Oracle® Hyperion Smart View for Office

Developer's Guide

Release 11.1.2.5
Chapter 3. General Functions .......................................................... 45

About General Functions .............................................................. 45
HypShowPanel ........................................................................ 45
HypGetVersion ....................................................................... 46
HypGetLastError ...................................................................... 48
HypShowPov ........................................................................... 48
HypSetMenu ............................................................................ 49
HypCopyMetaData .................................................................. 50
HypDeleteMetaData ................................................................. 50
HypIsDataModified ................................................................. 51
HypIsSmartViewContentViewPresent ....................................... 52
HypIsFreeForm ....................................................................... 53
HypUndo ................................................................................ 54
HypRedo ................................................................................ 54
HypPreserveFormatting ............................................................ 55
HypRemovePreservedFormats .................................................... 56
HypSetAliasTable ................................................................. 57
HypGetSubstitutionVariable ...................................................... 57
HypSetSubstitutionVariable ..................................................... 59
HypGetDatabaseNote ............................................................... 60
Chapter 10. Calculation, Consolidation, and Translation Functions

Chapter 11. Member Query Functions

Chapter 12. Options Functions
<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>HypSetSheetOption</td>
<td>162</td>
</tr>
<tr>
<td>HypGetOption</td>
<td>163</td>
</tr>
<tr>
<td>HypSetOption</td>
<td>168</td>
</tr>
<tr>
<td>HypDeleteAllMRUItems</td>
<td>168</td>
</tr>
</tbody>
</table>

**Chapter 13. Dynamic Link Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Dynamic Link Views</td>
<td>171</td>
</tr>
<tr>
<td>Setting Up Dynamic Link Views</td>
<td>172</td>
</tr>
<tr>
<td>Automating Macro Execution</td>
<td>172</td>
</tr>
<tr>
<td>HypUseLinkMacro</td>
<td>172</td>
</tr>
<tr>
<td>HypSetLinkMacro</td>
<td>173</td>
</tr>
<tr>
<td>HypGetLinkMacro</td>
<td>174</td>
</tr>
<tr>
<td>HypGetSourceGrid</td>
<td>174</td>
</tr>
<tr>
<td>HypDisplayToLinkView</td>
<td>175</td>
</tr>
<tr>
<td>HypGetConnectionInfo</td>
<td>176</td>
</tr>
<tr>
<td>HypSetConnectionInfo</td>
<td>178</td>
</tr>
<tr>
<td>HypGetRowCount</td>
<td>179</td>
</tr>
<tr>
<td>HypGetColCount</td>
<td>180</td>
</tr>
<tr>
<td>HypGetPOVCount</td>
<td>180</td>
</tr>
<tr>
<td>HypGetRowItems</td>
<td>181</td>
</tr>
<tr>
<td>HypSetRowItems</td>
<td>182</td>
</tr>
<tr>
<td>HypGetColItems</td>
<td>183</td>
</tr>
<tr>
<td>HypSetColItems</td>
<td>184</td>
</tr>
<tr>
<td>HypGetPOVItems</td>
<td>185</td>
</tr>
<tr>
<td>HypSetPOVItems</td>
<td>186</td>
</tr>
</tbody>
</table>

**Chapter 14. MDX Query Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About MDX</td>
<td>189</td>
</tr>
<tr>
<td>HypExecuteMDXEx</td>
<td>189</td>
</tr>
</tbody>
</table>

**Chapter 15. Oracle BI EE Functions**

<table>
<thead>
<tr>
<th>Function</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>About Oracle BI EE Functions</td>
<td>193</td>
</tr>
<tr>
<td>Preparing to Work with Oracle BI EE Functions</td>
<td>193</td>
</tr>
<tr>
<td>Instantiating an Oracle Smart View BI Extension Object</td>
<td>194</td>
</tr>
<tr>
<td>Oracle Smart View BI Extension Functions</td>
<td>194</td>
</tr>
<tr>
<td>InsertView</td>
<td>194</td>
</tr>
<tr>
<td>EditPrompts</td>
<td>198</td>
</tr>
<tr>
<td>EditPagePrompts</td>
<td>199</td>
</tr>
<tr>
<td>GetPagePrompts</td>
<td>200</td>
</tr>
<tr>
<td>DeleteView</td>
<td>201</td>
</tr>
</tbody>
</table>
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.
Assumed Knowledge

You can customize and automate common tasks using Microsoft Visual Basic for Applications (VBA) functions in Oracle Hyperion Smart View for Office using Microsoft Excel's Visual Basic Editor.

To use the information in this chapter to develop VBA applications for Smart View, you must have working knowledge of the following:

- Smart View and how it is used in your organization
- Visual Basic or VBA programming language
- Excel Visual Basic Editor as an environment for VBA development

VBA Functions Location

All Smart View VBA functions are contained in the file smartview.bas, located by default in EPM_ORACLE_HOME/smartview/bin. To access these functions, import smartview.bas into a Visual Basic Editor module and use this module as a source of VBA functions for your program.

Using VBA Function Code Samples

This guide provides examples for each VBA function. You can copy these code samples into a Visual Basic Editor Module; however Oracle recommends that you use smartview.bas
imported into a module as the source of the function declarations. This is particularly important for declarations that contain arrays. See “VBA Functions Location” on page 13.

If you do copy and paste code samples, always use the HTML version of this guide. Copying from a PDF file may cause characters in the code to be lost.

**VBA Functions in 64-Bit Versions**

If you are using the 64-bit version of Microsoft Office, VBA function declarations are slightly different from those in the 32-bit version. In 64-bit versions, the declarations include `PtrSafe` after the `Declare` keyword. For example:

- **32-bit version:**
  ```vba
  Public Declare Function HypMenuVAbout Lib "HsAddin" () As Long
  ```

- **64-bit version:**
  ```vba
  Public Declare PtrSafe Function HypMenuVAbout Lib "HsAddin" () As Long.
  ```

The `smartview.bas` file provided with your Smart View installation automatically contains the appropriate declaration statements.

**Note:** The code samples in this guide contain declarations for the 32-bit version of Office; if you have the 64-bit version, you must ensure that `PtrSafe` is included in the declarations.

**VBA Parameters**

Most VBA functions require you to supply values for one or more parameters. Table 1 lists the parameter types and the valid values for each type:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Text</td>
<td>A word or phrase or name in quotation marks. For example:</td>
</tr>
<tr>
<td></td>
<td>- “Smart View”</td>
</tr>
<tr>
<td></td>
<td>- “[Book2.xls]Sheet1”</td>
</tr>
<tr>
<td>Boolean</td>
<td>- True</td>
</tr>
<tr>
<td></td>
<td>- False</td>
</tr>
<tr>
<td>Range object</td>
<td>A cell, row or column, one or more selections of cells, or a three-dimensional range address, surrounded by quotation marks. For example:</td>
</tr>
<tr>
<td></td>
<td>- RANGE(“A1”)</td>
</tr>
<tr>
<td></td>
<td>- RANGE(“A1:B2”)</td>
</tr>
<tr>
<td></td>
<td>- RANGE(“G:G,I:I,K:K”)</td>
</tr>
<tr>
<td></td>
<td>- RANGE(“A1:B5,C1:C10,D5:L8”)</td>
</tr>
<tr>
<td></td>
<td>- RANGE(“Sheet1!C3:R20,Sheet2!C3:R20”)</td>
</tr>
<tr>
<td>Parameter</td>
<td>Value</td>
</tr>
<tr>
<td>--------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>Number</td>
<td>A number without quotation marks and without commas. For example:</td>
</tr>
<tr>
<td></td>
<td>• 1</td>
</tr>
<tr>
<td></td>
<td>• 2.5</td>
</tr>
<tr>
<td></td>
<td>• 50000</td>
</tr>
<tr>
<td>List of strings</td>
<td>A list of text values separated by commas. For example: “Qtr1”, “Actual”, “Oregon”</td>
</tr>
<tr>
<td>Constant</td>
<td>A predefined constant from <code>smartview.bas</code></td>
</tr>
<tr>
<td>Default value</td>
<td>• Null</td>
</tr>
<tr>
<td></td>
<td>• Empty</td>
</tr>
</tbody>
</table>

**Note:** Many parameters have default values or behavior that the function uses if you specify Null or Empty. If you do not specify a value for such parameters, use Null or Empty. See the description of each function for default values of such parameters.

# VBA Return Values

Smart View VBA functions may return any of the following values to indicate success or failure of the function. A return value of zero (0) indicates that the function ran successfully. Negative numbers represent client issues; positive numbers represent server issues. **Table 2** lists the return values.

**Table 2  Return Values and Their Descriptions**

<table>
<thead>
<tr>
<th>Return Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4</td>
<td>SS_ERR_ERROR</td>
</tr>
<tr>
<td></td>
<td>An error specific to the data provider or a generic error that cannot be mapped to a value.</td>
</tr>
<tr>
<td>2</td>
<td>SS_NO_GRID_ON_SHEET_BUT_FUNCTIONS_SUBMITTED</td>
</tr>
<tr>
<td></td>
<td>The value returned when a function sheet without a grid is submitted.</td>
</tr>
<tr>
<td>1</td>
<td>SS_SHEET_NOT_CONNECTED_BUT_FUNCTIONS_SUBMITTED</td>
</tr>
<tr>
<td></td>
<td>The value returned when a function sheet that is not connected is submitted.</td>
</tr>
<tr>
<td>0</td>
<td>SS_OK</td>
</tr>
<tr>
<td></td>
<td>The function ran successfully.</td>
</tr>
<tr>
<td>-1</td>
<td>SS_INIT_ERR</td>
</tr>
<tr>
<td></td>
<td>Initialization error.</td>
</tr>
<tr>
<td>-2</td>
<td>SS_TERM_ERR</td>
</tr>
<tr>
<td></td>
<td>Termination error.</td>
</tr>
<tr>
<td>-3</td>
<td>SS_NOT_INIT</td>
</tr>
<tr>
<td></td>
<td>Initialization error.</td>
</tr>
<tr>
<td>-4</td>
<td>SS_NOT_CONNECTED</td>
</tr>
<tr>
<td></td>
<td>The spreadsheet is not yet connected to the server.</td>
</tr>
<tr>
<td>-5</td>
<td>SS_NOT_LOCKED</td>
</tr>
<tr>
<td></td>
<td>The spreadsheet is not locked.</td>
</tr>
<tr>
<td>-6</td>
<td>SS_INVALID_SSTABLE</td>
</tr>
<tr>
<td></td>
<td>The spreadsheet has become unstable.</td>
</tr>
<tr>
<td>-7</td>
<td>SS_INVALID_SSDATA</td>
</tr>
<tr>
<td></td>
<td>The spreadsheet contains invalid data.</td>
</tr>
<tr>
<td>Return Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>-8</td>
<td>SS_NOUNDO_INFO No Undo information exists.</td>
</tr>
<tr>
<td>-9</td>
<td>SS_CANCELED Operation has been canceled.</td>
</tr>
<tr>
<td>-10</td>
<td>SS_GLOBALOPTS Not used.</td>
</tr>
<tr>
<td>-11</td>
<td>SS_SHEETOPTS Not used.</td>
</tr>
<tr>
<td>-12</td>
<td>SS_NOTENABLED Undo is not enabled.</td>
</tr>
<tr>
<td>-13</td>
<td>SS_NO_MEMORY Not enough memory resources are available.</td>
</tr>
<tr>
<td>-14</td>
<td>SS_DIALOG_ERROR Appropriate dialog box could not be displayed.</td>
</tr>
<tr>
<td>-15</td>
<td>SS_INVALID_PARAM Function contains an invalid parameter.</td>
</tr>
<tr>
<td>-16</td>
<td>SS_CALCULATING Calculation is in progress.</td>
</tr>
<tr>
<td>-17</td>
<td>SS_SQL_IN_PROGRESS Obsolete setting.</td>
</tr>
<tr>
<td>-18</td>
<td>SS_FORMULAPRESERVE Operation is not allowed because the spreadsheet is in formula preservation mode.</td>
</tr>
<tr>
<td>-19</td>
<td>SS_INTERNALSSERROR Operation cannot take place on the specified sheet.</td>
</tr>
<tr>
<td>-20</td>
<td>SS_INVALID_SHEET Current sheet cannot be determined.</td>
</tr>
<tr>
<td>-21</td>
<td>SS_NOACTIVESHEET Spreadsheet name was not specified and no active sheet is selected.</td>
</tr>
<tr>
<td>-22</td>
<td>SS_NOTCALCULATING Calculation cannot be canceled because no calculation is running.</td>
</tr>
<tr>
<td>-23</td>
<td>SS_INVALIDSELECTION Selection parameter is invalid.</td>
</tr>
<tr>
<td>-24</td>
<td>SS_INVALIDTOKEN Not used.</td>
</tr>
<tr>
<td>-25</td>
<td>SS_CASCADENOTALLOWED Cascade list file cannot be created, or you are attempting to cascade while the spreadsheet is embedded in another document.</td>
</tr>
<tr>
<td>-26</td>
<td>SS_NOMACROS Spreadsheet macros cannot be run due to a licensing agreement.</td>
</tr>
<tr>
<td>-27</td>
<td>SS_NOREADONLYMACROS Spreadsheet macros which update the database cannot be run due to a licensing constraint.</td>
</tr>
<tr>
<td>-28</td>
<td>SS_READONLYSS You have a read-only license and cannot update the database.</td>
</tr>
<tr>
<td>-29</td>
<td>SS_NOSQLACCESS Obsolete setting.</td>
</tr>
<tr>
<td>-30</td>
<td>SS_MENUALREADYREMOVED The menu is removed already.</td>
</tr>
<tr>
<td>-31</td>
<td>SS_MENUALREADYADDED The menu is added already.</td>
</tr>
<tr>
<td>Return Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>-32</td>
<td>SS_NOSpreadSheetACCESS</td>
</tr>
<tr>
<td>-33</td>
<td>SS_NOHANDLES</td>
</tr>
<tr>
<td>-34</td>
<td>SS_NOPREVCONNECTION</td>
</tr>
<tr>
<td>-35</td>
<td>SS_LROERROR</td>
</tr>
<tr>
<td>-36</td>
<td>SS_LROWINAPPACCESSERR</td>
</tr>
<tr>
<td>-37</td>
<td>SS_DATANAVINITERR</td>
</tr>
<tr>
<td>-38</td>
<td>SS_PARAMSETNOTALLOWED</td>
</tr>
<tr>
<td>-39</td>
<td>SS_SHEET_PROTECTED</td>
</tr>
<tr>
<td>-40</td>
<td>SS_CALCSCRIPT_NOTFOUND</td>
</tr>
<tr>
<td>-41</td>
<td>SS_NOSUPPORT_PROVIDER</td>
</tr>
<tr>
<td>-42</td>
<td>SS_INVALID_ALIAS</td>
</tr>
<tr>
<td>-43</td>
<td>SS_CONN_NOT_FOUND</td>
</tr>
<tr>
<td>-44</td>
<td>SS_APS_CONN_NOT_FOUND</td>
</tr>
<tr>
<td>-45</td>
<td>SS_APS_NOT_CONNECTED</td>
</tr>
<tr>
<td>-46</td>
<td>SS_APS_CANT_CONNECT</td>
</tr>
<tr>
<td>-47</td>
<td>SS_CONN_ALREADY_EXISTS</td>
</tr>
<tr>
<td>-48</td>
<td>SS_APS_URL_NOT_SAVED</td>
</tr>
<tr>
<td>-49</td>
<td>SS_MIGRATION_OF_CONN_NOT_ALLOWED</td>
</tr>
<tr>
<td>-50</td>
<td>SS_CONN_MGR_NOT_INITIALIZED</td>
</tr>
<tr>
<td>-51</td>
<td>SS_FAILED_TO_GET_APS_OVERRIDE_PROPERTY</td>
</tr>
<tr>
<td>-52</td>
<td>SS_FAILED_TO_SET_APS_OVERRIDE_PROPERTY</td>
</tr>
<tr>
<td>-53</td>
<td>SS_FAILED_TO_GET_APS_URL</td>
</tr>
<tr>
<td>-54</td>
<td>SS_APS_DISCONNECT_FAILED</td>
</tr>
<tr>
<td>-55</td>
<td>SS_OPERATION_FAILED</td>
</tr>
<tr>
<td>-56</td>
<td>SS_CANNOT_ASSOCIATE_SHEET_WITH_CONNECTION</td>
</tr>
<tr>
<td>-57</td>
<td>SS_REFRESH_SHEET_NEEDED</td>
</tr>
<tr>
<td>-58</td>
<td>SS_NO_GRID_OBJECT_ON_SHEET</td>
</tr>
<tr>
<td>-59</td>
<td>SS_NO_CONNECTION_ASSOCIATED</td>
</tr>
</tbody>
</table>
Using Spreadsheet Toolkit VBA Applications in Smart View

VBA applications created in Oracle Hyperion Essbase Spreadsheet Toolkit can be converted to Smart View by making the following modifications:

- Replace the `EssV` prefix of Spreadsheet Toolkit functions with `Hyp`; for example, change `EssVRemoveOnly` to `HypRemoveOnly`.
- Replace the `EssMenuV` prefix of Oracle Hyperion Essbase Spreadsheet Toolkit menu functions with `HypMenuV`; for example, change `EssMenuVZoomIn` to `HypMenuVZoomIn`.
- Replace the declarations in `essxlvba.txt` with the declarations in `smartview.bas`.

VBA Function Types

- **Menu** functions are identical to the equivalent commands on the Smart View menu and ribbon. See Chapter 2, “Menu Functions.”
- General functions perform actions, set options, or retrieve information typically performed from the Smart View ribbon or Options dialog box. See Chapter 3, “General Functions.”
- Connection functions perform actions related to connections to data providers. See Chapter 4, “Connection Functions.”
- Ad hoc functions perform ad hoc operations such as zooming, retrieving and submitting data, and pivoting. See Chapter 5, “Ad Hoc Functions.”
- The Form function opens a data form. See Chapter 6, “Form Functions.”
- Cell functions perform operations and retrieve information for data cells and their contents. See Chapter 7, “Cell Functions.”
- POV functions specify or retrieve settings for the POV. See Chapter 8, “POV Functions.”
- Calculation script and business rule functions retrieve lists of or execute calculation scripts and business rules. See Chapter 9, “Calculation Script and Business Rule Functions.”
- Calculation, consolidation, and translation functions executes these operations on data for Oracle Hyperion Financial Management and Oracle Hyperion Enterprise® applications. See Chapter 10, “Calculation, Consolidation, and Translation Functions.”
- Member query functions retrieve generation, level, attribute, and other information about members. See Chapter 11, “Member Query Functions.”
- Options functions set and retrieve information for global and/or sheet options, and enable deletion of MRU items. See Chapter 12, “Options Functions.”
- Dynamic link functions set or retrieve data point details that are displayed in separate windows via dynamic links. See Chapter 13, “Dynamic Link Functions.”
- The MDX query function executes an MDX query whose results are not displayed in a worksheet. See Chapter 14, “MDX Query Functions.”

Note: For an alphabetical list of VBA functions, see the index.
Menu Functions

In This Chapter

- About Menu Functions ................................................................. 22
- HypMenuVAbout .............................................................................. 22
- HypMenuVAdjust ............................................................................. 23
- HypMenuVBusinessRules ............................................................... 23
- HypMenuVCalculation ................................................................. 23
- HypMenuVCascadeNewWorkbook .................................................. 24
- HypMenuVCascadeSameWorkbook .................................................. 24
- HypMenuVCeilText ......................................................................... 25
- HypMenuVCollapse ......................................................................... 25
- HypMenuVConnect .......................................................................... 26
- HypMenuVCopyDataPoints ............................................................ 26
- HypMenuVExpand ............................................................................. 27
- HypMenuVFunctionBuilder ............................................................ 27
- HypMenuVInstruction ...................................................................... 28
- HypMenuVKeepOnly .......................................................................... 28
- HypMenuVLR0 .................................................................................. 29
- HypMenuVMemberInformation ....................................................... 29
- HypMenuVMemberSelection .......................................................... 29
- HypMenuVMigrate ............................................................................ 30
- HypMenuVOptions ........................................................................... 31
- HypMenuVPasteDataPoints ........................................................... 31
- HypMenuVPivot ............................................................................... 32
- HypMenuVPOVManger ................................................................. 32
- HypMenuVQueryDesigner ............................................................. 33
- HypMenuVRedo ............................................................................... 33
- HypMenuVRefresh ........................................................................... 34
- HypMenuVRefreshAll ....................................................................... 34
- HypMenuVRefreshOfflineDefinition ........................................... 34
- HypMenuVRemoveOnly ................................................................. 35
- HypMenuVRulesOnForm ............................................................... 35
- HypMenuVRunReport ....................................................................... 36
- HypMenuVSelectForm ..................................................................... 36
- HypMenuVShowHelpHtml ............................................................... 37
About Menu Functions

VBA menu functions are identical to the equivalent commands on the Smart View menu and ribbon. The requirements for the menu functions are the same as those for the menu commands. For example, if you must be logged in to an Oracle Essbase server to use a menu command, then you must also be logged in to an Essbase server to use the equivalent VBA command.

**HypMenuVAbout**

Data provider types: Essbase, Oracle Hyperion Planning, Financial Management, Hyperion Enterprise

Description

HypMenuVAbout() opens the Help About screen.

Syntax

HypMenuVAbout()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

```
Public Declare Function HypMenuVAbout Lib "HsAddin" () As Long
Sub MAbout()
    X=HypMenuVAbout()
End Sub
```
**HypMenuVAdjust**

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypMenuVAdjust() opens the Adjust Data dialog box.

Syntax

HypMenuVAdjust()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

```vba
Public Declare Function HypMenuVAdjust Lib "HsAddin" () As Long
Sub MAdjust()
    X=HypMenuVAdjust()
End Sub
```

**HypMenuVBusinessRules**

Data provider types: Planning

Description

HypMenuVBusinessRules() opens the Business Rules dialog box.

Syntax

HypMenuVBusinessRules()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

```vba
Public Declare Function HypMenuVBusinessRules Lib "HsAddin" () As Long
Sub MBusinessRules()
    X=HypMenuVBusinessRules()
End Sub
```

**HypMenuVCalculation**

Data provider types: Essbase, Financial Management (ad hoc only), Hyperion Enterprise
Description
HypMenuVCalculation() opens the **Calculation Scripts** dialog box.

Syntax
HypMenuVCalculation()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypMenuVCalculation Lib "HsAddin"() As Long
Sub MCalc()
    X=HypMenuVCalculation()
End Sub

---

**HypMenuVCascadeNewWorkbook**

Data provider types: Essbase, Planning, Hyperion Enterprise

Description
HypMenuVCascadeNewWorkbook() opens the **Member Selection** dialog box to begin the cascading process to worksheets of a newly-opened Excel workbook.

Syntax
HypMenuVCascadeNewWorkbook()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypMenuVCascadeNewWorkbook Lib "HsAddin" () As Long
Sub MCascadeNewWorkbook()
    X=HypMenuVCascadeNewWorkbook()
End Sub

---

**HypMenuVCascadeSameWorkbook**

Data provider types: Essbase, Planning, Hyperion Enterprise

Description
HypMenuVCascadeSameWorkbook() opens the **Member Selection** dialog box to begin the cascading process to the same workbook.
Syntax
HypMenuVCascadeSameWorkbook()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypMenuVCascadeSameWorkbook Lib "HsAddin" () As Long
Sub MCascadeSameWorkbook()
    X=HypMenuVCascadeSameWorkbook()
End Sub

HypMenuVCellText

Data provider types: Planning, Financial Management, Hyperion Enterprise (forms only)

Description
HypMenuVCellText() opens the Cell Comments dialog box.

Syntax
HypMenuVCellText()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypMenuVCellText Lib "HsAddin" () As Long
Sub MCellText()
    X=HypMenuVCellText()
End Sub

HypMenuVCollapse

Data provider types: Planning (forms only)

Description
HypMenuVCollapse() collapses all levels of detail for the selected cells.

Syntax
HypMenuVCollapse()
Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypMenuVCollapse Lib "HsAddin" () As Long
Sub MHypMenuVCollapse()
   X=HypMenuVCollapse()
End Sub

HypMenuVConnect
Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypMenuVConnect() opens the Smart View Panel and enables users to connect to a data provider.

Syntax
HypMenuVConnect()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypMenuVConnect Lib "HsAddin"() As Long
Sub MConn()
   X=HypMenuVConnect()
End Sub

HypMenuVCopyDataPoints
Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypMenuVCopyDataPoints() copies data points from Excel for pasting into Word or PowerPoint. See also “HypMenuVPasteDataPoints” on page 31.

Syntax
HypMenuVCopyDataPoints()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.
HypMenuVCopyDataPoints

Sub MCopyDataPoints
    X=HypMenuVCopyDataPoints()
End Sub

HypMenuVExpand

Data provider types: Planning (forms only)

Description
HypMenuVExpand() displays all levels of detail for the selected cells.

Syntax
HypMenuVExpand()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypMenuVExpand Lib "HsAddin" () As Long
Sub MExpand()
    X=HypMenuVExpand()
End Sub

HypMenuVFunctionBuilder

Data provider types: Essbase, Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypMenuVFunctionBuilder() opens the Function Builder.

Syntax
HypMenuVFunctionBuilder()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypMenuVFunctionBuilder Lib "HsAddin" () As Long
Sub MFunctionBuilder()
    X=HypMenuVFunctionBuilder()
End Sub
HypMenuVInstruction

Data provider types: Planning (forms only), Financial Management (forms only), Hyperion Enterprise (forms only)

Description

HypMenuVInstruction() opens the Instructions dialog box.

Syntax

Returns 0 if successful; otherwise, returns the appropriate error code.

Return Value

HypMenuVInstruction()

Example

Public Declare Function HypMenuVInstruction Lib "HsAddin" () As Long
Sub MInstruction()
    X=HypMenuVInstruction()
End Sub

HypMenuVKeepOnly

Data provider types: Essbase (ad hoc only), Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypMenuVKeepOnly() retains only the selected member (the active cell) or member range in the sheet.

Syntax

HypMenuVKeepOnly()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypMenuVKeepOnly Lib "HsAddin"() As Long
Sub MKeepOnly()
    X=HypMenuVKeepOnly()
End Sub
HypMenuVLRO

Data provider types: Essbase

Description
HypMenuVLRO() opens the **Linked Objects** dialog box.

Syntax
HypMenuVLRO()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
```vba
Public Declare Function HypMenuVLRO Lib "HsAddin" () As Long
Sub MLRO()
    X=HypMenuVLRO()
End Sub
```

HypMenuVMemberInformation

Data provider types: Essbase

Description
HypMenuVMemberInformation() opens the **Member Information** dialog box.

Syntax
HypMenuVMemberInformation()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
```vba
Public Declare Function HypMenuVMemberInformation Lib "HsAddin" () As Long
Sub MMemberInformation()
    X=HypMenuVMemberInformation()
End Sub
```

HypMenuVMemberSelection

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise
Description

HypMenuVMemberSelection() opens the **Member Selection** dialog box.

Syntax

HypMenuVMemberSelection()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

```vba
Public Declare Function HypMenuVMemberSelection Lib "HsAddin" () As Long
Sub MMemberSelection()
    X=HypMenuVMemberSelection()
End Sub
```

---

**HypMenuVMigrate**

**Data provider types:** Financial Management, Hyperion Enterprise

Description

HypMenuVMigrate() launches the Financial Management and Hyperion Enterprise migration utility for **Active Workbook Migration** and **Batch Migration**.

Syntax

HypMenuVMigrate (vtOption, vtOutput)

ByVal vtOption As Variant

ByRef vtOutput As Variant

Parameters

vtOption: Number that indicates the migration utility to be launched:

1—Financial Management Active Workbook Migration

2—Financial Management Batch Migration

3—Hyperion Enterprise Active Workbook Migration

4—Hyperion Enterprise Batch Migration

vtOutput: Output parameter. Returns the migration result.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.
Public Declare Function HypMenuVMigrate Lib "HsAddin" (ByVal vtOption As Variant, ByRef vtOutput As Variant) As Long

Sub MigrateHFM()
    sts = HypMenuVMigrate(1, out)
    MsgBox (out)
    MsgBox (sts)
End Sub

HypMenuVOptions

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypMenuVOptions() opens the Options dialog box.

Syntax

HypMenuVOptions()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypMenuVOptions Lib "HsAddin"() As Long
Sub MOptions()
    X=HypMenuVOptions()
End Sub

HypMenuVPasteDataPoints

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypMenuVPasteDataPoints() pastes data points that were copied from Excel into Word or PowerPoint. See also “HypMenuVCopyDataPoints” on page 26.

Syntax

HypMenuVPasteDataPoints()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.
Example

Public Declare Function HypMenuVPasteDataPoints Lib "HsAddin" () As Long
Sub MVPasteDataPoints()
    X=HypMenuVPasteDataPoints()
End Sub

### HypMenuVPivot

**Data provider types:** Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypMenuVPivot() pivots the members associated with the selected cell.

**Syntax**

HypMenuVPivot()

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypMenuVPivot Lib "HsAddin"() As Long
Sub MPivot()
    X=HypMenuVPivot()
End Sub

### HypMenuVPOVManager

**Data provider types:** Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypMenuVPOVManager() opens the POV Manager.

**Syntax**

HypMenuVPOVManager()

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Public Declare Function HypMenuVPOVManager Lib "HsAddin" () As Long
Sub MPOVManager()
HypMenuVQueryDesigner

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypMenuVQueryDesigner() opens the Query Designer.

Syntax
HypMenuVQueryDesigner()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypMenuVQueryDesigner Lib "HsAddin" () As Long
Sub MDesigner()
    X=HypMenuVQueryDesigner()
End Sub

HypMenuVRedo

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypMenuVRedo() reverses an Undo operation.

Syntax
HypMenuVRedo()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypMenuVRedo Lib "HsAddin" () As Long
Sub MRedo()
    X=HypMenuVRedo()
End Sub
**HypMenuVRefresh**

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypMenuVRefresh() refreshes the active worksheet.

Syntax

HypMenuVRefresh()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

```
Declare Function HypMenuVRefresh Lib "HsAddin"() As Long
Sub MRetrieve()
    X=HypMenuVRefresh()
End Sub
```

**HypMenuVRefreshAll**

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypMenuVRefreshAll() refreshes data in all connected worksheets in an Excel workbook.

Syntax

HypMenuVRefreshAll()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

```
Public Declare Function HypMenuVRefreshAll Lib "HsAddin" () As Long
Sub MRefreshAll()
    X=HypMenuVRefreshAll()
End Sub
```

**HypMenuVRefreshOfflineDefinition**

Data provider types: Planning
Description
HypMenuVRefreshOfflineDefinition() refreshes the Offline data form definition and data.

Syntax
HypMenuVRefreshOfflineDefinition()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypMenuVRefreshOfflineDefinition Lib "HsAddin" () As Long
Sub MRefreshOfflineDefinition()
    X=HypMenuVRefreshOfflineDefinition()
End Sub

HypMenuVRemoveOnly

Data provider types: Essbase Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypMenuVRemoveOnly() removes only the selected member or member range in the sheet.

Syntax
HypMenuVRemoveOnly()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypMenuVRemoveOnly Lib "HsAddin"() As Long
Sub MRemoveOnly()
    X=HypMenuVRemoveOnly()
End Sub

HypMenuVRulesOnForm

Data provider types: Planning (forms only)

Description
HypMenuVRulesOnForm() opens the Rules on Form dialog box.
Syntax

HypMenuVRulesOnForm()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypMenuVRulesOnForm Lib "HsAddin" () As Long
Sub MRulesOnForm()
    X=HypMenuVRulesOnForm()
End Sub

HypMenuVRunReport

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypMenuVRunReport() runs a report designed in the Query Designer.

Syntax

HypMenuVRunReport()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypMenuVRunReport Lib "HsAddin" () As Long
Sub MRunReport()
    X=HypMenuVRunReport()
End Sub

HypMenuVSelectForm

Data provider types: Planning, Financial Management, Hyperion Enterprise

Description

HypMenuVSelectForm() opens the Select Form dialog box.

Syntax

HypMenuVSelectForm()
Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypMenuVSelectForm Lib "HsAddin" () As Long
Sub MSelectForm()
    X=HypMenuVSelectForm()
End Sub

HypMenuVShowHelpHtml
Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypMenuVShowHelpHtml() launches the online help.

Syntax
HypMenuVShowHelpHtml(vtHelpPage)
ByVal vtHelpPage As Variant

Parameter
vtHelpPage: The name of the HTML file that launches the help.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypMenuVShowHelpHtml Lib "HsAddin" (ByVal vtHelpPage As Variant) As Long
Sub MShowHelpHtml()
    X=HypMenuVShowHelpHtml("launch.htm")
End Sub

HypMenuVSubmitData
Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypMenuVSubmitData() submits data that has been modified or marked as dirty with HypSetCellsDirty to the active database on the server.
**Syntax**

HypMenuVSubmitData()

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypMenuVSubmitData Lib "HsAddin" () As Long
Sub MSubmit()
    X=HypMenuVSubmitData()
End Sub

---

**HypMenuVSupportingDetails**

**Data provider types:** Planning

**Description**

HypMenuVSupportingDetails() opens the Supporting Details dialog box.

**Syntax**

HypMenuVSupportingDetails()

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Public Declare Function HypMenuVSupportingDetails Lib "HsAddin" () As Long
Sub MSupportingDetails()
    X=HypMenuVSupportingDetails()
End Sub

---

**HypMenuVSyncBack**

**Data provider types:** Planning

**Description**

HypMenuVSyncBack() synchronizes data from an offline Planning data form to the server.

**Syntax**

HypMenuVSyncBack()
Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypMenuVSyncBack Lib "HsAddin" () As Long
Sub MSyncBack()
    X=HypMenuVSyncBack()
End Sub

HypMenuVTakeOffline

Data provider types: Planning

Description
HypMenuVTakeOffline() launches the Take Offline wizard.

Syntax
HypMenuVTakeOffline()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypMenuVTakeOffline Lib "HsAddin" () As Long
Sub MTakeOffline()
    X=HypMenuVTakeOffline()
End Sub

HypMenuVUndo

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypMenuVUndo() restores the previous database view.

Syntax
HypMenuVUndo()

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.
Example

Public Declare Function HypMenuVUndo Lib "HsAddin" () As Long
Sub MUndo()
    X = HypMenuVUndo()
End Sub

HypMenuVVisualizeinExcel

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypMenuVVisualizeinExcel() retrieves the Excel spreadsheet from which data points were copied to Word or PowerPoint.

Syntax

HypMenuVVisualizeinExcel()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypMenuVVisualizeinExcel Lib "HsAddin" () As Long
Sub MVisualizeinExcel()
    X = HypMenuVVisualizeinExcel()
End Sub

HypMenuVZoomIn

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypMenuVZoomIn() expands the view of data according to the options specified in the Options dialog box.

Syntax

HypMenuVZoomIn()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.
Example

Declare Function HypMenuVZoomIn Lib "HsAddin"() As Long
Sub MZoomIn()
    X=HypMenuVZoomIn()
End Sub

HypMenuVZoomOut

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypMenuVZoomOut() collapses the view of data.

Syntax

HypMenuVZoomOut()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypMenuVZoomOut Lib "HsAddin"() As Long
Sub MZoomOut()
    X=HypMenuVZoomOut()
End Sub

HypExecuteMenu

Data provider types: All

Description

HypExecuteMenu() executes the specified menu or ribbon item.

You can use HypExecuteMenu only with these controls: button, split button, menu, dynamic menu, and toggle button (toggle buttons for extensions are not supported).

Syntax

HypExecuteMenu (vtSheetName, vtMenuName) As Long
ByVal vtSheetName As Variant
ByVal vtMenuName As Variant
Parameters

vtSheetName: Input parameter; the name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtMenuName: Input parameter; the name of the menu item to execute.

- For items that are displayed on multiple ribbons or menus, you must prepend the ribbon title (Office 2007 or later) to the item name using the characters -> to avoid ambiguity. For example, to distinguish between Refresh on the Smart View ribbon and Refresh on the Essbase ribbon, use Smart View->Refresh or Essbase->Refresh. Duplicate items within the same data provider or extension ribbon cannot be used.

- Only items associated with an action are supported. For example, Panel can be used, because it opens the Smart View Panel. Connections cannot be used, because it is not associated with an action.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code. Common error codes for this function are -15 (invalid parameter) and -73 (ambiguity: “Could not resolve menu name”).

Example

Public Declare Function HypExecuteMenu Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMenuName As Variant) As Long
Sub Example_ExecuteMenu()
sts = HypExecuteMenu("Sheet1", "Panel") 'returns 0
sts = HypExecuteMenu(Empty, "Smartview->Refresh") 'returns 0
sts = HypExecuteMenu("Sheet1", "Refresh") 'returns -73(ambiguity)
sts = HypExecuteMenu("Sheet1", "Connections") 'returns -15(invalid parameter because "Connections" is not associated with an action)
End Sub

HypHideRibbonMenu

Data provider types: All

Description

Hides ribbon menus and menu items. Also hides context ribbon menus and menu items based on sheet input.

Smart View ribbon customization is applicable for any sheet. Context ribbon customization is sheet-based.

This function is supported for Office 2007 and above.

Syntax

Public Declare Function HypHideRibbonMenu Lib "HsAddin" (ByVal vtSheetName As Variant, ParamArray vtMenus() As Variant) As Long
ByVal vtSheetName As Variant
ParamArray vtMenus() As Variant

Parameters

vtSheetName: Input variable containing the sheet name on which ribbon menus and menu items are to be hidden.

vtMenus: Input variable containing ribbon menu names and menu item names to be hidden.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code. A common error code for this function is -73 (ambiguity).

Example

Sub HideMenus()
    sts = HypHideRibbonMenu("Sheet1", "Smart View->Submit Data", "Panel")
    sts = HypHideRibbonMenu("Sheet1", "Smart View->Refresh->Refresh")
    'Hides the submenu item Refresh under the Refresh split button
    sts = HypHideRibbonMenu("Sheet1", "Essbase->POV", "Smart View->Copy", "Essbase->Same Workbook")
    'Hides menu items for different ribbons on the same sheet
End Sub

HypHideRibbonMenuReset

Data provider types: All

Description

Resets visibility of the ribbon menus and menu items hidden on the sheet using HypHideRibbonMenu.

This function is supported for Office 2007 and above.

Syntax

Public Declare Function HypHideRibbonMenuReset Lib "HsAddin" (ByVal vtSheetName As Variant) As Long

ByVal vtSheetName As Variant

Parameters

vtSheetName: Input variable containing the sheet name on which the hidden menus and hidden menu items are to be reset to visible state.
Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Sub HideMenuReset()

sts = HypHideRibbonMenuReset ("Sheet1")
' Resets the visibility of menus and menu items hidden on this sheet

End Sub
About General Functions

General VBA functions perform actions, set options, or retrieve information typically performed from the Smart View ribbon or Options dialog box.

HypShowPanel

Data provider types: All
Description
HypShowPanel () shows or hides the Smart View Panel. Once hidden, the Smart View Panel will be displayed only when the user selects Panel on the Smart View ribbon or runs HypShowPanel.

Syntax
HypShowPanel Lib (bShow)
ByVal bShow As Boolean

Parameters
bShow: Set to True to show the Smart View Panel. Set to False to hide the Smart View Panel

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Examples
To show the Smart View Panel:

Public Declare Function HypShowPanel Lib "HsAddin" (ByVal bShow As Boolean) As Long
Sub Example_HypShowPanel()
sts = HypShowPanel(True)
End Sub

To hide the Smart View Panel:

Public Declare Function HypShowPanel Lib "HsAddin" (ByVal bShow As Boolean) As Long
Sub Example_HypShowPanel()
sts = HypShowPanel(False)
End Sub

HypGetVersion
Data provider types: All

Description
HypGetVersion() retrieves any of the following information about the installed version of Smart View and creates a version information file:

- Product version number
- Build number
- Build date
- Build version
Syntax

HypGetVersion (vtID, vtValueList, vtVersionInfoFileCommand)

ByVal vtID As Variant
ByRef vtValueList As Variant
ByVal vtVersionInfoFileCommand As Variant

Parameters

vtID: Input parameter; the ID for which the information is required; can be one of the following constants or strings or empty:
- BUILD_DATE or “BUILD DATE”
- BUILD_NUMBER or “BUILD NO”
- BUILD_VERSION or “VERSION”
- PRODUCT_ID or “PRODUCT” ID
- Empty: If empty, the output list contains all information in the version information file with comma-separated values.

vtValueList: Output parameter; the array list or required value

vtVersionInfoFileCommand: Input parameter; a numerical command ID to save or launch the version information file if vtID is empty. Possible values:
- 0- Do nothing
- 1- Save the version information file
- 2- Launch the version information file

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code

Examples

To create a message box that displays the build version:

Public Declare Function HypGetVersion Lib "HsAddin" (ByVal vtID As Variant, ByRef vtValueList As Variant, ByVal vtVersionInfoFileCommand As Variant) As Long
Sub Example_HypGetVersion()
    sts = HypGetVersion(BUILD_VERSION, version, 0)
    MsgBox version(0)
End Sub

To retrieve and save version information in a version information file:

Public Declare Function HypGetVersion Lib "HsAddin" (ByVal vtID As Variant, ByRef vtValueList As Variant, ByVal vtVersionInfoFileCommand As Variant) As Long
Sub Example_HypGetVersion()
    sts = HypGetVersion(“”, versioninfo, 1) ‘saves version info file in user directory and gets array
inf = versioninfo(0) 'gets the information in 0th array element
End Sub

**HypGetLastError**

Data provider types: All

**Description**

HypGetLastError() returns the last error message stored in Smart View. It retrieves the error message as it is stored in the server (error messages returned via VBA functions may not match those retrieved from the server).

**Syntax**

HypGetLastError (vtErrorCode, vtErrorMessage, vtErrorDescription)

ByRef vtErrorCode As Variant
ByRef vtErrorMessage As Variant
ByRef vtErrorDescription As Variant

**Parameters**

- vtErrorCode: The error code number
- vtErrorMessage: The error message
- vtErrorDescription: A description of the error

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

```vbnet
Public Declare Function HypGetLastError Lib "HsAddin" (ByRef vtErrorCode As Variant, ByRef vtErrorMessage As Variant, ByRef vtErrorDescription As Variant) As Long
Sub Example_HypGetLastError
ReturnValue = HypGetLastError(ErrorMessageValue, ErrorMessageValue, ErrorDescriptionValue)
End Sub
```

**HypShowPov**

Data provider types: All

**Description**

HypShowPov() shows or hides the POV toolbar.
**HypShowPov**

Syntax

```vbnet
HypShowPov(bShowPov)
```

ByVal bShowPov As Boolean

**Parameters**

bShowPov: Set to True to show the POV toolbar. Set to False to hide the POV toolbar.

**Return Value**

Returns 0 if successful; otherwise, the appropriate error code.

**Example**

```vbnet
Public Declare Function HypShowPov Lib "HsAddin" (ByVal bShowPov As Boolean) As Long
Sub Example_HypShowPov()
    X=HypShowPov(True)
End Sub
```

---

**HypSetMenu**

**Data provider types:** Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

In Excel 2007 and 2010, HypSetMenu( ) shows or hides the Smart View and data provider ribbons.

**Syntax**

```vbnet
HypSetMenu(bSetMenu)
```

ByVal bSetMenu As Boolean

**Parameters**

bSetMenu: Set to True to show the ribbons or menu. Set to False to hide the menu or ribbons.

**Return Value**

Returns 0 if successful; otherwise, the appropriate error code

**Example**

```vbnet
Declare Function HypSetMenu Lib "HsAddin" (ByVal bSetMenu As Boolean) As Long
Sub Example_HypSetMenu()
    X=HypSetMenu(True)
End Sub
```
HypCopyMetaData

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypCopyMetaData() copies the metadata from one worksheet to another worksheet.

Syntax
HypCopyMetaData (vtSourceSheetName, vtDestinationSheetName)
ByVal vtSourceSheetName As Variant
ByVal vtDestinationSheetName As Variant

Parameters
vtSourceSheetName: The name of the worksheet that contains the data to be copied
vtDestinationSheetName: The name of the destination worksheet

Return Value
Returns 0 if successful; otherwise, the appropriate error code.

Example
Public Declare Function HypCopyMetaData Lib "HsAddin" (ByVal vtSourceSheetName As Variant, ByVal vtDestinationSheetName As Variant) As Long
Sub Example_HypCopyMetaData()
  Dim LRet As Long
  LRet = HypCopyMetaData ("Sheet1", "Sheet2")
End Sub

HypDeleteMetaData

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise, Oracle Hyperion Reporting and Analysis

Description
HypDeleteMetaData() deletes Smart View metadata from the workbook in any of three modes:

- Mode 1—Delete all Smart View metadata only from the provided worksheet storage
- Mode 2—Delete all Smart View metadata only from the provided workbook storage
- Mode 3—Delete all Smart View metadata from the provided workbook storage and from all the worksheets’ storage

Syntax
HypDeleteMetaData(vtDispObject, vtbWorkbook, vtbClearMetadataOnAllSheetsWithinWorkbook)
vtDispObject As Variant
vtbWorkbook As Variant
vtbClearMetadataOnAllSheetsWithinWorkbook As Variant

Parameters

vtDispObject: Dispatch object of worksheet or workbook that indicates where to delete metadata. If Null is passed, then vtbWorkbook determines the active worksheet or active workbook and will be operated upon.

vtbWorkbook: Boolean. Indicates that you passed worksheet dispatch or workbook dispatch. If Null is passed in vtDispObject, then this flag will determine that the user wants to delete metadata from active worksheet or active workbook.

vtbClearMetadataOnAllSheetsWithinWorkbook: Boolean. Specifies that Smart View metadata should be deleted from all sheets within the workbook. Used only if vtbWorkbook is True.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypDeleteMetaData Lib "HsAddin" (ByVal vtDispObject As Variant, ByVal vtbWorkbook As Variant, ByVal vtbClearMetadataOnAllSheetsWithinWorkbook As Variant) As Long

Sub Example_HypDeleteMetaData()
    Dim Ret As Long
    Dim Workbook As Workbook
    Dim Sheet As Worksheet
    Set Workbook = ActiveWorkbook
    Set Sheet = ActiveSheet

    'Ret = HypDeleteMetaData(oSheet, False, True)      'Mode 1
    Ret = HypDeleteMetaData(oWorkbook, True, False)  'Mode 2
    'Ret = HypDeleteMetaData(oWorkbook, True, True)   'Mode 3

    MsgBox (Ret)
End Sub

HypIsDataModified

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypIsDataModified() determines whether any data cells have been modified but not yet submitted.
Syntax
HypIsDataModified (vtSheetName)
By Val vtSheetName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

Return Value
Returns True if the worksheet contains any data cells that have been modified but not yet submitted; otherwise, False.

Example

Public Declare Function HypIsDataModified Lib "HsAddin" (ByVal vtSheetName As Variant)As Boolean
Sub Example_HypIsDataModified()
Dim oRet As Boolean
oRet = HypIsDataModified(Empty)
MsgBox (oRet)
End Sub

HypIsSmartViewContentPresent

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypIsSmartViewContentPresent() determines whether the sheet contains Smart View content.

Syntax
HypIsSmartViewContentPresent(vtSheetName, vtTypeOfContentsInSheet])
ByVal vtSheetName As Variant
ByRef vtTypeOfContentsInSheet

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtTypeOfContentsInSheet: Output parameter; returns the type of content on the worksheet. Possible values are in the enum as follows:

Enum TYPE_OF_CONTENTS_IN_SHEET
    EMPTY_SHEET
    ADHOC_SHEET
    FORM_SHEET

Return Value
Returns True if the worksheet contains Smart View content; otherwise, returns False.

Example

Public Declare Function HypIsSmartViewContentPresent Lib "HsAddin" (ByVal vtSheetName As Variant, ByRef vtTypeOfContentsInSheet As TYPE_OF_CONTENTS_IN_SHEET) As Boolean

Sub Example_HypIsSmartViewContentPresent()
    Dim Ret As Boolean
    Dim vtTypeOfContentsInSheet As TYPE_OF_CONTENTS_IN_SHEET
    Dim SheetName As String
    Dim SheetDisp As Worksheet

    SheetName = Empty
    Set SheetDisp = Worksheets("Sheet1")
    Ret = HypIsSmartViewContentPresent (Empty, ContentType)
End Sub

HypIsFreeForm

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypIsFreeForm() determine whether the worksheet is in free-form mode.

Syntax
HypIsFreeForm (vtSheetName)
By Val vtSheetName As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

Return Value
Returns True if the worksheet is in free-form state; otherwise, returns False.

Example

Public Declare Function HypIsFreeForm Lib "HsAddin" (ByVal vtSheetName As Variant) As Boolean
Sub Example_HypIsFreeForm()
    Dim oRet As Boolean
    oRet = HypIsFreeForm(Empty)
**HypUndo**

**Data provider types:** Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypUndo() reverts the database view of a worksheet to what it was before a Zoom In, Zoom Out, Keep Only, Remove Only, or Refresh operation.

**Syntax**

HypUndo (vtSheetName)

ByVal vtSheetName As Variant

**Parameters**

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypUndo Lib "HsAddin" (ByVal vtSheetName As Variant) As Long
Sub Example_Hypundo()
X=HypUndo(Sheet1)
End Sub

**HypRedo**

**Data provider types:** Essbase, Planning (ad hoc only) Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypRedo() reverts the database view to what it was before an Undo operation.

**Syntax**

HypRedo (vtSheetName)

ByVal vtSheetName As Variant
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is `Null` or `Empty`, the active worksheet is used.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

```vba
Declare Function HypRedo Lib "HsAddin" (ByVal vtSheetName As Variant) As Long
Sub Example_HypRedo()
X=HypRedo(Sheet1)
End Sub
```

**HypPreserveFormatting**

**Data provider types:** Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypPreserveFormatting() applies grid formatting to cells created by zooming in.

**Syntax**

HypPreserveFormatting (vtSheetName, vtSelectionRange)

- `ByVal vtSheetName As Variant`
- `ByVal vtSelectionRange As Variant`

**Parameters**

- **vtSheetName**: The name of worksheet on which to run the function. If vtSheetName is `Null` or `Empty`, the active worksheet is used.
- **vtSelectionRange**: The range of cell(s) in which formatting is to be preserved. Multiple ranges are supported.

**Return Value**

Returns 0 if successful; otherwise, the appropriate error code.

**Example**

```vba
Public Declare Function HypPreserveFormatting Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelectionRange As Variant) As Long
Sub Example_HypPreserveFormatting()
    Dim oRet As Long
    Dim oSheetName As String
```
Dim oSheetDisp As Worksheet

oSheetName = Empty
Set oSheetDisp = Sheet1
oRet = HypPreserveFormatting ("", oSheetDisp.Range("B2"))
MsgBox (oRet)

End Sub

**HypRemovePreservedFormats**

**Data provider types:** Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypRemovePreservedFormats() removes preserved formats.

**Note:** Users must refresh before the original formatting is applied.

**Syntax**

HypRemovePreservedFormats (vtSheetName, vtbRemoveAllCapturedFormats, vtSelectionRange)

*ByVal vtSheetName As Variant*

*ByVal vtbRemoveAllCapturedFormats As Variant*

*ByVal vtSelectionRange As Variant*

**Parameters**

**vtSheetName:** The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

**vtbRemoveAllCapturedFormats:** Set to True to remove all preserved formats in the selected range. Otherwise, set to False. If set to True, the next parameter value is not used, so users can pass Null for vtSelectionRange.

**vtSelectionRange:** The range of the cell(s) in which formatting is to be preserved. Multiple ranges are supported.

**Return Value**

Returns 0 if successful; otherwise, the appropriate error code.

**Example**

Public Declare Function HypRemovePreservedFormats Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtbRemoveAllCapturedFormats As Variant, ByVal vtSelectionRange As Variant) As Long
Sub Example_HypRemovePreservedFormats()
    Dim Ret As Long
    Dim SheetName As String
    Dim SheetDisp As Worksheet

    SheetName = "Sheet1"

    Set oSheetDisp = Worksheets(SheetName)
    'Ret = HypRemovePreservedFormats(Empty, False, SheetDisp.Range("B2"))
    Ret = HypRemovePreservedFormats(Empty, True, Null)
    MsgBox (oRet)
End Sub

HypSetAliasTable

Data provider types: Essbase, Planning

Description
HypSetAliasTable() sets the alias table for the selected worksheet.

Syntax
HypSetAliasTable (ByVal vtSheetName As Variant, ByVal vtAliasTableName As Variant)

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtAliasTableName: The text name of the alias table. vtAliasTableName is of the form “Default”, “Long Names” and so forth.

Return Value
0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypSetAliasTable Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtAliasTableName As Variant) As Long
Sub Example_SetAliasTable
    sts = HypSetAliasTable(Empty,"Long Names")
End sub

HypGetSubstitutionVariable

Data provider types: Essbase
Description

HypGetSubstitutionVariable() retrieves substitution variables and their current values from Essbase.

Syntax

HypGetSubstitutionVariable (vtSheetName, vtApplicationName, vtDatabaseName, vtVariableName, vtVariableNames, vtVariableValues)

ByVal vtSheetName As Variant
ByVal vtApplicationName As Variant
ByVal vtDatabaseName As Variant
ByVal vtVariableName As Variant
ByRef vtVariableNames As Variant
ByRef vtVariableValues As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtApplicationName: The name of the application from which to return substitution variables. If set to Null or Empty, all the applications are considered.

vtDatabaseName: The name of the database from which to return substitution variables. If set to Null or Empty, all the databases are considered.

vtVariableName: The name of the substitution variable to be retrieved. If set to Null or Empty, the entire list of variables is returned.

vtVariableNames: Output result vector that contains the list of the substitution variable names. Its contents are unknown if the macro fails.

vtVariableValues: Output result vector that contains the list of the substitution variable values corresponding to each variable returned. Its contents are unknown if the macro fails.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypGetSubstitutionVariable Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtApplicationName As Variant, ByVal vtDatabaseName As Variant, ByVal vtVariableName As Variant, ByRef vtVariableNames As Variant, ByRef vtVariableValues As Variant) As Long

Sub Example_HypGetSubstitutionVariable()
    Dim sts As Long
    sts = HypGetSubstitutionVariable(Empty, "Sample", "Basic", Empty, vtVarNameList, vtVarValueList)
End Sub
**HypSetSubstitutionVariable**

**Data provider types:** Essbase

**Description**

HypSetSubstitutionVariable() creates substitution variables in Essbase. If the variable already exists, then its value is set to the new specified value.

**Syntax**

HypSetSubstitutionVariable (vtSheetName, vtApplicationName, vtDatabaseName, vtVariableName, vtVariableValue)

- ByVal vtSheetName As Variant
- ByVal vtApplicationName As Variant
- ByVal vtDatabaseName As Variant
- ByVal vtVariableName As Variant
- ByVal vtVariableValue As Variant

**Parameters**

- **vtSheetName:** The name of worksheet on which to run the function. If set to Null or Empty, the active worksheet is used.

- **vtApplicationName:** The name of the application name in which to create the new substitution variable. If set to Null or Empty, the scope of the variable is global.

- **vtDatabaseName:** The name of the database in which to create the new variable. If set to Null or Empty, the scope of the variable created is global within the application specified.

- **vtVariableName:** The variable name to be created. Required.

- **vtVariableValue:** The value to be assigned to the variable. Required.

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypSetSubstitutionVariable Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtApplicationName As Variant, ByVal vtDatabaseName As Variant, ByVal vtVariableName As Variant, ByVal vtVariableValue As Variant) As Long

Sub Example_HypSetSubstitutionVariable
    Dim X As Long
    X = HypSetSubstitutionVariable(Empty, "Sample", "Basic", "Account", "100")
End Sub
HypGetDatabaseNote

Data provider types: Essbase

Description
HypGetDatabaseNote() retrieves Essbase database notes.

Syntax
HypGetDatabaseNote (vtSheetName, vtDBNote)
ByVal vtSheetName As Variant
ByRef vtDBNote As Variant

Parameters
vtSheetName: Input parameter; the name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtDBNote: Output parameter; the database note to be retrieved.

Example
Public Declare Function HypGetDatabaseNote Lib "HsAddin" (ByVal vtSheetName As Variant, ByRef vtDBNote As Variant) As Long
Sub Example_HypGetDatabaseNote()
sts = HypGetDatabaseNote(Empty, DBNote)
MsgBox DBNote
End Sub
Connection Functions

In This Chapter

- About Connection Functions ................................................................. 61
- HypConnect .......................................................................................... 61
- HypUIConnect ...................................................................................... 62
- HypConnected ..................................................................................... 63
- HypConnectionExists .......................................................................... 64
- HypCreateConnection ......................................................................... 65
- HypCreateConnectionEx ...................................................................... 66
- HypDisconnect .................................................................................... 68
- HypDisconnectAll ............................................................................... 69
- HypDisconnectEx ............................................................................... 69
- HypGetSharedConnectionsURL ............................................................ 70
- HypSetSharedConnectionsURL ............................................................ 70
- HypIsConnectedToSharedConnections ................................................ 71
- HypRemoveConnection ........................................................................ 72
- HypInvalidateSSO ............................................................................... 72
- HypResetFriendlyName ....................................................................... 72
- HypSetActiveConnection ...................................................................... 73
- HypSetAsDefault ................................................................................ 74
- HypSetConnAliasTable ........................................................................ 75

About Connection Functions

Connection functions perform actions related to connections to data providers.

HypConnect

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypConnect() logs into a data provider and associates the worksheet with that connection. HypConnect() must be called for each sheet in order to associate this connection with that sheet.
Syntax

HypConnect (vtSheetName, vtUserName, vtPassword, vtFriendlyName)
ByVal vtSheetName As Variant
ByVal vtUserName As Variant
ByVal vtPassword As Variant
ByVal vtFriendlyName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtUserName: A valid user name

vtPassword: The password for this user

vtFriendlyName: The friendly connection name of the data provider. This is the connection name created by HypCreateConnection.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypConnect Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtUserName As Variant, ByVal vtPassword As Variant, ByVal vtFriendlyName As Variant) As Long

Sub Example_HypConnect()
  X=HypConnect(Empty, UserName, Password, "My Sample Basic")
End Sub

HypUIConnect

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise, Oracle Business Intelligence Enterprise Edition

Description

HypUIConnect() prompts the user with the Connect to Data Source dialog box when the user name and password is not provided. It does not prompt if already connected.

Syntax

HypUIConnect Lib (vtSheetName, vtUserName, vtPassword, vtFriendlyName)
ByVal vtSheetName As Variant
ByVal vtUserName As Variant
ByVal vtPassword As Variant
ByVal vtFriendlyName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtUserName: A valid user name

vtPassword: The password for this user

vtFriendlyName: The connection name of the data provider

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare PtrSafe Function HypUIConnect Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtUserName As Variant, ByVal vtPassword As Variant, ByVal vtFriendlyName As Variant) As Long
HypUIConnect(Empty, UserName, Password, "My Connection")

HypConnected

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypConnected() provides the connection status of the sheet.

Syntax

HypConnected (vtSheetName)

ByVal vtSheetName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

Return Value

Returns True if the sheet is connected to a provider; False if it is not.

Example

Declare Function HypConnected Lib "HsAddin" (ByVal vtSheetName As Variant) As Variant

Sub Example_HypConnected
    Dim X As Variant

If the sheet is connected, a variant with a value of -1 is returned, which is interpreted as True by VBA. In order to get -1 as the return value, you must declare the variable (which takes a return value) as a number type (Long, Integer, Double, etc.). The script given below demonstrates this:

```
Declare Function HypConnected Lib "HsAddin" (ByVal vtSheetName As Variant) As Variant
Sub Example_HypConnected()
    Dim X As Integer 'Can also be Long or Double
    X = HypConnected(Empty)  'Value of X will become -1 if Sheet1 is connected
End Sub
```

If variable X is not defined, VBA interprets it (and any other variable which is not defined) as being of the type, Variant. Then, if Sheet1 is connected, X will be equal to True.

If variable X is defined as a boolean, the return value is correctly displayed as True.

---

**HypConnectionExists**

**Data provider types:** Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypConnectionExists() checks whether a particular connection name exists in the list of all connections as viewed in the Smart View Panel. The particular connection may or may not be active (connected).

**Syntax**

```
HypConnectionExists(vtFriendlyName)
```

**Parameters**

- **vtFriendlyName:** The name of the connection to search for in the list of all connections. It is not case-sensitive.

**Return Value**

Boolean. If successful, return value is TRUE; otherwise, return value is FALSE.

**Example**

```
Declare Function HypConnectionExists Lib "HsAddin" (ByVal vtFriendlyName As Variant) As Variant
Sub Example_HypConnectionExists
    Dim bIsConnection as Boolean
    bIsConnection = HypConnectionExists ("Demo_Basic")
End sub
```
**HypCreateConnection**

**Data provider types:** Essbase, Financial Management, Hyperion Enterprise

**Description**

HypCreateConnection() creates a connection to the data provider from the specified information. See also “HypCreateConnectionEx” on page 66.

**Note:** Planning users who want to add data providers in the Smart View Panel must use HypCreateConnectionEx.

**Note:** Use HypConnect to establish the connection.

**Syntax**

HypCreateConnection(vtSheetName, vtUserName, vtPassword, vtProvider, vtProviderURL, vtServerName, vtApplicationName, vtDatabaseName, vtFriendlyName, vtDescription)

ByVal vtSheetName As Variant
ByVal vtUserName As Variant
ByVal vtPassword As Variant
ByVal vtProvider As Variant
ByVal vtProviderURL As Variant
ByVal vtServerName As Variant
ByVal vtApplicationName As Variant
ByVal vtDatabaseName As Variant
ByVal vtFriendlyName As Variant
ByVal vtDescription As Variant

**Parameters**

- **vtSheetName:** Not used
- **vtUserName:** A valid user name
- **vtPassword:** The password for this user
- **vtProvider:** The data provider. Supported vtProvider types:
  - Global Const HYP_ESSBASE = "Essbase"
  - Global Const HYP_ENTERPRISE = "Hyperion Enterprise"
  - Global Const HYP_FINANCIAL_MANAGEMENT = "Hyperion Financial Management"
Note: The global constant HYP_ANALYTIC_SERVICES = "Analytic Provider Services" has been deprecated.

vtProviderURL: The URL of the data provider
vtServerName: The name of the server on which the application resides
vtApplicationName: The name of the application
vtDatabaseName: The name of the database
vtFriendlyName: The connection name of the data provider
vtDescription: A description of the data provider

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypCreateConnection Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtUserName As Variant, ByVal vtPassword As Variant, ByVal vtProvider As Variant, ByVal vtProviderURL As Variant, ByVal vtServerName As Variant, ByVal vtApplicationName As Variant, ByVal vtDatabaseName As Variant, ByVal vtFriendlyName As Variant, ByVal vtDescription As Variant) As Long
Sub Example_HypCreateConnection()
X = HypCreateConnection(Empty, UserName, Password, HYP_ESSBASE, "http://localhost:13080/smartview/SmartView", "localhost", "Sample", "Basic", "My Connection", "Essbase_1")
End Sub

HypCreateConnectionEx

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise, Reporting and Analysis

Description
HypCreateConnectionEx is a superset of HypCreateConnection; it has additional parameters that enable use of the Smart View Panel. Planning users who want to add data providers in the Smart View Panel must use HypCreateConnectionEx.

For Essbase, Planning, and Financial Management, HypCreateConnectionEx can be used to create private connections using a Workspace URL.

Syntax
HypCreateConnectionEx (vtProviderType, vtServerName, vtApplicationName, vtDatabaseName, vtFormName, vtProviderURL, vtFriendlyName, vtUserName, vtPassword, vtDescription, vtReserved1, ByVal vtReserved2)
ByVal vtProviderType As Variant
ByVal vtServerName As Variant
ByVal vtApplicationName As Variant
ByVal vtDatabaseName As Variant
ByVal vtFormName As Variant
ByVal vtProviderURL As Variant
ByVal vtFriendlyName As Variant
ByVal vtUserName As Variant
ByVal vtPassword As Variant
ByVal vtDescription As Variant
ByVal vtReserved1 As Variant (reserved for future use)
ByVal vtReserved2 As Variant (reserved for future use)

Parameters

vtProviderType: The data provider. Supported vtProviderType types:
  • Global Const HYP_ESSBASE = "Essbase"
  • Global Const HYP_PLANNING = "Planning"
  • Global Const HYP_FINANCIAL_MANAGEMENT = "Financial Management"
  • Global Const HYP_RA = "Hyperion Smart View Provider for Hyperion Reporting and Analysis"
  • Global Const HYP_ENTERPRISE = "Hyperion Enterprise"

vtServerName: The name of the server on which the application resides
vtApplicationName: The name of the application
vtDatabaseName: The name of the database
vtFormName: The name of the data form. Required to create Planning connection in Smart View Panel under Favorites
vtProviderURL: The data provider URL. Required to create Planning connection in Smart View Panel.
vtFriendlyName: The connection name of the data provider
vtUserName: A valid user name
vtPassword: The password for this user
vtDescription: Description for the data provider

Note: For Oracle Hyperion Reporting and Analysis, only the provider URL, provider type, and connection name are required.
**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Public Declare Function HypCreateConnectionEx Lib "HsAddin" (ByVal vtProviderType As Variant, ByVal vtServerName As Variant, ByVal vtApplicationName As Variant, ByVal vtDatabaseName As Variant, ByVal vtFormName As Variant, ByVal vtProviderURL As Variant, ByVal vtFriendlyName As Variant, ByVal vtUserName As Variant, ByVal vtPassword As Variant, ByVal vtDescription As Variant, ByVal vtReserved1 As Variant, ByVal vtReserved2 As Variant) As Long

Sub Example_HypCreateConnectionEx()

Dim lRet As Long
lRet = HypCreateConnectionEx("Essbase", "server12", "Demo", "Basic", ",", ",", "My Demo", "Username", "Password", ",", ",")
End Sub

**HypDisconnect**

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypDisconnect() logs out from the data provider.

**Syntax**

HypDisconnect(vtSheetName, bLogoutUser)

ByVal vtSheetName As Variant
ByVal bLogoutUser As Boolean

**Parameters**

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

bLogoutUser: Set to True to disconnect and log out from the provider session. Default value is False.

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.
Example

Declare Function HypDisconnect Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal bLogoutUser As Boolean) As Long

Sub Example_HypDisconnect()
    X=HypDisconnect(Empty, True)
End Sub

HypDisconnectAll

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypDisconnectAll is a security measure that disconnects all connected users and invalidates the user authentication. Equivalent of the Disconnect All menu item.

Syntax

HypDisconnectAll()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypDisconnectAll Lib "HsAddin" () As Long
Sub Example_HypDisconnectAll()
    sts = HypDisconnectAll()
End Sub

HypDisconnectEx

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypDisconnectEx disconnects the specified connection. This connection need not be associated as in HypDisconnect.

Syntax

HypDisconnectEx (vtFriendlyName )
ByVal vtFriendlyName As Variant

Parameters

vtFriendlyName: The friendly connection name to be disconnected
Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypDisconnectEx Lib "HsAddin" (ByVal vtFriendlyName As Variant) As Long
Sub Example_HypDisconnectEx()
    Dim lRet As Long
    lRet = HypDisconnectEx("My Sample")
End Sub

HypGetSharedConnectionsURL

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypGetSharedConnectionsURL() returns the Shared Connections URL to be used (also shown in the Options dialog box).

Syntax

HypGetSharedConnectionsURL (vtSharedConnURL)
ByRef vtSharedConnURL As Variant

Parameters

vtSharedConnURL: Output parameter; the Shared Connections URL

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypGetSharedConnectionsURL Lib "HsAddin" (ByRef vtSharedConnURL As Variant) As Long
Sub Example_HypGetSharedConnectionsURL()
    Dim lRet As Long
    Dim conn As Variant
    lRet = HypGetSharedConnectionsURL(conn)
    MsgBox (lRet)
    MsgBox (conn)
End Sub

HypSetSharedConnectionsURL

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise
HypSetSharedConnectionsURL() sets the Shared Connections URL in the config file and Options dialog box.

Syntax
HypSetSharedConnectionsURL (vtSharedConnURL)
ByVal vtSharedConnURL As Variant

Parameters
vtSharedConnURL: the new Shared Connections URL to be set.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypSetSharedConnectionsURL Lib "HsAddin" (ByVal vtSharedConnURL As Variant) As Long
Sub Example_HypSetSharedConnectionsURL()
Dim lRet As Long
lRet = HypSetSharedConnectionsURL("http://<server>:19000/workspace/SmartViewProviders")
End Sub

HypIsConnectedToSharedConnections

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypIsConnectedToSharedConnections() determines whether SmartView is connected to Shared Connections.

Syntax
HypIsConnectedToSharedConnections ()

Return Value
Return: True if Smart View is connected to Shared Connections, otherwise, False.

Example
Declare Function HypIsConnectedToSharedConnections Lib "HsAddin" () As Variant
Sub Example_HypIsConnectedToSharedConnections()
Dim vtRet As Variant
vtRet = HypIsConnectedToSharedConnections ()
MsgBox(vtRet)
End Sub
HypRemoveConnection

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypRemoveConnection() removes the specified connection from the list of available Smart View connections in the Smart View Panel.

Syntax
HypRemoveConnection(vtFriendlyName)
ByVal vtFriendlyName As Variant

Parameters
vtFriendlyName: The friendly connection name of the data provider

Return Value
Returns 0 if successful, otherwise, returns the appropriate error code.

Example
Declare Function HypRemoveConnection Lib "HsAddin" (ByVal vtFriendlyName As Variant) As Long
Sub Example_HypRemoveConnection()
    X = HypRemoveConnection("My Connection")
End Sub

HypInvalidateSSO

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypInvalidateSSO() discards the existing SSO token.

Example
Declare Function HypInvalidateSSO Lib "HsAddin" () As Long
Sub Example_HypInvalidateSSO()
    X = HypInvalidateSSO()
End Sub

HypResetFriendlyName

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise
Description
HypResetFriendlyName modifies the friendly name to a new one. To modify the friendly name of a connection in the Smart View Panel, Smart View must be connected to Oracle Hyperion Provider Services.

Syntax
HypResetFriendlyName (vtOldFriendlyName, vtNewFriendlyName)
By Val vtOldFriendlyName As Variant
By Val vtNewFriendlyName As Variant

Parameters
vtOldFriendlyName: The old friendly connection name
vtNewFriendlyName: The new friendly connection name

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypResetFriendlyName Lib "HsAddin" (ByVal vtOldFriendlyName As Variant, ByVal vtNewFriendlyName As Variant) As Long
Sub Example_HypResetFriendlyName()
    Dim lRet As Long
    lRet = HypResetFriendlyName("server2_Sample_Basic", "My Sample Basic")
End Sub

HypSetActiveConnection

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypSetActiveConnection() associates the current active worksheet with one of the active connections.

Note: HypSetActiveConnection does not work with worksheets that contain Report Designer objects

Syntax
HypSetActiveConnection (vtFriendlyName)
ByVal vtFriendlyName As Variant
Parameters

vtFriendlyName: The friendly name of the active connection to be associated with the current active worksheet. It is not case-sensitive.

Return Value

Long. If successful, return value is 0; otherwise, the appropriate error code is returned.

Example

Declare Function HypSetActiveConnection Lib "HsAddin" (ByVal vtFriendlyName As Variant) As Long

Sub Example_SetActiveConnection()
    sts = HypSetActiveConnection("Demo_Basic")
End sub

HypSetAsDefault

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypSetAsDefault() sets a connection default.

Syntax

HypSetAsDefault (vtFriendlyName)

ByVal vtFriendlyName As Variant

Parameters

vtFriendlyName: The name of the private active connection to be set as the default. It must be a private connection name whose value can be found in the Registry: HKCU\Software \Hyperion Solutions\HyperionSmartView\Connections

Return Value

If successful, return value is 0; otherwise, the appropriate error code is returned.

Example

Public Declare Function HypSetAsDefault Lib "HsAddin" (ByVal vtFriendlyName As Variant) As Long

Sub Example_SetAsDefault()
    sts = HypSetAsDefault("buildtie7_w32Simple_w32Simple")
    MsgBox (sts)
End Sub
HypSetConnAliasTable

Data provider types: Essbase, Planning

Description
HypSetConnAliasTable() sets the alias table for a connection. This function requires an active connection.

Syntax
HypSetConnAliasTable (ByVal vtFriendlyName As Variant, ByVal vtAliasTableName As Variant)

Parameters
vtFriendlyName: The connection name of the data provider; for example, "MyConnectionString1" or "SampleBasic". If vtFriendlyName is Null or Empty, an error is returned.

vtAliasTableName: The name of the alias table in the form “Default”, “Long Names”, “None”, and so forth. This parameter cannot be Null or Empty. If no alias needs to be applied, then you can use the parameter “None”.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypSetConnAliasTable Lib "HsAddin" (ByVal vtFriendlyName As Variant, ByVal vtAliasTableName As Variant) As Long

Sub Example_HypSetConnAliasTable
sts = HypSetConnAliasTable("SampleBasic","Long Names")
End sub
About Ad Hoc Functions

Ad hoc functions perform ad hoc operations such as zooming, retrieving and submitting data, and pivoting.

HypPerformAdhocOnForm

Data provider types: Planning

Description

HypPerformAdhocOnForm() enables ad hoc analysis in Excel worksheets for Planning web forms.

Syntax

HypPerformAdhocOnForm(vtSheetName, vtFormName)
ByVal vtSheetName As Variant
ByVal vtFormName As Variant

Parameters

vtSheetName: Input variable; the name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtFormName: Input variable; the name of the Planning web form, including its full path; for example, /Forms/Financials/Financials Summary

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypPerformAdhocOnForm Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtFormName As Variant) As Long
Sub Example_PerformAdhocOnForm
sts = HypPerformAdhocOnForm(Empty, "/Forms/Financials/Financials Summary")
End Sub

HypRetrieve

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypRetrieve() retrieves data from the database.

Syntax

HypRetrieve(vtSheetName)

ByVal vtSheetName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Examples

Public Declare Function HypRetrieve Lib "HsAddin" (ByVal vtSheetName As Variant) As Long
Sub Example_HypRetrieve()
X=HypRetrieve(Empty)
End Sub
Public Declare Function HypRetrieve Lib "HsAddin" (ByVal vtSheetName As Variant) As Long
Sub Example_HypRetrieve()
X=HypRetrieve(Empty)
If X = 0 Then
    MsgBox("Retrieve successful.")
Else
    MsgBox("Retrieve failed.")
End If
End Sub

HypRetrieveRange

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypRetrieveRange() enables users to refresh a selected or named range of cells in a grid or worksheet. If the range specified for this function contains more rows or columns than the actual grid has, the additional rows and columns are treated as comments and are thus part of the grid.

HypRetrieveRange clears the Undo buffer, therefore the Undo operation cannot be used afterward.

Syntax
HypRetrieveRange(vtSheetName,vtRange,vtFriendlyName)
ByVal vtSheetName As Variant
ByVal vtRange As Variant
ByVal vtFriendlyName As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtRange: The continuous range to be refreshed. This range must contain one or more member cells as well as data cells. If vtRange is Null, the entire worksheet is refreshed, and GetUsedRange is used on the worksheet specified to get the range to be refreshed.

vtFriendlyName: The friendly name of the connection to be used to refresh the range. If set to Null, the active connection associated with the worksheet is used to refresh the range on that worksheet. If no connection is associated, an error is returned.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.
Example

Public Declare Function HypRetrieveRange Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant, ByVal vtFriendlyName As Variant) As Long
Worksheets("Sheet2").Names.Add name:="MyRange", RefersTo:="=$E$11:$F$28"
Sub Example_RetrieveRange
    Worksheets("Sheet1").Names.Add name:="MyRange", RefersTo:="=$E$11:$F$28"
    sts = HypRetrieveRange(Empty, range("E11:F28"), "Samp1")
    'retrieve by regular range
    sts = HypRetrieveRange(Empty, range("MyRange"), "Samp1")
    'retrieve by named range
End sub

HypRetrieveNameRange

Data provider types: Essbase

Description

HypRetrieveNameRange refreshes the grid created by HypRetrieveRange. This function works only with Smart View multi-grid defined range names.

Syntax

HypRetrieveNameRange (vtSheetName, vtGridName)

ByVal vtSheetName As Variant
ByVal vtGridName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtGridName: Input parameter; the name of the named range or grid to be refreshed. Named ranges take the form: "'<Sheetname>'!<range name>"

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Examples

Example 1

Public Declare Function HypRetrieveNameRange Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtGridName As Variant) As Long

Sub RetrieveAllRange()
    'connect all required connections
    sts = HypConnect("Sheet1", "UserName", "Password", "stm10026_Sample_Basic")
    'get list of named grids available
    sts = HypGetNameRangeList("Sheet1", "", vtList)
'refresh each range one by one
For i = 0 To 2
sts = HypRetrieveNameRange("Sheet1", vtList(i))
Next i
End Sub

Example 2
If you know the name of the grid:

Public Declare Function HypRetrieveNameRange Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtGridName As Variant) As Long
Sub Example_HypRetrieveNameRange()
sts = HypRetrieveNameRange("Sheet1", "'Sheet1'!DMDemo_Basic_2")
End Sub

HypGetNameRangeList

Data provider types: Essbase

Description
HypGetNameRangeList returns a list of named grids for a given connection.

Syntax
HypGetNameRangeList (vtSheetName, vtFriendlyName, vtNameList)
ByVal vtSheetName As Variant
ByVal vtFriendlyName As Variant
ByRef vtNameList As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtFriendlyName: Input parameter; the connection name whose list of name ranges are to be retrieved. If set to Empty, all name range lists in the sheet are retrieved.

vtNameList: Output parameter; the list output.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypGetNameRangeList Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtFriendlyName As Variant, ByRef vtNameList As Variant) As Long
Sub Example_HypGetNameRangeList()
sts = HypGetNameRangeList("Sheet1", "stm10026_Sample_Basic", vtList)
End Sub
**HypRetrieveAllWorkbooks**

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypRetrieveAllWorkbooks() refreshes all open workbooks from the same instance of Excel.

Syntax

HypRetrieveAllWorkbooks()

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

```
Public Declare Function HypRetrieveAllWorkbooks Lib "HsAddin" () As Long

Sub Example_HypRetrieveAllWorkbooks()
    X=HypRetrieveAllWorkbooks()
End Sub
```

**HypExecuteQuery**

Data provider types: Essbase

Description

HypExecuteQuery() executes an MDX query and displays the results on a worksheet. (If you do not want to display the query results on a worksheet, use HypExecuteMDXEx instead.)

Syntax

HypExecuteQuery (ByVal vtSheetName As Variant, ByVal vtMDXQuery As Variant) As Long

ByVal vtSheetName As Variant
ByVal vtMDXQuery As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtMDXQuery: The MDX query statement to be executed on the worksheet

Return Value

Long. If successful, return value is 0; otherwise, returns the appropriate error code.
Example

Public Declare Function HypExecuteQuery Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMDXQuery As Variant) As Long

Sub Example_HypExecuteQuery ()
    Dim vtQuery As Variant
    vtQuery = "SELECT {{[Jan]}} on COLUMNS, {{[East]}} on ROWS"
    sts = HypConnect (Empty, "Username", "Password", "Sample_Basic")
    sts = HypExecuteQuery (Empty, vtQuery)
    sts = HypDisconnect (Empty, True)
End sub

HypSubmitData

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypSubmitData() updates the database with modified data from the specified spreadsheet.

Note: HypSubmitData() is not supported with aggregate storage databases or in a clustered environment.

Note: The ability to update the database depends on the access permissions of the submitter. To update data, you must have at least Write access to the database.

Syntax

HypSubmitData(vtSheetName)
ByVal vtSheetName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

Return Value

For forms: Returns 0 if form is submitted successfully; otherwise, returns the appropriate error code.

For ad hoc: Returns 0 if ad hoc grid is submitted successfully and HsSetVal functions, if any, were run. Returns 1 if the sheet was not connected but HsSetVal functions, if any, were run. Returns 2 if sheet had no ad hoc grid but HsSetVal functions, if any, were run. Otherwise, returns the appropriate error code.
**Example**

Declare Function HypSubmitData Lib "HsAddin" (ByVal vtSheetName As Variant) As Long
Sub Example_HypSubmitData()
Worksheets(Empty).range("B2").value = 8023
Worksheets(Empty).range("B2").Select
sts = HypSubmitData(Empty)
End Sub

**HypPivot**

**Data provider types:** Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypPivot() transposes spreadsheet rows and columns, based on the selected dimension.

**Syntax**

HypPivot(vtSheetName, vtStart, vtEnd)
ByVal vtSheetName As Variant
ByVal vtStart As Variant
ByVal vtEnd As Variant

**Parameters**

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtStart: The range object that refers to the single cell starting point of the pivot

vtEnd: The range object that refers to the single cell ending point of the pivot

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Public Declare Function HypPivot Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtStart As Variant, ByVal vtEnd As Variant) As Long
Sub Example_HypPivot()
X=HypPivot(Empty, RANGE("B2"), RANGE("D1"))
If X = 0 Then
    MsgBox("Pivot successful.")
Else
    MsgBox("Pivot failed.")
End If
End Sub
HypPivotToGrid

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypPivotToGrid() moves the selected dimension and members from the POV to the spreadsheet grid.

Syntax
HypPivotToGrid (vtSheetName, vtDimensionName, vtSelection)
ByVal vtSheetName As Variant
ByVal vtDimensionName As Variant
ByVal vtSelection As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtDimensionName: The currently selected dimension from the toolbar
vtSelection: The range object that refers to the single cell starting point of the pivot. Orientation is calculated based on the selection.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypPivotToGrid Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtDimensionName As Variant, ByVal vtSelection As Variant) As Long

Sub Example_PivotGrid()
X = HypPivotToGrid(Empty, "Product", Range("E6"))
If X = 0 Then
    MsgBox ("Pivot to grid successful.")
Else
    MsgBox ("Pivot to grid failed.")
End If
End Sub

HypPivotToPOV

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)
Description

HypPivotToPOV() pivots from the grid to the POV.

Syntax

HypPivotToPOV (vtSheetName, vtSelection)
ByVal vtSheetName As Variant
ByVal vtSelection As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtSelection: The range object that refers to the single cell starting point of the pivot. Orientation is calculated based on the selection.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypPivotToPOV Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelection As Variant) As Long
Sub Example_HypPivotToPOV()
X=HypPivotToPOV(Empty, RANGE("E6"))
If X = 0 Then
    MsgBox("Pivot to POV successful.")
Else
    MsgBox("Pivot to POV failed.")
End If
End Sub

HypKeepOnly

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypKeepOnly() retains only the selected member(s) in the sheet and removes unselected members.

Selection must include only member cells, not data cells.

Syntax

HypKeepOnly(vtSheetName, vtSelection)
ByVal vtSheetName As Variant
ByVal vtSelection As Variant

**Parameters**

**vtSheetName**: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

**vtSelection**: The range object that refers to the member(s) to be kept. If selection is Null or Empty, the active cell is used.

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Examples**

To keep only one member name:

```vba
Public Declare Function HypKeepOnly Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelection As Variant) As Long

Sub Example_HypKeepOnly()
    X=HypKeepOnly(Empty, RANGE("D2"))
    If X = 0 Then
        MsgBox("Keep Only successful.");
    Else
        MsgBox("Keep Only failed." + X)
    End If
End Sub
```

To keep multiple member names:

```vba
Sub Example_HypKeepOnly
    X=HypKeepOnly(Empty, RANGE("D2:A5"))
    If X = 0 Then
        MsgBox("Keep Only successful.");
    Else
        MsgBox("Keep Only failed." + X)
    End If
End Sub
```

**HypRemoveOnly**

**Data provider types**: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypRemoveOnly() removes only the selected member(s) in the worksheet.

Selection must include only member cells, not data cells.
Syntax
HypRemoveOnly(vtSheetName, vtSelection)
ByVal vtSheetName As Variant
ByVal vtSelection As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtSelection: The range object that refers to the member(s) to be removed. If selection is Null or Empty, the active cell is used.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Examples
To remove only one member name:
Public Declare Function HypRemoveOnly Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelection As Variant) As Long
Sub Example_HypRemoveOnly()
X=HypRemoveOnly(Empty, RANGE("D2"))
If X = 0 Then
    MsgBox("Remove Only successful.")
Else
    MsgBox("Remove Only failed." + X)
End If
End Sub

To remove multiple member names:
Public Declare Function HypRemoveOnly Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelection As Variant) As Long
Sub Example_HypRemoveOnly()
X=HypRemoveOnly(Empty, RANGE("D2, A5"))
If X = 0 Then
    MsgBox("Remove Only successful.")
Else
    MsgBox("Remove Only failed." + X)
End If
End Sub

HypZoomIn
Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)
Description
HypZoomIn() retrieves and expands data from Smart View based on the selected members.

Syntax
HypZoomIn(vtSheetName, vtSelection, vtLevel, vtAcross)
ByVal vtSheetName As Variant
ByVal vtSelection As Variant
ByVal vtLevel As Variant
ByVal vtAcross As Variant (not used)

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtSelection: The range object that refers to the members to be zoomed in on. If the selection is Null or Empty, the active cell is used.
vtLevel: The number that indicates the level of the zoom. Available levels:
  ● 0 = Next level
  ● 1 = All levels
  ● 2 = Bottom level
  ● 3 = Siblings (available only for Essbase 11.1.2.1.102 or later connections using Provider Services)
  ● 4 = Same Level (available only for Essbase 11.1.2.1.102 or later connections using Provider Services)
  ● 5 = Same generation (available only for Essbase 11.1.2.1.102 or later connections using Provider Services)
  ● 6 = Formulas (available only for Essbase 11.1.2.1.102 or later connections using Oracle Hyperion Provider Services)

If Null, Empty or an incorrect value is passed, the currently selected option is used.
vtAcross: Not used.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypZoomIn Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelection As Variant, ByVal vtLevel As Variant, ByVal vtAcross As Variant) As Long

Sub Example_HypZoomIn()
    X=HypZoomIn(Empty, RANGE("B3"), 1, FALSE)
If X = 0 Then
    MsgBox("Zoom successful.")
Else
    MsgBox("Zoom failed.")
End If
End Sub

HypZoomOut

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypZoomOut() collapses the view of data based on the selected members.

Syntax
HypZoomOut(vtSheetName, vtSelection)
ByVal vtSheetName As Variant
ByVal vtSelection As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtSelection: The range object that refers to the members to be zoomed out on. If the selection is Null or Empty, the active cell is used.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypZoomOut Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelection As Variant) As Long

Sub Example_HypZoomOut()
X=HypZoomOut(Empty, RANGE("B3"))
If X = 0 Then
    MsgBox("Zoom out successful.")
Else
    MsgBox("Zoom out failed.")
End If
End Sub
Form Functions

In This Chapter

About Forms ................................................................. 91
HypOpenForm ................................................................. 91

About Forms

Forms are grid displays that enable users to enter data into the database and to view and analyze data or related text. In Financial Management, Hyperion Enterprise, forms are called “data forms.”

“HypOpenForm ” on page 91

HypOpenForm

Data provider types: Planning, Financial Management, Hyperion Enterprise

Description

HypOpenForm () opens the specified form.

Syntax

HypOpenForm (vtSheetName, vtFolderPath, vtFormName, vtDimensionList(), vtMemberList())
ByVal vtSheetName As Variant
ByVal vtFolderPath As Variant
ByVal vtFormName As Variant
ByRef vtDimensionList() As Variant
ByRef vtMemberList() As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtFolderPath: The folder path name
vtFormName: The name of the data form

vtDimensionList(): not in use

vtMemberList(): not in use

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypOpenForm Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtFolderPath As Variant, ByVal vtFormName As Variant, ByRef vtDimensionList() As Variant, ByRef vtMemberList() As Variant) As Long

Sub Example_HypOpenForm()
    Dim DimList() As Variant
    Dim MemList() As Variant
    sts = HypOpenForm(Empty, "/Forms/data1", "data1", DimList, MemList)
    MsgBox (sts)
End Sub
About Cell Functions

Cell functions perform operations and retrieve information for data cells and their contents.

HypGetDimMbrsForDataCell

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypGetDimMbrsForDataCell() retrieves the entire set of dimension members for a data cell. These members must be in the grid.

Syntax

HypGetDimMbrsForDataCell (vtSheetName, vtCellRange, vtServerName, vtAppName, vtCubeName, vtFormName, vtDimensionNames, vtMemberNames)
ByVal vtSheetName As Variant
ByVal vtCellRange As Variant
ByRef vtServerName As Variant
ByRef vtAppName As Variant
ByRef vtCubeName As Variant
ByRef vtFormName As Variant
ByRef vtDimensionNames As Variant
ByRef vtMemberNames As Variant

Parameters

vtSheetName: Input variable; the name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtCellRange: Input variable; the range of the cell (one cell only)

vtServerName: Output variable; the name of the server the associated connection on the sheet is connected to

vtAppName: Output variable; the name of the application the associated connection on the sheet is connected to

vtCubeName: Output variable; the name of the cube/database (Plan Type in Planning) the associated connection on the sheet is connected to

vtFormName: Output variable; the name of the form the associated connection on the sheet is connected to (in ad hoc grids, this is returned as an empty string)

vtDimensionNames: Output variable; the array of dimension names

vtMemberNames: Output variable; the array of member names

Return Value

Returns 0 if successful; otherwise, the appropriate error code.

Example

In order to run the example below, the defined sheet in oSheetName must contain a valid grid, and the cell or cell range defined in oSheetDisp.Range must be a valid data cell within a grid.

Public Declare Function HypGetDimMbrsForDataCell Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtCellRange As Variant, ByRef vtServerName As Variant, ByRef vtAppName As Variant, ByRef vtCubeName As Variant, ByRef vtFormName As Variant, ByRef vtDimensionNames As Variant, ByRef vtMemberNames As Variant) As Long

Sub Example_HypGetDimMbrsForDataCell()

Dim oRet As Long
Dim oSheetName As String
Dim oSheetDisp As Worksheet

94  Cell Functions
Dim vtDimNames As Variant
Dim vtMbrNames As Variant
Dim vtServerName As Variant
Dim vtAppName As Variant
Dim vtCubeName As Variant
Dim vtFormName As Variant
Dim lNumDims As Long
Dim lNumMbrs As Long
Dim sPrintMsg As String

oSheetName = "Sheet1"
Set oSheetDisp = Worksheets("Sheet1")

oRet = HypGetDimMbrsForDataCell("", oSheetDisp.Range("valid data cell"), vtServerName, vtAppName, vtCubeName, vtFormName, vtDimNames, vtMbrNames)

If (oRet = SS_OK) Then
    If IsArray(vtDimNames) Then
        lNumDims = UBound(vtDimNames) - LBound(vtDimNames) + 1
    End If

    If IsArray(vtMbrNames) Then
        lNumMbrs = UBound(vtMbrNames) - LBound(vtMbrNames) + 1
    End If

    sPrintMsg = "Number of Dimensions = " & lNumDims & "  Number of Members = " & lNumMbrs & "  Cube Name - " & vtCubeName
    MsgBox (sPrintMsg)
End If

End Sub

---

**HypCell**

**Data provider types:** Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypCell() retrieves a cell value for a single member combination.

**Syntax**

HypCell(vtSheetName, ParamArray MemberList())

ByVal vtSheetName As Variant

ByVal ParamArray MemberList() As Variant

**Parameters**

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

MemberList: A list of strings that describe the member combination for which a data value will be retrieved. If MemberList is Null or Empty, the top level value is used. Represent members as "Dimension#Member"; for example, "Year#Jan" or "Market#East".
Return Value

Returns the value of the data point if successful. Returns #No Connection if the sheet cannot be determined or is not connected to a data provider. Returns “Invalid Member MemberName or dimension DimensionName” if a member is incorrect.

Example

Declare Function HypCell Lib "HsAddin" (ByVal vtSheetName As Variant, ParamArray MemberList() As Variant) As Variant

Sub Example_HypCell()
Dim X As String
X=HypCell(Empty, "Year#Qtr1", "Scenario#Actual", "Market#Oregon")
  If X = "#No Connection" Then
    MsgBox("Not logged in, or sheet not active.")
  Else
    If Left(X, 15) = "#Invalid member" then
      MsgBox("Member name incorrect.")
    Else
      MsgBox(X + " Value retrieved successfully.")
    End If
  End If
End Sub

Note: The value of the data point returned is not placed in a cell in the spreadsheet automatically. To place the value in a cell, use the Visual Basic select method and the ActiveCell property. See your Visual Basic documentation for more information.

HypFreeDataPoint

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypFreeDataPoint() frees any memory allocated by HypGetDataPoint.

Syntax

Syntax

HypFreeDataPoint(vtInfo)
ByRef vtInfo As Variant

Parameters

vtInfo: The variant array returned by HypGetDataPoint

Return Value

Returns 0 if successful; returns -15 ("Invalid Parameter") if not successful.
HypGetCellRangeForMbrCombination

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypGetCellRangeForMbrCombination() retrieves the cell range for the selected combination of members.

Syntax
HypGetCellRangeForMbrCombination (vtSheetName, vtDimNames, vtMbrNames, vtCellIntersectionRange)

By Val vtSheetName As Variant
ByRef vtDimNames As Variant
ByRef vtMbrNames As Variant
ByRef vtCellIntersectionRange As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtDimNames: Input variable; the array of dimension names
vtMbrNames: Input variable; the array of member names corresponding to the dimensions (in the same order)
vtCellIntersectionRange: Output variable; the range of the cell(s) on the grid

Return Value
Returns SS_OK if successful; otherwise, the appropriate error code.

Example

Public Declare Function HypGetCellRangeForMbrCombination Lib "HsAddin" (ByVal vtSheetName As Variant, ByRef vtDimNames() As Variant, ByRef vtMbrNames() As Variant, ByRef vtCellIntersectionRange As Variant) As Long
Sub Example_HypGetCellRangeForMbrCombination()
    Dim oRet As Long
    Dim oSheetName As String
    Dim oSheetDisp As Worksheet
    Dim vtDimNames(3) As Variant
    Dim vtMbrNames(3) As Variant
    Dim vtReturnCellRange As Variant

Dim oRange As Range

' oSheetName = Empty
' Set oSheetDisp = Worksheets(oSheetName$)

vtDimNames(0) = "Measures"
vtdimNames(1) = "Market"
vtdimNames(2) = "Year"
vtdimNames(3) = "Product"
vtdimNames(4) = ""

vtMbrNames(0) = "Sales"
vtmbrNames(1) = "New York"
vtmbrNames(2) = "Year"
vtmbrNames(3) = " Product"
vtmbrNames(4) = ""

oRet = HypGetCellRangeForMbrCombination ("", vtDimNames, vtMbrNames, vtReturnCellRange)

If (oRet = 0) Then
    Set oRange = vtReturnCellRange
End If
End Sub

**HypGetDataPoint**

**Data provider types:** Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypGetDataPoint() retrieves member information for a single data cell. For example, to find out the members that consist of the data intersection at cell B6, HypGetDataPoint may return the members January, California, Actual, Root Beer, Profit.

**Syntax**

HypGetDataPoint (vtSheetName, vtCell)

By Val vtSheetName As Variant

By Val vtCell As Variant

**Parameters**

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtCell: The reference cell for which to retrieve the member combination information

**Return Value**

Returns an array of member names.
Example

Declare Function HypGetDataPoint Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtCell As Variant) As Variant

Sub Example_HypGetDataPoint()
Dim vt As Variant
Dim cbItems As Variant
Dim i As Integer
Dim pMember As String
vt = HypGetDataPoint(Empty, range("B3"))
If IsArray(vt) Then
    cbItems = UBound(vt) - LBound(vt) + 1
    MsgBox("Number of elements = " + Str(cbItems))
    For i = LBound(vt) To UBound(vt)
        MsgBox("Member = " + vt(i))
    Next
    X = HypFreeDataPoint(vt)
Else
    MsgBox("Return Value = " + Str(vt))
End If
End Sub

HypIsCellWritable

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypIsCellWritable() checks to see whether a cell is writable.

Syntax
HypIsCellWritable (vtSheetName, vtCellRange)
ByVal vtSheetName As Variant
ByVal vtCellRange As Variant

Parameters
vtSheetName: Input parameter; the name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtCellRange: Output parameter; the range of the cell (one cell only) whose writability is to be checked

Return Value
Returns VARIANT_TRUE if the cell is writable; otherwise, VARIANT_FALSE.
Example

Public Declare Function HypIsCellWritable Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtCellRange As Variant) As Boolean

Sub Example_HypIsCellWritable()
    Dim oRet As Boolean
    Dim oSheetName As String
    Dim oSheetDisp As Worksheet
    oSheetName = "Sheet1"
    Set oSheetDisp = Worksheets(oSheetName$)
    oRet = HypIsCellWritable (Empty, oSheetDisp.Range("G2"))
End Sub

**HypSetCellsDirty**

*Data provider types:* Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypSetCellsDirty() marks selected data range dirty for submitting data.

**Syntax**

HypSetCellsDirty (vtSheetName, vtRange)

*ByVal vtSheetName As Variant*

*ByVal vtRange As Variant*

**Parameters**

*vtSheetName:* The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

*vtRange:* Variant data range to be marked as dirty

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypSetCellsDirty Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long

Sub Example_HypSetCellsDirty()
    X=HypSetCellsDirty (Empty, Range("A3:B3"))
End Sub
**HypDeleteAllLROs**

Data provider types: Essbase

Description

HypDeleteAllLROs() deletes all linked reporting objects from the cells specified by the vtSelectionRange parameter.

Syntax

HypDeleteAllLROs (vtSheetName, vtSelectionRange)

ByVal vtSheetName As Variant
ByVal vtSelectionRange As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtSelectionRange: The range of cells from which to delete all linked reporting objects

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypDeleteAllLROs Lib "HsAddIn" (ByVal vtSheetName As Variant, ByVal vtSelectionRange As Variant) As Long

Sub Example_HypDeleteAllLROs
   sts = HypDeleteAllLROs("Sheet1", Range("B3"))
End Sub

**HypDeleteLROs**

Description

HypDeleteLROs() deletes one or more linked reporting objects from the cells specified by the vtSelectionRange parameter.

Syntax

HypDeleteLROs (vtSheetName, vtSelectionRange, vtLROIDs())

ByVal vtSheetName As Variant
ByVal vtSelectionRange As Variant
ByRef vtLROIDs() As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtSelectionRange: Input variable; the range of cells from which to delete all linked reporting objects

vtLROIDs(): Input variable; the array of LRO Ids to be deleted

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypDeleteLROs Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelectionRange As Variant, ByRef vtLROIDs() As Variant) As Long

Sub Example_HypDeleteLROs()
    Dim LROIDs(1)
    LROIDs(0) = 1
    LROIDs(1) = 2
    sts = HypDeleteLROs("Sheet1", Range("B3"), LROIDs)
End Sub

HypAddLRO

Data provider types: Essbase

Description

HypAddLRO() adds linked reporting objects to the cells specified by the vtSelectionRange parameter. To see the added linked reporting objects, you must launch the Linked Reporting Objects dialog box or use HypListLRO.

Syntax

HypAddLRO(vtSheetName, vtSelectionRange, vtLType, vtName, vtDescription)

ByVal vtSheetName As Variant

ByVal vtSelectionRange As Variant

ByVal vtLType As Variant

ByVal vtName As Variant

ByVal vtDescription As Variant
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtSelectionRange: Input parameter; the range of cells to associate with the linked reporting object

vtlType: Input parameter; the linked reporting object type expressed as a constant

- 1—Cell note
- 2—File
- 3—URL

vtName: Input variable; the location of the file with filename and URL information. Not used for cell note.

vtDescription: Input variable; the description of the cell note, file, or URL

Return Value

Returns 0 if successful; otherwise, returns the corresponding error code.

Example

Public Declare Function HypAddLRO Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelectionRange As Variant, ByVal vtlType As Variant, ByVal vtName As Variant, ByVal vtDescription As Variant, ByRef vtLROIDs() As Variant) As Long
Sub Example_HypAddLRO()
  stx = HypAddLRO("Sheet1", Range("B3"), 1, ", " , "Hello World")
End Sub

HypUpdateLRO

Description

HypUpdateLRO() updates linked reporting objects associated with the cells specified by the vtSelectionRange parameter. To see the updates, you must launch the Linked Reporting Objects dialog box or use HypListLRO.

Syntax

HypUpdateLRO(vtSheetName, vtSelectionRange, vtID, vtlType, vtName, vtDescription)
ByVal vtSheetName As Variant
ByVal vtSelectionRange As Variant
ByVal vtID As Variant
ByVal vtlType As Variant
ByVal vtName As Variant
ByVal vtDescription As Variant
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtSelectionRange: Input variable; the range of cells to associate with the linked reporting object

vtID: Input variable; the ID of the linked reporting object to be updated

vtType: Input variable; the linked reporting object type expressed as a constant

- 1—Cell note
- 2—File
- 3—URL

vtName: Input variable; the location of the file with filename and URL information. Not used for cell note.

vtDescription: Input variable; the description of the cell note, file, or URL

Return Value

Returns 0 if successful; otherwise, returns the corresponding error code.

Example

Public Declare Function HypUpdateLRO Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelectionRange As Variant, ByVal vtID As Variant, ByVal vtType As Variant, ByVal vtName As Variant, ByVal vtDescription As Variant) As Long

Sub Example_HypUpdateLRO
sts = HypUpdateLRO("Sheet1", Range("B3"), "2", 2, "d:\test2.txt", "linked object")
End Sub

HypListLROs

Data provider types: Essbase

Description

HypListLROs() lists all linked reporting objects associated with the cells specified by the vtSelectionRange parameter.

Syntax

HypListLROs (vtSheetName, vtSelectionRange, vtLRO)

ByVal vtSheetName As Variant

ByVal vtSelectionRange As Variant

ByRef vtLRO As LRO_Info
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtSelectionRange: Input variable; the range of cells from which to list all linked reporting objects

vtLRO: Output variable; the 2-dimensional array of linked reporting objects

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypListLROs Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelectionRange As Variant, ByRef vtLRO) As Long

Dim ObjectList As LRO_Info
Sub Example_HypListLROs()
    sts = HypListLROs("Sheet1", Range("B3"), ObjectList)
End Sub

HypRetrieveLRO

Data provider types: Essbase

Description

HypRetrieveLRO() retrieves linked reporting objects associated with the cells specified by the vtSelectionRange parameter. To see the linked reporting objects, you must launch the Linked Reporting Objects dialog box or use HypListLRO.

Syntax

HypRetrieveLRO(vtSheetName, vtSelectionRange, vtID, vtType, vtName, vtDescription)

ByVal vtSheetName As Variant
ByVal vtSelectionRange As Variant
ByVal vtID As Variant
ByVal vtName As Variant
ByVal vtDescription As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtSelectionRange: Input variable; the range of cells to associate with the linked reporting object

vtID: Input variable; the ID of the linked reporting object to be retrieved. This is provided when you execute HypListLROs.
vtName: Output variable; the name of the linked reporting object

vtDescription: Output variable; the description of the retrieved linked reporting object

**Return Value**

Returns 0 if successful; otherwise, returns the corresponding error code.

**Example**

```vba
Public Declare Function HypRetrieveLRO Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelectionRange As Variant, ByVal vtID, ByRef vtName, ByRef vtDescription) As Long

Sub Example_HypRetrieveLRO
    sts = HypRetrieveLRO("Sheet1", Range("B3"), "1", vtName, vtDescription)
End Sub
```

**HypExecuteDrillThroughReport**

**Data provider types:** Essbase

**Description**

HypExecuteDrillThroughReport() executes the specified drill-through report. See also “HypGetDrillThroughReports” on page 107.

**Syntax**

HypExecuteDrillThroughReport(vtSheetName, vtSelectionRange, vtID, vtName, vtURL, vtURLTemplate, vtType)

- ByVal vtSheetName As Variant
- ByVal vtSelectionRange As Variant
- ByVal vtID As Variant
- ByVal vtName As Variant
- ByVal vtURL As Variant
- ByVal vtURLTemplate As Variant
- ByVal vtType As Variant

**Parameters**

- **vtSheetName**: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

- **vtSelectionRange**: Input variable; the range of cells in which to execute the drill-through report

- **vtID**: Input variable; the ID for the execution of the drill-through report. This is returned from the server when you run HypGetDrillThroughReports.
vtName: Input variable; the name of the drill-through report. This is returned from the server when you run HypGetDrillThroughReports.

vtURL: Input variable; the URL of the drill-through report. This is returned from the server when you run HypGetDrillThroughReports.

vtURLTemplate: Input variable; the URL template of the drill-through report. This is returned from the server when you run HypGetDrillThroughReports.

vtType: Input variable; the type of the drill-through report. This is returned from the server when you run HypGetDrillThroughReports.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypExecuteDrillThroughReport Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelectionRange As Variant, ByVal vtID As Variant, ByVal vtName As Variant, ByVal vtURL As Variant, ByVal vtURLTemplate As Variant, ByVal vtType As Variant) As Long

Sub Example_HypExecuteDrillThroughReport()
sts = HypExecuteDrillThroughReport("Sheet3", Range("B3"), ids(0), names(0), ", ", ", ", ")
End Sub

HypGetDrillThroughReports

Data provider types: Essbase

Description

HypGetDrillThroughReports() retrieves a list of drill-through reports. See also “HypExecuteDrillThroughReport” on page 106.

Syntax

HypGetDrillThroughReports(vtSheetName, vtSelectionRange, vtIDs, vtNames, vtURLs, vtURLTemplates, vtTypes)
ByVal vtSheetName As Variant
ByVal vtSelectionRange As Variant
ByVal vtIDs As Variant
ByVal vtNames As Variant
ByVal vtURLs As Variant
ByVal vtURLTemplates As Variant
ByVal vtTypes As Variant
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtSelectionRange: The range of cells that contain the drill-through reports to retrieve

vtIDs: Output variable; the array of the IDs returned from the server

vtNames: Output variable; the array of the names returned from the server

vtURLs: Output variable; the array of the URLs returned from the server

vtURLTemplates: Output variable; the array of the URL templates returned from the server

vtTypes: Output variable; the array of the types returned from the server

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypGetDrillThroughReports Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtSelectionRange As Variant, ByRef vtIDs As Variant, ByRef vtNames As Variant, ByRef vtURLs As Variant, ByRef vtURLTemplates As Variant, ByRef vtTypes As Variant) As Long

Sub Example_HypGetDrillThroughReports()
    sts = HypGetDrillThroughReports("Sheet3", Range("B3"), ids, names, urls, urltemplates, types)
End Sub
POV Functions

In This Chapter

About POV Functions ................................................................. 109
HypSetPOV.................................................................................. 109
HypGetBackgroundPOV.............................................................. 110
HypSetBackgroundPOV............................................................... 111
HypGetPagePOVChoices.............................................................. 111
HypSetPages .............................................................................. 112
HypGetMembers ....................................................................... 113
HypSetMembers......................................................................... 114
HypSetActiveMember................................................................. 115
HypSetActiveMember................................................................. 116
HypGetDimensions.................................................................... 116
HypSetDimensions.................................................................... 117

About POV Functions

POV functions specify or retrieve settings for the POV.

HypSetPOV

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypSetPOV() sets the POV for the selected ad hoc worksheet. This function does not support data forms; for forms, use HypSetPages instead (see “HypSetPages” on page 112).

Syntax

HypSetPOV(vtSheetName, ParamArray MemberList())
ByVal vtSheetName As Variant
ParamArray MemberList() As Variant
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

ParamArray MemberList(): A list of strings that describe the member combination for which a data value will be retrieved. If MemberList is null or empty, the top level value is used.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code. If you use this function on a form instead of an ad hoc worksheet, error -69 (deprecated VBA) is returned.

Example

Declare Function HypSetPOV Lib "HsAddin" (ByVal vtSheetName As Variant, ParamArray MemberList() As Variant) As Long
Sub Example_HypSetPOV()
    X=HypSetPOV (Empty,"Year#Qtr1", "Market#East")
End Sub

HypGetBackgroundPOV

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypGetBackgroundPOV() returns the list of background POV members as two-string arrays. One string array contains the POV dimension names; the other contains the member names.

Syntax

HypGetBackgroundPOV (vtFriendlyName, vtDimensionNames, vtMemberNames)
ByVal vtFriendlyName As Variant
ByRef vtDimensionNames As Variant
ByRef vtMemberNames As Variant

Parameters

vtFriendlyName: Input variable; the connection name of the data provide.
vtDimensionNames: Output variable; the dimension names array
vtMemberNames: Output variable; the member names array (one member per POV dimension)

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

This example assumes that the worksheet is connected and has a grid.
Public Declare Function HypGetBackgroundPOV Lib "HsAddin" (ByVal vtFriendlyName As Variant, ByRef vtDimensionNames As Variant, ByRef vtMemberNames As Variant) As Long
Sub Example_GetBackgroundPOV()
sts = con = HypGetBackgroundPOV("stm10026_Sample_Basic", vtDim, vtMem)
End Sub

**HypSetBackgroundPOV**

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypSetBackgroundPOV() sets the POV for the connection object in the POV Manager.

**Syntax**

HypSetBackgroundPOV(vtFriendlyName, ParamArray MemberList())
ByVal vtFriendlyName As Variant
ParamArray MemberList() As Variant

**Parameters**

vtFriendlyName: The connection name of the data provider.

MemberList: A list of strings that describe the member combination for which a data value will be retrieved. If MemberList is Null or Empty, the top level HypSetDimensions value is used.

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypSetBackgroundPOV Lib "HsAddin" (ByVal vtFriendlyName, ParamArray MemberList() As Variant) As Long
Sub Example_HypSetBackgroundPOV()
    X=HypSetBackgroundPOV ("My Connection","Year#Qtr1", "Market#East")
End Sub

**HypGetPagePOVChoices**

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypGetPagePOVChoices() returns the available member names and member description for a given dimension.
Syntax

HypGetPagePOVChoices(vtSheetName, vtDimensionName, vtMbrNameChoices, 
vtMbrDescChoices)

ByVal vtSheetName As Variant
ByVal vtDimensionName As Variant
ByRef vtMbrNameChoices As Variant
ByRef vtMbrDescChoices As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtDimensionName: The dimension names in the POV

vtMbrNameChoices: Output parameter; the array of member names

vtMbrDescChoices: Output parameter; the array of member descriptions

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypGetPagePOVChoices Lib "HsAddin" (ByVal vtSheetName As 
Variant, ByVal vtDimensionName As Variant, ByRef vtMbrNameChoices As Variant, ByRef 
vtMbrDescChoices As Variant) As Long
Sub Example_HypGetPagePOVChoices()
  Dim mbrName As Variant
  Dim mbrDesc As Variant
  sts = HypGetPagePOVChoices(Empty, "Product", vtMbrNameChoices, vtMbrDescChoices)
  MsgBox (sts)
End Sub

HypSetPages

Data provider types: Planning (forms only), Financial Management (forms only), Hyperion Enterprise (forms only)

Description

HypSetPages() sets the page members for the selected sheet.

Syntax

HypSetPages (ByVal vtSheetName, ParamArray MemberList())

ByVal vtSheetName As Variant
ParamArray MemberList() As Variant
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

ParamArray MemberList(): The list of desired page member items in the form Dimension#Current Member. If MemberList is Null or Empty, the top level value is used.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypSetPages Lib "HsAddin" (ByVal vtSheetName As Variant, ParamArray MemberList() As Variant) As Long

Sub Example_HypSetPages()
    X=HypSetPages (Empty,"Entity#Operations","Scenario#Current")
End Sub

HypGetMembers

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypGetMembers() gets the list of selected or used members for a given dimension present in the grid.

For Essbase and Planning, member names are based on the selected alias table.

For Financial Management, the second array returns the descriptions.

For POV (in forms), Page (in ad hoc) and user variables, a single member is returned.

To uniquely identify the user variable, provide the user variable name rather than the dimension name.

Syntax

HypGetMembers (vtSheetName, vtDimensionName, vtMbrNameChoices, vtMbrDescChoices)

ByVal vtSheetName As Variant
ByVal vtDimensionName As Variant
ByRef vtMbrNameChoices As Variant
ByRef vtMbrDescChoices As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
**vtDimensionName**: Input variable; the name of the dimension for which the selected member list is to be returned

**vtMbrNameChoices**: Output variable; the array of member names used

**vtMbrDescChoices**: Output variable; the array of member name descriptions. For Essbase and Planning, this is the same as member names. This list will be empty if the dimension is a row or column dimension.

**Return Value**
Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**
This example assumes that the worksheet is connected and has a grid.

```vba
Public Declare Function HypGetMembers Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtDimensionName As Variant, ByRef vtMbrNameChoices As Variant, ByRef vtMbrDescChoices As Variant) As Long
Sub Example_HypGetMembers()
    sts = HypGetMembers("Sheet1", "Year", vtMbr, vtDes)
End Sub
```

# HypSetMembers

**Data provider types**: Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**
HypSetMembers() sets the list of POV dimension choices in ad hoc grids and the Page list in Financial Management forms.

This function cannot be used to set the Page list in Planning forms, nor can it be used to set row or column members.

The member list submitted by the user is validated before it is set.

**Syntax**
HypSetMembers (vtSheetName, vtDimensionName, ParamArray MemberList())

ByVal vtSheetName As Variant
ByVal vtDimensionName As Variant
ParamArray MemberList() As Variant

**Parameters**

**vtSheetName**: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

**vtDimensionName**: Input variable; the name of the dimension for which the selected member list is to be set
**MemberList**: Input variable; the array of member names to be set as choices

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

This example assumes that the worksheet is connected and has a grid. Note: “InvalidMember” does not belong to the Entity dimension and therefore will not be included in the list of dimension choices.

```vbscript
Public Declare Function HypSetMembers Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtDimensionName As Variant, ParamArray MemberList() As Variant) As Long
Sub Example_HypSetMembers()
  sts = HypSetMembers("Sheet1", "Entity", "Regional", "InvalidMember", "None")
End Sub
```

---

**HypGetActiveMember**

**Data provider types**: Essbase, Planning, Financial Management, Hyperion Enterprise

**Description**

HypGetActiveMember () returns the active member name of the given dimension. The active member for page dimensions, POV dimensions, and user variables can be retrieved on ad hoc or form worksheets. Row and column dimensions are not returned.

**Syntax**

HypGetActiveMember (vtDimName, vtMember)

*ByVal vtDimName As Variant*

*ByRef vtMember As Variant*

**Parameters**

**vtDimName**: Input variable; the dimension name whose active member is to be retrieved

**vtMember**: Output variable; the active member name returned

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

This example assumes that the worksheet is connected and has a grid.

```vbscript
Public Declare Function HypGetActiveMember Lib "HsAddin" (ByVal vtDimName As Variant, ByRef vtMember As Variant) As Long
Sub Example_GetActiveMember()
  sts = HypGetActiveMember("Market", vtMem)
End Sub
```
HypSetActiveMember

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypSetActiveMember() sets the active member for a given dimension: page, POV, and user variables. Does not apply to row and column dimensions.

Syntax

HypSetActiveMember (vtDimName, vtMember)
ByVal vtDimName As Variant
ByVal vtMember As Variant

Parameters

vtDimName: Input variable; the dimension name whose active member is to be changed or set
vtMember: Input variable; the active member to be set

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

This example assumes that the worksheet is connected and has a grid.

Public Declare Function HypSetActiveMember Lib "HsAddin" (ByVal vtDimName As Variant, ByVal vtMember As Variant) As Long
Sub Example_HypSetActiveMember()
sts = HypSetActiveMember("Market", "Washington")
End Sub

HypGetDimensions

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypGetDimensions() returns an array containing the dimension names in the grid and an array containing their corresponding types.

Type array has five possible types (row, column, page, POV, user variable), which can be identified using the following enumeration:

Enum DIMENSION_TYPE
    ROW_DIM = 0
    COL = 1
    POV = 2
    PAGE = 3
End Enum
To uniquely identify the user variable, use the user variable name rather than the dimension name.

Syntax

HypGetDimensions (vtSheetName, vtMemberNames, vtType)
ByVal vtSheetName As Variant
ByRef vtMemberNames As Variant
ByRef vtType As Variant

Parameters

vtSheetName: Input parameter; the name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtMemberNames: Output variable; the dimension name array present in the grid

vtType: Output variable; the type information for the respective dimension

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

This example assumes that the worksheet is connected and has a grid.

Public Declare Function HypGetDimensions Lib "HsAddin" (ByVal vtSheetName As Variant, ByRef vtMemberNames As Variant, ByRef vtType As Variant) As Long
Sub Example_GetDimensions()
sts = HypGetDimensions("Sheet1", vtDim, vtType)
End Sub

HypSetDimensions

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypSetDimensions() specifies an ad hoc grid layout other than the default grid by rearranging the metadata of the grid. In this function, you specify an array containing the dimension names in the grid and an array containing their corresponding types.

If HypSetDimensions() is used on an existing ad hoc report, the entire grid layout is rearranged, and comments, formulas, and formatting are lost.
Syntax
HypSetDimensions(vtSheetName, vtDimNames(), vtType())
ByVal vtSheetName() As Variant
ByRef vtDimNames() As Variant
ByRef vtType() As Variant)

Parameters
vtSheetName: Input parameter; the name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtDimNames(): Input parameter; the dimension name array present in the grid
vtType(): Input parameter; the type information for the respective dimension. Possible values:
  • Row dimension (ROW_DIM) = 0
  • Column (COL) = 1
  • POV (POV) = 2
  • Page dimension (PAGE) = 3
  • User variable (USERVAR) = 5

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
This example assumes that the worksheet is connected.
Public Declare Function HypSetDimensions Lib "HsAddin" (ByVal vtSheetName() As Variant, ByRef vtDimNames() As Variant, ByRef vtType() As Variant) As Long
Sub Example_HypSetDimensions()
    Dim dims(3) As Variant
    Dim types(3) As Variant
    dims(0) = "Product"
    dims(1) = "Market"
    dims(2) = "Scenario"
    dims(3) = "Measures"
    types(0) = ROW_DIM
    types(1) = COL
    types(2) = POV
    types(3) = POV
    sts = HypSetDimensions("Sheet2", dims, types)
End Sub
Calculation Script and Business Rule Functions

About Calculation Script and Business Rule Functions
Calculation script and business rule functions retrieve or execute calculation scripts and business rules.

HypListCalcScripts

Data provider types: Essbase

Description
HypListCalcScripts() lists all calculation scripts present on an Essbase server.

Syntax
HypListCalcScripts (vtSheetName, vtScriptArray)
ByVal vtSheetName As Variant
ByRef vtScriptArray As Variant

Parameters
vtSheetName: Input parameter; the name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtScriptArray: Output parameter; the array of business rule scripts

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.
Example

Declare Function HypListCalcScripts Lib "HsAddin" (ByVal vtSheetName As Variant, ByRef vtScriptArray As Variant) As Long
Sub Example_HypListCalcScripts()
Dim sts As Long
Dim paramList As Variant
sts = HypListCalcScripts(Empty, paramList)
If IsArray(paramList) Then
    cbItems = UBound(paramList) - LBound(paramList) + 1
    MsgBox ("Number of elements = " + Str(cbItems))
    For i = LBound(paramList) To UBound(paramList)
        MsgBox ("Member = " + paramList(i))
    Next
Else
    MsgBox ("Return Value = " + sts)
End If
End Sub

HypExecuteCalcScript

Data provider types: Essbase

Description

HypExecuteCalcScript() uses a calculation script (business rule script) to initiate a calculation on the server.

Syntax

HypExecuteCalcScript (vtSheetName, vtCalcScript, vtSynchronous)

ByVal vtSheetName As Variant
ByVal vtCalcScript As Variant
ByVal vtSynchronous As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtCalcScript: The name of the calculation script on the server in the database directory to run. To run the default calculation script, use Default.

vtSynchronous: Not used

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypExecuteCalcScript Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtCalcScript As Variant, ByVal vtSynchronous As Variant) As Long
Sub Example_HypExecuteCalcScript()
    X = HypExecuteCalcScript (Empty, "Default", False)
    If X = 0 Then
        MsgBox("Calculation complete.")
    Else
        MsgBox("Calculation failed.")
    End If
End Sub

HypListCalcScriptsEx

Data provider types: Essbase, Planning

Description

HypListCalcScriptsEx() lists all business rules.

Note: See Usage under HypExecuteCalcScriptsEx for more information.

Syntax

HypListCalcScriptsEx (vtSheetName, vtbRuleOnForm, vtCubeNames, vtBRNames,
vtBRTypes, vtBRHasPrompts, vtBRNeedsPageInfo, vtBRHidePrompts)

ByVal vtSheetName As Variant
ByVal vtbRuleOnForm As Variant
ByRef vtCubeNames As Variant
ByRef vtBRNames As Variant
ByRef vtBRTypes As Variant
ByRef vtBRHasPrompts As Variant
ByRef vtBRNeedsPageInfo As Variant
ByRef vtBRHidePrompts As Variant

Parameters

vtSheetName: Input parameter; the name of worksheet on which to run the function. If
vtSheetName is Null or Empty, the active worksheet is used.

vtbRuleOnForm: Input parameter; the boolean to indicate whether to list business rules
associated only with the form opened on the sheet. If set to False, all business rules associated
with the application are returned.

vtCubeNames: Output parameter; the array of cube names (plan types in Planning) associated
with the business rules

vtBRNames: Output parameter; the array of business rule names
vtBRTypes: Output parameter; the array of business rule types

vtBRHasPrompts: Output parameter; the array of Booleans that indicate whether the business rule has runtime prompts (RTP)

vtBRNeedsPageInfo: Output parameter; the array of Booleans that indicate whether the business rule requires Page Information to be run on the sheet

vtBRHidePrompts: Output parameter; the array of Booleans that indicate whether the RTPs for the business rule are hidden

Return Value

Returns 0 if successful; otherwise, the appropriate error code.

Example

Public Declare Function HypListCalcScriptsEx Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtbRuleOnForm As Variant, ByRef vtCubeNames As Variant, ByRef vtBRNames As Variant, ByRef vtBRTypes As Variant, ByRef vtBRHasPrompts As Variant, ByRef vtBRNeedPageInfo As Variant, ByRef vtBRHidePrompts As Variant) As Long
Sub RunListCalcScriptsEx()
sts = HypListCalcScriptsEx(Empty, True, CubeName, BRNames, BRTypes, BRHasPrompts, BRNeedPageInfo, BRHidePrompts)
End Sub

HypExecuteCalcScriptEx

Data provider types: Essbase, Planning

Description

HypExecuteCalcScriptEx() executes the selected business rule.

Syntax

HypExecuteCalcScriptEx(vtSheetName, vtCubeName, vtBRName, vtBRTypes, vtbBRHasPrompts, vtbBRNeedsPageInfo, vtRTPNames(), vtRTPValues(), vtbShowRTPDlg, vtbRuleOnForm, vtbBRRanSuccessfully, vtCubeName, vtBRName, vtBRTypes, vtbBRHasPrompts, vtbBRNeedsPageInfo, vtbBRHidePrompts, vtRTPNamesUsed, vtRTPValuesUsed)

ByVal vtSheetName As Variant
ByVal vtCubeName As Variant
ByVal vtBRName As Variant
ByVal vtBRTypes As Variant
ByVal vtbBRHasPrompts As Variant
ByVal vtbBRNeedPageInfo As Variant
ByVal vtRTPNames() As Variant
ByVal vtRTPValues() As Variant
ByVal vtbShowRTPDlg As Variant
ByVal vtbRuleOnForm As Variant
ByVal vtbBRRanSuccessfully As Variant
ByRef vtRTPNamesUsed As Variant
ByRef vtRTPValuesUsed As Variant
ByRef vtRTPValues() As Variant
ByVal vtbShowRTPDlg As Variant
ByVal vtbRuleOnForm As Variant
ByRef vtbBRRanSuccessfully As Variant
ByRef vtCubeName As Variant
ByRef vtBRName As Variant
ByRef vtBRType As Variant
ByRef vtbBRHasPrompts As Variant
ByRef vtbBRNeedPageInfo As Variant
ByRef vtbBRHidePrompts As Variant
ByRef vtRTPNamesUsed As Variant
ByRef vtRTPValuesUsed As Variant

Parameters

vtSheetName: Input parameter; the name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtCubeName: Input parameter; the cube name (plan type in Planning) associated with the business rule

vtBRName: Input parameter; the name of the business rule to be run

vtBRType: Input parameter; the type of business rule to be run

vtbBRHasPrompts: Input parameter; the Boolean that indicates whether the business rule has RTPs

vtbNeedPageInfo: Input parameter; the Boolean that indicates whether the business rule requires Page Information to be run (this information is either from HypListCalcScriptsEx or from a prior run of HypExecuteCalcScriptEx)

vtRTPNames: Input parameter; the array of RTP names associated with the business rule

vtRTPValues: Input parameter; the array of RTP values corresponding to the RTP names

vtbShowBRDlg: Input parameter; the Boolean that indicates whether to display the Business Rules dialog to let users select the business rule (True) or to execute the business rule automatically (False). If set to True, all input parameters related to the business rule are ignored. Recommendation: Set to True when running the business rule for the first time, and thereafter set to false to automate the execution of the same business rule.

vtbRuleOnForm: Input parameter; the Boolean that indicates whether the business rule is to be associated to the form open on active sheet

vtbBRRanSuccessfully: Output parameter; the Boolean value that indicates whether the last business rule ran successfully
vtCubeName: Output parameter; the cube name (plan types in Planning) associated with the last run business rule

vtBRName: Output parameter; the name of the last run business rule

vtBRTyp: Output parameter; the type of the last run business rule

vtbBRHasPrompts: Output parameter; the Boolean that indicates whether the last run business rule has RTPs

vtbBRNeedPageInfo: Output parameter; the Boolean that indicates whether the last run business rule requires Page information to be run

vtbBRHidePrompts: Output parameter; the Boolean that indicates whether the last run business rule has hidden RTPs

vtRTPNames: Output parameter; the array of RTP names used to run last run business rule

vtRTPValues: Output parameter; the array of RTP values associated with RTP names used to run last run business rule

Return Value

Returns 0 if successful; otherwise, the appropriate error code.

Example

Public Declare Function HypExecuteCalcScriptEx Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtCubeName As Variant, ByVal vtBRName As Variant, ByVal vtBRTyp As Variant, ByVal vtBHasPrompts As Variant, ByVal vtBNeedPageInfo As Variant, ByRef vtRTPNames() As Variant, ByRef vtRTPValues() As Variant, ByVal vtBShowRTPDlg As Variant, ByVal vtBRuleOnForm As Variant, ByRef vtBRRanSuccessfully As Variant, ByRef vtCubeName As Variant, ByRef vtCubeName As Variant, ByRef vtBHasPrompts As Variant, ByRef vtBNeedPageInfo As Variant, ByRef vtBHidePrompts As Variant, ByRef vtRTPNamesUsed As Variant, ByRef vtRTPValuesUsed As Variant) As Long

Sub Example_HypExecuteCalcScriptEx()

    Dim oRet As Long
    Dim oSheetName As String Dim oSheet As Worksheet
    Dim vtCubeNames As Variant
    Dim vtBRNames As Variant
    Dim vtBRTypes As Variant
    Dim vtBHasPrompts As Variant
    Dim vtBNeedsPageInfo As Variant
    Dim vtBHidePrompts As Variant
    Dim sAllCalcs As String
    Dim sCalcName As String
    Dim bNeedPageInfo As Variant
    Dim vtInRTPNames() As Variant
    Dim vtInRTPValues() As Variant
    Dim vtOutRTPNames As Variant
    Dim vtOutRTPValues As Variant
    Dim vtBRRanSuccessfully As Variant
    Dim vtBRRanSuccessfully2 As Variant
    Dim vtOutCubeName As Variant
    Dim vtOutBRName As Variant
Dim vtOutBRType As Variant
Dim bBRHasPrompts As Variant
Dim bBRNeedPageInfo As Variant
Dim bBRHidePrompts As Variant
Dim bShowDlg As Variant
Dim bRuleOnForm As Variant

'Set oSheet = ActiveSheet
'oSheetName = oSheet.Name
oSheetName = "Sheet3"

oRet = HypListCalcScriptsEx (oSheetName, False, vtCubeNames, vtBRNames, vtBRTypes,
vtBRHasPrompts, vtBRNeedsPageInfo, vtBRHidePrompts)
If (oRet = 0) Then
    If IsArray(vtBRNames) Then
        lNumMbrs = (UBound(vtBRNames) - LBound(vtBRNames) + 1)
    End If

    sPrintMsg = "Number of Calc Scripts = " & lNumMbrs
    MsgBox (sPrintMsg)

'Start Executing the Calc Script

    bShowDlg = True
    bRuleOnForm = False
    iScript = 1

    oRet = HypExecuteCalcScriptEx (oSheetName, vtCubeNames(iScript), vtBRNames(iScript),
    vtBRTypes(iScript), vtBRHasPrompts(iScript), vtBRNeedsPageInfo(iScript), vtInRTPNames,
    vtInRTPValues, bShowDlg, bRuleOnForm, vtbBRRanSuccessfully, vtOutCubeName, vtOutBRName,
    vtOutBRType, bBRHasPrompts, bBRNeedPageInfo, bBRHidePrompts, vtOutRTPNames,
    vtOutRTPValues)
    If (oRet = 0) Then
        MsgBox ("Last BR ran successfully - " & vtbBRRanSuccessfully)

        If (vtbBRRanSuccessfully = True) Then
            bShowDlg = False
            bRuleOnForm = False

            If IsArray(vtOutRTPNames) And IsArray(vtOutRTPValues) Then
                lNumRTPNames = (UBound(vtOutRTPNames) - LBound(vtOutRTPNames) + 1)
                lNumRTPVals = (UBound(vtOutRTPValues) - LBound(vtOutRTPValues) + 1)
            End If

            If (lNumRTPNames > 0) Then
                ReDim vtInRTPNames(lNumRTPNames - 1) As Variant
                ReDim vtInRTPValues(lNumRTPNames - 1) As Variant

                For iRTPs = 0 To lNumRTPNames - 1
                    sBRName = vtOutRTPNames(iRTPs)
                    sBRVal = vtOutRTPValues(iRTPs)
                    vtInRTPNames(iRTPs) = sBRName
                    vtInRTPValues(iRTPs) = sBRVal
                Next iRTPs
            End If

        End If
    End If
oRet = HypExecuteCalcScriptEx (oSheetName, vtOutCubeName, vtOutBRName, vtOutBRTypen, bBRHasPrompts, bBRNeedPageInfo, vtInRTPNames, vtInRTPValues, bShowDlg, bRuleOnForm, vtbBRanSuccessfully2, vtOutCubeName, vtOutBRName, vtOutBRTypen, bBRHasPrompts, bBRNeedPageInfo, bBRHidePrompts, vtOutRTPNames, vtOutRTPValues)
    MsgBox *(Automated BR ran successfully - " & vtbBRanSuccessfully2)
End If
Else
    sPrintMsg = "Error - " & oRet
    MsgBox (sPrintMsg)
End If
Else
    sPrintMsg = "Error - " & oRet
    MsgBox (sPrintMsg)
End If
End Sub

Usage

You can use HypExecuteCalcScriptEx in four modes, depending on whether HypListCalcScriptsEx is called before HypExecuteCalcScriptEx.

If you do not call HypListCalcScriptsEx before HypExecuteCalcScriptEx, then the first time you call HypListCalcScriptsEx you should set vtbShowBRDlg to True for the first usage and to False thereafter.

- When vtbShowBRDlg is True (mode 1):
  - **Input Arguments:** vtSheetName, vtCubeName, vtbRuleOnForm are used. vtBRName, vtBRTypen, vtbBRHasPrompts, vtbNeedPageInfo, ppRTPNames, ppRTPValues are ignored.
  - **Behavior:** The Business Rules dialog box displays all possible rules depending upon the vtbRuleOnForm value. When the user, runs the selected business rule and exits the Business Rules dialog box, the details of that business rule are filled in the out arguments and returned to the caller.
  - **Output arguments:** All out arguments are filled and returned to the caller so that they can be used in subsequent calls.

- When vtbShowBRDlg argument is False (mode 2):
  - **Input arguments:** All input arguments are used.
  - **Behavior:** The Business Rules dialog box is not displayed. The business rule is run automatically, and the appropriate status is returned to the caller.
  - **Output arguments:** All output arguments are left unmodified, because nothing needs to be passed on to the caller, who already has all the information to run this particular business rule.

If you do call HypListCalcScriptsEx before HypExecuteCalcScriptEx, then when HypListCalcScriptsEx is called, users get information about all business rules and runtime prompts, if any.

If a user runs a business rule that has no RTP, HypExecuteCalcScriptEx can be called with vtbShowBRDlg argument as False and provides all other information as the input arguments.
If a user runs a business rule that has an RTP, HypExecuteCalcScriptEx must be called with vtbShowBRDlg as True so that the business rule and its RTPs can be displayed and the user can select the RTP values to run the business rule. (InPlanning, the RTP flag may be True for a business rule when there are no RTPs to be displayed.)

- If the cube name, business rule name and business rule type are passed as empty in HypExecuteCalcScriptEx (mode 3), the Business Rules dialog box is displayed and all business rules are shown, depending upon vtbRuleOnForm argument. All else is the same as mode 1.
- If the cube name, business rule name and business rule type are passed with filled values in HypExecuteCalcScriptEx (mode 4), the Business Rules dialog box is displayed and only the passed business rule (business rule name for the provided cube name) is displayed along with its RTPs. All else is the same as mode 1.

**HypDeleteCalc**

**Data provider types:** Essbase

**Description**

HypDeleteCalc() deletes a calculation script from an Essbase server.

**Syntax**

HypDeleteCalc (vtSheetName, vtApplicationName, vtDatabaseName, vtCalcScript)

ByVal vtSheetName As Variant
ByVal vtApplicationName As Variant
ByVal vtDatabaseName As Variant
ByVal vtCalcScript As Variant

**Parameters**

- **vtSheetName**: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
- **vtApplicationName**: The name of the application name that contains the calculation script
- **vtDatabaseName**: The name of the database that contains the calculation script
- **vtCalcScript**: The name of the calculation script to be deleted

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypDeleteCalc Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtApplicationName As Variant, ByVal vtDatabaseName As Variant, ByVal vtCalcScript As Variant) As Long
Sub Example_HypDeleteCalc
Dim X as Long
    X = HypDeleteCalc (Empty,"Sample","Basic","CalcYear")
End Sub
About Calculation, Consolidation, and Translation Functions

These functions execute calculation, consolidation, and translation operations on data for Financial Management and Hyperion Enterprise applications.

HypCalculate

Data provider types: Financial Management, Hyperion Enterprise

Description
HypCalculate() calls the Calculate method.

Syntax
HypCalculate (vtSheetName, vtRange)
ByVal vtSheetName As Variant
By Val vtRange As Variant
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtRange: The range that contains the data to be used. If Empty or Null, then the selected range in the worksheet is used.

Return Value

Returns 0 if successful; otherwise, returns the corresponding error code.

Example

Declare Function HypCalculate Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long
Sub Example_HypCalculate()
sts = HypCalculate (Empty, Empty)
End Sub

HypCalculateContribution

Data provider types: Financial Management (ad hoc only)

Description

HypCalculateContribution() calls the Calculate Contribution.

Syntax

HypCalculateContribution (vtSheetName, vtRange)

ByVal vtSheetName As Variant
By Val vtRange As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtRange: The range that contains the data to be used. If Empty or Null, then the selected range in the worksheet is used.

Return Value

Returns 0 if successful; otherwise, returns the corresponding error code.

Example

Declare Function HypCalculateContribution Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long
Sub Example_HypCalculateContribution()
sts = HypCalculateContribution (Empty, Empty)
End Sub
**HypConsolidate**

**Data provider types:** Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypConsolidate calls the Consolidate method.

**Syntax**

HypConsolidate (vtSheetName, vtRange)

ByVal vtSheetName As Variant

By Val vtRange As Variant

**Parameters**

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtRange: The range object that refers to the data to be used. If Empty or Null, the selected range in the worksheet is used.

**Return Value**

Returns 0 if successful; otherwise, returns the corresponding error code.

**Example**

Declare Function HypConsolidate Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long
Sub Example_HypConsolidate()
    sts = HypConsolidate (Empty, Empty)
End Sub

---

**HypConsolidateAll**

**Data provider types:** Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypConsolidateAll() calls the Consolidate All method.

**Syntax**

HypConsolidateAll (vtSheetName, vtRange)

ByVal vtSheetName As Variant

By Val vtRange As Variant
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtRange: The range that contains the data to be used. If Empty or Null, the selected range in the worksheet is used.

Return Value

Returns 0 if successful; otherwise, returns the corresponding error code.

Example

Declare Function HypConsolidateAll Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long
Sub Example_HypConsolidateAll
sts = HypConsolidateAll(Empty, Empty)
End Sub

HypConsolidateAllWithData

Data provider types: Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypConsolidateAllWithData calls the Consolidate All With Data method.

Syntax

HypConsolidateAllWithData (vtSheetName, vtRange)
ByVal vtSheetName As Variant
By Val vtRange As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtRange: The range that contains the data to be used. If Empty or Null, the selected range in the worksheet is used.

Return Value

Returns 0 if successful; otherwise, returns the corresponding error code.

Example

Declare Function HypConsolidateAllWithData Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long
Sub Example_HypConsolidateAllWithData()
sts = HypConsolidateAllWithData (Empty, Empty)
End Sub
**HypForceCalculate**

Data provider types: Financial Management

Description

HypForceCalculate() calls the Force Calculate method.

Syntax

HypForceCalculate(vtSheetName, vtRange)

ByVal vtSheetName As Variant

By Val vtRange As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtRange: The range that contains the data to be used. If Empty or Null, the selected range in the worksheet is used.

Return Value

Returns 0 if successful; otherwise, returns the corresponding error code.

Example

Declare Function HypForceCalculate Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long
Sub Example_HypForceCalculate()
sts = HypForceCalculate (Empty, Empty)
End Sub

**HypForceCalculateContribution**

Data provider types: Financial Management (ad hoc only)

Description

HypForceCalculateContribution calls the Force Calculate Contribution method.

Syntax

HypForceCalculateContribution (vtSheetName, vtRange)

ByVal vtSheetName As Variant

By Val vtRange As Variant
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtRange: The range that contains the data to be used. If Empty or Null, the selected range in the worksheet is used.

Return Value

Returns 0 if successful; otherwise, returns the corresponding error code.

Example

Declare Function HypForceCalculateContribution Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long
Sub Example_HypForceCalculateContribution()
    sts = HypForceCalculateContribution (Empty, Empty)
End Sub

HypForceTranslate

Data provider types: Financial Management (ad hoc only)

Description

HypForceTranslate calls the Force Translate method.

Syntax

HypForceTranslate (vtSheetName, vtRange)
ByVal vtSheetName As Variant
By Val vtRange As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtRange: The range that contains the data to be used. If Empty or Null, the selected range in the worksheet is used.

Return Value

Returns 0 if successful; otherwise, returns the corresponding error code.

Example

Declare Function HypForceTranslate Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long
Sub Example_HypForceTranslate()
    sts = HypForceTranslate (Empty, Empty)
End Sub
HypTranslate

Data provider types: Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypTranslate() calls the Translate method.

Syntax
HypTranslate (vtSheetName, vtRange)
ByVal vtSheetName As Variant
By Val vtRange As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtRange: The range that contains the data to be used. If Empty or Null, the selected range in the worksheet is used.

Return Value
Returns 0 if successful; otherwise, returns the corresponding error code.

Example
Declare Function HypTranslate Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtRange As Variant) As Long
Sub Example_HypTranslate()
sts = HypTranslate (Empty, Empty)
End Sub
About Member Query Functions

Member query functions retrieve generation, level, attribute, and other information about members.

HypFindMember

Data provider types: Essbase

Description
HypFindMember() retrieves dimension, alias, generation and level information for the specified member.
Syntax

HypFindMember (vtSheetName, vtMemberName, vtAliasTable, vtDimensionName, vtAliasName, vtGenerationName, vtLevelName)

ByVal vtSheetName As Variant
ByVal vtMemberName As Variant
ByVal vtAliasTable As Variant
ByRef vtDimensionName As Variant
ByRef vtAliasName As Variant
ByRef vtGenerationName As Variant
ByRef vtLevelName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtMemberName: Input parameter; the member for which to retrieve information. Required; there is no default value.

vtAliasTable: Input parameter; the name of the alias table to search for the alias name. If Null, the default alias table is used.

vtDimensionName: Output parameter; the dimension of the member

vtAliasName: Output parameter; the alias name of the member

vtGenerationName: Output parameter; the generation of the member

vtLevelName: Output parameter; the level of the member

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypFindMember Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal vtAliasTable As Variant, ByRef vtDimensionName As Variant, ByRef vtAliasName As Variant, ByRef vtGenerationName As Variant, ByRef vtLevelName As Variant) As Long

Sub Example_HypFindMember()
    X = HypFindMember(Empty, "100", "Default", dimName, aliasName, genName, levelName)
    MsgBox (dimName)
    MsgBox (aliasName)
    MsgBox (genName)
    MsgBox (levelName)
End Sub
**HypFindMemberEx**

Data provider types: Essbase

Description

HypFindMemberEx() retrieves dimension, alias, generation and level information for the specified member.

Syntax

```vba
HypFindMember (vtSheetName, vtMemberName, vtAliasTable, vtDimensionName, vtAliasName, vtGenerationName, vtLevelName)
```

- **ByVal vtSheetName As Variant**
- **ByVal vtMemberName As Variant**
- **ByVal vtAliasTable As Variant**
- **ByRef vtDimensionName As Variant**
- **ByRef vtAliasName As Variant**
- **ByRef vtGenerationName As Variant**
- **ByRef vtLevelName As Variant**

Parameters

- **vtSheetName**: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
- **vtMemberName**: The member for which to retrieve information. Required; there is no default value.
- **vtAliasTable**: The name of the alias table to search for the alias name. If Null, the default alias table is searched.
- **vtDimensionName**: Output parameter; the dimension of the member
- **vtAliasName**: Output parameter; the alias name of the member
- **vtGenerationName**: Output parameter; the generation of the member
- **vtLevelName**: Output parameter; the level of the member

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

```vba
Declare Function HypFindMemberEx Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal vtAliasTable As Variant, ByRef vtDimensionName As Variant, ByRef vtAliasName As Variant, ByRef vtGenerationName As Variant, ByRef vtLevelName As Variant) As Long
```
Sub Example_HypFindMemberEx()
    X = HypFindMemberEx(Empty, "100", "Default", dimName, aliasName, genName, levelName)
    MsgBox (dimName)
    MsgBox (aliasName)
    MsgBox (genName)
    MsgBox (levelName)
End Sub

HypGetAncestor

Data provider types: Essbase

Description
HypGetAncestor() returns the ancestor at any specific generation or level for the specified member.

Syntax
HypGetAncestor (vtSheetName, vtMemberName, vtLayerType, intLayerNum, vtAncestor)
ByVal vtSheetName As Variant
ByVal vtMemberName As Variant
ByVal vtLayerType As Variant
ByVal intLayerNum As Integer
ByRef vtAncestor As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtMemberName: The member for which to retrieve information. Required; there is no default value.
vtLayerType: Input parameter: Gen or Level. If set to Null or Empty, Gen is the default.
intLayerNum: Input parameter: the level or generation number. Required.
vtAncestor: Output parameter; the name of the ancestor

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypGetAncestor Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal vtLayerType As Variant, ByVal intLayerNumber As Integer, ByRef vtAncestor As Variant) As Long

Sub Example_HypGetAncestor
Dim X as Long
Dim vtAncestor As Variant
X = HypGetAncestor (Empty, "100-20", "Level", 1, vtAncestor)
End Sub

**HypGetChildren**

*Data provider types:* Essbase

**Description**

HypGetChildren() returns the children for the specified member.

**Syntax**

HypGetChildren (vtSheetName, vtMemberName, intChildCount, vtChildArray)

**Parameters**

- **vtSheetName**: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
- **vtMemberName**: Input parameter; the member name. Required.
- **intChildCount**: Input parameter; a restriction on the number of children returned.
  - ChildCount <=0. All children are returned.
  - ChildCount >0. The result set is limited to the number specified as the argument. If the result set is less than the specified argument, all results are returned.
- **vtChildArray**: Output result vector that contains the list of the children. Its contents are unknown if the macro fails.

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypGetChildren Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal intChildCount As Integer, ByRef vtChildArray As Variant) As Long

Sub Example_HypGetChildren
    Dim vtChildren As Variant
    Dim vtChild As Variant
    Dim X as Long
X = HypGetChildren (Empty, "Market", 0, vtChildren)
If IsArray (vtChildren) Then
    For i = LBound (vtChildren) To UBound (vtChildren)
        VtChild = vtChildren (i)
    Next
End If
End Sub

**HypGetParent**

**Data provider types:** Essbase

**Description**

HypGetParent() returns the name of the parent of the specified member.

**Syntax**

HypGetParent(vtSheetName, vtMemberName, vtParentName)
ByVal vtSheetName As Variant
ByVal vtMemberName As Variant
ByRef vtParentName As Variant

**Parameters**

- **vtSheetName:** The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
- **vtMemberName:** Input parameter; the member name. Required.
- **vtParentName:** Output parameter; the parent name

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypGetParent Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal vtParentName As Variant) As Long

Sub Example_HypGetParent
    Dim vtParent As Variant
    X = HypGetParent (Empty, "East", vtParent)
End sub

**HypIsAttribute**

**Data provider types:** Essbase

---

142  Member Query Functions
Description
HypIsAttribute() checks to see if the specified member has a specific attribute.

Syntax
HypIsAttribute(vtSheetName, vtDimensionName, vtMemberName, vtUDAString)
ByVal vtSheetName As Variant
ByVal vtDimensionName As Variant
ByVal vtMemberName As Variant
ByVal vtUDAString As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtDimensionName: The name of the dimension to which the member belongs
vtMemberName: The member for which to retrieve information. Required; there is no default value.
vtUDAString: Input string that is compared against the attributes of the member.

Return Value
Returns a variant in which -1 is true, 0 is false; otherwise, returns the appropriate error code.

Example
Declare Function HypIsAttribute Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtDimensionName As Variant, ByVal vtMemberName As Variant, ByVal vtUDAString As Variant) As Variant
Sub Example_HypIsAttribute()
vtret = HypIsAttribute(Empty, "Market", "Connecticut", "MyAttribute")
If vtret = -1 Then
    MsgBox ("Found MyAttribute")
ElseIf vtret = 0 Then
    MsgBox ("MyAttribute not available for Connecticut")
Else
    MsgBox ("Error value returned is" & vtret)
End If
End Sub

HypIsDescendant

Data provider types: Essbase
Description
HypIsDescendant() checks if the specified member is the descendant of another specified member.

Syntax
HypIsDescendant(vtSheetName, vtMemberName, vtAncestorName)
ByVal vtSheetName As Variant
ByVal vtMemberName As Variant
ByVal vtAncestorName As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtMemberName: The member for which to retrieve information. Required; there is no default value.
vtAncestorName: The name of the ancestor. Required.

Return Value
Returns a variant in which -1 is true, 0 is false; otherwise, returns the appropriate error code.

Example
Declare Function HypIsDescendant Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal vtDescendantName As Variant) As Boolean
Sub Example_HypIsDescendant
    Dim b as Boolean
    b = HypIsDescendant (Empty, "Year", "Jan")
End sub

HypIsAncestor

Data provider types: Essbase

Description
HypIsAncestor() checks whether the specified member is the ancestor of another specified member.

Syntax
HypIsAncestor(vtSheetName, vtMemberName, vtAncestorName)
ByVal vtSheetName As Variant
ByVal vtMemberName As Variant
ByVal vtAncestorName As Variant
ByVal vtAncestorName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtMemberName: The member for which to retrieve information. Required; there is no default value.

vtAncestorName: The name of the ancestor. Required.

Return Value

Returns a variant in which -1 is true, 0 is false; otherwise, returns the appropriate error code.

Example

Declare Function HypIsAncestor Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal vtAncestorName As Variant) As Variant

Sub Example_HypIsAncestor
    Dim b As Variant
    b = HypIsAncestor (Empty, "Year", "Jan")
End sub

HypIsExpense

Data provider types: Essbase

Description

HypIsExpense() verifies that the member specified has an Expense tag.

Syntax

HypIsExpense(vtSheetName, vtDimensionName, vtMemberName)

ByVal vtSheetName As Variant

ByVal vtDimensionName As Variant

ByVal vtMemberName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtDimensionName: The dimension of the member. If set to Null or Empty, the active dimension is used.

vtMemberName: The member for which to retrieve information. Required; there is no default value.
Return Value

Returns a variant in which -1 is true, 0 is false; otherwise, returns the appropriate error code.

Example

Declare Function HypIsExpense Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtDimensionName As Variant, ByVal vtMemberName As Variant) As Variant

Sub CheckExpense()
    vtret = HypIsExpense(Empty, "Measures", "Opening Inventory")
    If vtret = -1 Then
        MsgBox ("Opening Inventory has expense flag set")
    ElseIf vtret = 0 Then
        MsgBox ("Expense flag has not been set")
    Else
        MsgBox ("Error value returned is" & vtret)
    End If
End Sub

HypIsParent

Data provider types: Essbase

Description

HypIsParent() checks whether the specified member is the parent of another specified member.

Syntax

HypIsParent(vtSheetName, vtMemberName, vtParentName)

ByVal vtSheetName As Variant
ByVal vtMemberName As Variant
ByVal vtParentName As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtMemberName: The member for which to retrieve information. Required; there is no default value.

vtParentName: The name of the parent. Required.

Return Value

Returns a variant in which -1 is true, 0 is false; otherwise, returns the appropriate error code.

Example

Declare Function HypIsParent Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal ParentName As Variant) As Boolean
Sub Example_HypIsParent
    Dim b as Boolean
    b = HypIsParent (Empty, "East", "Market")
End Sub

HypIsChild

Data provider types: Essbase

Description
HypIsChild() determines whether a member is the child of a specified parent member. HypIsChild checks only for children, not for all descendants.

Syntax
HypIsChild(vtSheetName, vtParentName, vtChildName)
ByVal vtSheetName As Variant
ByVal vtParentName As Variant
ByVal vtChildName As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtParentName: The name of the parent. Required
vtChildName: The name of the child. Required

Return Value
Returns a variant in which -1 is true, 0 is false; otherwise, returns the appropriate error code.

Example
Declare Function HypIsChild Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtParentName As Variant, ByVal vtChildName As Variant) As Variant

Sub Example_HypIsChild
    Dim b as Boolean
    b = HypIsChild ("Sheet1", "Year", "Qtr1")
End Sub

HypIsUDA

Data provider types: Essbase
HypIsUDA() determines whether a member has a specific UDA.

Syntax
HypIsUDA (vtSheetName, vtDimensionName, vtMemberName, vtUDAString)
ByVal vtSheetName As Variant
ByVal vtDimensionName As Variant
ByVal vtMemberName As Variant
ByVal vtUDAString As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtDimensionName: The dimension of the member

vtMemberName: The member for which to retrieve information. Required; there is no default value.

vtUDAString: Input string that is compared against the attributes of the member.

Return Value
Returns a variant in which -1 is true, 0 is false; otherwise, returns the appropriate error code.

Example
Declare Function HypIsUDA Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtDimensionName As Variant, ByVal vtMemberName As Variant, ByVal vtUDAString As Variant) As Variant
Sub Example_HypIsUDA()
    vtret = HypIsUDA(Empty, "Market", "Connecticut", "MyUDA")
    If vtret = -1 Then
        MsgBox ("Found MyUDA")
    ElseIf vtret = 0 Then
        MsgBox ("Did not find MyUDA")
    Else
        MsgBox ("Error value returned is" & vtret)
    End If
End Sub

HypOtlGetMemberInfo

Data provider types: Essbase
Description
HypOtlGetMemberInfo() returns the comments, formulas, UDAs, and attributes associated with the selected member selection.

Syntax
HypOtlGetMemberInfo (vtSheetName, vtDimensionName, vtMemberName, vtPredicate, vtMemberArray)
ByVal vtSheetName As Variant
ByVal vtDimensionName As Variant
ByVal vtMemberName As Variant
ByVal vtPredicate As Variant
ByRef vtMemberArray As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtDimensionName: The dimension of the member. If set to Null, the predicate in the whole outline is searched.

vtMemberName: The member for which to retrieve information. Required; there is no default value.

vtPredicate: Member selection criteria:
● 1 = HYP_COMMENT
● 2 = HYP_FORMULA
● 3 = HYP_UDA
● 4 = HYP_ATTRIBUTE

vtMemberArray: Output parameter; the result of the query.

Return Value
Returns 0 if successful; otherwise returns the appropriate error code.

Example
Declare Function HypOtlGetMemberInfo Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtDimensionName As Variant, ByVal vtMemberName As Variant, ByVal vtPredicate As Variant, ByRef vtMemberArray As Variant) As Long
Sub Example_HypOtlGetMemberInfo()
    vtRet = HypOtlGetMemberInfo(Empty, "Year", "Jan", HYP_COMMENT, vt)
    If IsArray(vt) Then cbItems = UBound(vt) + 1
        MsgBox ("Number of elements = " + Str(cbItems))
    For i = 0 To UBound(vt)
        MsgBox ("Member = " + vt(i))
    Next
End Sub
HypQueryMembers

Data provider types: Essbase

Description
HypQueryMembers() executes the member selection query.

Syntax
HypQueryMembers (vtSheetName, vtMemberName, vtPredicate, vtOption, vtDimensionName, vtInput1, vtInput2, vtMemberArray)

ByVal vtSheetName As Variant
ByVal vtMemberName As Variant
ByVal vtPredicate As Variant
ByVal vtOption As Variant
ByVal vtDimensionName As Variant
ByVal vtInput1 As Variant
ByVal vtInput2 As Variant
ByRef vtMemberArray As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtMemberName: The member for which to retrieve information. Required; there is no default value.

vtPredicate: Member selection criteria (integer):

- 1 = HYP_CHILDREN
- 2 = HYP_DESCENDANTS
- 3 = HYP_BOTTOMLEVEL
- 4 = HYP_SIBLINGS
- 5 = HYPSAMELEVEL
- 6 = HYP_SAMEGENERATION
- 7 = HYP_PARENT
- 8 = HYP_DIMENSION
- 9 = HYP_NAMEDGENERATION
vtOption: (integer) Options are dependent on the predicate. For the predicate values, HYP_SEARCH and HYP_WILDSEARCH, specify query options:

- HYP_MEMBERSONLY
- HYP_ALIASESONLY
- HYP_MEMBERSANDALIASES

vtDimensionName: (string) Dimension to limit the scope of the query. It is used with the following query options and ignored otherwise: HYP_NAMEDGENERATION, HYP_NAMEDLEVEL, HYP_USERATTRIBUTE, HYP_SEARCH (set to Null to search through all dimensions), HYP_WILDSEARCH (set to Null to search through all dimensions).

vtInput1: (string) Input string that is determined by the option. It is used with the following query options and ignored otherwise:

- HYP_NAMEDGENERATION (The name of the generation)
- HYP_NAMEDLEVEL (The name of the level)
- HYP_SEARCH (The string to search for. The string is defined as an exact)
- HYP_WILDSEARCH (The string to search for. The string is defined as an exact search string with an optional '*' at the end to mean any set of characters)
- HYP_USERATTRIBUTE (The user-defined attribute)

vtInput2: (string) Input string that is determined by the option. It is used with the following query options and ignored otherwise:

- HYP_USERATTRIBUTE (The user-defined attribute)
- HYP_SEARCH, HYP_WILDSEARCH (If the options are set to search in the alias tables, this string specifies which alias table to search. If the string is Null, all alias tables will be searched).

vtMemberArray: Output that contains the result of the query. If unsuccessful, its contents are unknown.

Return Value

Returns a zero if successful; otherwise, returns the appropriate error code.
Example

Declare Function HypQueryMembers Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal vtPredicate As Variant, ByVal vtOption As Variant, ByVal vtDimensionName As Variant, ByVal vtInput1 As Variant, ByVal vtInput2 As Variant, ByRef vtMemberArray As Variant) As Long

Sub Example_HypQueryMembers()
    ' sts = HypQueryMembers(Empty, "Profit", HYP_CHILDREN, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Profit", HYP_DESCENDANTS, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Profit", HYP_BOTTOMLEVEL, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Sales", HYP_SIBLINGS, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Sales", HYP_SAMELEVEL, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Sales", HYP_SAMEGENERATION, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Sales", HYP_PARENT, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Sales", HYP_DIMENSION, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Year", HYP_NAMEDGENERATION, Empty, "Year", "Quarter", Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Product", HYP_NAMEDLEVEL, Empty, "Product", "SKU", Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Product", HYP_SEARCH, HYP_ALIASESONLY, "Product", "Cola", Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Year", HYP_WILDSEARCH, HYP_MEMBERSONLY, "Year", "J*", Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Market", HYP_USERATTRIBUTE, Empty, "Market", "Major Market", Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Sales", HYP_ANCESTORS, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Jan", HYP_DTSMEMBER, Empty, Empty, Empty, Empty, vArray)
    ' sts = HypQueryMembers(Empty, "Product", Empty, Empty, Empty, Empty, vArray)

    If IsArray(vt) Then
        cbItems = UBound(vt) + 1
        MsgBox ("Number of elements = " + Str(cbItems))
        For i = 0 To UBound(vt)
            MsgBox ("Member = " + vt(i))
        Next
    Else
        MsgBox ("Return Value = " + Str(vt))
    End If
End Sub

HypGetMemberInformation

Data provider types: Essbase
Description

HypGetMemberInformation returns the properties of a selected member.

Syntax

HypGetMemberInformation (vtSheetName, vtMemberName, vtPropertyName, vtPropertyValue, vtPropertyValueStrings)
ByVal vtMemberName As Variant
ByVal vtPropertyName As Variant
ByVal vtPropertyValue As Variant
ByRef vtPropertyValueStrings As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtMemberName: The member for which to retrieve information. Required; there is no default value.

vtPropertyName: Input parameter; the name of the property for which information is required. See Table 3.

vtPropertyValue: Output parameter; the property array for the member, returned as numerical value from the server.

vtPropertyValueStrings: Output parameter; the property array for the member, returned as string equivalent of numerical value for properties for which numerical values do not make sense.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypGetMemberInformation Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByVal vtPropertyName As Variant, ByRef vtPropertyValue As Variant, ByRef vtPropertyValueStrings As Variant) As Long
Sub Example_HypGetMemberInformation
sts = HypGetMemberInformation("Sheet1", "Jan", HYP_MI_NAME, vtValues, vtPropertyValueString)
End Sub

Table 3  Constants for Member Information

<table>
<thead>
<tr>
<th>Constants for Member Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Global Const HYP_MI_NAME = &quot;Name&quot;</td>
</tr>
<tr>
<td>Global Const HYP_MI_DIM = &quot;Dim&quot;</td>
</tr>
</tbody>
</table>
**Constants for Member Information**

- `Global Const HYP_MI_LEVEL = "Level"
- `Global Const HYP_MI_GENERATION = "Generation"
- `Global Const HYP_MI_PARENT_MEMBER_NAME = "ParentMbrName"
- `Global Const HYP_MI_CHILD_MEMBER_NAME = "ChildMbrName"
- `Global Const HYP_MI_PREVIOUS_MEMBER_NAME = "PrevMbrName"
- `Global Const HYP_MI_NEXT_MEMBER_NAME = "NextMbrName"
- `Global Const HYP_MI_CONsolidation = "Consolidation"
- `Global Const HYP_MI_IS_TWO_PASS_CAL_MEMBER = "IsTwoPassCalcMbr"
- `Global Const HYP_MI_IS_EXPENSE_MEMBER = "IsExpenseMbr"
- `Global Const HYP_MI_CURRENCY_CONVERSION_TYPE = "CurrencyConversionType"
- `Global Const HYP_MI_CURRENCY_CATEGORY = "CurrencyCategory"
- `Global Const HYP_MI_TIME_BALANCE_OPTION = "TimeBalanceOption"
- `Global Const HYP_MI_TIME_BALANCE_SKIP_OPTION = "TimeBalanceSkipOption"
- `Global Const HYP_MI_SHARE_OPTION = "ShareOption"
- `Global Const HYP_MI_STORAGE_CATEGORY = "StorageCategory"
- `Global Const HYP_MI_CHILD_COUNT = "ChildCount"
- `Global Const HYP_MI_ATTRIBUTED = "Attributed"
- `Global Const HYP_MI_RELATIONAL_DESCENDANT_PRESENT = "RelDescendantPresent"
- `Global Const HYP_MI_RELATIONAL_PARTITION_ENABLED = "RelPartitionEnabled"
- `Global Const HYP_MI_DEFAULT_ALIAS = "DefaultAlias"
- `Global Const HYP_MI_HIERARCHY_TYPE = "HierarchyType"
- `Global Const HYP_MI_DIM_SOLVE_ORDER = "DimSolveOrder"
- `Global Const HYP_MI_IS_DUPLICATE_NAME = "IsDuplicateName"
- `Global Const HYP_MI_UNIQUE_NAME = "UniqueName"
- `Global Const HYP_MI_ORIGINAL_MEMBER = "OrigMember"
- `Global Const HYP_MI_IS_FLOW_TYPE = "IsFlowType"
- `Global Const HYP_MI_AGGREGATE_LEVEL = "AggLevel"
- `Global Const HYP_MI_FORMAT_STRING = "FormatString"
### HypGetMemberInformationEx

Data provider types: Essbase

**Description**

HypGetMemberInformationEx returns all information about a member in an array.

**Syntax**

HypGetMemberInformationEx (vtSheetName, vtMemberName, vtPropertyNames, vtPropertyValueStrings)

`ByVal vtSheetName As Variant`  
`ByVal vtMemberName As Variant`  
`ByRef vtPropertyNames As Variant`  
`ByRef vtPropertyValueStrings As Variant`  

**Parameters**

- **vtSheetName**: The name of worksheet on which to run the function. If `vtSheetName` is `Null` or `Empty`, the active worksheet is used.

- **vtMemberName**: The member for which to retrieve information. This parameter is required because there is no default value.

- **vtPropertyNames**: The property name array

- **vtPropertyValueStrings**: The property string value array
Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypGetMemberInformationEx Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtMemberName As Variant, ByRef vtPropertyNames As Variant, ByRef vtPropertyValue As Variant, ByRef vtPropertyValueStrings As Variant) As Long

Sub Example_HypGetMemberInformationEx()
    sta = HypGetMemberInformationEx(Empty, "100-10", propertynames, propertyvalues, propertyvaluestrings)
End Sub
About Options Functions

Options functions set and retrieve information for global and/or sheet options, and enable deletion of MRU items.

HypGetGlobalOption

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypGetGlobalOption() returns information about Smart View global options. Global options are options that apply to the entire current workbook and to any workbooks and worksheets that are created henceforth.

See also "HypGetOption" on page 163.

Syntax

HypGetGlobalOption(vtItem)

ByVal vtItem As Long

Parameters

vtItem: The number that indicates which option is to be retrieved

Table 4 lists the numbers of options and their return data types.
<table>
<thead>
<tr>
<th>vtItem</th>
<th>Option</th>
<th>Return Data Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Use Excel formatting</td>
<td>Boolean</td>
</tr>
<tr>
<td>2</td>
<td>Use double-click for ad hoc operations</td>
<td>Boolean</td>
</tr>
<tr>
<td>3</td>
<td>Enable undo</td>
<td>Boolean</td>
</tr>
<tr>
<td>4</td>
<td>Not used</td>
<td>--</td>
</tr>
<tr>
<td>5</td>
<td>Specify message level setting:</td>
<td>Integer</td>
</tr>
<tr>
<td></td>
<td>● 0 = Information</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● 1 = Warnings</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● 2 = Errors</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● 3 = None</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● 4 = Extended info</td>
<td></td>
</tr>
<tr>
<td></td>
<td>● 5 = Profile</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Use thousands separator</td>
<td>Boolean</td>
</tr>
<tr>
<td>7</td>
<td>Route messages to log file</td>
<td>Boolean</td>
</tr>
<tr>
<td>8</td>
<td>Clear log file on next launch</td>
<td>Boolean</td>
</tr>
<tr>
<td>9</td>
<td>Navigate without data</td>
<td>Boolean</td>
</tr>
<tr>
<td>10</td>
<td>Not used</td>
<td>--</td>
</tr>
<tr>
<td>11</td>
<td>Not used</td>
<td>--</td>
</tr>
<tr>
<td>12</td>
<td>Specify Meaningless label</td>
<td>Text</td>
</tr>
<tr>
<td>13</td>
<td>Reduce Excel file size</td>
<td>Boolean</td>
</tr>
<tr>
<td>14</td>
<td>Enable formatted strings</td>
<td>Boolean</td>
</tr>
<tr>
<td>15</td>
<td>Retain numeric formatting</td>
<td>Boolean</td>
</tr>
<tr>
<td>16</td>
<td>Enable enhanced comment handling</td>
<td>Boolean</td>
</tr>
<tr>
<td>17</td>
<td>Enable retain ribbon context</td>
<td>Boolean</td>
</tr>
<tr>
<td>18</td>
<td>Display Smart View Panel on startup</td>
<td>Boolean</td>
</tr>
<tr>
<td>19</td>
<td>Always show on refresh (in Comment Edit dialog box; available only if Enhanced comment handling is enabled and the grid contains comments)</td>
<td>Boolean</td>
</tr>
</tbody>
</table>

**Return Value**

Returns the appropriate return data type as shown in Table 4, “HypGetGlobalOption Parameter Numbers and Options”; otherwise, returns the appropriate error code.
Example

The following example sets the message level option and checks whether the value set is valid.

```vba
Declare Function HypGetGlobalOption Lib "HsAddin" (ByVal vtItem As Long) As Variant

Sub Example_HypGetGlobalOption()
    sts = HypGetGlobalOption(5)
    If sts = -15 Then
        MsgBox ("Invalid Parameter")
    Else
        MsgBox ("Message level is set to" & sts)
    End If
End Sub
```

HypSetGlobalOption

**Data provider types:** Essbase, Financial Management, Planning, Hyperion Enterprise

**Description**

HypSetGlobalOption() sets global Smart View options. Global options are options that apply to the entire current workbook and to any workbooks and worksheets that are created henceforth.

**Note:** You can set only one option at a time.

See also “HypSetOption” on page 168.

**Syntax**

HypSetGlobalOption(vtItem, vtGlobalOption)

* ByVal vtItem As Long
* ByVal vtGlobalOption As Variant

**Parameters**

- **vtItem**: The number that indicates which option is to be set. See Table 4, “HypGetGlobalOption Parameter Numbers and Options,” on page 158 for values.
- **vtGlobalOption**: A variant which can take a Boolean, Number, or Text value denoting the option being set for vtItem. If Null or Empty, no action is performed.

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

The following example sets the option to display no messages.
Declare Function HypSetGlobalOption Lib "HsAddin" (ByVal vtItem As Long, ByVal vtGlobalOption As Variant) As Long

Sub Example_HypSetGlobalOption()
    X=HypSetGlobalOption(5, 3)
    If X=0 Then
        MsgBox("Message level is set to 3 - No messages")
    Else
        MsgBox("Error. Message level not set.")
    End If
End Sub

**HypGetSheetOption**

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description

HypGetSheetOption() returns information about sheet level options.

Syntax

HypGetSheetOption(vtSheetName, vtItem)

ByVal vtSheetName As Variant

ByVal vtItem As Variant

Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtItem: The number that indicates which option is to be retrieved. See Table 5 for a list of values.

<table>
<thead>
<tr>
<th>vtItem</th>
<th>Option</th>
<th>Data Type and Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Set zoom in level:</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>• 0 = Next level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1 = All levels</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 2 = Bottom level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 3 = Sibling level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 4 = Same level</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 5 = Same generation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 6 = Formulas</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Enable Include Selection setting</td>
<td>Boolean</td>
</tr>
<tr>
<td>3</td>
<td>Enable Within Selection Group setting</td>
<td>Boolean</td>
</tr>
<tr>
<td>4</td>
<td>Enable Remove Unselected Groups setting</td>
<td>Boolean</td>
</tr>
<tr>
<td>vtItem</td>
<td>Option</td>
<td>Data Type and Values</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------------</td>
<td>----------------------</td>
</tr>
<tr>
<td>5</td>
<td>Specify Indent setting:</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>• 0 = No indentation</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1 = Indent sub items</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 2 = Indent totals</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>Enable suppress missing setting</td>
<td>Boolean</td>
</tr>
<tr>
<td>7</td>
<td>Enable suppress zeros setting</td>
<td>Boolean</td>
</tr>
<tr>
<td>8</td>
<td>Enable suppress underscores setting</td>
<td>Boolean</td>
</tr>
<tr>
<td>9</td>
<td>Enable No Access setting</td>
<td>Boolean</td>
</tr>
<tr>
<td>10</td>
<td>Enable Repeated Member setting</td>
<td>Boolean</td>
</tr>
<tr>
<td>11</td>
<td>Enable Invalid setting</td>
<td>Boolean</td>
</tr>
<tr>
<td>12</td>
<td>Ancestor Position:</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>• 0 = Top</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1 = Bottom</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>Specify Missing Text label</td>
<td>Text</td>
</tr>
<tr>
<td>14</td>
<td>Specify No Access label</td>
<td>Text</td>
</tr>
<tr>
<td>15</td>
<td>Cell Status:</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>• 0 = Data</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1 = Calculation Status</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 2 = Process Management</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>Member Name Display options:</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td>• 0 = Name Only</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 1 = Name and Description</td>
<td></td>
</tr>
<tr>
<td></td>
<td>• 2 = Description only</td>
<td></td>
</tr>
</tbody>
</table>

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypGetSheetOption Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtItem As Variant) As Variant

Sub Example_HypGetSheetOption()
    sts = HypGetSheetOption("Sheet", 5)
    If sts = -15 then
        MsgBox ("Invalid Parameter")
    Else
        MsgBox ("Indentation is set to" & sts)
End Sub
HypSetSheetOption

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypSetSheetOption() sets sheet level options.

Note: You can set only one option at a time.

Syntax
HypSetSheetOption(vtSheetName, vtItem, vtOption)
ByVal vtSheetName As Variant
ByVal vtItem As Variant
ByVal vtOption As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtItem: The number that indicates which option is to be set. See Table 5 on page 160 for a list of values.

vtOption: The new value of the item.

Return Values
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypSetSheetOption Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtItem As Variant, ByVal vtOption As Variant) As Long

Sub Example_HypSetSheetOption()
X=HypSetSheetOption(Empty, 6, FALSE)
If X=0 Then
    MsgBox("#Missing values will appear. ")
Else
    MsgBox("Error. #Missing option not set.")
End If
End Sub
**HypGetOption**

Data provider types: Essbase, Financial Management, Planning, Hyperion Enterprise

Description

HypGetOption() retrieves Smart View options that are both default and sheet specific so you do not need separate VBA commands for the two types of options.

See also “HypGetGlobalOption” on page 157.

Syntax

HypGetOption (vtItem, vtRet, vtSheetName)

ByVal vtItem As Variant

ByRef vtRet As Variant

ByVal vtSheetName As Variant

Parameters

vtItem: The index or constant that refers to a specific option. See Table 6 on page 164 for descriptions of the options. Also, a list of available options is shown in smartview.bas under “Enumeration of options index to be used for HypGetOption/HypSetOption.”

vtRet: The output variable

vtSheetName: The sheet name of a sheet level option. If a valid sheet name is not provided, then the default option is used.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypGetOption Lib "HsAddin" (ByVal vtItem As Variant, ByRef vtRet As Variant, ByVal vtSheetName As Variant) As Long

Sub Example_HypGetOption()

    sts = HypGetOption(HSV_ZOOMIN, Var, "Sheet2") 'get zoom in option for sheet2
    sts = HypGetOption(1, Var, "") 'get default zoom in option

End Sub
<table>
<thead>
<tr>
<th>Constant</th>
<th>Data Type</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>HSV_ZOOMIN</td>
<td>Number</td>
<td>Sets zoom in level:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0 = Next level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 = All levels</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 = Bottom level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 3 = Sibling level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 4 = Same level</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 5 = Same generation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 6 = Formulas</td>
</tr>
<tr>
<td>HSV_INCLUDE_SELECTION</td>
<td>Boolean</td>
<td>Selects the Include Selection check box</td>
</tr>
<tr>
<td>HSV_WITHIN_SELECTEDGROUP</td>
<td>Boolean</td>
<td>Selects the Within Selected Group check box</td>
</tr>
<tr>
<td>HSV_REMOVE_UNSELECTEDGROUP</td>
<td>Boolean</td>
<td>Selects the Remove Unselected Groups check box</td>
</tr>
<tr>
<td>HSV_INDENTATION</td>
<td>Number</td>
<td>Selects an Indentation option</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0 = No indentation</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 = Indent sub items</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 2 = Indent totals</td>
</tr>
<tr>
<td>HSV_SUPPRESSROWS_MISSING</td>
<td>Boolean</td>
<td>Suppresses rows that contain no data or are missing data</td>
</tr>
<tr>
<td>HSV_SUPPRESSROWS_ZEROS</td>
<td>Boolean</td>
<td>Suppresses rows that contain only zeroes</td>
</tr>
<tr>
<td>HSV_SUPPRESSROWS_UNDERSCORE</td>
<td>Boolean</td>
<td>Suppresses rows that contain underscore characters in member names</td>
</tr>
<tr>
<td>HSV_SUPPRESSROWS_NOACCESS</td>
<td>Boolean</td>
<td>Suppress rows that contain data that the user does not have the security access to view</td>
</tr>
<tr>
<td>HSV_SUPPRESSROWS_REPEATEDMEMBERS</td>
<td>Boolean</td>
<td>Suppresses rows that contain repeated member names, regardless of grid orientation.</td>
</tr>
<tr>
<td>HSV_SUPPRESSROWS_INVALID</td>
<td>Boolean</td>
<td>Suppresses rows that contain only invalid values</td>
</tr>
<tr>
<td>HSV_ANCESTOR_POSITION</td>
<td>Number</td>
<td>Specifies an ancestor position in hierarchies:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 0 = Top</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• 1 = Bottom</td>
</tr>
<tr>
<td>HSV_MISSING_LABEL</td>
<td>Text</td>
<td>Displays #Missing, #Numeric Zero, or the text of your choice in data cells that contain missing data.</td>
</tr>
<tr>
<td>HSV_NOACCESS_LABEL</td>
<td>Text</td>
<td>Displays #NoAccess, #Numeric Zero, or the text of your choice in data cells that the user does not have permission to view.</td>
</tr>
<tr>
<td>Constant</td>
<td>Data Type</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-----------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HSV_CELL_STATUS</td>
<td>15</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>As an alternative to displaying actual data, displays the calculation or process status of the cells:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 = Data</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 = Calculation Status</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 = Process Management</td>
</tr>
<tr>
<td>HSV_MEMBER_DISPLAY</td>
<td>16</td>
<td>Number</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specifies how to display member names in cells:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 0 = Name Only</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 1 = Name and Description</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- 2 = Description only</td>
</tr>
<tr>
<td>HSV_INVALID_LABEL</td>
<td>17</td>
<td>Text</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Displays #Invalid, #Numeric Zero, or the text of your choice in data cells that contain invalid data.</td>
</tr>
<tr>
<td>HSV_SUBMITZERO</td>
<td>18</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If you specified #NumericZero for the HSV_MISSING_LABEL, HSV_NOACCESS_LABEL, or SV_INVALID_LABEL options, allows you to submit zeroes to the database.</td>
</tr>
<tr>
<td>HSV_19</td>
<td>19</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>HSV_20</td>
<td>20</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>HSV_PRESERVE_FORMULA_COMMENT</td>
<td>21</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Preserves formulas and comments on the grid during queries.</td>
</tr>
<tr>
<td>HSV_22</td>
<td>22</td>
<td>Reserved for future use</td>
</tr>
<tr>
<td>HSV_FORMULA_FILL</td>
<td>23</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Propagates formulas associated with member cells to the members retrieved as a result of zooming in.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>If HSV_PRESERVE_FORMULA_COMMENT and HSV_EXCEL_FORMATTING are both enabled, propagates cell formatting to the members retrieved as a result of zooming in.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Applies to formulas in both member and data cells.</td>
</tr>
<tr>
<td>HSV_EXCEL_FORMATTING</td>
<td>30</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Selects the Excel formatting check box</td>
</tr>
<tr>
<td>HSV_RETAIN_NUMERIC_FORMATTING</td>
<td>31</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When the user drills down in dimensions, uses the scale specified in HSV_SCALE and/or number of decimal places from HSV_DECIMALPLACES for data.</td>
</tr>
<tr>
<td>HSV_THOUSAND_SEPARATOR</td>
<td>32</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Uses a comma or other thousands separator in numerical data. Do not use # or $ as the thousands separator in Excel International Options.</td>
</tr>
<tr>
<td>HSV_NAVIGATE_WITHOUTDATA</td>
<td>33</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Enables the speeding up of operations such as Pivot, Zoom, Keep Only, and Remove Only by preventing the calculation of source data while you are navigating. When you are ready to retrieve data, disable Navigate without Data.</td>
</tr>
<tr>
<td>HSV_ENABLE_FORMATSTRING</td>
<td>34</td>
<td>Boolean</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Essbase-specific.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Essbase provides a format string to be associated with different data types.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Once enabled, shows user specific text instead of numbers.</td>
</tr>
<tr>
<td>Constant</td>
<td>Data Type</td>
<td>Comment</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>-----------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HSV_ENHANCED_COMMENT_HANDLING</td>
<td>35</td>
<td>Boolean Enables review and correction of comments and member names in ad hoc grids that contain comments.</td>
</tr>
<tr>
<td>HSV_ADJUSTCOLUMNWIDTH</td>
<td>36</td>
<td>Boolean Adjusts column widths to fit cell contents automatically.</td>
</tr>
<tr>
<td>HSV_DECIMALPLACES</td>
<td>37</td>
<td>Number Specifies the number of decimal places to display.</td>
</tr>
<tr>
<td>HSV_SCALE</td>
<td>38</td>
<td>Number Specifies the scaling of numeric data, which is displayed based on the scale selected.</td>
</tr>
<tr>
<td>HSV_MOVEFORMATS_ON_ADHOC</td>
<td>39</td>
<td>Boolean Copies parent cell formatting to zoomed in cells and retains this formatting even if the cell location changes after an operation.</td>
</tr>
<tr>
<td>HSV_DISPLAY_INVALIDDATA</td>
<td>40</td>
<td>Boolean Displays invalid data.</td>
</tr>
<tr>
<td>HSV_SUPPRESSCOLUMNS_MISSING</td>
<td>41</td>
<td>Boolean Suppresses columns that contain cells for which no data exists in the database (no data is not the same as zero. Zero is a data value.)</td>
</tr>
<tr>
<td>HSV_SUPPRESSCOLUMNS_ZEROS</td>
<td>42</td>
<td>Boolean Suppresses columns that contain only zeroes.</td>
</tr>
<tr>
<td>HSV_SUPPRESSCOLUMNS_NOACCESS</td>
<td>43</td>
<td>Boolean Suppresses columns that contain data that the user does not have the security access to view.</td>
</tr>
<tr>
<td>HSV_SUPPRESS_MISSINGBLOCKS</td>
<td>44</td>
<td>Boolean Suppresses blocks of cells for which no data exists in the database.</td>
</tr>
<tr>
<td>HSV_DOUBLECLICK_FOR_ADHOC</td>
<td>101</td>
<td>Boolean Specifies that double-clicking retrieves the default grid in a blank worksheet and thereafter zooms in or out on the cell contents.</td>
</tr>
<tr>
<td>HSV_UNDO_ENABLE</td>
<td>102</td>
<td>Boolean Enables and disables Undo. Specify the number undo operations allowed with the HSV_NUMBER_OF_UNDO_ACTION parameter.</td>
</tr>
<tr>
<td>HSV_103</td>
<td>103</td>
<td>Reserved for future use.</td>
</tr>
<tr>
<td>HSV_LOGMESSAGE_DISPLAY</td>
<td>104</td>
<td>Number Specifies message display level setting:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● 0 = Information</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● 1 = Warnings</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● 2 = Errors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● 3 = None</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● 4 = Extended info</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● 5 = Profile</td>
</tr>
<tr>
<td>HSV_ROUTE_LOGMESSAGE_TO_FILE</td>
<td>105</td>
<td>Boolean Enables and disables the Route Messages to File check box.</td>
</tr>
<tr>
<td>HSV_CLEAR_LOG_ON_NEXTLAUNCH</td>
<td>106</td>
<td>Boolean Clears the log file starting with the next log message generation, which will be seen after Excel is closed.</td>
</tr>
<tr>
<td>Constant</td>
<td>Data Type</td>
<td>Comment</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HSV_REDUCE_EXCEL_FILESIZE</td>
<td>Boolean</td>
<td>Should always be enabled except in the following cases, when it should not be used:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● You send an Excel workbook to users on Smart View releases earlier than 9.3.1.6 or to users on Microsoft Office regardless of Smart View release.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In these workbooks:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ Grids that contain functions must be refreshed before data can be displayed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>○ In ad hoc mode, POV settings are lost; the behavior is similar to that of a fresh ad hoc grid.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>● You open a workbook sent from users on Smart View release earlier than 9.3.1.6 or on Microsoft Office regardless of Smart View release.</td>
</tr>
<tr>
<td>HSV_ENABLE_RIBBON_CONTEXT</td>
<td>Boolean</td>
<td>Displays the active data provider ribbon automatically after you use a button on the Smart View ribbon.</td>
</tr>
<tr>
<td>HSV_DISPLAY_HOMEPANEL_ONSTARTUP</td>
<td>Boolean</td>
<td>Enables and disables the Display on Startup check box on the Smart View Home panel.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When enabled, shows the Smart View Home Panel when the Panel icon is selected in the Smart View ribbon.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When disabled, the last opened panel is shown.</td>
</tr>
<tr>
<td>HSV_SHOW_COMMENTDIALOG_ON_REFRESH</td>
<td>Boolean</td>
<td>When enabled, if the grid has comments, the comment editor is displayed to users upon refresh.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When disabled, users can launch the comment editor from the Smart View ribbon.</td>
</tr>
<tr>
<td>HSV_NUMBER_OF_UNDO_ACTION</td>
<td>Number</td>
<td>The number of Undo and Redo actions permitted on an operation (0 through 100).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Works in conjunction with the HSV_UNDO_ENABLE parameter.</td>
</tr>
<tr>
<td>HSV_NUMBER_OF_MRU_ITEMS</td>
<td>Number</td>
<td>The number, 15 or fewer, of your most recently used connections to be displayed on Smart View Home and the Open menu on the Smart View ribbon.</td>
</tr>
<tr>
<td>HSV_ROUTE_LOGMESSAGE_FILE_LOCATION</td>
<td>Text</td>
<td>Saves log messages in a file.</td>
</tr>
<tr>
<td>HSV_DISABLE_SMARTVIEW_IN_OUTLOOK</td>
<td>Boolean</td>
<td>Disables Smart View in Outlook if you do not want to use Smart View task lists in Outlook.</td>
</tr>
<tr>
<td>HSV_DISPLAY_SMARTVIEW_SHORTCUT_MENU_ONLY</td>
<td>Boolean</td>
<td>Displays only Smart View menu items on shortcut menus. Otherwise, shortcut menus display both Excel and Smart View items.</td>
</tr>
<tr>
<td>HSV_DISPLAY_DRILL_THROUGH_REPORT_TOOLTIP</td>
<td>Boolean</td>
<td>Displays by default lists of available drill-through reports for cells whenever you mouse over them.</td>
</tr>
<tr>
<td>HSV_SHOW_PROGRESSINFORMATION</td>
<td>Boolean</td>
<td>Specifies that the Smart View Progress status bar will appear when an operation begins after the number of seconds defined in HSV_PROGRESSINFO_TIMEDELAY.</td>
</tr>
<tr>
<td>HSV_PROGRESSINFO_TIMEDELAY</td>
<td>Number</td>
<td>The time, in seconds, after which the Smart View Progress status bar appears when an operation begins.</td>
</tr>
</tbody>
</table>
HypSetOption

Data provider types: Essbase, Financial Management, Planning, Hyperion Enterprise

Description

HypSetOption() enables you to set Smart View options as both default and sheet specific so you do not need separate VBA commands for the two types of options.

See also “HypSetGlobalOption” on page 159.

Syntax

HypSetOption (vtItem,vtOption,vtSheetName)
ByVal vtItem As Variant
ByVal vtOption As Variant
ByVal vtSheetName As Variant

Parameters

vtItem: The index or constant that refers to a specific option. See Table 6 on page 164 for descriptions of the options. Also, a list of available options is shown in smartview.bas under “Enumeration of options index to be used for HypGetOption/HypSetOption.”

vtOption: The input value to set for an option.

vtSheetName: The sheet name to set a sheet level option. If a valid sheet name is not provided, then the default option is used.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Public Declare Function HypSetOption Lib "HsAddin" ( ByVal vtItem As Variant, ByVal vtOption As Variant, ByVal vtSheetName As Variant) As Long

Sub Example_HypSetOption()

sts = HypSetOption(HSV_ZOOMIN, 2, "Sheet2") 'set zoom in option for sheet2
sts = HypSetOption(HSV_ZOOMIN, 1, "") 'set default zoom in

sts = HypSetOption(HSV_INVALID_LABEL, "#InvalidTest", "Sheet2") 'set invalid label for sheet2
sts = HypSetOption(17, "#globalinvalid", "") 'set default invalid label, numbers can be used instead of declared constants

End Sub

HypDeleteAllMRUItems

Data provider types: All
Description
HypDeleteAllMRUItems () deletes all items in the most recently used list, including those that are pinned to the list.

Syntax
HypDeleteAllMRUItems Lib "HsAddin" () As Long

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Public Declare Function HypDeleteAllMRUItems Lib "HsAddin" () As Long

Sub Example_HypDeleteAllMRUItems ()
    sts = HypDeleteAllMRUItems()
End Sub
About Dynamic Link Views

You can use static or dynamic link views to display details about a data point in an adjacent window without disturbing the contents in the main window. Static link views are predefined and are built into Smart View. With dynamic link views, you can use the VBA functions in this section to change row, column, POV, and connection information.

When the dynamic link query has been initialized, all the subsequent setinfo, getinfo, displaytolinkview calls are performed on that saved dynamic link query. If you change the grid on the worksheet and want to perform the dynamic link action on the new grid, you must again initialize the query using the setinfo calls available.
Setting Up Dynamic Link Views

Use dynamic link views to customize link behavior. With a dynamic link view, you can change the connection, row, column, POV, and column information.

To set up a dynamic link view:

1. Set the HypUseLinkMacro flag to True. (When HypUseLinkMacro is set to False, the predefined link query is performed.)

2. Set the macro name to run.
   The macro name you set should contain all the function calls to initialize the grid and to set the connection, row, POV, and column items as needed.

3. Connect the sheet and retrieve the appropriate grid onto the sheet.

4. Select a data point on the sheet.

5. From the Essbase ribbon, select Visualize, then Visualize in Excel.
   The macro set in step 2 is executed, and the link action is performed.

Automating Macro Execution

You can automate execution of a macro through the Smart View menu.

To set up a macro to execute manually through the Smart View menu:

1. Set the HypUseLinkMacro flag to false.

2. Connect the sheet and retrieve a grid.

3. Select a data point on the sheet.

4. Run the macro that contains all the function calls to initialize the grid and set the connection, row, column, and POV items.

HypUseLinkMacro

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypUseLinkMacro() specifies the type of link view: static or dynamic.

Note: Static and dynamic link views share the same menu option; therefore, you must turn the flag on before performing the dynamic link query. When you are finished with dynamic link views, turn the flag off.
Syntax
HypUseLinkMacro (bUse)
ByVal bUse as Boolean

Parameters
bUse: Set to True to perform dynamic link. Set to False to perform static link.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypUseLinkMacro Lib "HsAddin" (ByVal bUse As Boolean) As Long
Sub Example_HypUseLinkMacro()
    Sts = HypUseLinkMacro(True)
End sub

HypSetLinkMacro

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypSetLinkMacro() sets the macro name to be run to perform the dynamic link query action.

Note: When the link action is triggered from the Visualize in Excel menu item, the macro set by this function will be run.

Syntax
HypSetLinkMacro (vtMacroName)
ByVal vtMacroName As Variant

Parameters
vtMacroName: The name of the macro to be run

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.

Example
Declare Function HypSetLinkMacro Lib "HsAddin" (ByVal vtMacroName As Variant) As Long
Sub Example_HypSetLinkMacro()
Sts = HypUseLinkMacro(True)
Sts = HypSetLinkMacro("Sheet1.Macro8")
End Sub

**HypGetLinkMacro**

**Data provider types:** Essbase, Planning (ad hoc only), Financial Management, (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypGetLinkMacro() returns the macro name currently set to be run to perform the dynamic link query.

**Syntax**

HypGetLinkMacro (vtMacroName)

ByRef vtMacroName As Variant

**Parameters**

vtMacroName: Output parameter, returns the currently set macro name

**Return Value**

Returns 0 if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypGetLinkMacro Lib "HsAddin" (ByRef vtMacroName As Variant) As Long

Sub Example_HypGetLinkMacro()
    Dim Macroname As Variant
    Sts = HypUseLinkMacro(True)
    Sts = HypSetLinkMacro("Sheet1.Macro8")
    Sts = HypGetLinkMacro(Macroname)
    If (StrComp(Macroname, "Sheet1.Macro8")) Then
        MsgBox ("Error Occurred")
    End If
End Sub

**HypGetSourceGrid**

**Data provider types:** Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypGetSourceGrid() creates a query from the source grid for the dynamic link query.

This function applies to both static and dynamic link views.
Before you run HypGetSourceGrid, a connected grid must exist on the active worksheet and a valid data cell must be selected.

Syntax
HypGetSourceGrid(vtSheetName, vtGrid)
ByVal vtSheetName As Variant
ByRef vtGrid As Variant

Parameters
vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.
vtGrid: The grid XML returned

Return Value
Returns 0 if successful or the appropriate error code otherwise.

Example
Declare Function HypGetSourceGrid Lib "HsAddin" (ByVal vtSheetName As Variant, ByRef vtGrid As Variant) As Long

Sub Example_HypGetSourceGrid()
    Dim vtGrid As Variant
    Range ("B2").Select
    Sts = HypGetSourceGrid (Empty, vtGrid)
End sub

HypDisplayToLinkView

Data provider types: Essbase, Planning, Financial Management, Hyperion Enterprise

Description
HypDisplayToLinkView() displays Office documents to Word or PowerPoint or grids to Excel.

Note: The link action is performed with the latest content of the dynamic link query.

Syntax
HypDisplayToLinkView (vtDocumentType, vtDocumentPath)
ByVal vtDocumentType As Variant
ByVal vtDocumentPath As Variant
Parameters

tDocumentType: The destination for the link view. Valid values:

- EXCEL_APP
- WORD_APP
- PPOINT_APP

tDocumentPath: The path to the document. Required only for WORD_APP or PPOINT_APP.

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypDisplayToLinkView Lib "HsAddin" (ByVal vtDocumentType As Variant, ByVal vtDocumentPath As Variant) As Long

Sub Example_HypDisplayToLinkView()
   Dim vtGrid As Variant
   Sts = HypConnect(Empty, "UserName", "Password", "MyDemoBasic")
   Sts = HypRetrieve(Empty)
   Range("B2").Select
   Sts = HypGetSourceGrid(Empty, vtGrid)
   Sts = HypSetColItems(1, "Market", "East", "West", "South", "Central", "Market")
   Sts = HypDisplayToLinkView("EXCEL_APP", "")
End Sub

HypGetConnectionInfo

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypGetConnectionInfo() returns the connection information for the dynamic link query.

Note: It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

Syntax

HypGetConnectionInfo(vtServerName, vtUserName, vtPassword, vtApplicationName, vtDatabaseName, vtFriendlyName, vtURL, vtProviderType)

ByRef vtServerName As Variant
ByRef vtUserName As Variant
ByRef vtPassword As Variant
ByRef vtApplicationName As Variant
ByRef vtDatabaseName As Variant
ByRef vtFriendlyName As Variant
ByRef vtURL As Variant
ByRef vtProviderType As Variant

Parameters

vtServerName: Output parameter; the name of the server for the dynamic link query
vtUserName: Output parameter; the user name for the dynamic link query
vtPassword: Output parameter; the password for the dynamic link query. Note: The actual password is not returned for security reasons; it is returned as Empty.
vtApplicationName: Output parameter; the application name for the dynamic link query
vtDatabaseName: Output parameter; the database name for the dynamic link query
vtFriendlyName: Output parameter; the friendly connection name for the dynamic link query
vtURL: Output parameter; the URL for the dynamic link query
vtProviderType: Output parameter; the provider type for the dynamic link query

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypGetConnectionInfo Lib "HsAddin" (ByRef vtServerName As Variant, ByRef vtUserName As Variant, ByRef vtPassword As Variant, ByRef vtApplicationName As Variant, ByRef vtDatabaseName As Variant, ByRef vtFriendlyName As Variant, ByRef vtURL As Variant, ByRef vtProviderType As Variant) As Long

Sub Example_HypGetConnectionInfo()
    Dim vtGrid As Variant
    Dim server As Variant
    Dim user As Variant
    Dim app As Variant
    Dim db As Variant
    Dim provider As Variant
    Dim conn As Variant
    Dim url As Variant
    Sts = HypConnect(Empty, "UserName", "MyDemoBasic")
    Sts = HypRetrieve(Empty)
    Range ("B2").Select
    Sts = HypGetSourceGrid (Empty, vtGrid)
    Sts = HypGetConnectionInfo(server, user, pwd, app, db, conn, url, provider)
End sub
**HypSetConnectionInfo**

**Data provider types:** Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypSetConnectionInfo() modifies the connection information in the query.

The parameters passed for HypSetConnectionInfo() must match the connection information stored with that connection name.

**Note:** It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

**Syntax**

HypSetConnectionInfo (vtServerName, vtUserName, vtPassword, vtApplicationName, vtDatabaseName, vtFriendlyName, vtURL, vtProviderType)

ByVal vtServerName As Variant
ByVal vtUserName As Variant
ByVal vtPassword As Variant
ByVal vtApplicationName As Variant
ByVal vtDatabaseName As Variant
ByVal vtFriendlyName As Variant
ByVal vtURL As Variant
ByVal vtProviderType As Variant

**Parameters**

- **vtServerName:** The server name in the query
- **vtUserName:** The user name in the query
- **vtPassword:** The user password in the query
- **vtApplicationName:** The application name in the query
- **vtDatabaseName:** The database name in the query
- **vtFriendlyName:** The friendly connection name in the query
- **vtURL:** The provider URL in the query
- **vtProviderType:** The provider type in the query
Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypSetConnectionInfo Lib "HsAddin" (ByVal vtServerName As Variant, ByVal vtUserName As Variant, ByVal vtPassword As Variant, ByVal vtApplicationName As Variant, ByVal vtDatabaseName As Variant, ByVal vtFriendlyName As Variant, ByVal vtURL As Variant, ByVal vtProviderType As Variant) As Long

Sub Example_HypSetConnectionInfo()
    Dim vtGrid As Variant
    Sts = HypConnect(Empty, "UserName", "Password", "DemoBasic")
    Sts = HypRetrieve(Empty)
    Range("B2").Select
    Sts = HypGetSourceGrid(Empty, vtGrid)
End Sub

HypGetRowCount

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypGetRowCount() returns the number of row dimensions.

Note: It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

Syntax

HypGetRowCount()

Return Value

Returns number of row dimensions if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypGetRowCount Lib "HsAddin" () As Long

Sub Example_HypGetRowCount()
    Dim vtGrid As Variant
    Sts = HypConnect(Empty, "UserName", "Password", "MyDemoBasic")
    Sts = HypRetrieve(Empty)
    Range("B2").Select
    Sts = HypGetSourceGrid(Empty, vtGrid)
Sts = HypGetRowCount()
End sub

**HypGetColCount**

**Data provider types:** Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypGetColCount() returns the number of column dimensions.

**Note:** It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

**Syntax**

HypGetColCount()

**Return Value**

Returns the number of column dimensions if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypGetColCount Lib "HsAddin" () As Long

Sub Example_HypGetColCount()
    Dim vtGrid As Variant
    Sts = HypConnect(Empty, "UserName", "Password", "MyDemoBasic")
    Sts = HypRetrieve(Empty)
    Range ("B2").Select
    Sts = HypGetColCount()
End sub

**HypGetPOVCount**

**Data provider types:** Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypGetPOVCount() returns the number of dimensions in the POV from the dynamic link query.
Note: It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

**Syntax**

HypGetPOVCount()

**Return Value**

Returns the number of dimensions in the POV if successful; otherwise, returns the appropriate error code.

**Example**

Declare Function HypGetPOVCount Lib "HsAddin" () As Long

Sub Example_HypGetPOVCount()
    Dim vtGrid As Variant
    Sts = HypConnect(Empty, "UserName", "Password", "MyDemoBasic")
    Sts = HypRetrieve(Empty)
    Range ("B2").Select
    Sts = HypGetSourceGrid (Empty, vtGrid)
    Sts = HypGetPOVCount ()
End sub

**HypGetRowItems**

**Data provider types:** Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

**Description**

HypGetRowItems() returns the members present for the nth row dimension in the dynamic link query.

Note: It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

**Syntax**

HypGetRowItems(vtRowID, vtDimensionName, vtMemberNames)

ByVal vtRowID As Variant
ByRef vtDimensionName As Variant
ByRef vtMemberNames As Variant
Parameters

vtRowID: The row number \( n \).

vtDimensionName: Output parameter; the nth row dimension name

vtMemberNames: Output parameter; the members for the nth row dimensions

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypGetRowItems Lib "HsAddin" (ByVal vtRowID As Variant, ByRef vtDimensionName As Variant, ByRef vtMemberNames As Variant) As Long

Sub Example_HypGetRowItems()
    Dim vtGrid As Variant
    Dim vtDimName As Variant
    Dim vtMembers As Variant
    Sts = HypConnect(Empty, "UserName", "Password", "DemoBasic_Connection")
    Sts = HypRetrieve(Empty)
    Range("B2").Select
    Sts = HypGetSourceGrid (Empty, vtGrid)
    Sts = HypGetRowItems(1, vtDimName, vtMembers)
End sub

HypSetRowItems

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

Sets the members for the nth row dimension for this dynamic link query. If the nth row does not exist, a new row is appended.

Note: It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

Syntax

HypSetRowItems (vtRowID, vtDimensionName, ParamArray MemberList())

ByVal vtRowID As Variant
ByVal vtDimensionName As Variant
ParamArray MemberList() As Variant
Parameters

vtRowID: The row number \( n \)

vtDimensionName: The dimension name

ParamArray MemberList: The list of member names

Return Value

Long. Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypSetRowItems Lib "HsAddin" (ByVal vtRowID As Variant, ByVal vtDimensionName As Variant, ParamArray MemberList() As Variant) As Long

Sub Example_HypSetRowItems()
    Dim vtGrid As Variant
    Sts = HypConnect(Empty, "UserName", "Password", "DemoBasic")
    Sts = HypRetrieve(Empty)
    Range ("B2").Select
    Sts = HypGetSourceGrid (Empty, vtGrid)
    Sts = HypSetRowItems(1, "Product", "100", "200", "300", "400", "Diet", "Product")
End sub

HypGetColItems

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypGetColItems() returns the members present in the dynamic link query for the nth column dimensions.

Note: It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

Syntax

HypGetColItems(vtColID, vtDimensionName, vtMemberNames)

ByVal vtColID As Variant

ByRef vtDimensionName As Variant

ByRef vtMemberNames As Variant

Parameters

vtColID: The column number \( n \)
vtDimensionName: Returns the nth column dimension name

vtMemberNames: Returns members for the nth column dimensions

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypGetColItems Lib "HsAddin" (ByVal vtColID As Variant, ByRef vtDimensionName As Variant, ByRef vtMemberNames As Variant) As Long

Sub Example_HypGetColItems()
    Dim vtGrid As Variant
    Dim vtDimensionName As Variant
    Dim vtMembers As Variant
    Sts = HypConnect(Empty, "UserName", "Password", "AnamikaDemoBasic")
    Sts = HypRetrieve(Empty)
    Range ("B2").Select
    Sts = HypGetSourceGrid (Empty, vtGrid)
    Sts = HypGetColItems(1, vtDimensionName, vtMemberNames)
End sub

HypSetColItems

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description

HypSetColItems() sets the members for the nth column dimension for the dynamic link query. If the nth column does not exist, a new column is appended.

Note: It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

Syntax

HypSetColItems (vtColID, vtDimensionName, ParamArray MemberList())

ByVal vtColID As Variant
ByVal vtDimensionName As Variant
ParamArray MemberList() As Variant

Parameters

vtColID: The column number n

vtDimensionName: The dimension name
ParamArray MemberList: The list of member names

Return Value
Long. Returns 0 if successful, otherwise, returns the appropriate error code.

Example
Declare Function HypSetColItems Lib "HsAddin" (ByVal vtColID As Variant, ByVal vtDimensionName As Variant, ParamArray MemberList() As Variant) As Long
Sub Example_HypSetColItems()
    Dim vtGrid As Variant
    Sts = HypConnect(Empty, "Username", "Password", "SalesDemoBasic")
    Sts = HypRetrieve(Empty)
    Range("B2").Select
    Sts = HypGetSourceGrid(Empty, vtGrid)
    Sts = HypSetColItems(1, "Market", "East", "West", "South", "Central", "Market")
End Sub

HypGetPOVItems

Data provider types: Essbase, Planning (ad hoc only), Financial Management (ad hoc only), Hyperion Enterprise (ad hoc only)

Description
HypGetPOVItems() returns the dimensions in the POV and the currently selected member for each dimension.

Note: It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

Syntax
HypGetPOVItems(vtDimensionNames, vtPOVNames)
ByRef vtDimensionNames As Variant
ByRef vtPOVNames As Variant

Parameters
vtDimensionNames: The dimension names in the POV
vtPOVNames: The currently selected member for each dimension in the POV.

Return Value
Returns 0 if successful; otherwise, returns the appropriate error code.
Example

Declare Function HypGetPOVItems Lib "HsAddin" (ByRef vtDimensionNames As Variant, ByRef vtPOVNames As Variant) As Long

Sub Example_HypGetPOVItems()
    Dim vtGrid As Variant
    Dim vtDimNames As Variant
    Dim vtPOVNames As Variant
    Sts = HypConnect(Empty, "UserName", "Password", "MyDemoBasic")
    Sts = HypRetrieve(Empty)
    Range ("B2").Select
    Sts = HypGetSourceGrid (Empty, vtGrid)
    Sts = HypGetPOVItems (vtDimNames, vtPOVNames)
End sub

HypSetPOVItems

Data provider types: Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (ad hoc only), Oracle Hyperion Enterprise® (ad hoc only)

Description

HypSetPOVItems() sets the POV dimensions for the dynamic link query.

Note: It is assumed that a call has already been made to HypGetSourceGrid to initialize the dynamic link query, which contains the information about the active data provider and the grid on the worksheet.

Syntax

HypSetPOVItems (ParamArray MemberList())

ParamArray MemberList() As Variant

Parameters

ParamArray MemberList(): The list of desired POV items in the form Dimension#Current Member

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Example

Declare Function HypSetPOVItems Lib "HsAddin" (ParamArray MemberList() As Variant) As Long
Sub Example_HypSetPOVItems()
    Dim vtGrid As Variant
    Sts = HypConnect(Empty, "UserName", "Password", "MyDemoBasic")
    Sts = HypRetrieve(Empty)
    Range ("B2").Select
    Sts = HypGetSourceGrid (Empty, vtGrid)
End sub
Sts = HypGetSourceGrid (Empty, vtGrid)
Sts = HypSetPOVItems ("Scenario#Scenario", "Measures#Measures")
End sub
About MDX

Multidimensional Expressions (MDX) language is used to develop scripts or applications to query and report against data and metadata in Essbase databases. For information about MDX, see the Essbase documentation set.

“HypExecuteMDXEx” on page 189

HypExecuteMDXEx

Data provider types: Oracle Essbase

Description

HypExecuteMDXEx() executes an MDX query whose results are output in a data structure but are not displayed on the worksheet. (If you want to display the query results on a worksheet, use HypExecuteQuery instead.)

Syntax

HypExecuteMDXEx
()
ByVal vtSheetName As Variant,
ByVal vtQuery As Variant,
ByVal vtBoolHideData As Variant,
ByVal vtBoolDataLess As Variant,
ByVal vtBoolNeedStatus As Variant,
ByVal vtMbIDType As Variant,
ByVal vtAliasTable As Variant,
ByRef outResult As MDX_AXES_NATIVE
) As Long
Parameters

vtSheetName: The name of worksheet on which to run the function. If vtSheetName is Null or Empty, the active worksheet is used.

vtQuery: The MDX query to be executed

vtBoolHideData: The Boolean flag to hide or unhide data in the result

vtBoolDataLess: The Boolean flag to get or avoid data in the result

vtBoolNeedStatus: The Boolean flag to get or avoid status info in the result

vtMbrIDType: The member type identifier for the result (name or alias)

vtAliasTable: The alias table to be used

outResult: Pointer to a structure of type MDX_AXES. It contains the query output. (See Data Types Specific to HypExecuteMDXEx for data types and support functions for this API.)

Return Value

Returns 0 if successful; otherwise, returns the appropriate error code.

Data Types Specific to HypExecuteMDXEx

The following data types apply exclusively to HypExecuteMDXEx:

MDX_CELL: The data type corresponding to a cell

MDX_PROPERTY: The data type containing properties info for members and dimensions

MDX_MEMBER: The data type for members information

MDX_DIMENSION: The data type for dimensions information

MDX_CLUSTER: The data type for cluster information

MDX_AXIS: The data type representing an axis

MDX_AXES: The root level structure containing a collection of axes and cells

MDX_AXES_NATIVE: The data type used as an out parameter for HypExecuteMDXEx. This structure should be converted to MDX_AXES using procedure GetVBCompatibleMDXStructure.

Example

Sub GetVBCompatibleMDXStructure(ByRef inStruct As MDX_AXES_NATIVE, ByRef outStruct As MDX_AXES)

Public Declare Function HypExecuteMDXEx Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal vtQuery As Variant, ByVal vtBoolHideData As Variant, ByVal vtBoolDataLess As Variant, ByVal vtBoolNeedStatus As Variant, ByVal vtMbrIDType As Variant, ByVal vtAliasTable As Variant, ByRef outResult As MDX_AXES_NATIVE) As Long

Sub Example_HypExecuteMDXEx()

Dim Query As Variant
Dim vtBoolHideData As Variant

190 MDX Query Functions
Dim vtBoolDataLess As Variant
Dim vtBoolNeedStatus As Variant
Dim vtMbrIDType As Variant
Dim vtAliasTable As Variant
Dim result_Native As MDX_AXES_NATIVE
Dim result_VBCompatible As MDX_AXES

Query = "select {Jan} on COLUMNS, {Profit} on ROWS from Sample.Basic"
vtBoolHideData = True
vtBoolDataLess = True
vtBoolNeedStatus = True
vtMbrIDType = "alias"
vAliasTable = "none"

sts = HypConnect(Empty, "UserName", "Password", "SB")

If sts = 0 Then

sts = HypExecuteMDXEx(Empty, Query, vtBoolHideData, vtBoolDataLess, vtBoolNeedStatus, vtMbrIDType, vtAliasTable, result_Native)
sts = GetVBCompatibleMDXStructure(result_Native, result_VBCompatible)
sts = HypDisconnect(Empty, True)
Else
End If
End Sub
About Oracle BI EE Functions

The VBA functions in this chapter support Smart View operations when connected to an Oracle BI EE data source.

Preparing to Work with Oracle BI EE Functions

Before you begin creating and editing VBA functions for Oracle BI EE, you must first add references to the Oracle Smart View BI Extension type library and Oracle Hyperion Smart View for Office type library.

To add Oracle Smart View BI Extension and Smart View references:

1. Start the Visual Basic Editor from a Microsoft Office application; for example, from Excel.
2. Select Tools, then References.
3. In Available References, check the following items:
   - Oracle Smart View BI Extension
   - Oracle SmartView RC 1.0 Type Library
4. Click OK.

Continue with "Instantiating an Oracle Smart View BI Extension Object" on page 194.
Instantiating an Oracle Smart View BI Extension Object

The Oracle Smart View BI Extension exposes its automation interface through COM interface. To make an automation call to Oracle Smart View BI Extension, an Oracle Smart View BI Extension COM object must first be instantiated.

All Oracle BI EE automation functions are defined in the IBIReport interface, and the SmartViewOBIEEAutomation class implements those functions. Therefore, in any Oracle BI EE automation call, you must include the variable declarations that are described in the following procedure.

To create the variable declarations that will be included in all functions:

1. Declare a variable of type `IBIReport`.
2. Set the variable to an object of type `SmartViewOBIEEAutomation`.

   The resulting lines are:
   ```vba
   Dim obiee As IBIReport
   Set obiee = New SmartViewOBIEEAutomation
   ```

3. Include the lines from step 2 in each of your functions.

You are ready to begin creating and working with the Oracle Smart View BI Extension functions. See “Oracle Smart View BI Extension Functions” on page 194 for a complete listing of the functions available and their usage.

Oracle Smart View BI Extension Functions

Subtopics

- InsertView
- EditPrompts
- EditPagePrompts
- GetPagePrompts
- DeleteView
- AnalysisProperties
- DirProperties
- InvokeMenu
- CopyView
- PasteView

InsertView

Description

Insert an Oracle BI EE view into an Office application.
Syntax
Function InsertView(
    connectionContext As String,
    sourcePath As String,
    viewName As String,
    prompt() As BIReportPrompt,
    format As SVREPORT_RENDER_FORMAT,
    insertOption As SVREPORT_COMPOUND_VIEW_INSERT_OPTION) As Boolean

Parameters

connectionContext: The Oracle BI EE provider URL.

sourcePath: The location of the view in the Oracle BI EE Catalog.
To express the path of the view, in a web browser, access the Oracle BI EE Catalog, navigate to
the view folder, and note the URL of the folder. The path of the folder can then be derived after
decoding the folder URL (which is encoded with URL encoding). To specify a location of the
view, include the analysis name in the path. For example, in the browser, the URL of a folder in
Oracle BI EE is:

http://xxxx.com:xxxx/analytics/saw.dll?catalog#%7B%22location%22%3A%22%2Fusers
%2Fadministrator%2Fsvc_auto_bugs%22%7D

Decoding the URL and the URL is changed to:

http://xxxx.com:xxxx/analytics/saw.dll?catalog#{"location":"/users/administrator/
svc_auto_bugs"}

After getting the folder path, append the analysis name to the path. In the end, the path looks
like:

/users/administrator/svc_auto_bugs/AnalysisName

viewName: The name of the view.

prompt: The prompts for inserting the view.
Prompts are an array of BIReportPrompt. BIReportPrompt is a class with only one member
which is an array of strings. All prompt input should be converted to strings. The order of the
BIReportPrompt array should be same as the order of the prompts in the Prompt Selector dialog
box.

For example, to specify prompt values for the prompts in the Figure 1, you must create an array
of four BIReportPrompts:

- The first element contains the selection for “D1 Office”
- The second element is for “1 - Revenue”
- The third element is for “P3 LOB”
- The fourth element is for “T00 Calendar Date”
The sample code follows Figure 1.

**Figure 1**  Prompt Selector Dialog Box with Selections for Office, Line of business, and Calendar Date

```
Dim prompts(0 To 3) As BIReportPrompt

Dim firstPrompt(0 To 3) As String
firstPrompt(0) = "Madison Office"
firstPrompt(1) = "Merrimon Office"
firstPrompt(2) = "Spring Office"
firstPrompt(3) = "Tellaro Office"
prompts(0).Values = firstPrompt

Dim secondPrompt(0 To 0) As String
secondPrompt(0) = "500"
prompts(1).Values = secondPrompt

Dim ThirdPrompt(0 To 5) As String
ThirdPrompt(0) = "Communication"
ThirdPrompt(1) = "Digital"
ThirdPrompt(2) = "Electronics"
ThirdPrompt(3) = "Games"
ThirdPrompt(4) = "Services"
ThirdPrompt(5) = "TV"
prompts(2).Values = ThirdPrompt

Dim FourthPrompt(0 To 0) As String
ForthPrompt(0) = "5/15/2009"
prompts(3).Values = ForthPrompt

format: The format to be rendered. Valid render format values are described in Table 7.
```
Table 7  Render Formats and View Types

<table>
<thead>
<tr>
<th>Render Format Value</th>
<th>View Types to be Used</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default_Format</td>
<td>All Views</td>
</tr>
<tr>
<td>ExcelPivot</td>
<td>Pivot Table View Only</td>
</tr>
<tr>
<td>ExcelTable</td>
<td>Table View Only</td>
</tr>
<tr>
<td>Image</td>
<td>Chart View Only</td>
</tr>
</tbody>
</table>

**insertOption**: For compound views only. This option specifies how to insert all the views in a compound view and is ignored for individual views.

Valid values:
- **NewSheet**—Inserts each view in the compound view in a new sheet.
- **SameSheet**—Inserts each view in the compound view in the same sheet.

**Return Value**
Indicates if the operation succeeds or not.

**Example**
Sub InsertTableTest()

Dim obiee As IBIReport
Set obiee = New SmartViewOBIEEAutomation

Dim prompts() As BIReportPrompt


End Sub

Sub InsertPromptTableTest()

Dim obiee As IBIReport
Set obiee = New SmartViewOBIEEAutomation

Dim prompts(0 To 3) As BIReportPrompt

Dim firstPrompt(0 To 3) As String
firstPrompt(0) = "Madison Office"
firstPrompt(1) = "Merrimon Office"
firstPrompt(2) = "Spring Office"
firstPrompt(3) = "Tellaro Office"
prompts(0).Values = firstPrompt

Dim secondPrompt(0 To 0) As String
secondPrompt(0) = "500"
prompts(1).Values = secondPrompt

Dim ThirdPrompt(0 To 5) As String
ThirdPrompt(0) = "Communication"
ThirdPrompt(1) = "Digital"
ThirdPrompt(2) = "Electronics"
ThirdPrompt(3) = "Games"
ThirdPrompt(4) = "Services"
ThirdPrompt(5) = "TV"
prompts(2).Values = ThirdPrompt

Dim FourthPrompt(0 To 0) As String
ForthPrompt(0) = "5/15/2009"
prompts(3).Values = ForthPrompt

obiee.InsertView "http://xxx.com:xxxx/analytics/jbips","/shared/SmartView/SV_Michelle/promptAllTypes", "tableView!1", prompts, Default_Format, SameSheet

End Sub

**EditPrompts**

**Description**
Edit prompts of a view.

**Syntax**
Function EditPrompts(
objID As String,
prompt() As BIReportPrompt
) As Boolean

**Parameters**
- **objID**: The ID of the view to be edited. If an empty ID is passed, the selected view will be used.
- **prompt**: Same as the “prompt” parameter in InsertView.

**Return Value**
Indicates if the operation succeeds or not.

**Example**
Sub EditPromptTableTest()

Dim obiee As IBIReport
Set obiee = New SmartViewOBIEEAutomation

Dim prompts(0 To 3) As BIReportPrompt

Dim firstPrompt(0 To 3) As String
firstPrompt(0) = "Madison Office"
firstPrompt(1) = "Merrimon Office"
firstPrompt(2) = "Spring Office"
firstPrompt(3) = "Tellaro Office"
prompts(0).Values = firstPrompt

Dim secondPrompt(0 To 0) As String
secondPrompt(0) = "500"
prompts(1).Values = secondPrompt

Dim ThirdPrompt(0 To 5) As String
ThirdPrompt(0) = "Communication"
ThirdPrompt(1) = "Digital"
ThirdPrompt(2) = "Electronics"
ThirdPrompt(3) = "Games"
ThirdPrompt(4) = "Services"
ThirdPrompt(5) = "TV"

prompts(2).Values = ThirdPrompt

Dim ForthPrompt(0 To 0) As String
ForthPrompt(0) = "8/15/2009"
prompts(3).Values = ForthPrompt

obiee.EditPrompts Empty, prompts

**EditPagePrompts**

**Description**
Edit the page selections of a view.

**Syntax**
Function EditPagePrompts(
    objID As String,
    pageSelections() As String
) As Boolean

**Parameters**

- **objID**: The IDid of the view to be edited. If an empty ID is passed, the selected view will be used.
- **pageSelections**: The order of the page selection stored in the string array should be same as the order the page selections appear in the Page Selector dialog box. For example, to specify the page selections shown in Figure 2, use the sample code that follows the figure.
Figure 2  Page Selector Dialog Box with Selections for Region and Year

![Page Selector Dialog Box](image)

Dim pageSelections(0 To 1) As String
pageSelections (0) = "CENTRAL REGION"
pageSelections (1) = "2000"

**Return Value**
Indicates if the operation succeeds or not.

**Example**

Sub EditPagePromptTest()

Dim obiee As IBIReport
Set obiee = New SmartViewOBIEEAutomation
Dim pages(0 To 1) As String
pages(0) = "CENTRAL REGION"
pages(1) = "2000"

obiee.EditPagePrompts Empty, pages

End Sub

**GetPagePrompts**

**Description**
Get page selections of a view.

**Syntax**
Function GetPagePrompts(
objID As String,
GetPagePrompts() As Boolean

Parameters

objID: The ID of the view to get page selections from. If an empty ID is passed, the selected view will be used.

PageEdges: An output argument. Returns names of the page edges of the view.

PageSelections: An output argument. Returns the selected page values.

Return Value
Indicates if the operation succeeds or not.

Example

Sub TestGetPage()
    Dim obiee As IBIReport
    Set obiee = New SmartViewOBIEEAutomation

    Dim dims() As String
    Dim pageSelections() As String

    obiee.GetPagePrompts Empty, dims, pageSelections
End Sub

DeleteView

Description
Delete a view in an Office application.

Syntax

Function DeleteView( objID As String ) As Boolean

Parameters

objID: The ID of the view to be deleted. If an empty ID is passed, the selected view will be used.

Return Value
Indicates if the operation succeeds or not.

Example

Sub DeleteViewTest()
Dim obiee As IBIReport
Set obiee = New SmartViewOBIEEAutomation
obiee.DeleteView Empty

End Sub

**AnalysisProperties**

**Description**
Fetch the properties of an analysis.

**Syntax**
Function AnalysisProperties(
    connectionContext As String,
    sourcePath As String,
    analysisName As String
) As SVReportProperty()

**Parameters**
- **connectionContext**: The Oracle BI EE provider URL.
- **sourcePath**: The path of the analysis.
- **analysisName**: The name of the analysis.

**Return Value**
An array of SVReportProperty. Each element in the array represents one property of the analysis. SVReportProperty’s name member contains the name of the property, and the value member contains the value of the property.

**Example**
Sub TestAnalysisProp()

    Dim BIReport As IBIReport
    Set BIReport = New SmartViewOBIEEAutomation

    Dim result As Variant

    result = BIReport.AnalysisProperties("http://xxx.com:xxxx/analytics/jbips","/shared/SmartView/OBIEE", "reshma")

End Sub
**DirProperties**

**Description**
Fetch properties of a directory

**Syntax**
Function DirProperties (  
connectionContext As String,  
sourcePath As String,  
) As SVReportProperty()

**Parameters**
connectionContext: The Oracle Business Intelligence Enterprise Edition provider URL.  
sourcePath: The path of the directory.

**Return Value**
Same as the return values of AnalysisProperties. An array of SVReportProperty. Each element in the array represents one property of the analysis. SVReportProperty’s name member contains the name of the property, and the value member contains the value of the property.

**Example**
Sub TestDirProp()  
Dim BIReport As IBIReport  
Set BIReport = New SmartViewOBIEEAutomation  
Dim result As Variant  
result = BIReport.DirProperties("http://xxx.com:xxxx/analytics/jbips","/shared/SmartView/OBIEE/Yogini")  
End Sub

**InvokeMenu**

**Description**
Invoke Smart View Oracle BI EE extension menu.

**Syntax**
Sub InvokeMenu(  
menuID As String  
)
Parameters

menuID: The ID of the menu items. Valid values are listed in Table 8.

Table 8  Oracle BI EE Menu Items and IDs

<table>
<thead>
<tr>
<th>Menu</th>
<th>ID</th>
</tr>
</thead>
<tbody>
<tr>
<td>View Designer</td>
<td>ViewDesigner</td>
</tr>
<tr>
<td>Publish View</td>
<td>PublishView</td>
</tr>
<tr>
<td>Refresh</td>
<td>Refresh</td>
</tr>
<tr>
<td>Edit Prompts</td>
<td>EditPrompts</td>
</tr>
<tr>
<td>Edit Page Prompts</td>
<td>EditPage</td>
</tr>
<tr>
<td>Copy</td>
<td>CopyView</td>
</tr>
<tr>
<td>Paste</td>
<td>PasteView</td>
</tr>
<tr>
<td>Delete</td>
<td>DeleteView</td>
</tr>
<tr>
<td>Mask Data</td>
<td>MaskView</td>
</tr>
<tr>
<td>Mask Document Data</td>
<td>MaskDocumentView</td>
</tr>
</tbody>
</table>

Example

Sub TestMenuInvoke()
    Dim obiee As IBIReport
    Set obiee = New SmartViewOBIEEAutomation

    obiee.InvokeMenu "ViewDesigner"
End Sub

CopyView

The CopyView function is not supported in the current release.

PasteView

The PasteView function is not supported in the current release.
Index

Numbers
64-bit installations
   declarations in, 14

A
ad hoc functions
   about, 77
   HypExecuteQuery, 82
   HypGetNameRangeList, 81
   HypKeepOnly, 86
   HypPerformAdhocOnForm, 77
   HypPivot, 84
   HypPivotToGrid, 85
   HypPivotToPOV, 85
   HypRemoveOnly, 87
   HypRetrieve, 78
   HypRetrieveAllWorkbooks, 82
   HypRetrieveNameRange, 80
   HypRetrieveRange, 79
   HypSubmitData, 83
   HypZoomIn, 88
   HypZoomOut, 90
   assumed knowledge, 13

C
calculation script and business rule functions
   about, 119
   HypDeleteCalc, 127
   HypExecuteCalcScript, 120
   HypExecuteCalcScriptEx, 122
   HypListCalcScripts, 119

   calculation, consolidation, and translation functions
   about, 129
   HypCalculate, 129
   HypCalculateContribution, 130
   HypConsolidate, 131
   HypConsolidateAll, 131

   HypConsolidateAllWithData, 132
   HypForceCalculate, 133
   HypForceCalculateContribution, 133
   HypForceTranslate, 134
   HypTranslate, 135

   cell functions
   about, 93
   HypAddLRO, 102
   HypCell, 95
   HypDeleteAllLROs, 101
   HypDeleteLROs, 101
   HypExecuteDrillThroughReport, 106
   HypFreeDataPoint, 96
   HypGetCellRangeForMbrCombination, 97
   HypGetDataPoint, 98
   HypGetDimMbrsForDataCell, 93
   HypGetDrillThroughReports, 107
   HypIsCellWritable, 99
   HypListLROs, 104
   HypRetrieveLRO, 105
   HypSetCellsDirty, 100
   HypUpdateLRO, 103

   code samples
   copying and pasting, 13

   connection functions
   about, 61
   HypConnect, 61
   HypConnected, 63
   HypConnectionExists, 64
   HypCreateConnection, 65
   HypCreateConnectionEX, 66
   HypDisconnect, 68
   HypDisconnectAll, 69
   HypDisconnectEx, 69
   HypGetSharedConnectionsURL, 70
   HypInvalidateSSO, 72
   HypIsConnectedToSharedConnections, 71
   HypRemoveConnection, 72
HypResetFriendlyName, 72
HypSetActiveConnection, 73
HypSetAsDefault, 74
HypSetConnAliasTable, 75
HypSetSharedConnectionsURL, 70
HypUIConnect, 62

D
default, 15
dynamic link functions
  HypDisplayToLinkView, 175
  HypGetColCount, 180
  HypGetColItems, 183
  HypGetConnectionInfo, 176
  HypGetLinkMacro, 174
  HypGetPOVCount, 180
  HypGetPOVItems, 185
  HypGetRowCount, 179
  HypGetRowItems, 181
  HypGetSourceGrid, 174
  HypSetCollItems, 184
  HypSetConnectionInfo, 178
  HypSetLinkMacro, 173
  HypSetPOVItems, 186
  HypSetRowItems, 182
  HypUseLinkMacro, 172
dynamic link views
  about, 171
  macro execution, 172
  setting up, 172

Eempty, as default value, 15

Fform functions
  HypOpenForm, 91
forms, 91

Ggeneral functions
  about, 45
  HypCopyMetaData, 50
  HypDeleteMetadata, 50
  HypGetDatabaseNote, 60
  HypGetLastError, 48
  HypGetSubstitutionVariable, 57
  HypGetVersion, 46
  HypIsDataModified, 51
  HypIsFreeForm, 53
  HypIsSmartViewContentPresent, 52
  HypPreserveFormatting, 55
  HypRedo, 54
  HypRemovePreservedFormats, 56
  HypSetAliasTable, 57
  HypSetMenu, 49
  HypSetSubstitutionVariable, 59
  HypShowPanel, 45
  HypShowPov, 48
  HypUndo, 54

L
Legacy VBA applications, migrating to Smart View, 18

MMDX
  about, 189
MDX query functions
  HypExecuteMDXEx, 189
member query functions
  about, 137
  HypFindMember, 137
  HypFindMemberEx, 139
  HypGetAncestor, 140
  HypGetChildren, 141
  HypGetMemberInformation, 152
  HypGetMemberInformationEX, 155
  HypGetParent, 142
  HypIsAncestor, 144
  HypIsAttribute, 142
  HypIsChild, 147
  HypIsDescendant, 143
  HypIsExpense, 145
  HypIsParent, 146
  HypIsUDA, 147
  HypOtlGetMemberInfo, 148
menu functions
  about, 22
  HypExecuteMenu, 41
Multidimensional Expressions. See MDX

null, as default value, 15
Spreadsheet Toolkit VBA applications, migrating to Smart View, 18
VBA function types, 18
VBA functions
  about, 13
  HypGetMembers, 113
  HypListCalcScriptsEx, 121
  HypSetBackgroundPOV, 111
VBA functions file, 13
VBA functions, alphabetical
  AnalysisProperties, 202
  CopyView, 204
  DirProperties, 203
  EditPagePrompts, 199
  EditPrompts, 198
 getPagePrompts, 200
  HypAddLRO, 102
  HypCalculate, 129
  HypCalculateContribution, 130
  HypCell, 95
  HypConnect, 61
  HypConnected, 63
  HypConnectionExists, 64
  HypConsolidate, 131
  HypConsolidateAll, 131
  HypConsolidateAllWithData, 132
  HypCopyMetaData, 50
  HypCreateConnection, 65
  HypCreateConnectionEx, 66
  HypDeleteAllLROs, 101
  HypDeleteAllMRUItems, 168
  HypDeleteCalc, 127
  HypDeleteLROs, 101
  HypDeleteMetadata, 50
  HypDisconnect, 68
  HypDisconnectAll, 69
  HypDisconnectEx, 69
  HypDisplayToLinkView, 175
  HypExecuteCalcScript, 120
  HypExecuteCalcScriptEx, 122
  HypExecuteDrillThroughReport, 106
  HypExecuteMDXEx, 189
  HypExecuteMenu, 41
  HypExecuteQuery, 82
  HypFindMember, 137
  HypFindMemberEx, 139
  HypForceCalculate, 133
  HypForceCalculateContribution, 133
  HypForceTranslate, 134
  HypFreeDataPoint, 96
  HypGetActiveMember, 115
  HypGetAncestor, 140
  HypGetBackgroundPOV, 110
  HypGetCellRangeForMbrCombination, 97
  HypGetChildren, 141
  HypGetColCount, 180
  HypGetColItems, 183
  HypGetConnectionInfo, 176
  HypGetDatabaseNote, 60
  HypGetDataSource, 98
  HypGetDimensions, 116
  HypGetDimMbrsForDataCell, 93
  HypGetDrillThroughReports, 107
  HypGetGlobalOption, 157
  HypGetLastError, 48
  HypGetLinkMacro, 174
  HypGetMemberInformation, 152
  HypGetMemberInformationEx, 155
  HypGetMembers, 113
  HypGetNameRangeList, 81
  HypGetOption, 163
  HypGetPagePOVChoices, 111
  HypGetParent, 142
  HypGetPOVCount, 180
  HypGetPOVItems, 185
  HypGetRowCount, 179
  HypGetRowItems, 181
  HypGetSharedConnectionsURL, 70
  HypGetSheetOption, 160
  HypGetSourceGrid, 174
  HypGetSubstitutionVariable, 57
  HypGetVersion, 46
  HypHideRibbonMenu, 42
  HypHideRibbonMenuReset, 43
  HypInvalidateSSO, 72
  HypIsAncestor, 144
  HypIsAttribute, 142
  HypIsCellWritable, 99
  HypIsChild, 147
  HypIsConnectedToSharedConnections, 71
  HypIsDataModified, 51
  HypIsDescendant, 143
| HypIsExpense, 145 |
| HypIsFreeForm, 53 |
| HypIsParent, 146 |
| HypIsSmartViewContentPresent, 52 |
| HypIsUDA, 147 |
| HypKeepOnly, 86 |
| HypListCalcScripts, 119 |
| HypListCalcScriptsEx, 121 |
| HypListLROs, 104 |
| HypMenuVAbout, 22 |
| HypMenuVAdjust, 23 |
| HypMenuVBusinessRules, 23 |
| HypMenuVCalculation, 23 |
| HypMenuVCascadeNewWorkbook, 24 |
| HypMenuVCascadeSameWorkbook, 24 |
| HypMenuVCellText, 25 |
| HypMenuVCollapse, 25 |
| HypMenuVConnect, 26 |
| HypMenuVCopyDataPoints, 26 |
| HypMenuVExpand, 27 |
| HypMenuVFunctionBuilder, 27 |
| HypMenuVInstruction, 28 |
| HypMenuVKeepOnly, 28 |
| HypMenuVLRO, 29 |
| HypMenuVMemberInformation, 29 |
| HypMenuVMemberSelection, 29 |
| HypMenuVMigrate, 30 |
| HypMenuVOptions, 31 |
| HypMenuVPasteDataPoints, 31 |
| HypMenuVPivot, 32 |
| HypMenuVPOVManager, 32 |
| HypMenuVQueryDesigner, 33 |
| HypMenuVRedo, 33 |
| HypMenuVRefresh, 34 |
| HypMenuVRefreshAll, 34 |
| HypMenuVRefreshOfflineDefinition, 34 |
| HypMenuVRemoveOnly, 35 |
| HypMenuVRulesOnForm, 35 |
| HypMenuVRunReport, 36 |
| HypMenuVSelectForm, 36 |
| HypMenuVShowHelpHtml, 37 |
| HypMenuVSubmitData, 37 |
| HypMenuVSupportingDetails, 38 |
| HypMenuVSyncBack, 38 |
| HypMenuVTakeOffline, 39 |
| HypMenuVUndo, 39 |
| HypMenuVZoomIn, 40 |
| HypMenuVZoomOut, 41 |
| HypOpenForm, 91 |
| HypOtlGetMemberInfo, 148 |
| HypPerformAdhocOnForm, 77 |
| HypPivot, 84 |
| HypPivotToGrid, 85 |
| HypPivotToPOV, 85 |
| HypPreserveFormatting, 55 |
| HypQueryMembers, 150 |
| HypRedo, 54 |
| HypRemoveConnection, 72 |
| HypRemoveOnly, 87 |
| HypRemovePreservedFormats, 56 |
| HypResetFriendlyName, 72 |
| HypRetrieve, 78 |
| HypRetrieveAllWorkbooks, 82 |
| HypRetrieveLRO, 105 |
| HypRetrieveNameRange, 80 |
| HypRetrieveRange, 79 |
| HypSetActiveConnection, 73 |
| HypSetActiveMember, 116 |
| HypSetAliasTable, 57 |
| HypSetAsDefault, 74 |
| HypSetCellsDirty, 100 |
| HypSetCollItems, 184 |
| HypSetConnAliasTable, 75 |
| HypSetConnectionInfo, 178 |
| HypSetDimensions, 117 |
| HypSetGlobalOption, 159 |
| HypSetLinkMacro, 173 |
| HypSetMembers, 114 |
| HypSetMenu, 49 |
| HypSetOption, 168 |
| HypSetPages, 112 |
| HypSetPOV, 109 |
| HypSetPOVItems, 186 |
| HypSetRowItems, 182 |
| HypSetSharedConnectionsURL, 70 |
| HypSetSheetOption, 162 |
| HypSetSubstitutionVariable, 59 |
| HypShowPanel, 45 |
| HypShowPov, 48 |
| HypSubmitData, 83 |
| HypTranslate, 135 |
| HypUIConnect, 62 |
| HypUndo, 54 |
| HypUpdateLRO, 103 |
HypUseLinkMacro, 172
HypZoomIn, 88
HypZoomOut, 90
InsertView, 194
InvokeMenu, 203
PasteView, 204