

# Oracle<sup>®</sup> Smart View for Office Developer's Guide



Release 20.100  
F32631-01  
June 2020

The Oracle logo, consisting of a solid red square with the word "ORACLE" in white, uppercase, sans-serif font centered within it.

ORACLE<sup>®</sup>

Oracle Smart View for Office Developer's Guide, Release 20.100

F32631-01

Copyright © 2004, 2020, Oracle and/or its affiliates.

Primary Author: EPM Information Development Team

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish, or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

If this is software or related documentation that is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, then the following notice is applicable:

U.S. GOVERNMENT END USERS: Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs) and Oracle computer documentation or other Oracle data delivered to or accessed by U.S. Government end users are "commercial computer software" or "commercial computer software documentation" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, reproduction, duplication, release, display, disclosure, modification, preparation of derivative works, and/or adaptation of i) Oracle programs (including any operating system, integrated software, any programs embedded, installed or activated on delivered hardware, and modifications of such programs), ii) Oracle computer documentation and/or iii) other Oracle data, is subject to the rights and limitations specified in the license contained in the applicable contract. The terms governing the U.S. Government's use of Oracle cloud services are defined by the applicable contract for such services. No other rights are granted to the U.S. Government.

This software or hardware is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications that may create a risk of personal injury. If you use this software or hardware in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy, and other measures to ensure its safe use. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software or hardware in dangerous applications.

Oracle and Java are registered trademarks of Oracle and/or its affiliates. Other names may be trademarks of their respective owners.

Intel and Intel Inside are trademarks or registered trademarks of Intel Corporation. All SPARC trademarks are used under license and are trademarks or registered trademarks of SPARC International, Inc. AMD, Epyc, and the AMD logo are trademarks or registered trademarks of Advanced Micro Devices. UNIX is a registered trademark of The Open Group.

This software or hardware and documentation may provide access to or information about content, products, and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third-party content, products, and services unless otherwise set forth in an applicable agreement between you and Oracle. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third-party content, products, or services, except as set forth in an applicable agreement between you and Oracle.

# Contents

Documentation Accessibility

---

Documentation Feedback

---

## 1 About the Oracle Smart View for Office Developer's Guide

---

## 2 About VBA Functions

---

Assumed Knowledge	2-1
VBA Functions Location	2-1
Using VBA Function Code Samples	2-1
VBA Functions in 64-Bit Versions	2-2
VBA Parameters	2-2
VBA Return Values (Error Codes)	2-3
Using Spreadsheet Toolkit VBA Applications in Smart View	2-6
VBA Function Types	2-6
VBA Functions—Alphabetical List	2-7

## 3 Menu Functions

---

About Menu Functions	3-2
HypMenuVAbout	3-2
HypMenuVAdjust	3-3
HypMenuVBusinessRules	3-3
HypMenuVCalculation	3-4
HypMenuVCascadeNewWorkbook	3-4
HypMenuVCascadeSameWorkbook	3-5
HypMenuVCellText	3-5
HypMenuVCollapse	3-6
HypMenuVConnect	3-6
HypMenuVCopyDataPoints	3-7

HypMenuVExpand	3-7
HypMenuVFunctionBuilder	3-8
HypMenuVInstruction	3-8
HypMenuVKeepOnly	3-9
HypMenuVLRO	3-9
HypMenuVMemberInformation	3-10
HypMenuVMemberSelection	3-10
HypMenuVMigrate	3-11
HypMenuVOptions	3-11
HypMenuVPasteDataPoints	3-12
HypMenuVPivot	3-12
HypMenuVPOVManager	3-13
HypMenuVQueryDesigner	3-13
HypMenuVRedo	3-14
HypMenuVRefresh	3-14
HypMenuVRefreshAll	3-15
HypMenuVRefreshOfflineDefinition	3-15
HypMenuVRemoveOnly	3-16
HypMenuVRulesOnForm	3-16
HypMenuVRunReport	3-17
HypMenuVSelectForm	3-17
HypMenuVShowHelpHtml	3-18
HypMenuVSubmitData	3-19
HypMenuVSupportingDetails	3-19
HypMenuVSyncBack	3-20
HypMenuVTakeOffline	3-20
HypMenuVUndo	3-20
HypMenuVVisualizeinExcel	3-21
HypMenuVZoomIn	3-22
HypMenuVZoomOut	3-22
HypExecuteMenu	3-23
HypHideRibbonMenu	3-24
HypHideRibbonMenuReset	3-25

## 4 General Functions

---

About General Functions	4-1
HypShowPanel	4-1
HypGetVersion	4-2
HypListApplications	4-4
HypListDatabases	4-5

HypGetLastError	4-6
HypShowPov	4-7
HypSetMenu	4-8
HypCopyMetaData	4-8
HypDeleteMetaData	4-9
HypIsDataModified	4-11
HypIsSmartViewContentPresent	4-11
HypIsFreeForm	4-12
HypUndo	4-13
HypRedo	4-14
HypPreserveFormatting	4-14
HypRemovePreservedFormats	4-15
HypSetAliasTable	4-17
HypGetSubstitutionVariable	4-17
HypSetSubstitutionVariable	4-18
HypGetDatabaseNote	4-19
HypGetSheetInfo	4-20
HypListDocuments	4-21

## 5 Connection Functions

---

About Connection Functions	5-1
HypConnect	5-1
HypUIConnect	5-3
HypConnected	5-5
HypConnectionExists	5-6
HypCreateConnection	5-7
HypCreateConnectionEx	5-9
HypSetCustomHeader	5-11
HypUpdateConnection	5-12
HypModifyConnection	5-15
HypDisconnect	5-16
HypDisconnectAll	5-17
HypDisconnectEx	5-17
HypGetSharedConnectionsURL	5-18
HypSetSharedConnectionsURL	5-19
HypIsConnectedToSharedConnections	5-19
HypRemoveConnection	5-20
HypSetSSO	5-21
HypInvalidateSSO	5-21
HypResetFriendlyName	5-22

HypSetActiveConnection	5-22
HypSetAsDefault	5-23
HypSetConnAliasTable	5-24

## 6 Ad Hoc Functions

---

About Ad Hoc Functions	6-1
HypPerformAdhocOnForm	6-1
HypRetrieve	6-2
HypRetrieveRange	6-3
HypRetrieveNameRange	6-4
HypCreateRangeGrid	6-5
HypModifyRangeGridName	6-7
HypGetNameRangeList	6-7
HypRetrieveAllWorkbooks	6-8
HypExecuteQuery	6-9
HypSubmitData	6-10
HypSubmitSelectedRangeWithoutRefresh	6-11
HypSubmitSelectedDataCells	6-12
HypPivot	6-13
HypPivotToGrid	6-14
HypPivotToPOV	6-15
HypKeepOnly	6-16
HypRemoveOnly	6-17
HypZoomIn	6-18
HypZoomOut	6-20

## 7 Form Functions

---

About Forms	7-1
HypOpenForm	7-1

## 8 Cell Functions

---

About Cell Functions	8-1
HypGetDimMbrsForDataCell	8-1
HypCell	8-3
HypFreeDataPoint	8-4
HypGetCellRangeForMbrCombination	8-5
HypGetDataPoint	8-6
HypIsCellWritable	8-7
HypSetCellsDirty	8-8

HypDeleteAllLROs	8-9
HypDeleteLROs	8-10
HypAddLRO	8-11
HypUpdateLRO	8-12
HypListLROs	8-13
HypRetrieveLRO	8-14
HypExecuteDrillThroughReport	8-15
HypGetDrillThroughReports	8-16

## 9 POV Functions

---

About POV Functions	9-1
HypSetPOV	9-1
HypGetBackgroundPOV	9-2
HypSetBackgroundPOV	9-3
HypGetPagePOVChoices	9-4
HypSetPages	9-5
HypGetMembers	9-5
HypSetMembers	9-6
HypGetActiveMember	9-7
HypSetActiveMember	9-8
HypGetDimensions	9-9
HypSetDimensions	9-10

## 10 Calculation Script and Business Rule Functions

---

About Calculation Script and Business Rule Functions	10-1
HypListCalcScripts	10-1
HypExecuteCalcScript	10-2
HypListCalcScriptsEx	10-3
HypExecuteCalcScriptEx	10-4
HypGetCalcScript	10-9
HypExecuteCalcScriptString	10-10
HypDeleteCalc	10-11

## 11 Calculation, Consolidation, and Translation Functions

---

About Calculation, Consolidation, and Translation Functions	11-1
HypCalculate	11-1
HypCalculateContribution	11-2
HypConsolidate	11-2
HypConsolidateAll	11-3

HypConsolidateAllWithData	11-4
HypForceCalculate	11-4
HypForceCalculateContribution	11-5
HypForceTranslate	11-6
HypTranslate	11-6

## 12 Member Query Functions

---

About Member Query Functions	12-1
HypFindMember	12-1
HypFindMemberEx	12-3
HypGetAncestor	12-4
HypGetChildren	12-5
HypGetParent	12-6
HypIsAttribute	12-7
HypIsDescendant	12-8
HypIsAncestor	12-8
HypIsExpense	12-9
HypIsParent	12-10
HypIsChild	12-11
HypIsUDA	12-12
HypOtlGetMemberInfo	12-13
HypQueryMembers	12-14
HypGetMemberInformation	12-17
HypGetMemberInformationEx	12-19

## 13 Options Functions

---

About Options Functions	13-1
HypGetGlobalOption	13-1
HypSetGlobalOption	13-3
HypGetSheetOption	13-4
HypSetSheetOption	13-6
HypGetOption	13-7
HypSetOption	13-14
HypDeleteAllMRUItems	13-15

## 14 Dynamic Link Functions

---

About Dynamic Link Views	14-1
Setting Up Dynamic Link Views	14-2
Automating Macro Execution	14-2



HypUseLinkMacro	14-2
HypSetLinkMacro	14-3
HypGetLinkMacro	14-4
HypGetSourceGrid	14-5
HypDisplayToLinkView	14-6
HypGetConnectionInfo	14-7
HypSetConnectionInfo	14-8
HypGetRowCount	14-10
HypGetColCount	14-11
HypGetPOVCount	14-11
HypGetRowItems	14-12
HypSetRowItems	14-13
HypGetCollItems	14-15
HypSetCollItems	14-16
HypGetPOVItems	14-17
HypSetPOVItems	14-18

## 15 MDX Query Functions

---

About MDX	15-1
HypExecuteMDXEx	15-1

## 16 Oracle Journals for Financial Management Functions

---

About Oracle Journals for Financial Management Functions	16-1
Registering the Oracle Journals VBA Functions Using RegAsm	16-2
Preparing to Work with Oracle Journals for Financial Management Functions	16-3
Instantiating an Oracle Journals for Financial Management Extension Object	16-3
Oracle Journals for Financial Management Extension Functions	16-4
ListJournals	16-4
OpenJournal	16-6
SetJournalProperty	16-6
ListTemplates	16-8
CreateJournal	16-9
SaveJournal	16-10
PerformAction	16-12
ValidateJournal	16-13

## 17 Oracle Analytics Cloud Functions

---

About Oracle Analytics Cloud Functions	17-1
Preparing to Work with Oracle Analytics Cloud Functions	17-1

Instantiating an Oracle Smart View BI Extension Object	17-2
Oracle Smart View BI Extension Functions	17-2
InsertView	17-2
EditPrompts	17-6
EditPagePrompts	17-7
GetPagePrompts	17-8
Retrieve	17-9
DeleteView	17-10
AnalysisProperties	17-11
DirProperties	17-12
InvokeMenu	17-13
CopyView	17-13
PasteView	17-13

## 18 Oracle BI EE Functions

---

About Oracle BI EE Functions	18-1
Preparing to Work with Oracle BI EE Functions	18-1
Instantiating an Oracle Smart View BI Extension Object	18-2
Oracle Smart View BI Extension Functions	18-2
InsertView	18-2
EditPrompts	18-6
EditPagePrompts	18-7
GetPagePrompts	18-8
Retrieve	18-9
DeleteView	18-10
AnalysisProperties	18-11
DirProperties	18-12
InvokeMenu	18-13
CopyView	18-13
PasteView	18-13

# 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 that have purchased support 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.

# Documentation Feedback

To provide feedback on this documentation, send email to [epmdoc\\_ww@oracle.com](mailto:epmdoc_ww@oracle.com), or, in an Oracle Help Center topic, click the Feedback button located beneath the Table of Contents (you may need to scroll down to see the button).

Follow EPM Information Development on these social media sites:

LinkedIn - [http://www.linkedin.com/groups?gid=3127051&goback=.gmp\\_3127051](http://www.linkedin.com/groups?gid=3127051&goback=.gmp_3127051)

Twitter - <http://twitter.com/hyperionepminfo>

Facebook - <http://www.facebook.com/pages/Hyperion-EPM-Info/102682103112642>

YouTube - <https://www.youtube.com/oracleepminthecloud>

# 1

## About the *Oracle Smart View for Office Developer's Guide*

The *Oracle Smart View for Office Developer's Guide* describes the Microsoft Visual Basic for Applications (VBA) functions that you can use to develop applications for Oracle Smart View for Office. This reference is intended for advanced users who need detailed information and examples for supported VBA functions.

# 2

## About VBA Functions

### Related Topics

- [Assumed Knowledge](#)
- [VBA Functions Location](#)
- [Using VBA Function Code Samples](#)
- [VBA Functions in 64-Bit Versions](#)
- [VBA Parameters](#)
- [VBA Return Values \(Error Codes\)](#)
- [Using Spreadsheet Toolkit VBA Applications in Smart View](#)
- [VBA Function Types](#)
- [VBA Functions—Alphabetical List](#)

## Assumed Knowledge

You can customize and automate common tasks using Microsoft Visual Basic for Applications (VBA) functions in Oracle 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 Oracle Smart View for Office 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](#).

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

The `smartview.bas` file provided with your Oracle Smart View for Office 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 2-1 VBA Parameters**

Parameter	Value
Text	A word or phrase or name in quotation marks. For example: <ul style="list-style-type: none"> <li>• "Smart View"</li> <li>• "[Book2.xls]Sheet1"</li> </ul>
Boolean	<ul style="list-style-type: none"> <li>• True</li> <li>• False</li> </ul>
Range object	A cell, row or column, one or more selections of cells, or a three-dimensional range address, surrounded by quotation marks. For example: <ul style="list-style-type: none"> <li>• RANGE("A1")</li> <li>• RANGE("A1:B2")</li> <li>• RANGE("G:G,I:I,K:K")</li> <li>• RANGE("A1:B5,C1:C10,D5:L8")</li> <li>• RANGE("Sheet1!C3:R20,Sheet2!C3:R20")</li> </ul>
Number	A number without quotation marks and without commas. For example: <ul style="list-style-type: none"> <li>• 1</li> <li>• 2.5</li> <li>• 50000</li> </ul>
List of strings	A list of text values separated by commas. For example: "Qtr1", "Actual", "Oregon"
Constant	A predefined constant from <code>smartview.bas</code>

**Table 2-1 (Cont.) VBA Parameters**

Parameter	Value
Default value	<ul style="list-style-type: none"> <li>Null</li> <li>Empty</li> </ul>

 **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 (Error Codes)

Oracle Smart View for Office VBA functions may return any of the values in [Table 1](#) to indicate success or failure of the function.

- A return value of zero (0) indicates that the function ran successfully.
- A return value of 1 or 2 indicates that the function ran successfully, but with a condition.
- Negative return values indicate client issues.
- A return value of 4 indicates a server issue.

 **Note:**

Return values are often referred to as error codes.

**Table 2-2 Return Values (Error Codes) and Their Descriptions**

Return Value	Name	Description
4	SS_ERR_ERROR	An error specific to the data provider or a generic error that cannot be mapped to a value.
2	SS_NO_GRID_ON_SHEET_BUT_FUNCTIONS_SUBMITTED	The function ran successfully; however, the function sheet that was submitted contained no grid.
1	SS_SHEET_NOT_CONNECTED_BUT_FUNCTIONS_SUBMITTED	The function ran successfully; however, the function sheet that was submitted was not connected.
0	SS_OK	The function ran successfully.
-1	SS_INIT_ERR	Initialization error.
-2	SS_TERM_ERR	Termination error.
-3	SS_NOT_INIT	Initialization error.
-4	SS_NOT_CONNECTED	The spreadsheet is not yet connected to the server.



Table 2-2 (Cont.) Return Values (Error Codes) and Their Descriptions

Return Value	Name	Description
-5	SS_NOT_LOCKED	The spreadsheet is not locked.
-6	SS_INVALID_SSTABLE	The spreadsheet has become unstable.
-7	SS_INVALID_SSDATA	The spreadsheet contains invalid data.
-8	SS_NOUNDO_INFO	No Undo information exists.
-9	SS_CANCELED	Operation has been canceled.
-10	SS_GLOBALOPTS	Not used.
-11	SS_SHEETOPTS	Not used.
-12	SS_NOTENABLED	Undo is not enabled.
-13	SS_NO_MEMORY	Not enough memory resources are available.
-14	SS_DIALOG_ERROR	Appropriate dialog box could not be displayed.
-15	SS_INVALID_PARAM	Function contains an invalid parameter.
-16	SS_CALCULATING	Calculation is in progress.
-17	SS_SQL_IN_PROGRESS	Obsolete setting.
-18	SS_FORMULAPRESERVE	Operation is not allowed because the spreadsheet is in formula preservation mode.
-19	SS_INTERNALSSERROR	Operation cannot take place on the specified sheet.
-20	SS_INVALID_SHEET	Current sheet cannot be determined.
-21	SS_NOACTIVESHEET	Spreadsheet name was not specified and no active sheet is selected.
-22	SS_NOTCALCULATING	Calculation cannot be canceled because no calculation is running.
-23	SS_INVALIDSELECTION	Selection parameter is invalid.
-24	SS_INVALIDTOKEN	Not used.
-25	SS_CASCADENOTALLOWED	Cascade list file cannot be created, or you are attempting to cascade while the spreadsheet is embedded in another document.
-26	SS_NOMACROS	Spreadsheet macros cannot be run due to a licensing agreement.
-27	SS_NOREADONLYMACROS	Spreadsheet macros which update the database cannot be run due to a licensing constraint.
-28	SS_READONLYSS	You have a read-only license and cannot update the database.
-29	SS_NOSQLACCESS	Obsolete setting.
-30	SS_MENUALREADYREMOVED	The menu is removed already.
-31	SS_MENUALREADYADDED	The menu is added already.
-32	SS_NOSPREADSHEETACCESS	Not used.
-33	SS_NOHANDLES	Not used.
-34	SS_NOPREVCONNECTION	Not used.
-35	SS_LROERROR	Not used.
-36	SS_LROWINAPPACCESSERR	Not used.
-37	SS_DATANAVINITERR	Not used.
-38	SS_PARAMSETNOTALLOWED	Not used.
-39	SS_SHEET_PROTECTED	The specified worksheet is protected. Unprotect the worksheet and try the operation again.
-40	SS_CALCSCRIPT_NOTFOUND	Calc script not found.

**Table 2-2 (Cont.) Return Values (Error Codes) and Their Descriptions**

Return Value	Name	Description
-41	SS_NOSUPPORT_PROVIDER	Provider not supported.
-42	SS_INVALID_ALIAS	Invalid alias.
-43	SS_CONN_NOT_FOUND	Connection not found.
-44	SS_APS_CONN_NOT_FOUND	Provider Services connection not found.
-45	SS_APS_NOT_CONNECTED	Provider Services not connected.
-46	SS_APS_CANT_CONNECT	Provider Services cannot connect.
-47	SS_CONN_ALREADY_EXISTS	Connection already exists.
-48	SS_APS_URL_NOT_SAVED	Provider Services URL not saved.
-49	SS_MIGRATION_OF_CONN_NOT_ALLOWED	Migration of connection not allowed.
-50	SS_CONN_MGR_NOT_INITIALIZE D	Connection manager not initialized.
-51	SS_FAILED_TO_GET_APS_OVERRIDE_PROPERTY	Failed to get Provider Services override property.
-52	SS_FAILED_TO_SET_APS_OVERRIDE_PROPERTY	Failed to set Provider Services override property.
-53	SS_FAILED_TO_GET_APS_URL	Failed to get Provider Services URL.
-54	SS_APS_DISCONNECT_FAILED	Provider Services disconnect failed.
-55	SS_OPERATION_FAILED	Operation failed.
-56	SS_CANNOT_ASSOCIATE_SHEET_WITH_CONNECTION	Cannot associate sheet with connection.
-57	SS_REFRESH_SHEET_NEEDED	Worksheet refresh needed.
-58	SS_NO_GRID_OBJECT_ON_SHEET	No grid object on sheet.
-59	SS_NO_CONNECTION_ASSOCIATED	No connection associated.
-60	SS_NON_DATA_CELL_PASSED	Non-data cell passed.
-61	SS_DATA_CELL_IS_NOT_WRITEABLE	Data cell is not writable.
-62	SS_NO_SVC_CONTENT_ON_SHEET	No Smart View content on sheet.
-63	SS_FAILED_TO_GET_OFFICE_OBJECT	Failed to get Office object.
-64	SS_OP_FAILED_AS_CHART_IS_SELECTED	Operation failed because chart is selected.
-65	SS_EXCEL_IN_EDIT_MODE	Excel in edit mode.
-66	SS_SHEET_NON_SMARTVIEW_COMPATIBLE	Sheet not compatible with Smart View.
-67	SS_APP_NOT_STANDALONE	Application not stand alone.
-68	SS_SMART_VIEW_DISABLED	Smart View is disabled.
-69	SS_VBA_DEPRECATED	The function has been deprecated.
-70	SS_OPERATION_NOT_SUPPORTED_IN_MULTIGRID_MODE	The operation is not supported in worksheets that are in multiple-grid mode.
-71	SS_INVALID_MEMBER	The member name is invalid. Used with HypGetMemberInformation.
-72	SS_NO_SV_NAME_RANGE	No named ranges are available. Used with HypGetNameRangeList.

Table 2-2 (Cont.) Return Values (Error Codes) and Their Descriptions

Return Value	Name	Description
-73	SS_AMBIGUOUS_MENU	The menu item is ambiguous and could not be resolved. Used with HypExecuteMenu, HypHideRibbonMenu, and HypHideRibbonMenuReset.

## Using Spreadsheet Toolkit VBA Applications in Smart View

VBA applications created in Oracle Hyperion Essbase Spreadsheet Toolkit can be converted to Oracle Smart View for Office 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 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 Oracle Smart View for Office menu and ribbon. See [Menu Functions](#).
- **General** functions perform actions, set options, or retrieve information typically performed from the Smart View ribbon or Options dialog box. See [General Functions](#).
- **Connection** functions perform actions related to connections to data providers. See [Connection Functions](#).
- **Ad hoc** functions perform ad hoc operations such as zooming, retrieving and submitting data, and pivoting. See [Ad Hoc Functions](#).
- The **Form** function opens a data form. See [Form Functions](#).
- **Cell** functions perform operations and retrieve information for data cells and their contents. See [Cell Functions](#).
- **POV** functions specify or retrieve settings for the POV. See [POV Functions](#).
- **Calculation script and business rule** functions retrieve lists of or execute calculation scripts and business rules. See [Calculation Script and Business Rule Functions](#).
- **Calculation, consolidation, and translation** functions executes these operations on data for Oracle Hyperion Financial Management applications. See [Calculation, Consolidation, and Translation Functions](#).
- **Member query** functions retrieve generation, level, attribute, and other information about members. See [Member Query Functions](#).
- **Options** functions set and retrieve information for global and/or sheet options, and enable deletion of MRU items. See [Options Functions](#).
- **Dynamic link** functions set or retrieve data point details that are displayed in separate windows via dynamic links. See [Dynamic Link Functions](#).

- The **MDX query** function executes an MDX query whose results are not displayed in a worksheet. See [MDX Query Functions](#).
- **Oracle Journals** functions support Financial Management Journals functionality in Smart View. See [Oracle Journals for Financial Management Functions](#).
- Oracle Analytics Cloud functions support Smart View operations when connected to an Oracle Analytics Cloud data source. See [Oracle Analytics Cloud Functions](#)
- **Oracle BI EE** functions support Smart View operations when connected to an Oracle Business Intelligence Enterprise Edition data source. See [Oracle BI EE Functions](#).

 **Note:**

See also [VBA Functions—Alphabetical List](#).

## VBA Functions—Alphabetical List

**Table 2-3 VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">AnalysisProperties</a>	Oracle Analytics Cloud
<a href="#">AnalysisProperties</a>	Oracle Business Intelligence Enterprise Edition
<a href="#">CopyView</a>	Oracle Analytics Cloud
<a href="#">CopyView</a>	Oracle BI EE
<a href="#">CreateJournal</a>	Oracle Hyperion Financial Management and Oracle Journals Extension for Financial Management
<a href="#">DeleteView</a>	Oracle Analytics Cloud
<a href="#">DeleteView</a>	Oracle BI EE
<a href="#">DirProperties</a>	Oracle Analytics Cloud
<a href="#">DirProperties</a>	Oracle BI EE
<a href="#">EditPagePrompts</a>	Oracle Analytics Cloud
<a href="#">EditPagePrompts</a>	Oracle BI EE
<a href="#">EditPrompts</a>	Oracle Analytics Cloud
<a href="#">EditPrompts</a>	Oracle BI EE
<a href="#">GetPagePrompts</a>	Oracle Analytics Cloud
<a href="#">GetPagePrompts</a>	Oracle BI EE
<a href="#">HypAddLRO</a>	Oracle Analytics Cloud - Essbase, Oracle Essbase
<a href="#">HypCalculate</a>	Financial Management
<a href="#">HypCalculateContribution</a>	Financial Management (ad hoc only)

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypCell</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypConnect</a>	<ul style="list-style-type: none"> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypConnected</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypConnectionExists</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypConsolidate</a>	Financial Management (ad hoc only)
<a href="#">HypConsolidateAll</a>	Financial Management (ad hoc only)
<a href="#">HypConsolidateAllWithData</a>	Financial Management (ad hoc only)
<a href="#">HypCreateConnection</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> <li>• Oracle Hyperion Reporting and Analysis</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypCreateConnectionEx</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> <li>• Reporting and Analysis</li> </ul>
<a href="#">HypCopyMetadata</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypDeleteAllLROs</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypDeleteAllMRUIItems</a>	All
<a href="#">HypDeleteCalc</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypDeleteLROs</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypDeleteMetadata</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> <li>• Reporting and Analysis</li> </ul>
<a href="#">HypDisconnect</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypDisconnectAll</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypDisconnectEx</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Essbase</li> <li>Financial Management</li> <li>Oracle Hyperion Planning</li> </ul>
<a href="#">HypDisplayToLinkView</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Essbase</li> <li>Financial Management</li> <li>Oracle Hyperion Planning</li> </ul>
<a href="#">HypExecuteCalcScript</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypExecuteCalcScriptEx</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Essbase</li> <li>Oracle Hyperion Planning</li> </ul>
<a href="#">HypExecuteCalcScriptString</a>	Essbase
<a href="#">HypExecuteMenu</a>	All
<a href="#">HypExecuteDrillThroughReport</a>	Essbase
<a href="#">HypExecuteMDXEx</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypExecuteQuery</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypFindMember</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypFindMemberEx</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypForceCalculate</a>	Financial Management
<a href="#">HypForceCalculateContribution</a>	Financial Management (ad hoc only)
<a href="#">HypForceTranslate</a>	Financial Management (ad hoc only)

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypFreeDataPoint</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetActiveMember</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetAncestor</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypGetBackgroundPOV</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetCalcScript</a>	Essbase
<a href="#">HypGetCellRangeForMbrCombination</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetChildren</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypGetColCount</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>



**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypGetCollItems</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning (ad hoc only)</li> <li>Planning Modules (ad hoc only)</li> <li>Financial Consolidation and Close (ad hoc only)</li> <li>Tax Reporting (ad hoc only)</li> <li>Essbase</li> <li>Financial Management (ad hoc only)</li> <li>Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypGetConnectio nInfo</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning (ad hoc only)</li> <li>Planning Modules (ad hoc only)</li> <li>Financial Consolidation and Close (ad hoc only)</li> <li>Tax Reporting (ad hoc only)</li> <li>Essbase</li> <li>Financial Management (ad hoc only)</li> <li>Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypGetDatabaseN ote</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypGetDataPoint</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Essbase</li> <li>Financial Management</li> <li>Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetDimension s</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Essbase</li> <li>Financial Management</li> <li>Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetDimMbrsF orDataCell</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Essbase</li> <li>Financial Management</li> <li>Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetDrillThroug hReports</a>	Essbase

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypGetGlobalOption</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetLastError</a>	All
<a href="#">HypGetLinkMacro</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypGetMemberInformation</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypGetMemberInformationEx</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypGetMembers</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetNameRangeList</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypGetOption</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetPagePOVChoices</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetParent</a>	Oracle Analytics Cloud - Essbase , Essbase

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypGetPOVCount</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypGetPOVItems</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypGetRowCount</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypGetRowItems</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypGetSharedConnectionsURL</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetSheetInfo</a>	All

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypGetSheetOption</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypGetSourceGrid</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypGetSubstitutionVariable</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypGetVersion</a>	All
<a href="#">HypHideRibbonMenu</a>	All
<a href="#">HypHideRibbonMenuReset</a>	All
<a href="#">HypInvalidateSSO</a>	All providers that support Single Signon (SSO)
<a href="#">HypIsAncestor</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypIsAttribute</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypIsCellWritable</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypIsChild</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypIsConnectedToSharedConnections</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypIsDataModified</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypIsDescendant</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypIsExpense</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypIsFreeForm</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypIsParent</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypIsSmartViewContentPresent</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypIsUDA</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypKeepOnly</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypListApplications</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypListCalcScripts</a>	Oracle Analytics Cloud - Essbase , Essbase

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypListCalcScripts Ex</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypListDatabases</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypListDocuments</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypListLROs</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypMenuVAbout</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVAdjust</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVBusinesRules</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Oracle Hyperion Planning</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypMenuVCalculation</a>	Oracle Analytics Cloud - Essbase Essbase Financial Management (ad hoc only)
<a href="#">HypMenuVCascadeNewWorkbook</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVCascadeSameWorkbook</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVCellText</a>	<ul style="list-style-type: none"> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVCollapse</a>	<ul style="list-style-type: none"> <li>• Planning (forms only)</li> <li>• Planning Modules (forms only)</li> <li>• Financial Consolidation and Close (forms only)</li> <li>• Tax Reporting (forms only)</li> <li>• Oracle Hyperion Planning (forms only)</li> </ul>
<a href="#">HypMenuVConnect</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVCopyDataPoints</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypMenuVExpand</a>	<ul style="list-style-type: none"> <li>• Planning (forms only)</li> <li>• Planning Modules (forms only)</li> <li>• Financial Consolidation and Close (forms only)</li> <li>• Tax Reporting (forms only)</li> <li>• Oracle Hyperion Planning (forms only)</li> </ul>
<a href="#">HypMenuVFunctionBuilder</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVInstruction</a>	<ul style="list-style-type: none"> <li>• Planning (forms only)</li> <li>• Planning Modules (forms only)</li> <li>• Financial Consolidation and Close (forms only)</li> <li>• Tax Reporting (forms only)</li> <li>• Financial Management (forms only)</li> <li>• Oracle Hyperion Planning (forms only)</li> </ul>
<a href="#">HypMenuVKeepOnly</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypMenuVLRO</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypMenuVMemberInformation</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypMenuVMemberSelection</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVMigrate</a>	Financial Management
<a href="#">HypMenuVOptions</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>



**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypMenuVPastedDataPoints</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVPivot</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypMenuVPOVMManager</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVQueryDesigner</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypMenuVRedo</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypMenuVRefresh</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypMenuVRefreshAll</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVRefreshOfflineDefinition</a>	<ul style="list-style-type: none"> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVRemoveOnly</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypMenuVRulesOnlineForm</a>	<ul style="list-style-type: none"> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVRunReport</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypMenuVSelectForm</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypMenuVShowHelpHtml</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVSubmitData</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVSupportingDetails</a>	<ul style="list-style-type: none"> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypMenuVSyncBack</a>	Planning, Oracle Hyperion Planning
<a href="#">HypMenuVTakeOffline</a>	Planning, Oracle Hyperion Planning
<a href="#">HypMenuVUndo</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypMenuVVisualizeInExcel</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypMenuVZoomIn</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning (ad hoc only)</li> <li>Planning Modules (ad hoc only)</li> <li>Financial Consolidation and Close (ad hoc only)</li> <li>Tax Reporting (ad hoc only)</li> <li>Essbase</li> <li>Financial Management (ad hoc only)</li> <li>Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypMenuVZoomOut</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning (ad hoc only)</li> <li>Planning Modules (ad hoc only)</li> <li>Financial Consolidation and Close (ad hoc only)</li> <li>Tax Reporting (ad hoc only)</li> <li>Essbase</li> <li>Financial Management (ad hoc only)</li> <li>Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypModifyConnection</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning (ad hoc only)</li> <li>Planning Modules (ad hoc only)</li> <li>Financial Consolidation and Close (ad hoc only)</li> <li>Tax Reporting (ad hoc only)</li> <li>Essbase</li> <li>Financial Management (ad hoc only)</li> <li>Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypModifyRangeGridName</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypOpenForm</a>	<ul style="list-style-type: none"> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Financial Management</li> <li>Oracle Hyperion Planning</li> </ul>
<a href="#">HypOtlGetMemberInfo</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypPerformAdhocOnForm</a>	<ul style="list-style-type: none"> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Oracle Hyperion Planning</li> </ul>
<a href="#">HypPivot</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning (ad hoc only)</li> <li>Planning Modules (ad hoc only)</li> <li>Financial Consolidation and Close (ad hoc only)</li> <li>Tax Reporting (ad hoc only)</li> <li>Essbase</li> <li>Financial Management (ad hoc only)</li> <li>Oracle Hyperion Planning (ad hoc only)</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypPivotToGrid</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypPivotToPOV</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypPreserveFormating</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypQueryMembers</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypRedo</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypRemoveConnection</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypRemoveOnly</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypRemovePreservedFormats</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypResetFriendlyName</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypRetrieve</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypRetrieveAllWorkbookbooks</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypRetrieveLRO</a>	Oracle Analytics Cloud - Essbase , Essbase
<a href="#">HypRetrieveNameRange</a>	Oracle Analytics Cloud - Essbase , Essbase

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypRetrieveRange</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypSetActiveConnection</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetActiveMember</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetAliasTable</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetAsDefault</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetBackgroundPOV</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypSetCellsDirty</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetCollItems</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetConnAliasTable</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetConnectionInfo</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypSetCustomHeader</a>	Oracle Analytics Cloud - Essbase
<a href="#">HypSetDimensions</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>



**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypSetGlobalOption</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetLinkMacro</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypSetMembers</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetMenu</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetOption</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
<a href="#">HypSetPages</a>	<ul style="list-style-type: none"> <li>• Planning (forms only)</li> <li>• Planning Modules (forms only)</li> <li>• Financial Consolidation and Close (forms only)</li> <li>• Tax Reporting (forms only)</li> <li>• Financial Management (forms only)</li> <li>• Oracle Hyperion Planning (forms only)</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
HypSetPOV	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
HypSetPOVItems	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
HypSetRowItems	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
HypSetSharedConnectionsURL	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
HypSetSheetOption	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning</li> <li>• Planning Modules</li> <li>• Financial Consolidation and Close</li> <li>• Tax Reporting</li> <li>• Essbase</li> <li>• Financial Management</li> <li>• Oracle Hyperion Planning</li> </ul>
HypSetSSO	All providers that support Single Signon (SSO)
HypSetSubstitutionVariable	Oracle Analytics Cloud - Essbase, Essbase
HypShowPanel	All
HypShowPov	All

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypSubmitData</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Essbase</li> <li>Financial Management</li> <li>Oracle Hyperion Planning</li> </ul>
<a href="#">HypSubmitSelectedDataCells</a>	Oracle Analytics Cloud - Essbase, Essbase
<a href="#">HypSubmitSelectedRangeWithoutRefresh</a>	Oracle Analytics Cloud - Essbase, Essbase
<a href="#">HypTranslate</a>	Financial Management (ad hoc only)
<a href="#">HypUIConnect</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning</li> <li>Planning Modules</li> <li>Financial Consolidation and Close</li> <li>Tax Reporting</li> <li>Essbase</li> <li>Financial Management</li> <li>Oracle Hyperion Planning</li> <li>Oracle BI EE</li> </ul>
<a href="#">HypUndo</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning (ad hoc only)</li> <li>Planning Modules (ad hoc only)</li> <li>Financial Consolidation and Close (ad hoc only)</li> <li>Tax Reporting (ad hoc only)</li> <li>Essbase</li> <li>Financial Management (ad hoc only)</li> <li>Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypUpdateLRO</a>	Oracle Analytics Cloud - Essbase, Essbase
<a href="#">HypUseLinkMacro</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning (ad hoc only)</li> <li>Planning Modules (ad hoc only)</li> <li>Financial Consolidation and Close (ad hoc only)</li> <li>Tax Reporting (ad hoc only)</li> <li>Essbase</li> <li>Financial Management (ad hoc only)</li> <li>Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">HypZoomIn</a>	<ul style="list-style-type: none"> <li>Oracle Analytics Cloud - Essbase</li> <li>Planning (ad hoc only)</li> <li>Planning Modules (ad hoc only)</li> <li>Financial Consolidation and Close (ad hoc only)</li> <li>Tax Reporting (ad hoc only)</li> <li>Essbase</li> <li>Financial Management (ad hoc only)</li> <li>Oracle Hyperion Planning (ad hoc only)</li> </ul>

**Table 2-3 (Cont.) VBA Functions—Alphabetical List**

VBA Functions	Applicable Data Sources
<a href="#">HypZoomOut</a>	<ul style="list-style-type: none"> <li>• Oracle Analytics Cloud - Essbase</li> <li>• Planning (ad hoc only)</li> <li>• Planning Modules (ad hoc only)</li> <li>• Financial Consolidation and Close (ad hoc only)</li> <li>• Tax Reporting (ad hoc only)</li> <li>• Essbase</li> <li>• Financial Management (ad hoc only)</li> <li>• Oracle Hyperion Planning (ad hoc only)</li> </ul>
<a href="#">InsertView</a>	Oracle Analytics Cloud
<a href="#">InsertView</a>	Oracle BI EE
<a href="#">InvokeMenu</a>	Oracle Analytics Cloud
<a href="#">InvokeMenu</a>	Oracle BI EE
<a href="#">ListJournals</a>	Financial Management and Oracle Journals Extension for Financial Management
<a href="#">ListTemplates</a>	Financial Management and Oracle Journals Extension for Financial Management
<a href="#">OpenJournal</a>	Financial Management and Oracle Journals Extension for Financial Management
<a href="#">PasteView</a>	Oracle Analytics Cloud
<a href="#">PasteView</a>	Oracle BI EE
<a href="#">PerformAction</a>	Financial Management and Oracle Journals Extension for Financial Management
<a href="#">Retrieve</a>	Oracle BI EE
<a href="#">SaveJournal</a>	Financial Management and Oracle Journals Extension for Financial Management
<a href="#">SetJournalProperty</a>	Financial Management and Oracle Journals Extension for Financial Management
<a href="#">ValidateJournal</a>	Financial Management and Oracle Journals Extension for Financial Management

# 3

## Menu Functions

### Related Topics

- [About Menu Functions](#)
- [HypMenuVAbout](#)
- [HypMenuVAdjust](#)
- [HypMenuVBusinessRules](#)
- [HypMenuVCalculation](#)
- [HypMenuVCascadeNewWorkbook](#)
- [HypMenuVCascadeSameWorkbook](#)
- [HypMenuVCellText](#)
- [HypMenuVCollapse](#)
- [HypMenuVConnect](#)
- [HypMenuVCopyDataPoints](#)
- [HypMenuVExpand](#)
- [HypMenuVFunctionBuilder](#)
- [HypMenuVInstruction](#)
- [HypMenuVKeepOnly](#)
- [HypMenuVLRO](#)
- [HypMenuVMemberInformation](#)
- [HypMenuVMemberSelection](#)
- [HypMenuVMigrate](#)
- [HypMenuVOptions](#)
- [HypMenuVPasteDataPoints](#)
- [HypMenuVPivot](#)
- [HypMenuVPOVManager](#)
- [HypMenuVQueryDesigner](#)
- [HypMenuVRedo](#)
- [HypMenuVRefresh](#)
- [HypMenuVRefreshAll](#)
- [HypMenuVRefreshOfflineDefinition](#)
- [HypMenuVRemoveOnly](#)
- [HypMenuVRulesOnForm](#)
- [HypMenuVRunReport](#)

- [HypMenuVSelectForm](#)
- [HypMenuVShowHelpHtml](#)
- [HypMenuVSubmitData](#)
- [HypMenuVSupportingDetails](#)
- [HypMenuVSyncBack](#)
- [HypMenuVTakeOffline](#)
- [HypMenuVUndo](#)
- [HypMenuVVisualizeinExcel](#)
- [HypMenuVZoomIn](#)
- [HypMenuVZoomOut](#)
- [HypExecuteMenu](#)
- [HypHideRibbonMenu](#)
- [HypHideRibbonMenuReset](#)

## About Menu Functions

VBA menu functions are identical to the equivalent commands on the Oracle Smart View for Office 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypMenuVAdjust() opens the **Adjust Data** dialog box.

### Syntax

```
HypMenuVAdjust()
```

### Return Value

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

### Example

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

## HypMenuVBusinessRules

**Cloud data provider types:** Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Hyperion Planning

### Description

HypMenuVBusinessRules() opens the **Business Rules** dialog box.

### Syntax

```
HypMenuVBusinessRules()
```

### Return Value

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

### Example

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

## HypMenuVCalculation

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Financial Management (ad hoc only)

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning

### 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

**Cloud data provider types:** Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Planning (forms only), Planning Modules (forms only), Financial Consolidation and Close (forms only), Tax Reporting (forms only)

**On-premises data provider types:** Oracle Hyperion 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypMenuVCopyDataPoints() copies data points from Excel for pasting into Word or PowerPoint. See also [HypMenuVPasteDataPoints](#).

### Syntax

```
HypMenuVCopyDataPoints()
```

### Return Value

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

### Example

```
Public Declare Function HypMenuVCopyDataPoints Lib "HsAddin" () As Long  
Sub MCopyDataPoints()  
    X=HypMenuVCopyDataPoints()  
End Sub
```

## HypMenuVExpand

**Cloud data provider types:** Planning (forms only), Planning Modules (forms only), Financial Consolidation and Close (forms only), Tax Reporting (forms only)

**On-premises data provider types:** Oracle Hyperion 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()  
End Sub
```

```
X=HypMenuVExpand()  
End Sub
```

## HypMenuVFunctionBuilder

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Planning (forms only), Planning Modules (forms only), Financial Consolidation and Close (forms only), Tax Reporting (forms only)

**On-premises data provider types:** Oracle Hyperion Planning (forms only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### Description

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

### Syntax

```
HypMenuVLRO()
```

### Return Value

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

### Example

```
Public Declare Function HypMenuVLRO Lib "HsAddin" () As Long  
Sub MLRO()
```

```
X=HypMenuVLR0()  
End Sub
```

## HypMenuVMemberInformation

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### Description

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

### Syntax

```
HypMenuVMemberInformation()
```

### Return Value

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

### Example

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

## HypMenuVMemberSelection

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

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

### Syntax

```
HypMenuVMemberSelection()
```

### Return Value

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

### Example

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

## HypMenuVMigrate

**Data provider types:** Oracle Hyperion Financial Management

### Description

HypMenuVMigrate() launches the Financial Management 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

vtOutput: Output parameter. Returns the migration result.

### Return Value

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

### Example

```
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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**Data provider types:**

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**Description**

HypMenuVPasteDataPoints() pastes data points that were copied from Excel into Word or PowerPoint. See also [HypMenuVCopyDataPoints](#).

**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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only).Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**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()
    X=HypMenuVPOVManager()
End Sub
```

## HypMenuVQueryDesigner

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**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

**Cloud data provider types:** Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Hyperion 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Planning (forms only), Planning Modules (forms only), Financial Consolidation and Close (forms only), Tax Reporting (forms only)

**On-premises data provider types:** Oracle Hyperion 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close ( ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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("SVPBC-over_olh_6")
End Sub
```

## HypMenuVSubmitData

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Hyperion 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:** Oracle Hyperion Planning

### Description

HypMenuVSyncBack() synchronizes data from an offline Oracle Hyperion 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:** Oracle Hyperion 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)



**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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 Oracle Smart View for Office ribbon and **Refresh** on the Oracle 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").

### Examples

#### For Refresh

```
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, "Smart View->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
```

If you are working with a non-English language, then vtMenuName requires the localized value the menu name in Smart View.

For example, if you are working with French, then vtMenuName would use the French value of Refresh in Smart View. The examples below compare the English and French parameter definitions for Refresh:

English:

```
sts = HypExecuteMenu(Empty, "Smart View->Refresh")
```

French:

```
sts = HypExecuteMenu(Empty, "Smart View->Actualiser")
```

### For Submit Without Refresh

```
Sub TestEssbaseSubmitData()  
sts = HypExecuteMenu(Empty, "Essbase->Submit Data Without Refresh")  
Debug.Print (sts)  
End Sub
```

### For Submit Data Range

```
Sub TestEssbaseSubmitData()  
sts = HypExecuteMenu(Empty, "Essbase->Submit Data Range")  
Debug.Print (sts)  
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. If vtSheetName is Null or Empty, the active worksheet is used.

**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. If vtSheetName is Null or Empty, the active worksheet is used.

### 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
```

# 4

## General Functions

### Related Topics

- [About General Functions](#)
- [HypShowPanel](#)
- [HypGetVersion](#)
- [HypListApplications](#)
- [HypListDatabases](#)
- [HypGetLastError](#)
- [HypShowPov](#)
- [HypSetMenu](#)
- [HypCopyMetaData](#)
- [HypDeleteMetaData](#)
- [HypIsDataModified](#)
- [HypIsSmartViewContentPresent](#)
- [HypIsFreeForm](#)
- [HypUndo](#)
- [HypRedo](#)
- [HypPreserveFormatting](#)
- [HypRemovePreservedFormats](#)
- [HypSetAliasTable](#)
- [HypGetSubstitutionVariable](#)
- [HypSetSubstitutionVariable](#)
- [HypGetDatabaseNote](#)
- [HypGetSheetInfo](#)
- [HypListDocuments](#)

## 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 Oracle Smart View for Office 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
```

## HypListApplications

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypListApplications() gets the list of applications and their descriptions for the requested provider.

### Syntax

HypListApplications (vtURL, vtServerName, vtUserName, vtPassword, vtApplications, vtAppsDescriptions)

ByVal vtURL As Variant

ByVal vtServerName As Variant

ByVal vtUserName As Variant

ByVal vtPassword As Variant

ByRef vtApplications As Variant

ByRef vtAppsDescriptions As Variant

### Parameters

**vtURL:** URL of the provider

**vtServerName:** Server name

**vtUserName:** User name

**vtPassword:** Password

**vtApplications:** List of applications

**vtAppsDescriptions:** List of application descriptions

### Return Value

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

### Example

This example combines the HypListApplications and HypListDatabases functions.

```
Sub test()

Dim url As Variant
```

```
Dim srv As Variant
Dim SSO As Variant
Dim username As Variant
Dim password As Variant
Dim app As Variant
Dim Applist As Variant
Dim AppDescList As Variant
Dim AppCubeList As Variant

'Essbase
url = "http://<server>:<port>/aps/SmartView"
app = "Sample"
srv = "EssbaseCluster-1"
username = "<username>"
password = "<password>"

ss = HypConnect("Sheet1", "admin", "ppp", "Conn123")
ss = HypListApplications(url, srv, <username>, <password>, Applist,
AppDescList)
ss = HypListDatabases(url, srv, <username>, <password>, app,
AppCubeList)

MsgBox (ss)

End Sub
```

## HypListDatabases

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning

### Description

HypListDatabases() gets the list of cubes (databases) present under an application for the requested provider.

### Syntax

HypListDatabases (vtURL, vtServerName, vtUserName, vtPassword, vtApplication, vtApplicationCubeList)

ByVal vtURL As Variant

ByVal vtServerName As Variant

ByVal vtUserName As Variant

ByVal vtPassword As Variant

ByVal vtApplication As Variant

ByRef vtApplicationCubeList As Variant

**Parameters****vtURL:** URL of the provider**vtServerName:** Server name**vtUserName:** User name**vtPassword:** Password**vtApplication:** Application name**vtApplicationCubeList:** List of cubes (databases)**Return Value**

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

**Example**

This example combines the HypListApplications and HypListDatabases functions.

```
Sub test()

Dim url As Variant
Dim srv As Variant
Dim SSO As Variant
Dim username As Variant
Dim password As Variant
Dim app As Variant
Dim Applist As Variant
Dim AppDescList As Variant
Dim AppCubeList As Variant

'Essbase
url = "http://<server>:<port>/aps/SmartView"
app = "Sample"
srv = "EssbaseCluster-1"
username = "<username>"
password = "<password>"

ss = HypConnect("Sheet1", "admin", "ppp", "Conn123")
ss = HypListApplications(url, srv, <username>, <password>, Applist,
AppDescList)
ss = HypListDatabases(url, srv, <username>, <password>, app,
AppCubeList)

MsgBox (ss)

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

```
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(ErrorCodeValue, ErrorMessageValue,
ErrorDescriptionValue)
End Sub
```

## HypShowPov

**Data provider types:** All

### Description

HypShowPov() shows or hides the POV toolbar.

### Syntax

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**

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

## HypSetMenu

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**Description**

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

**Syntax**

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**

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

## HypCopyMetaData

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management, Oracle Hyperion Reporting and Analysis

### Description

HypDeleteMetaData() deletes Oracle Smart View for Office 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypIsSmartViewContentPresent() determines whether the sheet contains Oracle Smart View for Office 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  
    INTERACTIVE_REPORT_SHEET  
End Enum
```

## 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**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)
MsgBox (oRet)
End Sub
```

## HypUndo

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

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

## HypPreserveFormatting

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

```
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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

## 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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 If
End Sub
```

## HypSetSubstitutionVariable

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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
```

## HypGetSheetInfo

**Data provider types:** All

### Description

HypGetSheetInfo() retrieves detailed information about the requested worksheet.

### Syntax

```
HypGetSheetInfo(vtSheetName, itemNameList, itemValueList)
```

```
ByVal vtSheetName As Variant
```

```
ByRef vtItemNameList As Variant
```

```
ByRef vtItemValueList As Variant
```

### Parameters

**vtSheetName:** Sheet name. If "Empty" is passed, then current worksheet information will be retrieved

**vtItemNameList:** List of the titles of sheet information

**vtItemValueList:** List of corresponding values of titles of Sheet information

### Return Value

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

### Example

```
Sub test()  
Dim namelist As Variant  
Dim vallist As Variant  
  
ss = HypConnect("Sheet1", "admin", "password", "Conn123")  
ss = HypRetrieve("Sheet1")  
sts = HypGetSheetInfo("Sheet1", namelist, vallist)  
  
End Sub
```

## HypListDocuments

**Cloud data provider types:** Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypListDocuments returns a list of folders and forms that are saved on the provider server, in the path specified by the user.

 **Note:**

In descriptions for this function, folders and forms are referred to as documents.

Unlike other VBAs, this is not a sheet-specific function. Sheet information is optional for this function. Refer to [Parameters](#).

### Syntax

HypListDocuments (vtSheetName, vtUserName, vtPassword, vtConnInfo, vtCompletePath, vtDocs)

ByVal vtSheetName As Variant

ByVal vtUserName As Variant

ByVal vtPassword As Variant

ByVal vtConnInfo As Variant

ByVal vtCompletePath As Variant

ByRef vtDocs As DOC\_Info

DOC\_Info is in turn defined as the following type :

Type DOC\_Info

numDocs As Long  
docTypes As Variant  
docNames As Variant  
docDescriptions As Variant  
docPlanTypes As Variant  
docAttributes As Variant

End Type

See further descriptions below in [Parameters](#).

### Parameters

**vtSheetName:** Optional. The name of the worksheet which will be used to obtain connection information, if vtConnInfo is empty. If vtSheetName is also empty, the active data source will be used to obtain connection information.

**vtUserName:** Optional input parameter. Used to connect using the given connection info (vtConnInfo/vtSheetName). Not necessary to be given if user is sure that the connection already exists.

**vtPassword:** Optional input parameter. Used to connect using the given connection info (vtConnInfo/vtSheetName). Not required if user is sure that the connection already exists.

**vtConnInfo:** Optional input parameter. Connection information, given either in the form of a friendly private connection name, or in the format accepted by HypConnect() or HypUIConnect(). If vtConnInfo is empty, vtSheetName is used to obtain connection info. If both are empty, the active data source is used.

**vtCompletePath:** Mandatory input parameter. The folder path within the server for which the document list is needed.

**vtDocs:** Mandatory output parameter. The list of documents (files and folders) present inside the given vtCompletePath, returned from the VBA.

### DOC\_Info Structure Definitions

**numDocs:** Number of folders plus the number of forms in the given folder. Defines the length of each of the arrays noted below.

**docTypes:** An array of strings. Array defining the type of each of the documents. Two types are supported, "DOC\_FORM" and "DOC\_FOLDER". Two global constants define the two types, which can be used to compare these values with the ones in the array

```
Global Const HYP_LIST_DOC_FORM = "DOC_FORM"  
Global Const HYP_LIST_DOC_FOLDER = "DOC_FOLDER"
```

See [Example](#) for usage.

**docNames:** An array of strings. Array containing names of each document, in the same order as above.

**docDescriptions:** An array of strings. Document descriptions, if any, in the same order as above. For folders, this field is empty.

**docPlanTypes:** An array of strings. Plan type or cube information for each document, in the same order as above. For folders, this field is empty.

**docAttributes:** An array of strings. Attribute information for each document, in the same order as above. For folders, this field is empty. For forms, this field can be any of the following:

Enum FORM\_ATTRIBUTES

NO\_ATTRIBUTE = -1 (for a folder)

HFM\_BASIC\_FORM = 0

ADHOC\_ENABLED = 8 (basic form)

COMPOSITE\_FORM = 16

SMART\_FORM = 128

SAVED\_ADHOC\_GRID = 40

SAVED\_ADHOC\_EXCLUSIVE\_GRID = 104

SMART\_FORM\_ADHOC\_ENABLED = 136

End Enum

HypListDocuments sends these values back as strings. See [Example](#).

Note that a folder has only a name associated with it. Other information is only available for forms.

### Return Value

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

### Example

The following VBA can be used directly, using each of the invocations one by one.

```
Sub testListDocs()  
  
Dim ret As Integer  
Dim firstDocType  
Dim vtDocs As DOC_Info  
Dim vtAttr  
  
'Usage with connection information as given for HypConnect()  
HypUIConect()  
ret = HypListDocuments("", "<user_name>", "<password>", "http://  
<server_url>:<port>/HyperionPlanning/SmartView|<server>|EPBCS|", "/"  
<path>", vtDocs)  
  
'Usage with friendly private connection name.
```

```
'ret = HypListDocuments("", "<user_name", "<password>", "<server>", "/"
<path>", vtDocs)

'Usage with connection information present in Sheet1.
'ret = HypListDocuments("Sheet1", "<user_name>", "<password>", "", "/"
<path>", vtDocs)

'Usage with active data source
'ret = HypListDocuments("", "<user_name>", "<password>", "", "/"<path>",
vtDocs)

MsgBox "Total no. of docs is : " & vtDocs.numDocs

If vtDocs.numDocs > 0 Then
    'First, folder info is sent, and then forms info.
    firstDocType = vtDocs.docTypes(0)

    If vtDocs.docTypes(0) = HYP_LIST_DOC_FORM Then
        MsgBox "First doc is a form."
    Else
        MsgBox "First doc is a folder."
    End If

    MsgBox "First doc name is : " & vtDocs.docNames(0)

    MsgBox "First doc attribute is : " & vtDocs.docAttributes(0)

    'Need to convert attribute string to integer before comparison.
    vtAttr = Cint(vtDocs.docAttributes(0))

    If vtAttr <> NO_ATTRIBUTE Then

        If vtAttr = ADHOC_ENABLED Then
            MsgBox "This form is adhoc-enabled"
        End If

        If vtAttr = SAVED_ADHOC_GRID Then
            MsgBox "This is a saved ad-hoc grid"
        End If

        If vtAttr = SAVED_ADHOC_EXCLUSIVE_GRID Then
            MsgBox "This is a saved ad-hoc exclusive grid"
        End If

        If vtAttr = COMPOSITE_FORM Then
            MsgBox "This is a composite form."
        Else
            MsgBox "This is not a composite form."
        End If

    End If

End If
End Sub
```

# 5

## Connection Functions

### Related Topics

- [About Connection Functions](#)
- [HypConnect](#)
- [HypUIConnect](#)
- [HypConnected](#)
- [HypConnectionExists](#)
- [HypCreateConnection](#)
- [HypCreateConnectionEx](#)
- [HypSetCustomHeader](#)
- [HypUpdateConnection](#)
- [HypModifyConnection](#)
- [HypDisconnect](#)
- [HypDisconnectAll](#)
- [HypDisconnectEx](#)
- [HypGetSharedConnectionsURL](#)
- [HypSetSharedConnectionsURL](#)
- [HypIsConnectedToSharedConnections](#)
- [HypRemoveConnection](#)
- [HypSetSSO](#)
- [HypInvalidateSSO](#)
- [HypResetFriendlyName](#)
- [HypSetActiveConnection](#)
- [HypSetAsDefault](#)
- [HypSetConnAliasTable](#)

## About Connection Functions

Connection functions perform actions related to connections to data providers.

## HypConnect

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

 **Note:**

**Cloud data sources:** HypConnect is *not* supported for cloud data sources. To connect to a cloud data source (for example, a Planning source), you must use the HypUIConnect function. See [HypUIConnect](#).

## 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.

HypConnect() can be used in two ways:

- Using a friendly connection name.  
The friendly connection name is first created using [HypCreateConnection](#).
- Using a connection string, in the place of a friendly name, consisting of URL, server, application, database name.

## 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 connection name of the data provider. The friendly name parameter can accept either of the following:

- A connection name created using [HypCreateConnection](#)
- A connection string consisting of a URL, server name, application name, and database name, in the format `URL|server|app|db`.

The URL component of the connection string follows the guidelines in [Private Connection URL Syntax](#) in the *Oracle Smart View for Office User's Guide*.

## Return Value

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



### Example Using a Friendly Name

In this example, we connect to the Essbase Sample application, Basic database, with the friendly name of "My Sample Basic."

```
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
```

### Essbase Example Using a Connection String

In this example, we connect to the Essbase Sample application, Basic database.

```
HypConnect("Sheet1", "UserName", "Password", "http://hostname:19000/aps/SmartView|servername|Sample|Basic")
```

### Oracle Hyperion Planning Example Using a Connection String

In this example, we connect to the Oracle Hyperion Planning Vision application, Plan1 database.

```
HypConnect("Sheet1", "UserName", "Password", "http://hostname:9000/HyperionPlanning/SmartView|servername|Vision|Plan1")
```

### Financial Management Example Using a Connection String

In this example, we connect to the Financial Management STATJP4 application.

```
HypConnect("Sheet1", "UserName", "Password", "http://hostname:19000/hfmadf/officeprovider|<servername>|STATJP4|STATJP4")
```

Note that for Financial Management, the db component can be empty; for example:

```
HypConnect("Sheet1", "UserName", "Password", "http://hostname.com:19000/hfmadf/officeprovider|<servername>|STATJP4|")
```

## HypUIConnect

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management, Oracle Business Intelligence Enterprise Edition

## Description

**On-premises data sources:** For on-premises data sources, HypUIConnect() prompts the user with the **Connect to Data Source** dialog box when the user name and password are not provided. It does not prompt if the user name and password are already provided.

**Cloud data sources and on-premises form-based authentication:** For cloud data sources and for on-premises sources using form-based authentication, you must use the HypUIConnect VBA function to connect to an existing private connection. HypUIConnect() *always* prompts for authentication where the user must provide the user name, password, and domain to log in.

HypUIConnect() can be used in two ways:

- Using a friendly connection name.  
The friendly connection name is first created using [HypCreateConnection](#).
- Using a connection string, in the place of a friendly name, consisting of URL, server, application, database name.

## Syntax

```
HypUIConnect (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. The friendly name parameter can accept either of the following:

- A connection name created using [HypCreateConnection](#)
- A connection string consisting of a URL, server name, application name, and database name, in the format `URL|server|app|db`.

The URL component of the connection string follows the guidelines in [Private Connection URL Syntax](#) in the *Oracle Smart View for Office User's Guide*.

## Return Value

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

### Example

In this example, we connect to the Oracle Analytics Cloud - Essbase Sample application, Basic database, with the friendly name of "My Sample Basic."

```
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 Sample Basic")
```

### Essbase Example Using a Connection String

In this example, we connect to the Oracle Analytics Cloud - Essbase Sample application, Basic database.

```
HypUIConnect("Sheet1", "UserName", "Password", "http://hostname:19000/aps/SmartView|servername|Sample|Basic")
```

### Planning Example Using a Connection String

In this example, we connect to the Planning Vision application, Plan1 database.

```
HypUIConnect("Sheet1", "UserName", "Password", "http://hostname.com:9000/HyperionPlanning/SmartView|servername|Vision|Plan1")
```

### Financial Management Example Using a Connection String

In this example, we connect to the Financial Management STATJP4 application.

```
HypUIConnect("Sheet1", "UserName", "Password", "http://hostname.com:19000/hfmadf/officeprovider|<servername>|STATJP4|STATJP4")
```

Note that for Financial Management, the db component can be empty; for example:

```
HypUIConnect("Sheet1", "UserName", "Password", "http://hostname.com:19000/hfmadf/officeprovider|<servername>|STATJP4|")
```

## HypConnected

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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  
    X = HypConnected(Empty)  
End sub
```

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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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)
```

```
ByVal vtFriendlyName As Variant
```

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypCreateConnection() creates a private connection to the data provider from the specified information.

 **Note:**

HypCreateConnection is not supported for shared connections.  
HypCreateConnection is for private connections only.

## 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\_FINANCIAL\_MANAGEMENT = "Hyperion Financial Management"
- Global Const HYP\_PLANNING = "Planning"
- Global Const HYP\_RA = "Hyperion Smart View Provider for Hyperion Reporting and Analysis"



### Note:

The global constant HYP\_ANALYTIC\_SERVICES = "Analytic Provider Services" is 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

 **Note:**

**Financial Management only:** This parameter is required, but you cannot use the name of the application. Instead, use `Empty` or `Null` for this parameter.

**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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

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

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

**Note:**

HypCreateConnectionEx is not supported for shared connections.  
HypCreateConnectionEx is for private connections only.

**Syntax**

HypCreateConnectionEx (vtProviderType, vtServerName, vtApplicationName, vtDatabaseName, vtFormName, vtProviderURL, vtFriendlyName, vtUserName, vtPassword, vtDescription, vtReserved1, 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"

**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 Oracle Hyperion Planning connection in Smart View Panel under Private Connections.

**vtProviderURL:** The data provider URL. Required to create Oracle Hyperion 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", "", "", "")

lRet = HypCreateConnectionEx("Planning", "plange14", "TotPlan", "",
"/Forms/Smart View Forms/01 Product Revenue", "http://plange14:8300/
HyperionPlanning/SmartView", "My Planning VBA Conn", "UserName",
>Password", "", "", "")

End Sub
```

## HypSetCustomHeader

**Data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

### Description

Add custom HTTP headers using this function. Once set, all requests made by Oracle Smart View for Office will include this header as part of the request.

This function may be used for signing on to providers that use the OAuth 2.0 token for authentication, such as Oracle Analytics Cloud - Essbase.

For example, with Oracle Analytics Cloud - Essbase, users can generate an OAuth bearer token by connecting to Oracle Identity Services Cloud and providing necessary information (Client ID, Client Secret, etc.). Once the token is generated, this token can be added to the header.

The HypCustomHeader function also supports Basic Authentication via Oracle Enterprise Performance Management Cloud for Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting.

### Syntax

```
HypSetCustomHeader Lib "HsAddin" (vtHeader) As Long
```

```
ByVal vtHeader As Variant
```

### Parameters

**vtHeader:** Content of this parameter will be added to existing HTTP headers. Parameter content must conform to HTTP header semantics. This parameter can then be added to the authorization header. See the [Example - OAuth](#) and [Example - Basic Authentication](#) for more information.

### Return Value

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

### Example - OAuth

```
Public Declare PtrSafe Function HypSetCustomHeader Lib "HsAddin" (ByVal  
vtHeader As Variant) As Long
```

```
AuthHeader = "Authorization: Bearer " + token sts =  
HypSetCustomHeader(AuthHeader)
```

### Example - Basic Authentication

The HypCustomHeader function also supports Basic Authentication in the format, / HyperionPlanning/rest/SmartView. In this example, the token is a Base64 Encoded String.

```
Public Declare PtrSafe Function HypSetCustomHeader Lib "HsAddin" (ByVal  
vtHeader As Variant) As Long
```

```
AuthHeader = "Authorization: Basic " + token sts =  
HypSetCustomHeader(AuthHeader)
```

## HypUpdateConnection

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

## Description

HypUpdateConnection updates the existing connection.

 **Note:**

When updating a connection in an existing multiple-grid sheet, only use HypUpdateConnection if the old connection name exists. If the connection name given in HypUpdateConnection does not exist, then the multiple-grid sheet becomes a single-grid ad hoc sheet.

## Syntax

```
Public Declare PtrSafe Function HypUpdateConnection Lib "HsAddin" (ByVal  
vtProviderType As Variant,  
ByVal vtServerName As Variant  
ByVal vtApplicationName As Variant  
ByVal vtDatabaseName As Variant  
ByVal vtProviderURL As Variant  
ByVal vtFriendlyName As Variant  
ByVal vtUserName As Variant  
ByVal vtPassword As Variant  
ByVal vtDescription As Variant  
ByVal bCreateNewIfConnectionDoesNotExist As Boolean) As Long
```

## Parameters

**vtServerName:** Server name

**vtApplicationName:** The name of the application

**vtDatabaseName:** The name of the database

**vtProviderURL:** The URL of the data provider

**vtFriendlyName:** The connection name of the data provider

**vtUserName:** User name

**vtPassword:** Password

**vtDescription:** A description of the data provider

## Return Value

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

**Example 1**

Connection does not exist and the new connection creation is False, resulting in error code -43.

```
Sub UpdateConnection()
sts = HypUpdateConnection("Essbase", "EssbaseCluster-1", "Sample",
"Basic",
"http://<server>:<port>/aps/SmartView", "NonExistingConnection",
"<username>", "<password>", "test", False) 'will return -43, connection
not found
```

**Example 2**

Connection exists but new connection creation is True. Result is successful and returns code 0 (if all parameters are correct).

```
sts = HypUpdateConnection("Essbase", "EssbaseCluster-1", "Sample",
"Basic",
"http://<server>:<port>/aps/SmartView", "NewConnection", "<username>",
"<password>", "test", True) 'will return 0 as it will internally create
a new connection if it does not exist as last parameter is true
sts = HypUpdateConnection("Essbase", "EssbaseCluster-1", "Sample",
"Basic", "<server>:<port>/aps/SmartView", "NewConnection", "<username>",
"<password>", "newTestConnection", False)
End Sub
```

**Example 3**

Connection exists but new connection creation is True. Result is successful and returns code 0 (if all parameters are correct).

```
Public Const serverName = "EssbaseCluster-1"
Public Const ProviderURL = "http://<server>:<port>/aps/SmartView"
Public Const HWL_application = "Sample"
Public Const HWL_db = "Basic"

Sub Test()
    HWL_Connection = "test_Ess"
    sts = HypDisconnectAll()
    'X = HypRemoveConnection(HWL_Connection)
    UserId = "<username>"
    UserPwd = "<password>"
    'X = HypCreateConnection(Empty, UserId, UserPwd, HYP_ESSBASE,
ProviderURL, _
        serverName, HWL_application, HWL_db, HWL_Connection,
"User_Description")
    X = HypUpdateConnection("Essbase", serverName, HWL_application,
HWL_db,
ProviderURL, HWL_Connection, UserId, UserPwd, "test", True)
    z1 = HypConnect("Sheet1", UserId, UserPwd, HWL_Connection) 'To login
```

End Sub

## HypModifyConnection

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypModifyConnection() is used to modify any connection information for a workbook, sheet, range, or grid. Applies to:

- Regular ad hoc sheet
- Multiple-grid ad hoc sheet
- Form-based sheet

### Syntax

```
Private Declare PtrSafe Function HypModifyConnection Lib "HsAddin"  
(vtDocumentName, vtSheetName, vtGridName As Variant, vtServer, vtURL, vtApp,  
vtDB, vtConnParam) As Long
```

ByVal vtDocumentName As Variant

ByVal vtSheetName As Variant

ByVal vtGridName As Variant

ByVal vtServer As Variant

ByVal vtURL As Variant

ByVal vtApp As Variant

ByVal vtDB As Variant

ByVal vtConnParam As Variant

### Parameters

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

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

**vtGridName:** Name range of a grid on a multiple-grid worksheet. If vtGridName is `Null` or `Empty`, and the sheet is a multiple-grid sheet, then the connection information of all grids on the sheet will be modified.

**vtServer:** The name of the new server; the application must reside in the new server

**vtURL:** The new data provider URL

**vtApp:** The new application name

**vtDB:** The new cube or database name

**vtConnParam:** Any additional provider parameters



#### Note:

The user must save the workbook for the connection changes to persist.

### Example

```
Sub testModifyConnection()
    'modify url in a particular workbook for all SV sheets
    s = HypModifyConnection("testmultigrid.xlsm", "", "", "", "http://
<server>:<port>/aps
        /SmartView", "", "", "")
    'modify app/db for a specific multi-grid in a workbook
    s = HypModifyConnection("testmultigrid.xlsm", "Sheet1",
        "Demol5FCFBC11_9D65_4555_94AC_6EDD429438B0_1", "", "",
"NoUniq", "NoUniq", "")
    'modify url for all sheets in active workbook
    s = HypModifyConnection("", "", "", "", "http://<server>:<port>/aps/
        SmartView", "", "", "")
    'modify url in a particular sheet for active workbook
    s = HypModifyConnection("", "Sheet1", "", "", "http://
<server>:<port>/aps/
        SmartView", "", "", "")
End Sub
```

## HypDisconnect

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

#### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypRemoveConnection() removes the specified connection from the list of available Oracle Smart View for Office 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
```

## HypSetSSO

**Data provider types:** All providers that support Single Sign on (SSO)

### Description

HypSetSSO() sets the SSO token in Smart View. When set, the SSO token takes priority over user name and password.

### Syntax

```
HypSetSSO(vtSSO)
```

ByVal vtSSO As Variant

### Parameters

**vtSSO:** SSO token

### Return Value

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

### Example

```
Sub test()  
  
SSO = <SSO token>  
sts = HypDisconnectAll()  
sts = HypSetSSO(SSO)  
MsgBox (sts)  
  
End Sub
```

## HypInvalidateSSO

**Data provider types:** All providers that support Single Sign on (SSO)

### Description

HypInvalidateSSO() discards the existing SSO token.

### Example

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

## HypResetFriendlyName

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypResetFriendlyName modifies the friendly name to a new one. To modify the friendly name of a connection in the Smart View Panel, Oracle Smart View for Office must be connected to the data source provider.

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion 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, "MyConnection1" 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
```

# 6

## Ad Hoc Functions

### Related Topics

- [About Ad Hoc Functions](#)
- [HypPerformAdhocOnForm](#)
- [HypRetrieve](#)
- [HypRetrieveRange](#)
- [HypRetrieveNameRange](#)
- [HypCreateRangeGrid](#)
- [HypModifyRangeGridName](#)
- [HypGetNameRangeList](#)
- [HypRetrieveAllWorkbooks](#)
- [HypExecuteQuery](#)
- [HypSubmitData](#)
- [HypSubmitSelectedRangeWithoutRefresh](#)
- [HypSubmitSelectedDataCells](#)
- [HypPivot](#)
- [HypPivotToGrid](#)
- [HypPivotToPOV](#)
- [HypKeepOnly](#)
- [HypRemoveOnly](#)
- [HypZoomIn](#)
- [HypZoomOut](#)

### About Ad Hoc Functions

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

### HypPerformAdhocOnForm

**Cloud data provider types:** Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Hyperion 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 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (ad hoc only)

### Description

HypRetrieve() retrieves data from the database.

**Essbase only:** HypRetrieve does not support creating a multiple-grid worksheet. Use [HypRetrieveRange](#) instead.

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypRetrieveRange() enables users to refresh a selected or named range of cells in a grid or worksheet. The selected or named range of cells should form a valid grid layout. 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.

**Essbase only:** To refresh or create a multiple-grid sheet, use [HypCreateRangeGrid](#). Starting in release 11.1.2.5.610, the behavior of the HypRetrieveRange VBA function is changed to only refresh the selected range. In 11.1.2.5.610 and later, it will no longer create multiple-grid sheets or create additional ranges.

**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**

This example assumes that the required data sources are already connected.

```
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"), "Sample")
        'retrieve by regular range
    sts = HypRetrieveRange(Empty, range("MyRange"), "Sample")
        'retrieve by named range
End sub
```

## HypRetrieveNameRange

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

**Description**

HypRetrieveNameRange refreshes the grid created by HypCreateRangeGrid. This function works only with Oracle Smart View for Office multiple-grid defined range names.

See also [HypCreateRangeGrid](#).

### 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",  
"myserver_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
```

## HypCreateRangeGrid

**Data provider types:** Oracle Essbase

## Description

HypCreateRangeGrid() enables users to refresh multiple selected or named ranges of cells in a multiple-grid worksheet. You can also use it to add a grid in the selected range if there is not already a grid in the particular location, thus creating a multiple-grid sheet. If the ranges specified for this function contain 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.

Users can refresh selected continuous cell ranges from more than one grid in a multiple-grid worksheet. HypCreateRangeGrid can also be used to create a multiple-grid sheet, as shown in [Example: Creating and Refreshing a Multiple-grid Sheet](#). Once the user runs HypCreateRangeGrid, the sheet becomes a multiple-grid ad hoc sheet.

To refresh selected continuous cell ranges from a single-grid worksheet, use [HypRetrieveRange](#).

## Syntax

```
HypCreateRangeGrid(vtSheetName,vtRange,vtFriendlyName)
```

```
ByVal vtSheetName As Variant
```

```
ByVal vtRange As Variant
```

```
ByVal vtFriendlyName As Variant
```

## Parameters

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

**vtRange:** The continuous ranges to be refreshed. Each 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 ranges to be refreshed.

**vtFriendlyName:** The friendly name of the connection to be used to refresh the ranges. If set to Null, the active connection associated with the worksheet is used to refresh the ranges 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: Creating and Refreshing a Multiple-grid Sheet

This example assumed that the required data sources are already connected.

```
Sub Example_CreateMultiGrid
    'create sample-basic range grid
    sts = HypCreateRangeGrid (Empty, range("E11:F13"), "SampleBasic1")
    'create demo-basic range grid
    sts = HypCreateRangeGrid (Empty, range("E17:G20"), "DemoBasic1")
End Sub
```

Once the grids are created, HypCreateRangeGrid can be called to refresh selected ranges in grids in the sheet one at a time. Additionally, HypCreateRangeGrid can be used to refresh all grids in the sheet.

## HypModifyRangeGridName

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### Description

HypModifyRangeGridName() is used to modify the name of any given grid on a multiple-grid worksheet present in the active workbook.

### Syntax

```
Private Declare PtrSafe Function HypModifyRangeGridName Lib "HsAddin"  
(vtSheetName, vtGridName, vtNewGridName) As Long
```

```
ByVal vtSheetName As Variant
```

```
ByVal vtGridName As Variant
```

```
ByVal vtNewGridName As Variant
```

### Parameters

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

**vtGridName:** Name range of a grid on a multiple-grid worksheet. This parameter cannot be Null or Empty.

**vtNewGridName:** New name range of a grid on a multiple-grid worksheet. This parameter cannot be Null or Empty.

### Example

```
Sub modifyName()  
    s = HypModifyRangeGridName("Sheet1",  
    "Demo15FCFBC11_9D65_4555_94AC_6EDD429438B0_1", "someNewGridName")  
End Sub
```

## HypGetNameRangeList

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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
```

 **Note:**

For better performance, when using this function, disable all Oracle Smart View for Office extensions.

## HypExecuteQuery

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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
```

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

## Description

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



### 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.
- For more guidelines on submitting data, see [About the Submit Data Options](#).

## 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
```

# HypSubmitSelectedRangeWithoutRefresh

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

## Description

HypSubmitSelectedRangeWithoutRefresh() updates the database with data, as is, from the selected data range; it ignores cells outside the selected range and does not overwrite (or refresh) them. The selected range does not automatically get refreshed after submit; the user must manually refresh the grid to retrieve the updated data. For a successful submit, the user must select a range which constitutes a valid grid. Some of the behaviors can be controlled with the parameters noted below.

### 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.
- For the HypSubmitSelectedRangeWithoutRefresh function to work, the sheet must already be connected to a data source, and a valid range selection must be made before calling the function.
- For a regular submit, Oracle recommends using the HypSubmitData function.

## Syntax

```
HypSubmitSelectedRangeWithoutRefresh(vtSheetName)
```

```
ByVal vtSheetName As Variant
```

```
ByVal vtSubmitBlankCellsAsMissing As Variant
```

```
ByVal vtRefreshGridAfterSubmit As Variant
```

```
ByVal vtUseWholeSheet 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.

**vtSubmitBlankCellsAsMissing:** Unused. Should be set to False until the supported version of Oracle Hyperion Provider Services is available. If set to True results in an "Unsupported Provider Services" error.

**vtRefreshGridAfterSubmit:** If set to True, displays the selected grid result after a submit; if there is no submit operation, then sheet is not updated. If set to False, then submits only, and does not refresh the selected contents.

**vtUseWholeSheet:** Ignored in multiple-grid sheet. For single ad hoc grid sheet, if set to True, the whole sheet content is used. If set to False, uses the selected range as grid range.

### Return Value

Returns 0 if the selected cells are submitted successfully; otherwise, returns the appropriate error code.

### Example

```
Public Declare PtrSafe Function HypSubmitSelectedRangeWithoutRefresh
Lib "HsAddin" (ByVal vtSheetName As Variant, ByVal
vtSubmitBlankCellsAsMissing As Variant, ByVal vtRefreshGridAfterSubmit
As Variant, ByVal vtUseWholeSheet) As Long

Sub SubmitFreeform()
    sts = HypSubmitSelectedRangeWithoutRefresh("Sheet1", False, True,
True)
    'submits the whole grid in the sheet and refreshes the grid with
result
    sts = HypSubmitSelectedRangeWithoutRefresh("Sheet1", False, False,
False)
    'submits only the selected grid range and no refresh is performed,
'so the sheet content is left as is
End Sub
```

## HypSubmitSelectedDataCells

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### Description

HypSubmitSelectedDataCells() allows the selected data cells to be submitted.

#### Note:

- For free-form grids, this VBA function allows selected blank cells to be submitted as #Missing.
- 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.
- For the HypSubmitSelectedDataCells function to work, the sheet must already be connected to a data source, and a valid range selection must be made before calling the function.
- For a regular submit, Oracle recommends using the HypSubmitData function.

## Syntax

HypSubmitSelectedDataCells(vtSheetName)

ByVal vtSheetName As Variant

ByVal vtDataRange as Variant

ByVal vtSubmitBlankCellsAsMissingInFreeFormGrid 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.

**vtDataRange:** Unused. If vtDataRange is provided, it will be ignored. If this parameter is set to Empty, then the selected cells on the worksheet are submitted.

Currently, you must select data cells manually on the sheet or by using a Visual Basic Range Select macro, and then run the HypSubmitSelectedDataCells() function.

**vtSubmitBlankCellsAsMissingInFreeFormGrid:** Applies only to free-form grids. When set to "True", any blank cells in the selected range are submitted as #Missing. When set to false, blank cells will revert back to the value that was last stored with the provider.

## Return Value

Returns 0 if the selected cells are 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

```
Sub SubmitRange()  
    'Example assumes sheet is already connected and the data  
    'cells to be submitted are already selected  
    sts = HypSubmitSelectedDataCells("Sheet1", Empty, True)  
End Sub
```

# HypPivot

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning(ad hoc only), Oracle Hyperion Financial Management (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:

```
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:

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

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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (ad hoc only)

## Description

HypZoomIn() retrieves and expands data from Oracle Smart View for Office 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 Oracle Hyperion 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 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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
```

# 7

## Form Functions

### Related Topics

- [About Forms](#)
- [HypOpenForm](#)

### 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 Oracle Hyperion Financial Management, forms are called "data forms."

### HypOpenForm

**Cloud data provider types:** Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Hyperion Planning, Oracle Hyperion Financial Management

#### 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
```

# 8

## Cell Functions

### Related Topics

- [About Cell Functions](#)
- [HypGetDimMbrsForDataCell](#)
- [HypCell](#)
- [HypFreeDataPoint](#)
- [HypGetCellRangeForMbrCombination](#)
- [HypGetDataPoint](#)
- [HypIsCellWritable](#)
- [HypSetCellsDirty](#)
- [HypDeleteAllLROs](#)
- [HypDeleteLROs](#)
- [HypAddLRO](#)
- [HypUpdateLRO](#)
- [HypListLROs](#)
- [HypRetrieveLRO](#)
- [HypExecuteDrillThroughReport](#)
- [HypGetDrillThroughReports](#)

## About Cell Functions

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

## HypGetDimMbrsForDataCell

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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 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
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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**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.

**Example**

See [HypGetDataPoint](#) for an example of HypFreeDataPoint.

## HypGetCellRangeForMbrCombination

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

## 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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
```

## HypDeleteAllROs

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

**Description**

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

**Syntax**

HypDeleteAllROs (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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, vtType, vtName, vtDescription)
```

ByVal vtSheetName As Variant

ByVal vtSelectionRange As Variant

ByVal vtType 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

**vtType:** 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 vtType As  
Variant, ByVal vtName As Variant, ByVal vtDescription As Variant, ByRef  
vtLROIDs() As Variant) As Long
```



```
Sub Example_HypAddLRO()  
    sts = HypAddLRO("Sheet1", Range("B3"), 1, "", "Hello World")  
End Sub
```

## HypUpdateLRO

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### 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,vtIType, vtName, vtDescription)

ByVal vtSheetName As Variant

ByVal vtSelectionRange As Variant

ByVal vtID As Variant

ByVal vtIType 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

**vtIType:** 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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,vtIType, 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

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

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

# HypExecuteDrillThroughReport

**Data provider types:** Oracle Essbase

## Description

HypExecuteDrillThroughReport() executes the specified drill-through report. See also [HypGetDrillThroughReports](#).

## 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:** Oracle Essbase

### Description

HypGetDrillThroughReports() retrieves a list of drill-through reports. See also [HypExecuteDrillThroughReport](#).

### 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
```

# 9

## POV Functions

### Related Topics

- [About POV Functions](#)
- [HypSetPOV](#)
- [HypGetBackgroundPOV](#)
- [HypSetBackgroundPOV](#)
- [HypGetPagePOVChoices](#)
- [HypSetPages](#)
- [HypGetMembers](#)
- [HypSetMembers](#)
- [HypGetActiveMember](#)
- [HypSetActiveMember](#)
- [HypGetDimensions](#)
- [HypSetDimensions](#)

## About POV Functions

POV functions specify or retrieve settings for the POV.

## HypSetPOV

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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](#).

To set the POV more efficiently, [HypSetDimensions](#) may be used instead of HypSetPOV.

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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_ypSetBackgroundPOV()  
X=HypSetBackgroundPOV ("My Connection", "Year#Qtr1", "Market#East")  
End Sub
```

# HypGetPagePOVChoices

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

## 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

**Cloud data provider types:** Planning (forms only), Planning Modules (forms only), Financial Consolidation and Close (forms only), Tax Reporting (forms only)

**On-premises data provider types:** Oracle Hyperion Planning (forms only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

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

For Essbase, Oracle Analytics Cloud - Essbase, Oracle Hyperion Planning, 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, Oracle Analytics Cloud - Essbase, Oracle Hyperion Planning, 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.

```
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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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.

```
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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**Data provider types:****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.

```
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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypGetDimensions() returns an array containing the dimension names in the grid and an array containing their corresponding types. HypGetDimensions() can be used in place of the deprecated HypGetPOV() function.

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  
USERVAR = 5  
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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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
```

# 10

## Calculation Script and Business Rule Functions

### Related Topics

- [About Calculation Script and Business Rule Functions](#)
- [HypListCalcScripts](#)
- [HypExecuteCalcScript](#)
- [HypListCalcScriptsEx](#)
- [HypExecuteCalcScriptEx](#)
- [HypGetCalcScript](#)
- [HypExecuteCalcScriptString](#)
- [HypDeleteCalc](#)

## About Calculation Script and Business Rule Functions

Calculation script and business rule functions retrieve or execute calculation scripts and business rules.

## HypListCalcScripts

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning

### Description

HypListCalcScriptsEx() lists all business rules.

 **Note:**

See [Usage](#) in [HypExecuteCalcScriptEx](#) 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 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 vtBRNeedsPageInfo As
Variant, ByRef vtBRHidePrompts As Variant) As Long
Sub RunListCalcScriptsEx()
sts = HypListCalcScriptsEx(Empty, True, CubeName, BRNames, BRTypes,
BRHasPrompts, BRNeedsPageInfo, BRHidePrompts)
End Sub
```

# HypExecuteCalcScriptEx

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning

## Description

HypExecuteCalcScriptEx() executes the selected business rule.

## Syntax

HypExecuteCalcScriptEx(vtSheetName, vtCubeName, vtBRName, vtBRType, vtbBRHasPrompts, vtbBRNeedPageInfo, vtRTPNames(), vtRTPValues(), vtbShowRTPDlg, vtbRuleOnForm, vtbBRRanSuccessfully, vtCubeName,

vtBRName, vtBRType, vtbBRHasPrompts, vtbBRNeedPageInfo, vtbBRHidePrompts, vtRTPNamesUsed, vtRTPValuesUsed )

ByVal vtSheetName As Variant

ByVal vtCubeName As Variant

ByVal vtBRName As Variant

ByVal vtBRType As Variant

ByVal vtbBRHasPrompts As Variant

ByVal vtbBRNeedPageInfo As Variant

ByRef vtRTPNames() 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 types in Oracle Hyperion 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. Valid values are "graphical", "ecs", "sequence", and "native".

**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 Oracle Hyperion Planning) associated with the last run business rule

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

**vtBRType:** 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 vtBRType As Variant, ByVal vtbBRHasPrompts As Variant,
ByVal vtbBRNeedPageInfo 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 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) As Long

Sub Example_HypExecuteCalcScriptEx()

Dim oRet As Long
Dim oSheetName As StringDim oSheet As Worksheet
```

```
Dim vtCubeNames As Variant
Dim vtBRNames As Variant
Dim vtBRTypes As Variant
Dim vtBRHasPrompts As Variant
Dim vtBRNeedsPageInfo As Variant
Dim vtBRHidePrompts 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 vtbBRRanSuccessfully As Variant
Dim vtbBRRanSuccessfully2 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
        End If
    End If
End If
```



```

        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

        oRet = HypExecuteCalcScriptEx (oSheetName, vtOutCubeName,
            vtOutBRName, vtOutBRType, bBRHasPrompts, bBRNeedPageInfo, vtInRTPNames,
            vtInRTPValues, bShowDlg, bRuleOnForm, vtbBRRanSuccessfully2,
            vtOutCubeName, vtOutBRName, vtOutBRType, bBRHasPrompts,
            bBRNeedPageInfo, bBRHidePrompts, vtOutRTPNames, vtOutRTPValues)
            MsgBox ("Automated BR ran successfully - " &
                vtbBRRanSuccessfully2)
            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.

### **Not Calling HypListCalcScriptsEx 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.

- Mode 1: When vtbShowBRDlg is **True**:

- **Input Arguments:** vtSheetName, vtCubeName, vtbRuleOnForm are used. vtBRName, vtBRType, 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.
- Mode 2: When vtShowBRDlg argument is **False**:
  - **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.

### Calling HypListCalcScriptsEx Before HypExecuteCalcScriptEx

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 vtShowBRDlg 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 vtShowBRDlg 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. (In Oracle Hyperion Planning and EPM Cloud, the RTP flag may be True for a business rule when there are no RTPs to be displayed.)

- Mode 3: If the cube name, business rule name and business rule type are passed as empty in HypExecuteCalcScriptEx, 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.
- Mode 4: If the cube name, business rule name and business rule type are passed with filled values in HypExecuteCalcScriptEx, 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.

## HypGetCalcScript

**Data provider types:** Oracle Essbase

### Description

HypGetCalcScript() gets the calculation script string for a given calculation script.

Use with [HypExecuteCalcScriptString](#).

This function requires Oracle Hyperion Provider Services 11.1.2.4.017 or higher.

**Syntax**

HypGetCalcScript (vtSheetName, vtCalcScriptName, vtType, vtCalcScriptOutput)

ByVal vtSheetName As Variant

ByVal vtCalcScriptName As Variant

ByVal vtType As Variant

ByRef vtCalcScriptOutput 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, and the data source of the given sheet name will be used for connection.

**vtCalcScriptName:** Input parameter; the name of the calculation script for which the output is needed.

**vtType:** Input parameter; the type of calculation script file. Valid values are 1 or 2. Type 1 represents a `csc` file; type 2 represents a `rep` file. If input is incorrect, then treated as type 1.

**vtCalcScriptOutput:** Output parameter; the string which returns the calculation script string.

**Return Value**

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

**Example**

```
Sub calcScrVBATest()  
  
Sts = HypGetCalcScript("Sheet1", "rule1", 1, Script)  
  
Param = "_mySales=222;"  
  
Sts = HypExecuteCalcScriptString("Sheet1", Script, Param)  
  
End Sub
```

## HypExecuteCalcScriptString

**Data provider types:** Oracle Essbase

**Description**

HypExecuteCalcScriptString () executes a calculation script along with substitution variables.

**Description**

HypExecuteCalcScriptString () executes a calculation script, including any substitution variables.

Use with [HypGetCalcScript](#).

This function requires Oracle Hyperion Provider Services 11.1.2.4.017 or higher.

### Syntax

HypExecuteCalcScriptString (vtSheetName, vtCalcScript, vtSubstitutionVarList)

ByVal vtSheetName As Variant

ByVal vtCalcScript As Variant

ByVal vtSubstitutionVarList 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, and the data source of the given sheet name will be used for connection.

**vtCalcScript:** Input parameter; the calculation script string being executed.

**vtSubstitutionVarList:** Input parameter; the list of substitution variables to be used during execution.

### Return Value

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

### Example

```
Sub calcScrVBATest2()  
Script = "SET RUNTIMESUBVARS{salesNum  
=400;_mySales=300;myRTVar=@CHILDREN(~100~);myCOGS=30;};FIX  
(@INTERSECT(@CHILDREN(~100~), ~100-10~)) Sales =  
&_mySales;COGS=555;ENDFIX;"  
  
Script = Replace(Script, Chr(126), Chr(34)) 'replace ~ with "  
  
Param = "_mySales=222;"  
  
HypExecuteCalcScriptString("Sheet1", Script, Param)  
  
End Sub
```

## HypDeleteCalc

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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
```

# 11

## Calculation, Consolidation, and Translation Functions

### Related Topics

- [About Calculation, Consolidation, and Translation Functions](#)
- [HypCalculate](#)
- [HypCalculateContribution](#)
- [HypConsolidate](#)
- [HypConsolidateAll](#)
- [HypConsolidateAllWithData](#)
- [HypForceCalculate](#)
- [HypForceCalculateContribution](#)
- [HypForceTranslate](#)
- [HypTranslate](#)

### About Calculation, Consolidation, and Translation Functions

These functions execute calculation, consolidation, and translation operations on data for Oracle Hyperion Financial Management applications.

### HypCalculate

**Data provider types:** Oracle Hyperion Financial Management

#### 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:** Oracle Hyperion 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:** Oracle Hyperion Financial Management (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:** Oracle Hyperion Financial Management (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:** Oracle Hyperion Financial Management (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:** Oracle Hyperion 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:** Oracle Hyperion 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:** Oracle Hyperion 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:** Oracle Hyperion Financial Management (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
```

# 12

## Member Query Functions

### Related Topics

- [About Member Query Functions](#)
- [HypFindMember](#)
- [HypFindMemberEx](#)
- [HypGetAncestor](#)
- [HypGetChildren](#)
- [HypGetParent](#)
- [HypIsAttribute](#)
- [HypIsDescendant](#)
- [HypIsAncestor](#)
- [HypIsExpense](#)
- [HypIsParent](#)
- [HypIsChild](#)
- [HypIsUDA](#)
- [HypOtlGetMemberInfo](#)
- [HypQueryMembers](#)
- [HypGetMemberInformation](#)
- [HypGetMemberInformationEx](#)

## About Member Query Functions

Member query functions retrieve generation, level, attribute, and other information about members.

## HypFindMember

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

## Description

HypFindMemberEx() 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:** 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

```
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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### Description

HypGetChildren() returns the children for the specified member.

### Syntax

HypGetChildren (vtSheetName, vtMemberName, intChildCount, vtChildArray)

ByVal vtSheetName As Variant

ByVal vtMemberName As Variant

ByVal intChildCount As Integer

ByRef vtChildArray 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.

**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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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, ByRef vtParentName As Variant) As Long
```

```
Sub Example_HypGetParent
    Dim vtParent As Variant
    X = HypGetParent (Empty, "East", vtParent)
End sub
```

## HypIsAttribute

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### 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" & vtrret)
    End If
End Sub
```

## HypIsDescendant

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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
```

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### Description

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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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
MsgBox ("Return Value = " + vtRet)
End Sub
```

## HypQueryMembers

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle 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 = HYP\_SAMELEVEL
- 6 = HYP\_SAMEGENERATION
- 7 = HYP\_PARENT
- 8 = HYP\_DIMENSION
- 9 = HYP\_NAMEDGENERATION
- 10 HYP\_NAMEDLEVEL
- 11 HYP\_SEARCH
- 12 HYP\_WILDSEARCH
- 13 HYP\_USERATTRIBUTE
- 14 HYP\_ANCESTORS
- 15 HYP\_DTSMEMBER
- 16 HYP\_DIMUSERATTRIBUTES

**vtOption:** (integer) Options are dependent on the predicate. For the predicate values, HYP\_SEARCH and HYP\_WILDSEARCH, specify query options:

- HYP\_MEMBERONLY
- 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", HYP_DIMUSERATTRIBUTES, Empty,
Empty, Empty, Empty, vArray)

If IsArray(vArray) Then
    cbItems = UBound(vArray) + 1
    MsgBox ("Number of elements = " + Str(cbItems))
    For i = 0 To UBound(vArray)
        MsgBox ("Member = " + vArray(i))
    Next
Else
    MsgBox ("Return Value = " + Str(vArray))
End If
End Sub
```

## HypGetMemberInformation

**Data provider types:** Oracle 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 [Constants for Member Information](#).

**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
```

## Constants for Member Information

Following is a list of constants and strings for member information properties:

```
Global Const HYP_MI_NAME = "Name"  
Global Const HYP_MI_DIM = "Dim"  
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_CALC_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"  
Global Const HYP\_MI\_ATTRIBUTE\_DIMENSIONS = "AttributeDims"  
Global Const HYP\_MI\_ATTRIBUTE\_MEMBERS = "AttributeMbrs"  
Global Const HYP\_MI\_ATTRIBUTE\_TYPES = "AttributeTypes"  
Global Const HYP\_MI\_ALIAS\_NAMES = "AliasNames"  
Global Const HYP\_MI\_ALIAS\_TABLES = "AliasTables"  
Global Const HYP\_MI\_FORMULA = "Formula"  
Global Const HYP\_MI\_COMMENT = "Comment"  
Global Const HYP\_MI\_LAST\_FORMULA = "LastFormula"  
Global Const HYP\_MI\_UDAS = "Udas"

## HypGetMemberInformationEx

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises data provider types:** Oracle Essbase

### Description

HypGetMemberInformationEx returns all information about a member in an array.

### Syntax

HypGetMemberInformationEx (vtSheetName, vtMemberName, vtPropertyNames, vtPropertyValues, vtPropertyValueStrings)

ByVal vtSheetName As Variant

ByVal vtMemberName As Variant

ByRef vtPropertyNames As Variant

ByRef vtPropertyValues As Variant

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

**vtPropertyValues:** The property value 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 vtPropertyValues As Variant, ByRef
    vtPropertyValueStrings As Variant) As Long

sub Example_HypGetMemberInformationEx()
    sts = HypGetMemberInformationEx(Empty, "100-10", propertynames,
    propertyvalues, propertyvaluestrings)
End Sub
```



# 13

## Options Functions

### Related Topics

- [About Options Functions](#)
- [HypGetGlobalOption](#)
- [HypSetGlobalOption](#)
- [HypGetSheetOption](#)
- [HypSetSheetOption](#)
- [HypGetOption](#)
- [HypSetOption](#)
- [HypDeleteAllMRUItems](#)

## About Options Functions

Options functions set and retrieve information for global and/or sheet options, and enable deletion of MRU items.

## HypGetGlobalOption

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypGetGlobalOption() returns information about Oracle Smart View for Office global (default) options. Global options are options that apply to the entire current workbook and to any workbooks and worksheets that are created henceforth.

#### Tip:

Use [HypGetOption](#) to set both global (default) and sheet specific Smart View options so that you do not need separate VBA commands for the two option types.

### Syntax

```
HypGetGlobalOption(vtItem)
```

```
ByVal vtItem As Long
```

## Parameters

**vtItem:** The number that indicates which option is to be retrieved

[Table 1](#) lists the numbers of options and their return data types.

**Table 13-1 HypGetGlobalOption Parameter Numbers and Options**

vtItem	Option	Return Data Type
1	Use Excel formatting	Boolean
2	Use double-click for ad hoc operations	Boolean
3	Enable undo	Boolean
4	Not used	--
5	Specify message level setting: <ul style="list-style-type: none"> <li>• 0 = Information</li> <li>• 1 = Warnings</li> <li>• 2 = Errors</li> <li>• 3 = None</li> <li>• 4 = Extended info</li> </ul> See <a href="#">Notes</a> in <a href="#">HypSetGlobalOption</a> for information about this option and backward compatibility.	Integer
6	Use thousands separator	Boolean
7	Route messages to log file	Boolean
8	Clear log file on next launch	Boolean
9	Navigate without data	Boolean
10	Not used	--
11	Not used	--
12	Specify Meaningless label	Text
13	Reduce Excel file size	Boolean
14	Enable formatted strings	Boolean
15	Retain numeric formatting	Boolean
16	Enable enhanced comment handling	Boolean
17	Enable retain ribbon context	Boolean
18	Display Smart View Panel on startup	Boolean
19	Always show on refresh (in Comment Edit dialog box; available only if <b>Enhanced comment handling</b> is enabled and the grid contains comments)	Boolean
20	Enable profiling. Includes extended Info log entries and most function calls. Creates XML files for each Office application with active Smart View. Intended for debugging. Severely impacts performance. See <a href="#">Notes</a> in <a href="#">HypSetGlobalOption</a> for information about backward compatibility.	Boolean

## Return Value

Returns the appropriate return data type as shown in [Table 1](#); otherwise, returns the appropriate error code.

### Example

The following example sets the message level option and checks whether the value set is valid.

```
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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypSetGlobalOption() sets global Oracle Smart View for Office 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.

#### Tip:

Use [HypSetOption](#) to set both global (default) and sheet specific Smart View options so that you do not need separate VBA commands for the two option types.

### 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 1](#) 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.

### Notes

For backward compatibility, HypSetGlobalOption(5, 5) is supported for setting the profiling option only on sheets created *earlier* than Smart View 11.1.2.5.000. This VBA statement is not supported to set the profiling option on sheets created in Smart View 11.1.2.5.000 and *later*.

In Smart View 11.1.2.5.000 and later, using HypSetGlobalOption(5,5) to set the profiling option returns a value of -69, SS\_VBA\_DEPRECATED. Instead, use the following functions to get or set the profiling option:

- HypGetGlobalOption(20) and HypSetGlobalOption(20, True or False)
- HypGetOption(119, Var, "") and HypSetOption(119, True or False, "")

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypGetSheetOption() returns information about sheet level options.

 **Tip:**

Use [HypGetOption](#) to set both global (default) and sheet specific Oracle Smart View for Office options so that you do not need separate VBA commands for the two option types.

**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 1](#) for a list of values.

**Table 13-2 Options for vtItem**

vtItem	Option	Data Type and Values
1	Set zoom in level: <ul style="list-style-type: none"> <li>0 = Next level</li> <li>1 = All levels</li> <li>2 = Bottom level</li> <li>3 = Sibling level</li> <li>4 = Same level</li> <li>5 = Same generation</li> <li>6 = Formulas</li> </ul>	Number
2	Enable Include Selection setting	Boolean
3	Enable Within Selection Group setting	Boolean
4	Enable Remove Unselected Groups setting	Boolean
5	Specify Indent setting: <ul style="list-style-type: none"> <li>0 = No indentation</li> <li>1 = Indent sub items</li> <li>2 = Indent totals</li> </ul>	Number
6	Enable suppress missing setting	Boolean
7	Enable suppress zeros setting	Boolean
8	Enable suppress underscores setting	Boolean
9	Enable No Access setting	Boolean
10	Enable Repeated Member setting	Boolean
11	Enable Invalid setting	Boolean
12	Ancestor Position: <ul style="list-style-type: none"> <li>0 = Top</li> <li>1 = Bottom</li> </ul>	Number
13	Specify Missing Text label	Text
14	Specify No Access label	Text

Table 13-2 (Cont.) Options for vtItem

vtItem	Option	Data Type and Values
15	Cell Status: <ul style="list-style-type: none"> <li>• 0 = Data</li> <li>• 1 = Calculation Status</li> <li>• 2 = Process Management</li> </ul>	Number
16	Member Name Display options: <ul style="list-style-type: none"> <li>• 0 = Name Only</li> <li>• 1 = Name and Description</li> <li>• 2 = Description only</li> </ul>	Number

**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 If
End Sub
```

## HypSetSheetOption

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**Description**

HypSetSheetOption() sets sheet level options.

**Note:**

You can set only one option at a time.

 **Tip:**

Use [HypSetOption](#) to set both global (default) and sheet specific Oracle Smart View for Office options so that you do not need separate VBA commands for the two option types.

**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 1](#) 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

## Description

HypGetOption() retrieves Oracle Smart View for Office options that are both global (default) and sheet specific so that you do not need separate VBA commands for the two option types.

See also [HypGetGlobalOption](#) and [HypGetSheetOption](#).

## 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 1](#) 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 13-3 Option Constants for HypGetOption and HypSetOption**

Option	Constant	Data Type	Comment
HSV_ZOOMIN	1	Number	Sets zoom in level: <ul style="list-style-type: none"> <li>• 0 = Next level</li> <li>• 1 = All levels</li> <li>• 2 = Bottom level</li> <li>• 3 = Sibling level</li> <li>• 4 = Same level</li> <li>• 5 = Same generation</li> <li>• 6 = Formulas</li> </ul>
HSV_INCLUDE_SELECTION	2	Boolean	Selects the Include Selections check box
HSV_WITHIN_SELECTEDGROUP	3	Boolean	Selects the Within Selected Group check box
HSV_REMOVE_UNSELECTEDGROUP	4	Boolean	Selects the Remove Unselected Groups check box
HSV_INDENTATION	5	Number	Selects an Indentation option <ul style="list-style-type: none"> <li>• 0 = No indentation</li> <li>• 1 = Indent sub items</li> <li>• 2 = Indent totals</li> </ul>
HSV_SUPPRESSROWS_MISSING	6	Boolean	Suppresses rows that contain no data or are missing data
HSV_SUPPRESSROWS_ZEROS	7	Boolean	Suppresses rows that contain only zeroes
HSV_SUPPRESSROWS_UNDERSCORE	8	Boolean	Suppresses rows that contain underscore characters in member names
HSV_SUPPRESSROWS_NOACCESS	9	Boolean	Suppress rows that contain data that the user does not have the security access to view
HSV_SUPPRESSROWS_REPEATEDMEMBERS	10	Boolean	Suppresses rows that contain repeated member names, regardless of grid orientation.
HSV_SUPPRESSROWS_INVALID	11	Boolean	Suppresses rows that contain only invalid values
HSV_ANCESTOR_POSITION	12	Number	Specifies an ancestor position in hierarchies: <ul style="list-style-type: none"> <li>• 0 = Top</li> <li>• 1 = Bottom</li> </ul>
HSV_MISSING_LABEL	13	Text	Displays #Missing, #Numeric Zero, or the text of your choice in data cells that contain missing data.
HSV_NOACCESS_LABEL	14	Text	Displays #NoAccess, #Numeric Zero, or the text of your choice in data cells that the user does not have permission to view.
HSV_CELL_STATUS	15	Number	As an alternative to displaying actual data, displays the calculation or process status of the cells: <ul style="list-style-type: none"> <li>• 0 = Data</li> <li>• 1 = Calculation Status</li> <li>• 2 = Process Management</li> </ul>

Table 13-3 (Cont.) Option Constants for HypGetOption and HypSetOption

Option	Constant	Data Type	Comment
HSV_MEMBER_DISPLAY	16	Number	Specifies how to display member names in cells: <ul style="list-style-type: none"> <li>• 0 = Name Only</li> <li>• 1 = Name and Description</li> <li>• 2 = Description only</li> </ul>
HSV_INVALID_LABEL	17	Text	Displays #Invalid, #Numeric Zero, or the text of your choice in data cells that contain invalid data.
HSV_SUBMITZERO	18	Boolean	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.
HSV_MOVEESSBASEMEMBERFOR MULAONZOOM	19	Boolean	When set to True, moves member formulas on zoom in and zoom out. When enabled, this behavior can impact performance during zoom in and zoom out. Therefore, the default setting is False.  This option becomes irrelevant when the grid contains any data or non-data formulas, or a zoom-in is performed in a free-form grid, in which case, the member formula will move by default.
HSV_PRESERVE_ESSBASECOMMENT_UNKNOWNMEMBERS	20	Boolean	Preserves Essbase comments. If set to false, an "unknown member" error message from Essbase is displayed.
HSV_PRESERVE_FORMULA_COMMENT	21	Boolean	Preserves formulas and comments on the grid during queries.
HSV_22	22		Reserved for future use
HSV_FORMULA_FILL	23	Boolean	Propagates formulas associated with member cells to the members retrieved as a result of zooming in. 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.  Applies to formulas in both member and data cells.
HSV_PRESERVE_FORMULA_ONPOVCHANGE	24	Boolean	Specific to form sheets. Preserves formulas in cells when user refreshes or makes changes to the POV. When set to False, any formulas in the sheet are lost.
HSV_EXCEL_FORMATTING	30	Boolean	Selects the Excel formatting check box

**Table 13-3 (Cont.) Option Constants for HypGetOption and HypSetOption**

Option	Constant	Data Type	Comment
HSV_RETAIN_NUMERIC_FORMATTING	31	Boolean	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.
HSV_THOUSAND_SEPARATOR	32	Boolean	Uses a comma or other thousands separator in numerical data. Do not use # or \$ as the thousands separator in Excel International Options.
HSV_NAVIGATE_WITHOUTDATA	33	Boolean	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.
HSV_ENABLE_FORMATSTRING	34	Boolean	Essbase-specific. Essbase provides a format string to be associated with different data types. Once enabled, shows user specific text instead of numbers.
HSV_ENHANCED_COMMENT_HANDLING	35	Boolean	Enables review and correction of comments and member names in ad hoc grids that contain comments.
HSV_ADJUSTCOLUMNWIDTH	36	Boolean	Adjusts column widths to fit cell contents automatically.
HSV_DECIMALPLACES	37	Number	Specifies the number of decimal places to display.
HSV_SCALE	38	Number	Specifies the scaling of numeric data, which is displayed based on the scale selected.
HSV_MOVEFORMATS_ON_ADHOC	39	Boolean	Copies parent cell formatting to zoomed in cells and retains this formatting even if the cell location changes after an operation.
HSV_DISPLAY_INVALIDDATA	40	Boolean	Displays invalid data.
HSV_SUPPRESSCOLUMNS_MISSING	41	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.)
HSV_SUPPRESSCOLUMNS_ZEROES	42	Boolean	Suppresses columns that contain only zeroes.
HSV_SUPPRESSCOLUMNS_NOACCESS	43	Boolean	Suppresses columns that contain data that the user does not have the security access to view.
HSV_SUPPRESS_MISSINGBLOCKS	44	Boolean	Suppresses blocks of cells for which no data exists in the database.
HSV_REPEATMEMBERS_IN_FORMS	45	Boolean	Facilitates the readability of forms by allowing member names to appear on each row of data.

Table 13-3 (Cont.) Option Constants for HypGetOption and HypSetOption

Option	Constant	Data Type	Comment
HSV_DOUBLECLICK_FOR_ADHOC	101	Boolean	Specifies that double-clicking retrieves the default grid in a blank worksheet and thereafter zooms in or out on the cell contents.
HSV_UNDO_ENABLE	102	Boolean	Enables and disables Undo. Specify the number undo operations allowed with the HSV_NUMBER_OF_UNDO_ACTION parameter.
HSV_103	103		Reserved for future use.
HSV_LOGMESSAGE_DISPLAY	104	Number	Specifies message display level setting: <ul style="list-style-type: none"> <li>• 0 = Information</li> <li>• 1 = Warnings</li> <li>• 2 = Errors</li> <li>• 3 = None</li> <li>• 4 = Extended info</li> </ul>
HSV_ROUTE_LOGMESSAGE_TO_FILE	105	Boolean	Enables and disables the Route Messages to File check box.
HSV_CLEAR_LOG_ON_NEXTLAUNCH	106	Boolean	Clears the log file starting with the next log message generation, which will be seen after Excel is closed.
HSV_REDUCE_EXCEL_FILESIZE	107	Boolean	Should always be enabled except in the following cases, when it should not be used: <ul style="list-style-type: none"> <li>• 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. In these workbooks: <ul style="list-style-type: none"> <li>– Grids that contain functions must be refreshed before data can be displayed.</li> <li>– In ad hoc mode, POV settings are lost; the behavior is similar to that of a fresh ad hoc grid.</li> </ul> </li> <li>• 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</li> </ul>
HSV_ENABLE_RIBBON_CONTEXT	108	Boolean	Displays the active data provider ribbon automatically after you use a button on the Smart View ribbon.

**Table 13-3 (Cont.) Option Constants for HypGetOption and HypSetOption**

Option	Constant	Data Type	Comment
HSV_DISPLAY_HOME_PANEL_ONS TARTUP	109	Boolean	Enables and disables the Display on Startup check box on the Smart View Home panel. When enabled, shows the Smart View Home Panel when the Panel icon is selected in the Smart View ribbon. When disabled, the last opened panel is shown.
HSV_SHOW_COMMENT_DIALOG_O N_REFRESH	110	Boolean	When enabled, if the grid has comments, the comment editor is displayed to users upon refresh. When disabled, users can launch the comment editor from the Smart View ribbon.
HSV_NUMBER_OF_UNDO_ACTION	111	Number	The number of Undo and Redo actions permitted on an operation (0 through 100). Works in conjunction with the HSV_UNDO_ENABLE parameter.
HSV_NUMBER_OF_MRU_ITEMS	112	Number	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.
HSV_ROUTE_LOGMESSAGE_FILE _LOCATION	113	Text	Saves log messages in a file.
HSV_DISABLE_SMARTVIEW_IN_O UTLOOK	114	Boolean	Disables Smart View in Outlook if you do not want to use Smart View task lists in Outlook.
HSV_DISPLAY_SMARTVIEW_SHOR TCUT_MENU_ONLY	115	Boolean	Displays only Smart View menu items on shortcut menus. Otherwise, shortcut menus display both Excel and Smart View items.
HSV_DISPLAY_DRILL_THROUGH_ REPORT_TOOLTIP	116	Boolean	Displays by default lists of available drill-through reports for cells whenever you mouse over them.
HSV_SHOW_PROGRESSINFORMA TION	117	Boolean	Specifies that the Smart View Progress status bar will appear when an operation begins after the number of seconds defined in HSV_PROGRESSINFO_TIMEDELAY.
HSV_PROGRESSINFO_TIMEDELA Y	118	Number	The time, in seconds, after which the Smart View Progress status bar appears when an operation begins.
HSV_ENABLE_PROFILING	119	Boolean	Creates extended Info log entries and most function calls. Creates XML files for each Office application with active Smart View. Intended for debugging. Severely impacts performance.

Table 13-3 (Cont.) Option Constants for HypGetOption and HypSetOption

Option	Constant	Data Type	Comment
HSV_REFRESH_SELECTED_DEPENDENT_FUNCTIONS	121	Boolean	Executes dependent functions on the same sheet before executing the selected functions.
HSV_IMPROVE_METADATA_STORAGE	122	Boolean	<p>Allows for more efficient storage of internal data structures.</p> <p>This option pertains to interoperability between different versions of Smart View.</p> <p>When this option is set to True, Smart View maintains two copies of metadata for compatibility purpose, which may result in slower overall performance.</p> <p>If all users in your organization are on Smart View 9.3.1.6 or higher, then this option should always be set to True.</p> <p>Set this option to False in the following situations:</p> <ul style="list-style-type: none"> <li>You send an Excel workbook to users on Smart View releases earlier than 9.3.1.6, or to users on Microsoft Office 2002 and earlier, regardless of Smart View release</li> <li>You open a workbook sent from users on Smart View releases earlier than 9.3.1.6, or from users on Microsoft Office 2002 and earlier, regardless of Smart View release</li> </ul>

## HypSetOption

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypSetOption() enables you to set Oracle Smart View for Office options as both global (default) and sheet specific so that you do not need separate VBA commands for the two option types.

See also [HypSetGlobalOption](#) and [HypSetSheetOption](#).

### 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 1](#) 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

Sub SetOptn()
    HypSetOption (HSV_REFRESH_SELECTED_DEPENDENT_FUNCTIONS, False, "")
    HypSetOption (HSV_IMPROVE_METADATASTORAGE, False, "")
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
```



# 14

## Dynamic Link Functions

### Related Topics

- [About Dynamic Link Views](#)  
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.
- [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.
- [Automating Macro Execution](#)  
Automate execution of a macro through the Oracle Smart View for Office menu.
- [HypUseLinkMacro](#)
- [HypSetLinkMacro](#)
- [HypGetLinkMacro](#)
- [HypGetSourceGrid](#)
- [HypDisplayToLinkView](#)
- [HypGetConnectionInfo](#)
- [HypSetConnectionInfo](#)
- [HypGetRowCount](#)
- [HypGetColCount](#)
- [HypGetPOVCount](#)
- [HypGetRowItems](#)
- [HypSetRowItems](#)
- [HypGetCollItems](#)
- [HypSetCollItems](#)
- [HypGetPOVItems](#)
- [HypSetPOVItems](#)

## About Dynamic Link Views

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 Oracle Smart View for Office. 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](#)

[Automating Macro Execution](#)

## 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

Automate execution of a macro through the Oracle Smart View for Office 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

**Data provider types:**



## Note:

The link action is performed with the latest content of the dynamic link query.

## Description

HypDisplayToLinkView() displays Office documents to Word or PowerPoint or grids to Excel.

## Syntax

HypDisplayToLinkView (vtDocumentType, vtDocumentPath)

ByVal vtDocumentType As Variant

ByVal vtDocumentPath As Variant

## Parameters

**vtDocumentType:** The destination for the link view. Valid values:

- EXCEL\_APP
- WORD\_APP
- PPOINT\_APP

**vtDocumentPath:** 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only),

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (ad hoc only)

### 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.

### Description

HypGetConnectionInfo() returns the connection information for the dynamic link query.

### 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### 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)
Sts = HypSetConnectionInfo("localhost", "UserName", "Password",
"Sample", "Basic", "SampleBasic", "http://localhost:13080/aps/
SmartView", provider)
End Sub

```

## HypGetRowCount

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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 *n*th row dimension name

**vtMemberNames:** Output parameter; the members for the *n*th 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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (ad hoc only)

## Description

Sets the members for the *n*th row dimension for this dynamic link query. If the *n*th 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
```

# HypGetCollItems

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (ad hoc only)

## Description

HypGetCollItems() 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

```
HypGetCollItems(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
```

## HypSetCollItems

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning, Planning Modules, Financial Consolidation and Close, Tax Reporting

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning, Oracle Hyperion Financial Management

### Description

HypSetCollItems() sets the members for the *n*th column dimension for the dynamic link query. If the *n*th 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

HypSetCollItems (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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

**Cloud data provider types:** Oracle Analytics Cloud - Essbase, Planning (ad hoc only), Planning Modules (ad hoc only), Financial Consolidation and Close (ad hoc only), Tax Reporting (ad hoc only)

**On-premises data provider types:** Oracle Essbase, Oracle Hyperion Planning (ad hoc only), Oracle Hyperion Financial Management (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)
```

```
    Sts = HypSetPOVItems ("Scenario#Scenario", "Measures#Measures")  
End sub
```

# 15

## MDX Query Functions

### Related Topics

- [About MDX](#)  
Multidimensional Expressions (MDX) language is used to develop scripts or applications to query and report against data and metadata in Oracle Essbase databases.
- [HypExecuteMDXEx](#)

### About MDX

Multidimensional Expressions (MDX) language is used to develop scripts or applications to query and report against data and metadata in Oracle Essbase databases.

For information about MDX, see the Essbase documentation set.

### HypExecuteMDXEx

**Cloud data provider types:** Oracle Analytics Cloud - Essbase

**On-premises 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 vtMbrIDType 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
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"
vtAliasTable = "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
```

# Oracle Journals for Financial Management Functions

## Related Topics

- [About Oracle Journals for Financial Management Functions](#)  
VBA functions support operations when connected to an Oracle Hyperion Financial Management data source.
- [Registering the Oracle Journals VBA Functions Using RegAsm](#)  
Before you begin creating and editing VBA functions for Oracle Journals for Financial Management, you must register the Oracle Journals for Financial Management DLL using RegAsm.
- [Preparing to Work with Oracle Journals for Financial Management Functions](#)  
Continue the setup process for creating and editing VBA functions for Oracle Journals for Financial Management by adding references to the Journals type library.
- [Instantiating an Oracle Journals for Financial Management Extension Object](#)  
The Oracle Journals for Oracle Hyperion Financial Management extension exposes its automation interface through COM.
- [Oracle Journals for Financial Management Extension Functions](#)

## About Oracle Journals for Financial Management Functions

VBA functions support operations when connected to an Oracle Hyperion Financial Management data source.

To use the VBA functions in this chapter, Financial Management 11.1.2.4.204 with the 11.1.2.4.204 PSE 25575478 is required. Additionally, you must install Oracle Smart View for Office 11.1.2.5.700 and the Oracle Journals for Financial Management extension that is included with the 11.1.2.5.700 release.

You must complete the following procedures in order before you begin working with the VBA functions for Oracle Journals for Financial Management.

1. [Registering the Oracle Journals VBA Functions Using RegAsm](#)
2. [Preparing to Work with Oracle Journals for Financial Management Functions](#)
3. [Instantiating an Oracle Journals for Financial Management Extension Object](#)

# Registering the Oracle Journals VBA Functions Using RegAsm

Before you begin creating and editing VBA functions for Oracle Journals for Financial Management, you must register the Oracle Journals for Financial Management DLL using RegAsm.

To register the Oracle Journals DLL using RegAsm:

1. Note the file location of the Oracle Journals DLL (`Journals.dll`); generally, located in:

```
%APPDATA%  
\Oracle\SmartView\extensions\Bin\Oracle.SmartView.SVFMJournal\bin
```

2. From Windows Explorer, navigate to the Oracle Smart View for Office installation directory; generally located in:

```
C:\Oracle\SmartView\bin
```

3. Copy the `Oracle.SmartView.Interop.SVRC.dll` file from this location to the Oracle Journals file location noted in [step 1](#).
4. As administrator, open a command prompt and navigate to one of the following locations:

- For Microsoft Office 64-bit:

```
C:\Windows\Microsoft.NET\Framework64\v4.0.30319
```

- For Microsoft Office 32-bit:

```
C:\Windows\Microsoft.NET\Framework\v4.0.30319
```

5. Execute the following command:

```
regasm /codebase /tlb <file path from Step 1>\Journals.dll  
You should see this message:
```

```
"Assembly exported to %APPDATA%  
\Oracle\SmartView\extensions\Bin\Oracle.SmartView.SVFMJournal\bin\Jo  
urnals.tlb and the type library was registered successfully."
```

Any warnings can be ignored.

Continue with [Preparing to Work with Oracle Journals for Financial Management Functions](#).



## Preparing to Work with Oracle Journals for Financial Management Functions

Continue the setup process for creating and editing VBA functions for Oracle Journals for Financial Management by adding references to the Journals type library.

### Note:

Before you begin this procedure, you must complete the steps in [Registering the Oracle Journals VBA Functions Using RegAsm](#).

To add Oracle Journals for Oracle Hyperion Financial Management 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 reference **Journals**.

Continue with [Instantiating an Oracle Journals for Financial Management Extension Object](#).

## Instantiating an Oracle Journals for Financial Management Extension Object

The Oracle Journals for Oracle Hyperion Financial Management extension exposes its automation interface through COM.

To make an automation call, an Oracle Journals for Financial Management COM object must first be instantiated.

All Oracle Journals for Financial Management automation functions are defined in the IJournalVBA interface, and the JournalVBA class implements those functions. Therefore, in any Oracle Journals automation call, you must include the variable declarations that are described in the following procedure.

### Note:

Before you begin this procedure, you must first complete the steps in [Registering the Oracle Journals VBA Functions Using RegAsm](#), and second, [Preparing to Work with Oracle Journals for Financial Management Functions](#).

To create the variable declarations that will be included in all functions:

1. Declare a variable of type IJournalVBA.
2. Set the variable to an object of type JournalVBA.

The resulting lines are:

```
Dim jObj As IJournalVBA  
  
Set jObj = New JournalVBA
```

3. Include the lines from the previous step in each of your functions.

You are ready to begin creating and working with the Oracle Journals for Financial Management functions. See [Oracle Journals for Financial Management Extension Functions](#) for a complete listing of the functions available and their usage.

## Oracle Journals for Financial Management Extension Functions

### Related Topics

- [ListJournals](#)
- [OpenJournal](#)
- [SetJournalProperty](#)
- [ListTemplates](#)
- [CreateJournal](#)
- [SaveJournal](#)
- [PerformAction](#)
- [ValidateJournal](#)

## ListJournals

### Description

ListJournals() lists all the available Oracle Hyperion Financial Management journals.

### Syntax

```
Function ListJournals(  
    dims() As String,  
    dimVals() As String,  
    jrnIDs() As String,  
    jrnLabels() As String  
) As Long
```

### Parameters

**dims:** An input argument. Provide the list of dimensions as an array of strings.

**dimVals:** An input argument. Provide the list of dimension values as an array of strings.

**jrnlIDs:** An output argument. Returns the Journal IDs as an array of strings.

**jrnlLabels:** An output argument. Returns the Journal Labels as an array of strings.

### Return Value

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

### Example

The following example sets the option to display no messages.

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

Sub TestListJournals
'Connect to an HFM data source
HypConnect "Sheet1", "admin", "password", "connName"

Set jObj = New JournalVBA
    jObj.UseActiveConnectionContext

    dims(0) = HFM_JOURNAL_DIM_SCENARIO
dims(1) = HFM_JOURNAL_DIM_YEAR
    dims(2) = HFM_JOURNAL_DIM_PERIOD
    dims(3) = HFM_JOURNAL_DIM_VALUE

    dimVals(0) = "Actual"
    dimVals(1) = "2007"
    dimVals(2) = "January"
    dimVals(3) = "<Entity Curr Adjs>"

    Dim jrnlIds() As String
    Dim jrnlLabels() As String

    Dim retVal as Long
    retVal = jObj.ListJournals(dims, dimVals, jrnlIds, jrnlLabels)

    If retVal = 0 Then
        Debug.Print "Following are the Journal IDs and their Labels..."
        Debug.Print "Journal Id          Name"
        Dim i As Integer
        For i = 0 To UBound(jrnlIds)
            Debug.Print Spc(5); jrnlIds(i); Spc(10); jrnlLabels(i)
        Next
    Else
        Debug.Print "ListJournals Failed!!!"
    End If

End Sub
```

## OpenJournal

### Description

OpenJournal() Opens the specified journal.

### Syntax

```
Function OpenJournal(  
    jrnID As String,  
    dims() As String,  
    dimVals () As String  
    ) As Long
```

### Parameters

**jrnIID:** An input argument. Provide the journal ID of the journal to be opened. Journals IDs can be obtained by calling ListJournals().

**dims:** An input argument. Provide the list of dimensions as an array of strings.

**dimVals:** An input argument. Provide the list of dimension values as an array of strings.

### Return Value

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

### Example

```
Sub OpenJournal  
    'Firstly, get the list of Journals  
    ListJournals  
  
    Dim jrnIID As String  
    jrnIID = 1  
  
    retVal = jObj.OpenJournal(jrnIID, dims, dimVals)  
  
    If retVal = 0 Then  
        Debug.Print "OpenJournal Succeeded"  
    Else  
        Debug.Print "OpenJournal Failed!!!"  
    End If  
  
End Sub
```

## SetJournalProperty

### Description

SetJournalProperty() sets the specified properties for the currently open journal.

**Syntax**

```
Function SetJournalProperty(
sheetName As String,
props() As String,
propVals() As String
) As Long
```

**Parameters**

**sheetName:** An input argument. Provide the name of the sheet in which a Journal is open.

**props:** An input argument. Provides the list of properties as an array of strings.

**propVals:** An input argument. Provide the list of property values as an array of strings.

**Return Value**

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

**Example**

```
Public Declare Function HypRetrieve Lib "HsAddin" (ByVal vtSheetName As
Variant) As Long
Sub SetJournalProperty()

    Dim props(6) As String
    props(0) = HFM_JOURNALPROP_LABEL
    props(1) = HFM_JOURNALPROP_DESCRIPTION
    props(2) = HFM_JOURNALPROP_TYPE
    props(3) = HFM_JOURNALPROP_BALANCE_TYPE
    props(4) = HFM_JOURNALPROP_GROUP
    props(5) = HFM_JOURNALPROP_SECURITY
    props(6) = HFM_JOURNALPROP_READONLY

    Dim propVals(6) As String
    propVals(0) = "J001"
    propVals(1) = "Test1"
    propVals(2) = HFM_JOURNALPROP_TYPE_REGULAR
    propVals(3) = HFM_JOURNALPROP_BALANCETYPE_BALANCED
    propVals(4) = HFM_JOURNALPROP_GROUP_ALLOCATION
    propVals(5) = HFM_JOURNALPROP_SECURITY_ACCOUNTS
    propVals(6) = "0"

    Dim retVal As Long
    Set jObj = New JournalVBA
    retVal = jObj.SetJournalProperty("Sheet1", props, propVals)

    If retVal = 0 Then
        Debug.Print "SetJournalProperty Succeeded"

        Dim status As Long
```

```

        status = HypRetrieve(Empty)
        Debug.Print "HypRetrieve returned Status as "; status
    Else
        Debug.Print "SetJournalProperty Failed"
    End If
End Sub

```

## ListTemplates

### Description

ListTemplates() lists all the available journal templates.

### Syntax

```

Function ListTemplates(
    templateType() As String,
    templateName() As String
) As Long

```

### Parameters

**templateType:** An output argument. Returns the available template types(Standard or Recurring) as an array of strings.

**templateName:**An output argument. Returns the template names an array of strings.

### Return Value

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

### Example

```

Public Declare Function HypConnect Lib "HsAddin" (ByVal vtSheetName
As Variant, ByVal vtUserName As Variant, ByVal vtPassword As Variant,
ByVal vtFriendlyName As Variant) As Long
Sub TestListTemplates
    'Connect to an HFM data source
    HypConnect "Sheet1", "admin", "password", "connName"

    Set jObj = New JournalVBA
        jObj.UseActiveConnectionContext

        Dim templateType() As String
        Dim templateName() As String

        retVal = jObj.ListTemplates(templateType, templateName)

        If retVal = 0 Then
            Debug.Print "Following are the Template types and their
names..."
            Debug.Print "Type          Name"

```

```

        Dim i As Integer
        For i = 0 To UBound(templateType)
            Debug.Print Spc(5); templateType(i); Spc(10);
templateName(i)
        Next
    Else
        Debug.Print "ListTemplates Failed!!!"
    End If
End Sub

```

## CreateJournal

### Description

CreateJournal() creates a blank journal or a journal based on a Standard or Recurring template.

### Syntax

```

Function CreateJournal(
    dims() As String,
    dimVals () As String,
    templateType As String,
    templateNames() As String
) As Long

```

### Parameters

**dims:** An input argument. Provide the list of dimensions as an array of strings.

**dimVals:** An input argument. Provide the list of dimension values as an array of strings.

**templateType:** An input argument. Value is one of the following:

- HFM\_JOURNAL\_TEMPLATE\_TYPE\_BLANK
- HFM\_JOURNAL\_TEMPLATE\_TYPE\_STANDARD
- HFM\_JOURNAL\_TEMPLATE\_TYPE\_RECURRING

These are defined in HFMJournalVBA.bas.

**templateNames:** An input argument. Provide the template names as an array of strings.

### Return Value

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

### Example

```

Sub CreateJournal
    'Connect to an HFM data source

```

```

HypConnect "Sheet1", "admin", "password", "connName"

Set jObj = New JournalVBA
jObj.UseActiveConnectionContext

dims(0) = HFM_JOURNAL_DIM_SCENARIO
dims(1) = HFM_JOURNAL_DIM_YEAR
    dims(2) = HFM_JOURNAL_DIM_PERIOD
    dims(3) = HFM_JOURNAL_DIM_VALUE

    dimVals(0) = "Actual"
    dimVals(1) = "2007"
    dimVals(2) = "January"
dimVals(3) = "<Entity Curr Adjs>"

    Dim templateNames(0) As String
    templateNames(0) = "Templatel"
retVal = jObj.CreateJournal(dims, dimVals,
HFM_JOURNAL_TEMPLATE_TYPE_STANDARD, templateNames)

    If retVal = 0 Then
    Debug.Print "Create Journal from Template Succeeded"
    Else
    Debug.Print "Create Journal from Template Failed!!!"
    End If
End Sub

```

## SaveJournal

### Description

SaveJournal() saves a journal on the Oracle Hyperion Financial Management server.

### Syntax

```

Function SaveJournal(
props() As String,
propVals() As String,
dims() As String,
dimVals () As String
) As Long

```

### Parameters

**props:** An input argument. Provide the list of properties as an array of strings.

**propVals:** An input argument. Provide the list of property values as an array of strings.

**dims:** An input argument. Provide the list of dimensions as an array of strings.

**dimVals:** An input argument. Provide the list of dimension values as an array of strings.



**Return Value**

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

**Example**

```

Sub SaveJournal()

Connect to an HFM data source
HypConnect "Sheet1", "admin", "password", "connName"

Set jObj = New JournalVBA
jObj.UseActiveConnectionContext

    Dim props(6) As String
    props(0) = HFM_JOURNALPROP_LABEL
    props(1) = HFM_JOURNALPROP_DESCRIPTION
    props(2) = HFM_JOURNALPROP_TYPE
    props(3) = HFM_JOURNALPROP_BALANCE_TYPE
    props(4) = HFM_JOURNALPROP_GROUP
    props(5) = HFM_JOURNALPROP_SECURITY
    props(6) = HFM_JOURNALPROP_READONLY

    Dim propVals(6) As String
    propVals(0) = "J001"
    propVals(1) = "Test1"
    propVals(2) = HFM_JOURNALPROP_TYPE_REGULAR
    propVals(3) = HFM_JOURNALPROP_BALANCETYPE_BALANCED
    propVals(4) = HFM_JOURNALPROP_GROUP_ALLOCATION
    propVals(5) = HFM_JOURNALPROP_SECURITY_ACCOUNTS
    propVals(6) = "0"

    dims(0) = HFM_JOURNAL_DIM_SCENARIO
    dims(1) = HFM_JOURNAL_DIM_YEAR
    dims(2) = HFM_JOURNAL_DIM_PERIOD
    dims(3) = HFM_JOURNAL_DIM_VALUE

    dimVals(0) = "Actual"
    dimVals(1) = "2007"
    dimVals(2) = "March"
    dimVals(3) = "<Entity Curr Adjs>"

    retVal = jObj.SaveJournal(props, propVals, dims, dimVals)
    If retVal = 0 Then
        Debug.Print "SaveJournal Succeeded"
    Else
        Debug.Print "SaveJournal Failed!!!"
    End If
End Sub

```

## PerformAction

### Description

PerformAction() performs one of the following operations: POST, UNPOST, SUBMIT, UNSUBMIT, APPROVE, REJECT or DELETE.

### Syntax

```
Function PerformAction(  
action As String  
) As Long
```

### Parameters

action : An input argument. Value is one of the following:

HFM\_JOURNAL\_ACTION\_POST or  
HFM\_JOURNAL\_ACTION\_UNPOST or  
HFM\_JOURNAL\_ACTION\_SUBMIT or  
HFM\_JOURNAL\_ACTION\_UNSUBMIT or  
HFM\_JOURNAL\_ACTION\_APPROVE or  
HFM\_JOURNAL\_ACTION\_REJECT or  
HFM\_JOURNAL\_ACTION\_DELETE.

These are defined in HFMJournalVBA.bas.

### Return Value

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

### Example

```
Sub TestPerformAction()  
    Set jObj = New JournalVBA  
  
    retVal = jObj.PerformAction(HFM_JOURNAL_ACTION_UNSUBMIT)  
    If retVal = 0 Then  
        Debug.Print "PerformAction Succeeded."  
    Else  
        Debug.Print "PerformAction Succeeded!!!"  
    End If  
End Sub
```

## ValidateJournal

### Description

ValidateJournal() validates dimensions, members, and journal properties by communicating with Oracle Hyperion Financial Management server.

### Syntax

```
Function ValidateJournal(
    errNames() As String,
    errValues() As String
) As Long
```

### Parameters

**errNames:** An output argument. Returns the categories of validations that failed as an array of strings.

**errValues:** An output argument. Returns which validation failed as an array of strings.

### Return Value

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

### Example

```
Sub TestValidateJournal()
    'Call TestCreateJournal
    TestCreateJournal

    Dim errNames() As String
    Dim errValues() As String

    retVal = jObj.ValidateJournal(errNames, errValues)
    If retVal = 0 Then
        Debug.Print "Validation Succeeded"
        If (Not Not errNames) <> 0 Then
            Debug.Print "Following are the Validation Errors"
            Debug.Print "Error Type      Description"
            Dim i As Integer
            For i = 0 To UBound(errNames)
                Debug.Print errNames(i); Spc(10); errValues(i)
            Next
        Else
            Debug.Print "No Validation Errors."
        End If
    Else
        Debug.Print "ValidateJournal Failed : " + GetErrorMessage(retVal)
    End If
End Sub
```

# Oracle Analytics Cloud Functions

## Related Topics

- [About Oracle Analytics Cloud Functions](#)
- [Preparing to Work with Oracle Analytics Cloud Functions](#)  
Before you begin creating and editing VBA functions for Oracle Analytics Cloud, you must first add references to the Oracle Smart View BI Extension type library and Oracle Smart View for Office type library.
- [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.
- [Oracle Smart View BI Extension Functions](#)

## About Oracle Analytics Cloud Functions

VBA functions support Oracle Smart View for Office operations when connected to an Oracle Analytics Cloud data source.

## Preparing to Work with Oracle Analytics Cloud Functions

Before you begin creating and editing VBA functions for Oracle Analytics Cloud, you must first add references to the Oracle Smart View BI Extension type library and Oracle 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](#).

## 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 Analytics Cloud automation functions are defined in the IBIReport interface, and the SmartViewOBIEEAutomation class implements those functions. Therefore, in any Oracle Analytics Cloud 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:

```
Dim obiee As IBIReport  
Set obiee = New SmartViewOBIEEAutomation
```

3. Include the lines from step [ep 2](#) in each of your functions.

You are ready to begin creating and working with the Oracle Smart View BI Extension functions. See [Oracle Analytics Cloud Functions](#) for a complete listing of the functions available and their usage.

## Oracle Smart View BI Extension Functions

### Related Topics

- [InsertView](#)
- [EditPrompts](#)
- [EditPagePrompts](#)
- [GetPagePrompts](#)
- [Retrieve](#)
- [DeleteView](#)
- [AnalysisProperties](#)
- [DirProperties](#)
- [InvokeMenu](#)
- [CopyView](#)
- [PasteView](#)

## InsertView

### Description

Insert an Oracle Analytics Cloud 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 Analytics Cloud provider URL.

**sourcePath:** The location of the view in the Oracle Analytics Cloud Catalog.

To express the path of the view, in a web browser, access the Oracle Analytics Cloud 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 Analytics Cloud 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.

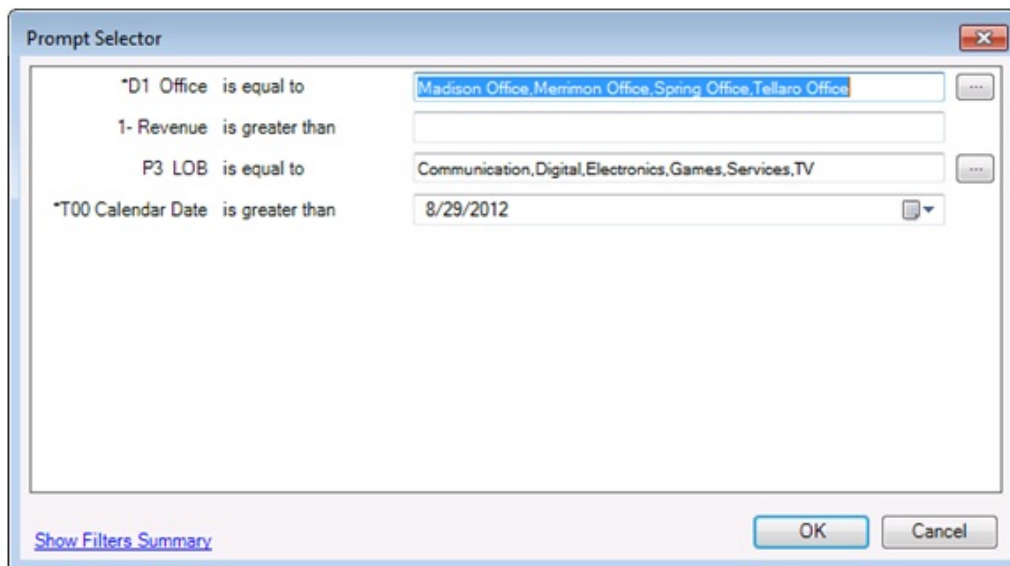
For example, to specify prompt values for the prompts in the [Figure 1](#), you must create an array of four BIReportPrompts:

- 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 17-1 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 1](#).

**Table 17-1 Render Formats and View Types**

Render Format Value	View Types to be Used
Default_Format	All Views
ExcelPivot	Pivot Table View Only
ExcelTable	Table View Only
Image	Chart View Only

**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

obiee.InsertView "http://xxx.com:xxxx/analytics/jbips", "/shared/SmartView/OBIEE/sv_vba_dev", "tableView!1", prompts, Default_Format, NewSheet

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) = "Tellarro 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_vba_dev/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) = "Tellarro 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

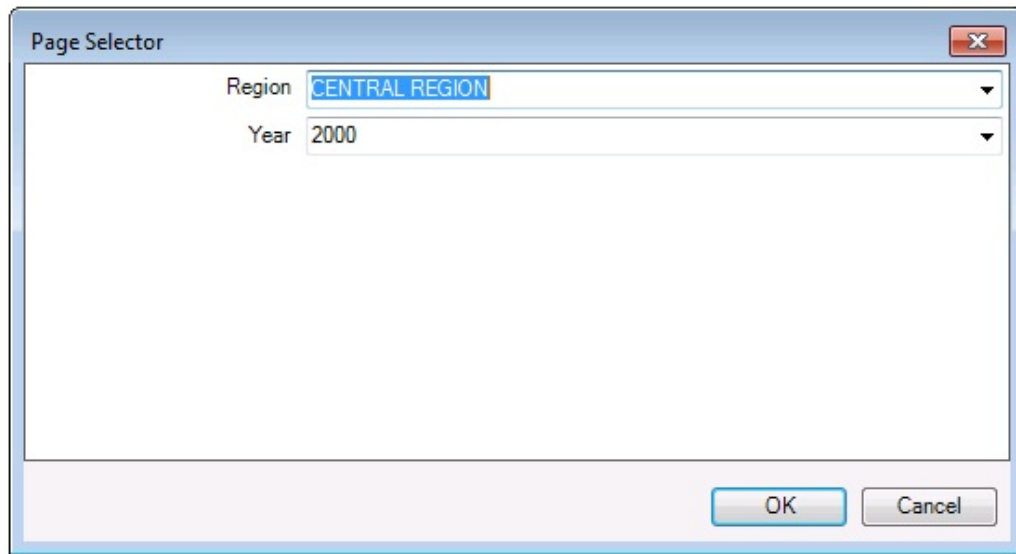
```
Function EditPagePrompts(
objID As String,
pageSelections() As String
) As Boolean
```

### 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 1](#), use the sample code that follows the figure.

**Figure 17-2 Page Selector Dialog Box with Selections for Region and Year**



```
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,  
PageEdges() As String,  
PageSelections() As String  
) 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
```

## Retrieve

#### Description

Refreshes Oracle Analytics Cloud content on the active sheet or workbook.

#### Syntax

```
Function Retrieve(  
refreshAll As Boolean)  
As BIRefreshStatus()
```

#### Parameters

**refreshAll:** If the value is True, the function will refresh all Oracle Analytics Cloud content in the entire workbook.

If the value is False, the function will only refresh the active document.

### Return Value

An array of BIRefreshStatus. Each element in the array represents the result of refreshing of an Oracle Analytics Cloud view:

- BIRefreshStatus viewName member contains the name of the refreshed view
- isSuccess member indicates if the refresh succeeds or fails
- errMsg member contains the error message if refresh fails

### Example

```
Sub testRefresh()

Dim obiee As IBIReport3
Set obiee = New SmartViewOBIEEAutomation

'Call Refresh
Dim result As Variant
result = obiee.Retrieve(True)

'Check for failed refresh
Dim success As Boolean
success = True
Dim i As Integer
For i = LBound(result) To UBound(result)
    Dim status As BIRefreshStatus
    status = result(i)
    If status.isSuccess = False Then
        success = False
    End If
Next

If success = True Then
    MsgBox "Succeeded"
Else
    MsgBox "Failed"
End If

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 Analytics Cloud 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 BIRReport As IBIReport  
Set BIRReport = New SmartViewOBIEEAutomation
```

```
Dim result As Variant

result = BIReport.AnalysisProperties("http://xxx.com:xxxx/analytics/
jbips", "/shared/SmartView/OBIEE", "svdevusr")

End Sub
```

## DirProperties

### Description

Fetch properties of a directory

### Syntax

```
Function DirProperties (
connectionContext As String,
sourcePath As String,
) As SVReportProperty()
```

### Parameters

**connectionContext:** The Oracle Analytics Cloud 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/sv_vba_dev")

End Sub
```

## InvokeMenu

### Description

Invoke Smart View Oracle BI EE extension menu ribbon.

### Syntax

```
Sub InvokeMenu(  
menuID As String  
)
```

### Parameters

**menuID:** The ID of the menu items. Valid values are listed in [Table 1](#).

**Table 17-2 Oracle BI EE Menu Items and IDs**

Menu	ID
View Designer	ViewDesigner
Publish View	PublishView
Refresh	Refresh
Edit Prompts	EditPrompts
Edit Page Prompts	EditPage
Copy	CopyView
Paste	PasteView
Delete	DeleteView
Mask Data	MaskView
Mask Document Data	MaskDocumentView

### Example

```
Sub TestMenuInvoke()  
  
    Dim obiee As IBIRreport  
    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.



# Oracle BI EE Functions

## Related Topics

- [About Oracle BI EE Functions](#)
- [Preparing to Work with Oracle BI EE Functions](#)  
Before you begin creating and editing VBA functions for Oracle Business Intelligence Enterprise Edition, you must first add references to the Oracle Smart View BI Extension type library and Oracle Smart View for Office type library.
- [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.
- [Oracle Smart View BI Extension Functions](#)

## About Oracle BI EE Functions

VBA functions support Oracle Smart View for Office operations when connected to an Oracle Business Intelligence Enterprise Edition data source.

## Preparing to Work with Oracle BI EE Functions

Before you begin creating and editing VBA functions for Oracle Business Intelligence Enterprise Edition, you must first add references to the Oracle Smart View BI Extension type library and Oracle 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](#).

# 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 Business Intelligence Enterprise Edition 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:

```
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](#) for a complete listing of the functions available and their usage.

## Oracle Smart View BI Extension Functions

### Related Topics

- [InsertView](#)
- [EditPrompts](#)
- [EditPagePrompts](#)
- [GetPagePrompts](#)
- [Retrieve](#)
- [DeleteView](#)
- [AnalysisProperties](#)
- [DirProperties](#)
- [InvokeMenu](#)
- [CopyView](#)
- [PasteView](#)

## InsertView

### Description

Insert an Oracle Business Intelligence Enterprise Edition 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.

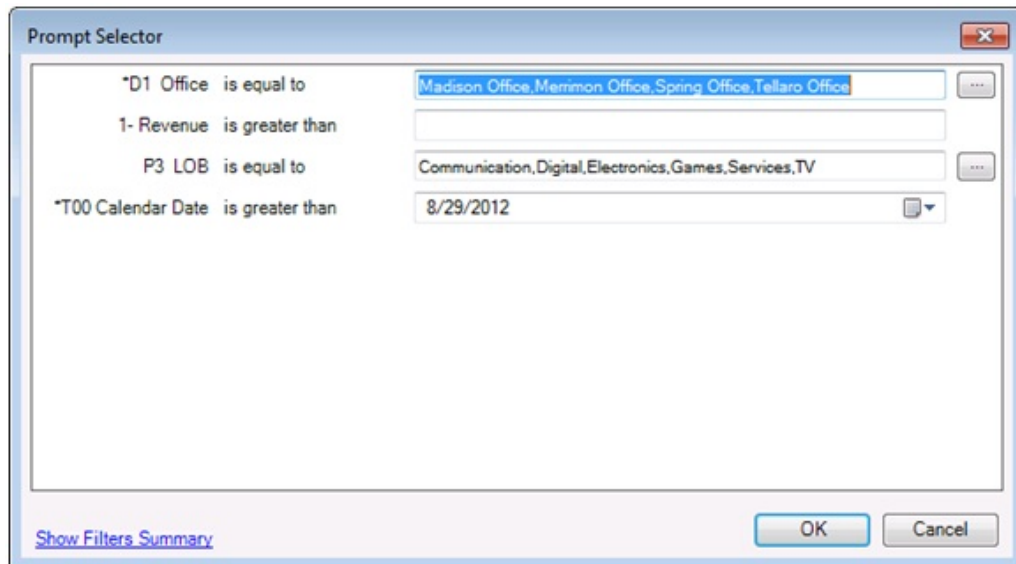
For example, to specify prompt values for the prompts in the [Figure 1](#), you must create an array of four BIReportPrompts:

- 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 18-1 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 ForthPrompt(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 1](#).

**Table 18-1 Render Formats and View Types**

Render Format Value	View Types to be Used
Default_Format	All Views
ExcelPivot	Pivot Table View Only
ExcelTable	Table View Only
Image	Chart View Only

**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

obiee.InsertView "http://xxx.com:xxxx/analytics/jbips", "/shared/SmartView/OBIEE/sv_vba_dev", "tableView!1", prompts, Default_Format, NewSheet

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) = "Tellarro 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_vba_dev/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) = "Tellarro 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

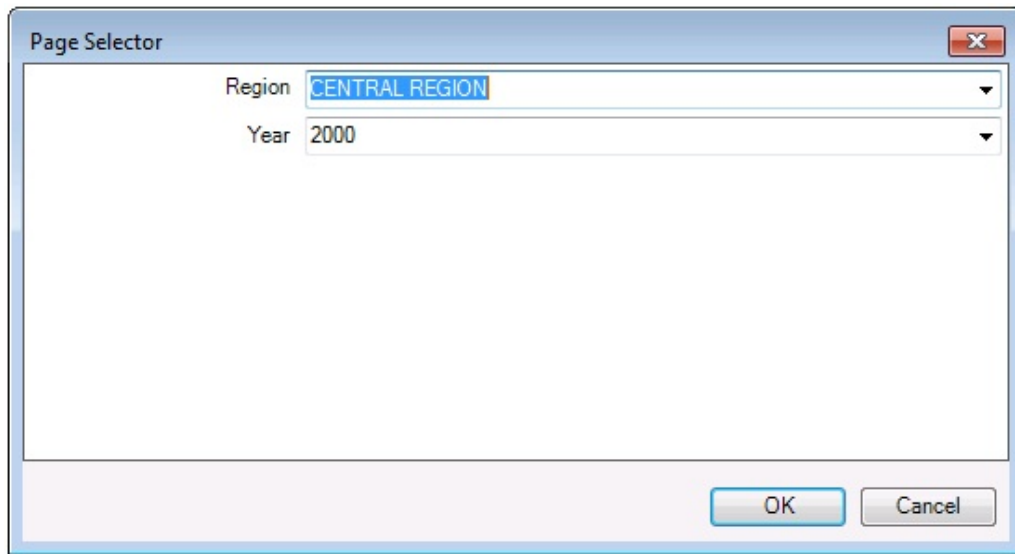
```
Function EditPagePrompts(  
objID As String,  
pageSelections() As String  
) As Boolean
```

### 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 1](#), use the sample code that follows the figure.

**Figure 18-2 Page Selector Dialog Box with Selections for Region and Year**



```
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,  
PageEdges() As String,  
PageSelections() As String  
) 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
```

## Retrieve

#### Description

Refreshes Oracle Business Intelligence Enterprise Edition content on the active sheet or workbook.

#### Syntax

```
Function Retrieve(  
refreshAll As Boolean)  
As BIRefreshStatus()
```

#### Parameters

**refreshAll:** If the value is True, the function will refresh all Oracle BI EE content in the entire workbook.

If the value is False, the function will only refresh the active document.

### Return Value

An array of `BIRefreshStatus`. Each element in the array represents the result of refreshing of an Oracle BI EE view:

- `BIRefreshStatus` `viewName` member contains the name of the refreshed view
- `isSuccess` member indicates if the refresh succeeds or fails
- `errMsg` member contains the error message if refresh fails

### Example

```
Sub testRefresh()  
  
Dim obiee As IBIReport3  
Set obiee = New SmartViewOBIEEAutomation  
  
'Call Refresh  
Dim result As Variant  
result = obiee.Retrieve(True)  
  
'Check for failed refresh  
Dim success As Boolean  
success = True  
Dim i As Integer  
For i = LBound(result) To UBound(result)  
    Dim status As BIRefreshStatus  
    status = result(i)  
    If status.isSuccess = False Then  
        success = False  
    End If  
Next  
  
If success = True Then  
    MsgBox "Succeeded"  
Else  
    MsgBox "Failed"  
End If  
  
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 Business Intelligence Enterprise Edition 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 BIRReport As IBIReport
```

```
Set BIReport = New SmartViewOBIEEAutomation

Dim result As Variant

result = BIReport.AnalysisProperties("http://xxx.com:xxxx/analytics/
jbips", "/shared/SmartView/OBIEE", "svdevusr")

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 IBIRReport
Set BIReport = New SmartViewOBIEEAutomation

Dim result As Variant

result = BIReport.DirProperties("http://xxx.com:xxxx/analytics/jbips", "/
shared/SmartView/OBIEE/sv_vba_dev")

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 1](#).

**Table 18-2 Oracle BI EE Menu Items and IDs**

Menu	ID
View Designer	ViewDesigner
Publish View	PublishView
Refresh	Refresh
Edit Prompts	EditPrompts
Edit Page Prompts	EditPage
Copy	CopyView
Paste	PasteView
Delete	DeleteView
Mask Data	MaskView
Mask Document Data	MaskDocumentView

### 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.