClient object, 178
HsxClientUI object, 184
RegisterServer
HsxClient object, 178
HsxClientUI object, 184
Reject, 583
RejectJournals, 440
ReleaseHsxServer, 534
RemoveAll
HsvMDDataBuffer object, 631
HsvMDDataBufferLite object, 653
RemoveAllJournalGroups, 441
RemoveApplicationAdministrator, 489
RemoveApplicationAdministrator2, 489
RemoveJournalGroup, 441
RemoveQueryOrder, 672
RemoveUser, 490
RemoveUser2, 490
RemoveUserFromRole, 490
RemoveUserFromRole2, 490
RenameSecurityClass, 491
reports
journals, 456, 457
reports, 605
reports property, 192
Resource Manager, 834
resource strings
returning the default language for, 527
setting the default language, 537
setting the default language for a session, 536
review levels, constants for, 878
RLE file
getting contents of
  GetVBScriptCalcRules, 531
  GetVBScriptRules, 401
loading and validating
  LoadCalcRules, 792
  SetVBScriptCalcRules, 538
  SetVBScriptRules, 402
role IDs, 462
rules
  extracting, 790
  loading, 792

S
SaveColumnFilter, 752
SaveDocument, 606
SaveDocumentEx, 607
SaveICReasonCode, 752
SaveICTransaction, 752
SaveICTransactionExtDim, 753
SaveJournal, 441
SaveJournalExtDim, 441
SavePeriodsSettings, 763
SaveRowFilter, 754
SaveTemplate, 443
SaveTemplateExtDim, 443
SaveTextJournal, 445
SaveTextTemplate, 448
SaveTextTemplateExtDim, 448
SavetLastModifiedDateAndModifiedBy, 534
Scenarios, 232
ScriptableEnumDataSources, 190
ScriptableEnumRegisteredClusterNames, 178
ScriptableEnumRegisteredDSNs, 190
ScriptableEnumRegisteredServerNames, 178
ScriptableGetLogonInfo, 178
ScriptableLoadCalcRules, 795
ScriptableLoadCalcRules2, 795
ScriptableLoadMemberListRules, 795
ScriptableOpenApplication, 178
security
  connected user, 464
  extract options, 778
  extracting
    example, 777
  overview, 776
  load options, 777
  loading
    example, 775
    overview, 775
  role IDs, 462
  security identifiers (SIDs), 464
  task IDs, 462
  type library for, 461
  security class, Default, constant for, 908
  security identifier
    returning username for, 487
    user, getting for, 487
  security identifiers, 464
  Security property, 192
  SecurityAsPartner attribute, 293
  SecurityClass attribute, returning, 293
  Select Member List dialog box, 817
  SelectDimension, 824
  SelectGridViewDimensionIndex, 706
  SelectionChanged event, 831
  SelectMemberSourceAncestors, 706
  SelectMemberSourceAncestorsID, 706
  SelectMemberSourceBase, 707
  SelectMemberSourceBaseID, 707
  SelectMemberSourceChildren, 708
  SelectMemberSourceChildrenID, 708
  SelectMemberSourceDescendants, 709
  SelectMemberSourceHierarchy, 709
  SelectMemberSourceHierarchyExpanded, 710
  SelectMemberSourceHierarchyExpandedID, 710
  SelectMemberSourceHierarchyID, 711
  SelectMemberSourceMemberID, 712
  SelectMemberSourceMembers, 712
  SelectMemberSourceParents, 713
  SelectMemberSourceParentsID, 713
  SelectMemberSourceUserDefinedMemberList, 714
  SelectMemberSourceUserDefinedMemberList2, 714
  SelectPageSourceMemberIDForDim, 716
  SelectPageSourceMemberListRules, 716
sequence IDs, intercompanytransactions, 668
Server-Side Application components, instantiating, 193
servers, application
applications on, getting, 186
data link file, getting, 188
names of, getting, 167
registering
  HsxClient object, 178
  HsxClientUI object, 184
system file, getting, 188
unregistering
  HsxClient object, 180
  HsxClientUI object, 184
SetAccessRight, 660
SetAccessRights, 672
SetAllDims, 146
SetApplicationDirectory, 534
SetApplicationFolder, 178
SetCalcStatusLocked, 380
SetCalcStatusLockedForMultipleProcessUnits, 381
SetCalcStatusUnlocked, 381
SetCalcStatusUnlockedForMultipleProcessUnits, 382
SetCell, 382
SetCellDescriptions, 383
SetCellLineItems, 383, 384
SetCells, 384
SetCells2, 385
SetCellsLineItems, 385
SetCheckedItems, 824
SetCheckLineItemDetailsForCaseInsensitiveDuplicates
  HsvMDDataBuffer object, 632
  HsvMDDataBufferLite object, 654
SetColsForPageControl, 717
SetCOMDLLCalcRules, 534
SetCOMDLLRules, 402
SetCurrentActivity, 534
SetCurrentModule, 534
SetCurrentModuleEx, 535
SetCustomMemberAt, 148
SetDefaults, 148
SetData
  HsvMDDataBuffer object, 632
  HsvMDDataBufferLite object, 654
SetDataExtDim
  HsvMDDataBuffer object, 632
  HsvMDDataBufferLite object, 654
SetDefaultTablePrefix, 731
SetDimensions
  HsvMDDataBuffer object, 633
  HsvMDDataBufferLite object, 655
SetDimensionMember, 149
SetDimUnk, 825
SetEdgeRepeatedMetaDataSuppression, 717
SetEntityPartnerOption, 673
SetErrorStatus, 673
SetExtractFileEncoding, 535
SetFileForLoad
  HsvcDataLoad object, 806
  HsvData object, 386
SetFilterOptions, 673
SetFixedMember, 154
SetGridEdge, 717
SetGridNavigationType, 718
SetGridOptionDescriptionEnabled, 718
SetGridOptionDimensionExpansion, 719
SetGridOptionDisplayParentForEntityHierarchy, 719
SetGridPageControl, 720
SetGridSuppressionType, 720
SetGrowByAmount
  HsvMDDataBuffer object, 633
  HsvMDDataBufferLite object, 655
SetICTransactionData, 673
SetICTransactionDataExtDim, 673
SetLanguageUserParameters, 535
SetLeftHandDims, 149
SetLineItems
  HsvData object, 386
  HsvMDDataBuffer object, 633
  HsvMDDataBufferLite object, 655
SetLineItemsExtDim
  HsvMDDataBuffer object, 634
  HsvMDDataBufferLite object, 656
SetListInfo, 826
SetLogonInfo, 179
SetManySecurityClassRightsForUser, 491
SetManySecurityClassRightsForUser2, 491
SetMember, 826
SetMemberArrayForDim, 155
SetMemberAt, 155
SetMinCubesInCache, 657
SetNumberFormattingUserParameters, 536
SetOBPInfo, 827
SetPageMovement, 721
SetPagingOption, 673
SetPartnerAsEntityList, 673
SetPartnerQueryDimensionMemberIDs, 673
SetPersistedSettings, 731
SetPersistedSettingsExtDim, 731
SetPhaseSubmissionGridForGivenScenarioPeriod, 387
SetQueryDimensionMemberIDs, 674
SetQueryItem, 661
SetQueryItemExtDim, 661
SetReasonCodeToICTransactions, 754
SetReport, 607
SetResourceAndLanguage, 828
SetResourceLanguageForCurrentSession, 536
SetResourceLanguageUserParameters, 537
SetRightHandDims, 150
SetRolesForUser, 492
SetRowsForPageControl, 721
SetRulesMode, 492
SetSecurityClassLabel, 492
SetSecurityClassLabelWithAccessCode, 493
SetSecurityClassRightsForManyUsers, 493
SetSecurityClassRightsForManyUsers2, 493
SetSecurityClassRightsForUser, 494
SetSecurityClassRightsForUser2, 494
SetSession, 731
  HsvcDataLoad, HsvJournalLoadACV, HsvMetadataLoadACV, and
  HsvSecurityLoadACV objects, 773
  HsvRulesLoadACV object, 795
SetSessionAndResource
  HsvRulesLoadACV object, 796
SetSortedNature
  HsvMDDataBuffer object, 634
  HsvMDDataBufferLite object, 656
SetSourceNotSuppressible, 722
SetSourceOptionFillDefaultParentForEntity, 722
SetSourceSubType, 722
SetTemplate, 733
SetTextCell, 387
SetTextCellLineItems, 388
SetTextCells, 388, 389
SetTextCellsLineItems, 389
SetTextLineItems, 390
SetTotalTransactions, 674
SetTransactionData, 661
SetTransactionDataExtDim, 661
SetUserClassAccess, 494
SetUserParameter, 537
SetVBScriptCalcRules, 538
SetVBScriptMemberListRules, 539
SetVBScriptRules, 402
SetVBScriptRules2, 404
Shared Services
  returning provisioning project names, 166
ShowCheckBoxes, 828
SID, returning username for, 487
SIDs, 464
SignOff, 585
SignOff2, 586
single-select mode, 828
Sort
  HsvMDDataBuffer object, 634
  HsvMDDataBufferLite object, 656
SortByCell, 674
SortMembersBasedOnList, 273
SS_PUSH_OPTIONS, Enum, 727
star schema, 725
  creating or updating, 726
  deleting, 728
Start, 587
Start2, 589
StartEx, 590
StartLoad
  HsvcDataLoad object, 806
  HsvData object, 390
StartLoadTask
  HsvData object, 390
statuses, cells
  getting with GetStatus, 365
  overview, 319
statutory consolidations, returning transaction data for, 657
StopConsolidation, 406
StopRunningTask, 541
storage folder, local
  getting, 172
  setting, 178
subcube
  accessing with HsvDataCubes type library, 675
  calculation status, getting with GetCalcStatus, 352
  data synchronization and, 54
  items, 675
overview, 53
Submit, 591
Submit2, 592
SubmitJournals, 450
SupportsAccountAllocations, 310
SupportsEmailAlerting, 310
SupportsProcessFlow, 310
SupportsProcessManagement attribute, testing for, 310
SupportsTargetSetting, 311
SupportsTimePeriodAllocations, 311
SwitchSignForFlow attribute, returning, 290
SwitchTypeForFlow attribute, returning, 290
system changes, testing for with HasSystemChanged, 194
system data link file, getting path of, 188
system file, getting path of, 188
system folder, 522
SystemInfo, 192

T
TakeOwnership, 495
task audit
  adding task to audit history, 501
delete audit history, 503
  getting audit history, 507
task IDs, 462
TestSQLConnection, 732
thousands separator
  getting, 526
  setting, 536
token, external authentication, getting, 173
TraceLog, 541
transaction data, statutory consolidations, 657
transaction status
  constants for, 871
  overview, 319
Translate, 406
Translate2, 407
TranslateApplicationAttributeForDisplay, 229
TranslateApplicationAttributeForDisplayExtDim, 229
TranslateAttributeValueForDisplay, 274
tripplets, Value dimension, 53
type libraries
  HsvCalculate, 391
  HsvcDataLoad, 796

HsvData, 319
HsvDataCubes, 675
HsvDQI, 685
HsvICM, 737
HsvJournalLoadACV, 806
HsvJournals, 409
HsvMDArrays, 611
HsvMetadata, 197
HsvMetadataLoadACV, 780
HsvPOVSelection, 817
HsvProcessFlow, 543
HsvReports, 595
HsvResourceManager, 833
HsvRulesLoadACV, 790
HsvSecurityAccess, 461
HsvSecurityLoadACV, 774
HsvSession, 191
HsvStarSchemaACM, 725
HsvSystemInfo, 501
HsxClient, 157
HsxClientUI, 181
HsxServer, 185
referencing in projects, 57
type mappings, 50

U
UnInitialize, 674
UnlockICEntity, 763
UnMatchICTransactions, 755
UnpostJournals, 451
UnregisterAllClusters, 179
UnregisterCluster, 179
UnregisterServer, 180
UnsubmitJournals, 452
update mode, 320
UpdateDataUsingMDDataBuffer, 390
UpdatePeriodSettings, 763
UpdatePeriodSettings2, 764
UpdateRunningTaskLogFilePathName, 542
UpdateRunningTaskPOV, 542
UpdateRunningTaskProgress, 542
UpdateRunningTaskProgressDetails, 542
UpdateRunningTaskStatus, 542
UpdateUserAppPreferences, 180
user preferences
  decimal character
getting, 526
setting, 536
examples, number formatting, getting and setting, 139
language
getting, 525
setting, 535
thousands separator
getting, 526
setting, 536
user, connected, 464
UserDefined1 attribute, returning
accounts, 282
entities, 294
scenarios, 307
UserDefined2 attribute, returning
accounts, 283
entities, 294
scenarios, 308
UserDefined3 attribute, returning
accounts, 283
entities, 295
scenarios, 308
username
getting, 173
setting, 179
single sign on
getting, 173
users, enumerating, 477
UsesLineItems
HsvAccounts object, 287
HsvScenarios object, 311

W
WarnUsersForShutDown
HsvSystemInfo object, 542
HsxClient object, 180

X
XBRLTags attribute, returning, 283

Y
Years, 233

Z
ZeroViewForAdj attribute, getting, 305
ZeroViewForNonadj attribute, getting, 306