insightsoftware

Add-in for Office Guide

Longview

Version 26

Document Information

Notices

Copyright

The information contained in this document represents the current view of insightsoftware.com on the issues discussed as of the date of publication. This document is for informational purposes only. insightsoftware.com makes no representation, guarantee or warranty, expressed or implied, that the content of this document is accurate, complete or up to date.

Disclaimer

This guide is designed to help you to use the Longview applications effectively and efficiently. All data shown in graphics are provided as examples only. The example companies and calculations herein are fictitious. No association with any real company or organization is intended or should be inferred.

Contents

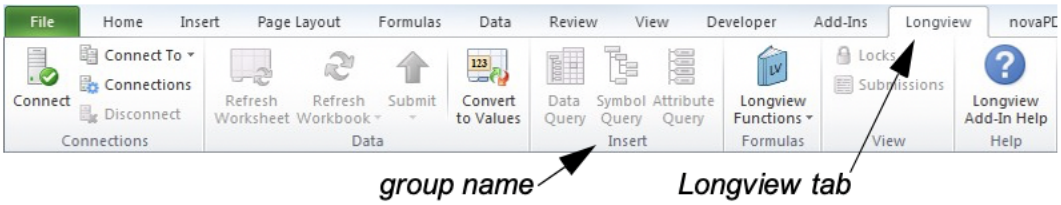
Longview Add-In For Office Overview

The Longview Add-In for Office is a fully integrated Add-In that allows you to make use of the familiar interface of Microsoft Excel in conjunction with the database used by your Longview system. You can use the Longview Add-In for Office to access Longview data servers to query data and other aspects of the data server repository and submit data to the data server repository in worksheets and workbooks within Microsoft Excel. The Longview Add-In for Office can also connect to a Tidemark Data Source. For more information on this topic, see Connecting to a Tidemark Data Source.

If you have already installed the Longview Add-In for Office, you can also open any Excel file and connect to the data server using the method described in Connecting to the data server.

Accessing the Longview Add-In For Office

After you have installed the Longview Add-In for Office, the Longview tab appears in the ribbon in Microsoft Excel.



Most Longview Add-In for Office features are unavailable until you connect to a data server. However, if you are not connected to a Longview data server, you can still view the Longview Add-In for Office Help. For more information, see Viewing Longview Add-In for Office Help.

Most of the Longview Add-In for Office functionality is available from the ribbon, which is divided into the following groups:

- **Connections**The Connections group contains all functionality related to connecting to and disconnecting from the data server and managing existing connections.
- **Data**The Data group contains functionality related to data in the Microsoft Excel workbook. For example, you can refresh, lock, or submit data in the active worksheet or workbook, or convert data.
- **Insert**The Insert group contains functionality related to querying data or information from the Longview data server. Click one of the buttons in the Insert group to insert a query into your Microsoft Excel workbook.
- **Formulas**The Formulas group contains functionality related to working with Longview Functions using the Function Arguments dialog. Longview Functions allow you to query different types of data into worksheet cells and submit data from worksheet cells to the Longview data server repository.
- **View**The View group contains links to the Data Locks and User Submissions tools:
  - Locks — Opens the Data Locks tool, to view and delete any locks that are applied to the data server repository.
  - Submissions — Opens the User Submissions tool, to view the status of batches that have been created in the system.

  For more information on the Data Locks or User Submissions tools, see the *Longview Data Locks Help* or *Longview User Submissions Help*.
- **Help**The Help group contains a link to the Longview Add-In for Office Help.

Using Longview Add-In With A Tidemark Data Source

The Longview Add-In for Office can also connect to a Tidemark Data Source. Connecting to a Tidemark Data Source is equivalent to connecting to a Tidemark Application. Once connected, data security, hierarchy and member selections will be restricted to those available in the Tidemark Application to which you are connected. Some, but not all the functionality is available for Tidemark Data Sources. Any functionality unavailable for Tidemark Data Sources will appears disabled in Longview menu options within Microsoft Excel.

This guide will describes how to use these features. However, note that there may be some slight differences in terminology. Most notably, Longview Data Sources use the term "symbol" and Tidemark Data Sources use the term "member".

The main features available for Tidemark Data Sources include retrieving data, refreshing data, and retrieving metadata.

The following lists the features applicable to Tidemark Data Sources:

- Data Query
- Data Extract
- Symbol Query
- Refresh Worksheet
- Refresh Workbook
- Convert to Values
- Longview Functions: LVCELL, LVDESC, LVCHILD, LVDIMDESC, LVDIMNAME, LVIDENTIFIER, LVPARENT, LVREFRESHDATETIME, LVSUBMIT, LVSYMBOLSELECTOR, LVUSER, LVUSERDESC

  Caution: The following caveats apply when connecting to a Tidemark Data Source:

- Longview Add-In for Office is only supported on Windows operating systems.
- The link to download the Longview Add-In for Office is available from the Tidemark main menu. Once the Add-In is installed, you may subsequently launch the Longview Add-In for Office from the Tidemark main menu or by launching Microsoft Excel from your desktop and navigating to the LONGVIEW tab.
- Longview Add-In for Office queries are made directly against Tidemark data. Data security is enforced through a Tidemark data security profile that customers must configure in each Tidemark application they want to make available to users from Longview Add-In for Office. The security profile must be named ExcelProfile. Users cannot connect to Tidemark applications unless the ExcelProfile is present and configured properly.
- Single Sign-On is not supported for Tidemark Data Sources.

- Longview Add-In for Office can handle a maximum of 64 dimensions when connecting to a Tidemark Data Source. If your Tidemark Data Source has more than 64 dimensions, then you will be unable to use the Longview Add-In for Office to connect to it.
- When retrieving data using Data Query, Data Extract, or LVCELL() and LVSUBMIT(); you must specify a co-ordinate for each dimension
- Some messages, tooltips and user interfaces may be Longview-centric. For example, in some cases, you may see the term "symbol" instead of "member". You may think of these two terms as equivalents.
- Automation APIs are not supported for Tidemark Data Sources.

Longview Add-In For Office Version And Session Information

It may be useful at times to view version and session information about the Longview Add-In for Office.

> **Note:** If you are not connected to a data server, only the version information for Longview Add-In for Office displays. (Not connected) displays in all other fields.

To view Longview Add-In for Office information:

1. Click the **File** tab. The Backstage view displays.
2. Click **Longview**.
3. Review version information in the About Longview Add-In for Office section for the following:
   - Longview Add-In for Office
   - Longview Data Server

   > **Note:** To copy version information to your Windows clipboard, click **Copy Info to Clipboard**.
4. Review session information in the Session Information section for the following:
   a. Username: Your current username and description.
   b. Group: Your current user group name and description.
   c. Name: The name of your current connection.
   d. Identifier: The Longview Identifier (LID) of the system you are currently connected to.
   e. Host: The host name of the data server you are currently connected to.
   f. Port: The port number of the data server you are currently connected to.

Managing Connections

Before you can start working with the Longview Add-In for Office, you must use the Manage Connections dialog to add a connection to the Longview data server.

For more information, see the following:

- Understanding authentication methods
- Adding a connection

In addition, you can use the Manage Connections dialog to complete several tasks involving data server connections, including the following:

- Modifying connections
- Deleting connections

After you have added a connection, you can complete the following tasks from the Connections group:

- Connecting to the data server
- Changing an expired password
- Upgrading the Longview Add-In for Office
- Disconnecting from the data server

Understanding authentication methods

Before you can start working with the Longview Add-In for Office, you must know the authentication method required by your company.

Currently, the Longview Add-In for Office supports the following authentication methods:

Authentication Method	Description
Windows authentication	Windows authentication is currently supported for all components of the Longview system. If your Longview system uses Windows authentication, you are authorized to access the Longview system as soon as you sign on to Windows. You do not have to supply a password specific to the Longview system.
Longview authentication	If your company uses Longview authentication, you will be required to enter a specific user name and password to access the Longview system.
Single Sign-On (Web)	Single Sign-On (Web) allows your company to use a standard authentication across multiple applications. This allows you to use a single password for both the Longview system and other applications.

Adding a connection

You can use the Manage Connections dialog to add a connection to the Longview data server. The first time you access the Longview Add-In for Office, you may have to add a connection, depending on how the Add-In was installed:

- If you installed the Longview Add-In for Office from a Longview Dashboard link, a connection is automatically created for you with the connection settings you used to log in to Longview Dashboard.
- If you installed Longview Add-In for Office as a stand-alone installation, you must add a connection before you can sign on.

You can add up to 12 connections.

To add a connection:

1. Click the Longview tab.
2. In the Connections group, click Connections. The Manage Connections dialog opens.

   Note: Connections is available only if you are disconnected from the data server. For more information, see Disconnecting from the data server.
3. Click Add. The Add Connection page appears.
4. Complete the following fields:

Field	Description
Name	Enter a name for the connection. Each connection name must be unique.
Category	Select a color to represent the connection from the drop-down list. Longview recommends that you select a different color for each connection so that they can be easily identified.
Identifier	Enter the Longview Identifier (LID) for the system you want to connect to. **Note:** If the Longview Identifier policy is enforced (as set by the LV_IDENTIFIER_POLICY parameter), the Longview Identifier set for Identifier must match the data server Longview Identifier. For more information, see the *Longview Installation Guide*.
Host	Enter the host name of the data server you want to connect to.
Port	This field is unavailable if you specified a URL (starting with http) for an HTTP Proxy Server for the Host field. For more information on HTTP Proxy servers, see the *Longview Installation Guide*. Enter the port number of the data server you want to connect to.

Field	Description
Web Bridge	Enter the web bridge URL, such as http://ca127docsql01:80/cgi-bin/LongviewCPM/lvwebsso.cgi or http://ca127docsql01:80/cgi-bin/LongviewCPM/ks_iwebsso.dll. This field cannot be longer than 256 characters.

5. Click Add. The Manage Connections dialog comes into view, and the connection appears in the Connections list.

Modifying connections

You can use the Manage Connections dialog to modify existing data server connections.

To modify connections:

1. Click the Longview tab.
2. In the Connections group, click Connections. The Manage Connections dialog opens.

   Note: Manage Connections is available only if you are disconnected from the data server. For more information, see Disconnecting from the data server.

3. Select the connection that you want to modify from the Connections list.
4. Click Edit. The Edit Connection page appears.
5. Modify the connection settings as necessary.
6. When you are finished, click Save. The Manage Connections dialog comes into view.

Deleting connections

You can use the Manage Connections dialog to delete existing data server connections.

To delete connections:

1. Click the Longview tab.
2. In the Connections group, click Connections. The Manage Connections dialog opens.

   Note: Manage Connections is available only if you are disconnected from the data server. For more information, see Disconnecting from the data server.

3. Select the connection that you want to delete from the Connections list.
4. Click Delete.

Connecting to the data server

To use the Longview Add-In for Office to work with system data, you must first connect to a Longview data server.

> **Note:** If you opened Excel by clicking a link to a Longview Add-In for Office file on the Longview Dashboard or in the Tax Provision navigation pane, you will automatically be connected to the data server using the same credentials as the ones used to sign on to the Longview Dashboard.

To connect to the data server:

1. Click the Longview tab.
2. In the Connections group, click Connect. The Longview Add-In for Office dialog opens.

   > **Note:** If this is your first time connecting to a data server from the Longview Add-In for Office, and you installed the Add-In as a stand-alone installation, you must add a connection before you can sign on. For more information, see Adding a connection.

3. Your current Connection displays in the Connection section. If you want to use a different connection, select it from the Connection drop-down list.
4. In the Authentication section, select the authentication method used by your organization. For more information, see Understanding authentication methods.
5. Do one of the following:
   - If you selected Windows authentication in step 4, the Username and Password fields are unavailable. Click Sign On to connect to the data server.
   - If you selected Single Sign-On (Web) in step 4, the Username and Password fields are unavailable. Click Sign On to connect to the data server.
   - If you selected Longview authentication in step 4, complete the following fields:

Field	Description
Username	Enter your username.
Password	Enter your password.

6. Select the Remember Me checkbox if you wish to save your username with the connection once you have successfully connected.
7. Click Sign On. The Group page appears.

   > **Note:** If you are a member of only one user group, the Group page does not appear. You are now connected to the data server.

8. Select the appropriate user group and click OK. You are now connected to the data server.

   Note: When you are connected to a data server, the Longview Identifier of the data server displays to the right of the workbook name in the Microsoft Excel title bar.

   If a later version of the Longview Add-In for Office is detected, an upgrade dialog opens.

   For more information, see Upgrading the Longview Add-In for Office.

Changing an expired password

If you are a Longview authenticated user and your password has expired, the system prompts you to change it the first time you attempt to connect to the Add-In for Office after your password has expired.

To change your password:

1. Click the Longview tab.
2. In the Connections group, click Connect. The Longview Add-In for Office dialog opens.
3. Your current Connection displays in the Connection section. If you want to use a different connection, select it from the Connection drop-down list.
4. Complete the following fields:

Field	Description
Username	Enter your username.
Password	Enter your password.

5. Click Sign On. The Change Password dialog appears.
6. Complete the following fields:

Field	Description
Old Password	Type your expired password.
Password	Type your new password. **Note:** Passwords cannot contain any of the following characters: double quotation marks ("), pipes (|), dollar sign ($), square brackets ([ ),( ]) or spaces ( ), and cannot start with the at sign (@).
Confirm Password	Retype your new password.

7. Click **OK**. Your password is updated, and you are connected to the data server.

Upgrading the Longview Add-In for Office

If a later version of the Longview Add-In for Office is detected when you connect to a Longview data server, you can upgrade the Longview Add-In for Office.

To upgrade the Longview Add-In for Office:

1. Connect to a data server (as described in Connecting to the data server). If a later version of the Longview Add-In for Office is detected, an upgrade dialog opens.
2. Click Upgrade.
3. Depending on your security settings, the Application Run - Security Warning dialog may appear. Click Run.
4. A Downloading LongviewAddIn.Office dialog opens, informing you of the status of the download. Once the download is complete, an installation dialog opens.
5. Close Microsoft Excel, and then click Continue Installation. Once the installation is complete, Excel launches.

Disconnecting from the data server

When you are finished working with the Longview Add-In for Office, you can disconnect from the Longview data server.

To disconnect from the data server:

1. Click the Longview tab.
2. In the Connections group, click Disconnect.

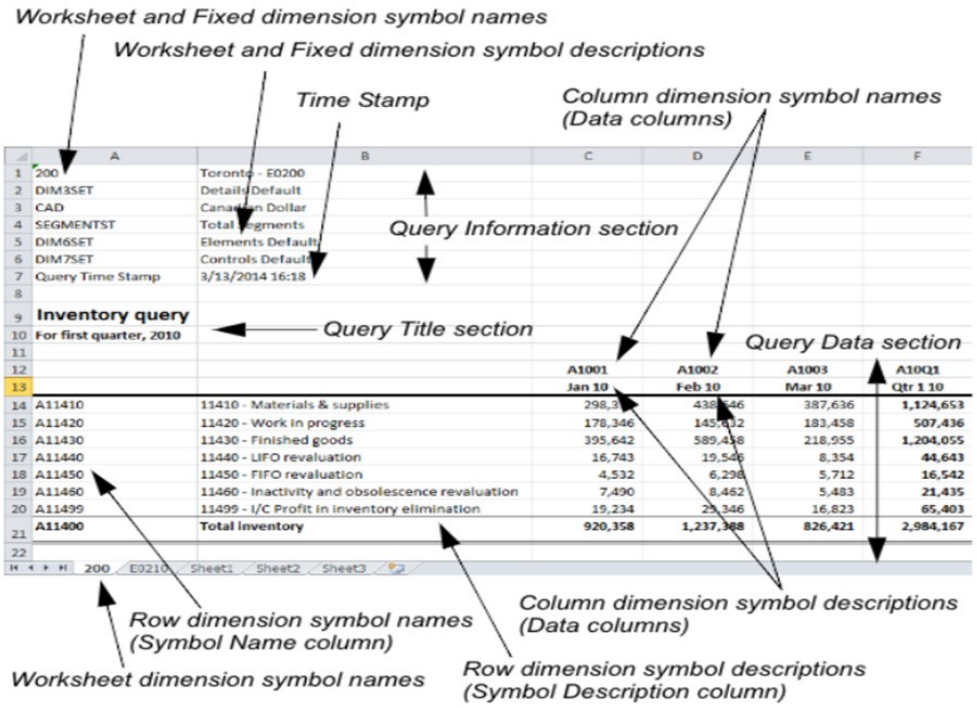
Querying Data

Once you are connected to the Longview data server, you are ready to work with the data contained within it. This chapter explains how to use the Longview Add-In for Office to query data.

For information on analyzing data queries, see Analyzing data query results.

Understanding data query output

You can use the Longview Add-In for Office to query data from the data server repository and place it in a Microsoft Excel workbook. The results of a data query are divided into several main sections (displayed depending on the Format Options specified for the data query) and appear in the following layout:



Data query results in the Longview Add-In for Office are divided into the following main sections:

- The Query Information section
- The Query Title section
- The Query Data section
- The worksheet dimension symbol names (worksheet tabs)

Each section is separated from the others by a blank row. If none of the elements for the Query Information and Query Title sections display in the data query results, the Query Data section appears at the top left cell of the query results.

Note: If you manually modify any character string or data value in the data query results, or if you insert or delete a row or column within the range of the query results, you will not be able to rerun, reorient, analyze, or drill into that data query.

The Query Information section includes the following elements:

Element	Description
Worksheet and Fixed dimension symbol names and Worksheet and Fixed dimension symbol descriptions	Fixed dimension symbols – The symbols as specified for the fixed dimensions (all dimensions other than the Row, Column, and Worksheet dimensions). Worksheet dimension symbol – The Worksheet dimension symbol for the active worksheet.
Time Stamp	The value for the time stamp depends on whether you selected one of the Values options or one of the Formulas options for the As field when you ran the data query. Values – The time stamp appears as Query Time Stamp and is the date and time that the data query was run, in the date and time format for your region (as set in your system settings). The Query Time Stamp is a hard-coded string value. Formulas – The time stamp appears as Refresh Time Stamp and is the date and time when the data query was last refreshed or run if it has not been refreshed. The Refresh Time Stamp is a LVREFRESHDATETIME function, with the date and time in the short format, as specified in your system. For more information, see LVREFRESHDATETIME.

The Query Title section includes the following elements:

Element	Description
Query title	The query title, as specified in the Format Options page.
Query subtitle	The query subtitle, as specified in the Format Options page.

The Query Data section includes the following elements:

Element	Description
Row dimension symbol names and Row dimension symbol descriptions	The symbols as specified for the Row dimension.
Column dimension symbol names and Column dimension symbol descriptions	The symbols as specified for the Column dimension.

If duplicate symbols are specified in the Worksheet dimension (for example, when a data query is rerun), a new worksheet is created for each duplicate symbol. In each new worksheet, the symbol name is appended with (N), where N is a unique number corresponding to the number of duplicates created.

For example, if two new worksheets are created (for two duplicate symbols), the symbol names in the worksheet tabs appear as follows:

- SymName
- SymName (2)
- SymName (3)

Querying base data

You can use the Longview Add-In for Office to query data from the data server repository and place it in a Microsoft Excel workbook.

When you run a data query, you can specify whether to output Values or Longview Formulas in the data query results. You should select an output type depending on how you plan to use the data query results:

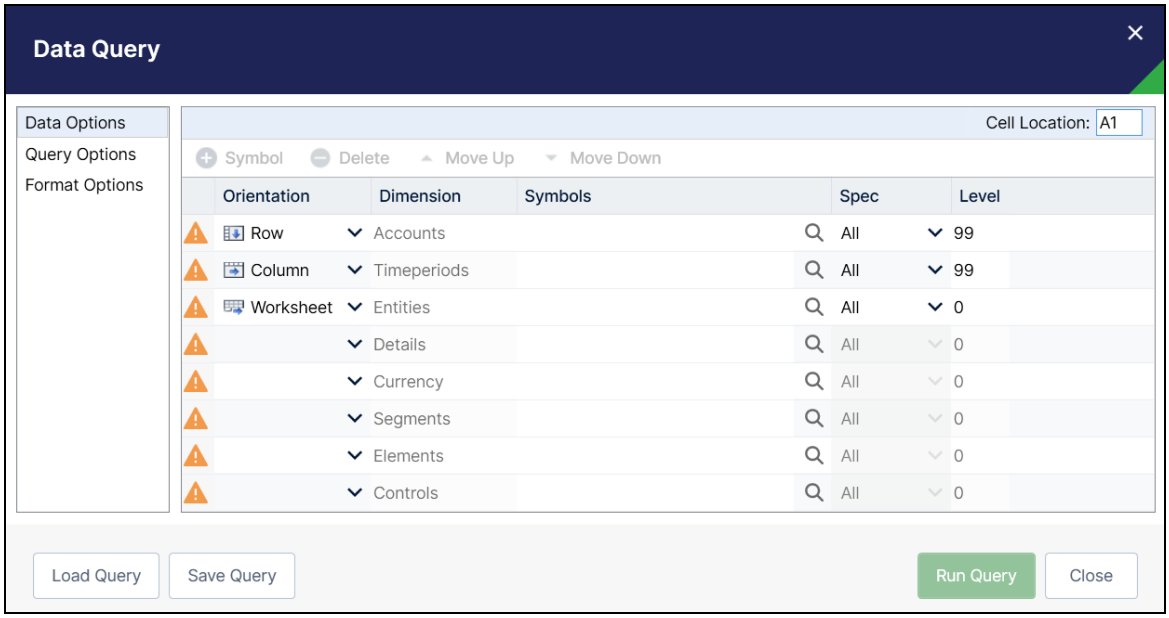
- Select one of the Values options for the As field if you do not plan to customize the data query results.
- Select one of the Formulas options for the As field if you want to customize the data query results (for example, by adding commentary, or an additional column, or by inserting Microsoft Excel functions) and you want to maintain your customizations when you refresh the data values in the query results. To refresh data values in customized data query results, use the Refresh Worksheet or Refresh Workbook buttons in the ribbon. For more information, see Refreshing data queries.

  Note: Some changes to data query results will prevent you from rerunning the data query or reorienting and drilling into the query results. For more information, see Formatting query results.

To query data:

1. Click the Longview tab.
2. In the Insert group, click Data Query. The Data Query dialog opens with the Data Options page in view.

   Note: You must be connected to the data server to run a data query.



3. Complete the following field:

Field	Description
Cell Location	Type the location of the top left cell of the query results. Keep in mind that the query results will overwrite any data in the worksheet cells. If you must place multiple queries in a single worksheet, Longview recommends that you ensure that the query results do not overlap.

4. Specify the active dimensions for your query:

Dimension	Description
Row	Data for the specified symbol or symbols in the Row dimension displays in rows on the left side of the worksheet.
Column	Data for the specified symbol or symbols in the Column dimension displays in columns across the top of the worksheet.
Worksheet	Data for the specified symbol or symbols in the Worksheet dimension displays in separate worksheets, one for each symbol.

5. For each dimension in your system, specify the following symbol specifications:

> **Note:** Certain combinations of Symbols, Spec, and Level are restricted because they will not return a valid set of results. If you attempt to use one of these combinations, either the combination will be restricted (for example, if you specify Leaf for Spec, the Level field is unavailable), or you will receive an error message.

Specification	Description
Symbols	Type a symbol name or click the symbol selector button to search the hierarchy for the selected dimension. For more information, see Using the Symbol Selector. If you specify more than one Root symbol for the Row, Column, or Worksheet dimensions, you cannot change the orientation of that dimension to a fixed dimension unless you delete the additional symbols. **Note:** Default values for the data query symbols for each dimension are set by the UGPDDimNQueryDefault user attributes, where DimN is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For more information on attributes, see the *Longview Application Administrator Guide*.
Spec	This field is unavailable for symbols that are not in the Row, Column, or Worksheet dimensions. Specify the type of symbol to query, using one of the following options from the drop-down list: ■ All — To query all symbol types. ■ Leaf — To query only leaf symbols. ■ Parent — To query only parent symbols. ■ Root and Parent — To query only root and parent symbols. The default value for this field is All.
Level	This field is unavailable for symbols that are not in the Row, Column, or Worksheet dimensions. Type the number of levels of descendants of the selected symbol to appear in the results. You can select up to 99 levels of symbol detail. If you type 0, only the symbol you selected will appear. To include the selected symbol and one level of symbol detail below it, type 1. If you select 99 levels of symbol detail, all levels of symbol detail are included in the results.

Note: Large queries take longer to complete than smaller queries. Be as specific as possible with the data intersection you are querying.

6. For the dimensions you specified as the Row, Column, and Worksheet dimensions in step 4, you can specify additional symbols to query. To add additional symbols to query for a dimension, complete the following steps:
   a. Select the applicable dimension row.
   b. Click Symbol. A new row appears.

c. For the new symbol, specify the symbol specifications, as described in step 5.

d. Repeat step a to step c for each additional symbol that you want to include in your data query.

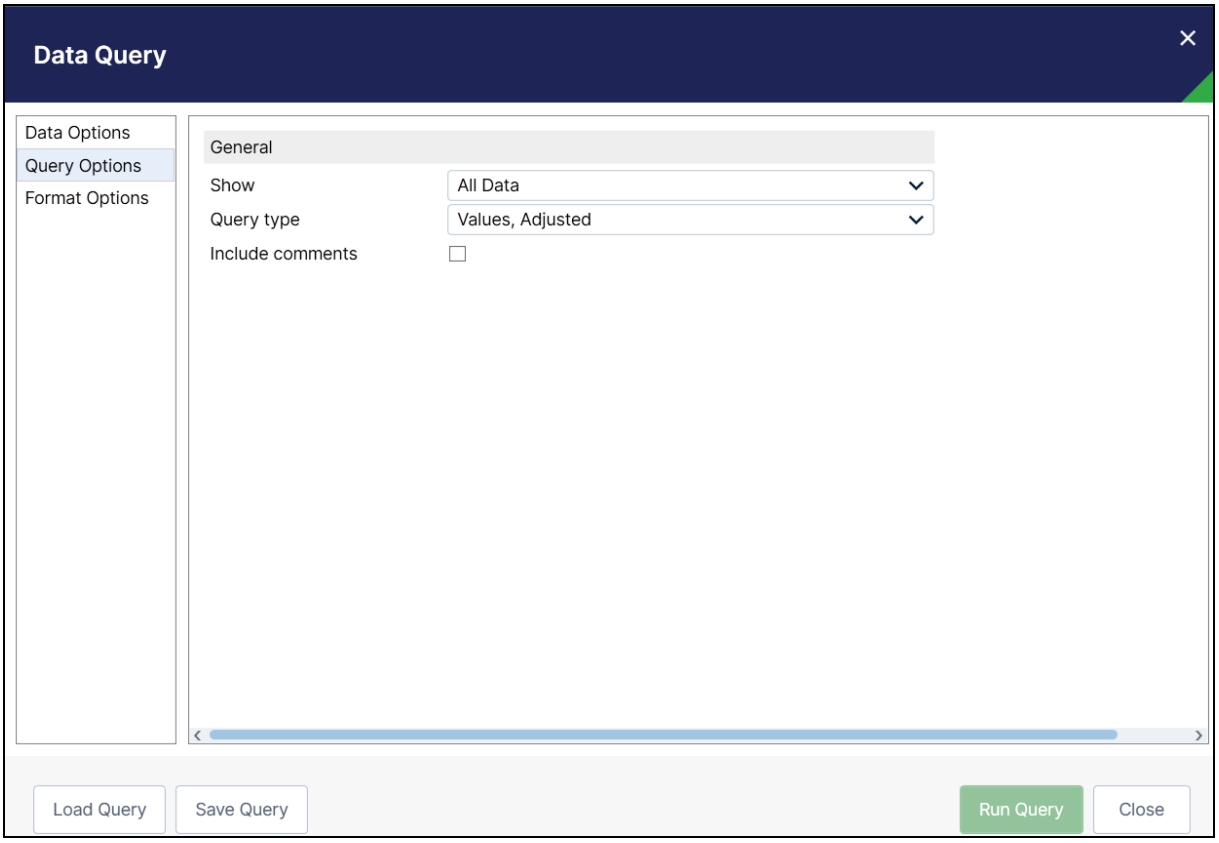
> **Note:** To delete additional symbols that you have added, select the row containing the symbol and click Delete. You can also change the order in which the symbols will appear in the query results by selecting the row containing the symbol and clicking Move Up or Move Down.

7. Do one of the following:

- To specify the format options for your data query, proceed to Specifying format options for data queries.
- To run the data query with the default format options, proceed to Running a data query.

> **Note:** You may also quickly populate the Data Query dialog with query selections that you have previously saved by using the Load Query button. For more information, see Saving and loading data query selections.

Specifying Query options for Data queries



Complete the following fields:

Field	Description
Show	Specify the type of hierarchy data to display in the data query results using one of the following options from the drop-down list: ■ All Data — To display all types of data. ■ Leaf Data — To display data for leaf cells only. ■ CTA Data — To display Cumulative Translation Adjustment (CTA) data only. This option displays data from CTA symbols containing data on the net gain or loss resulting from the data translation of another currency. **Note:** If you specify Leaf Data or CTA Data, you must specify either Values, Adjusted or Values, Unadjusted for As. The default value for this field is All Data.

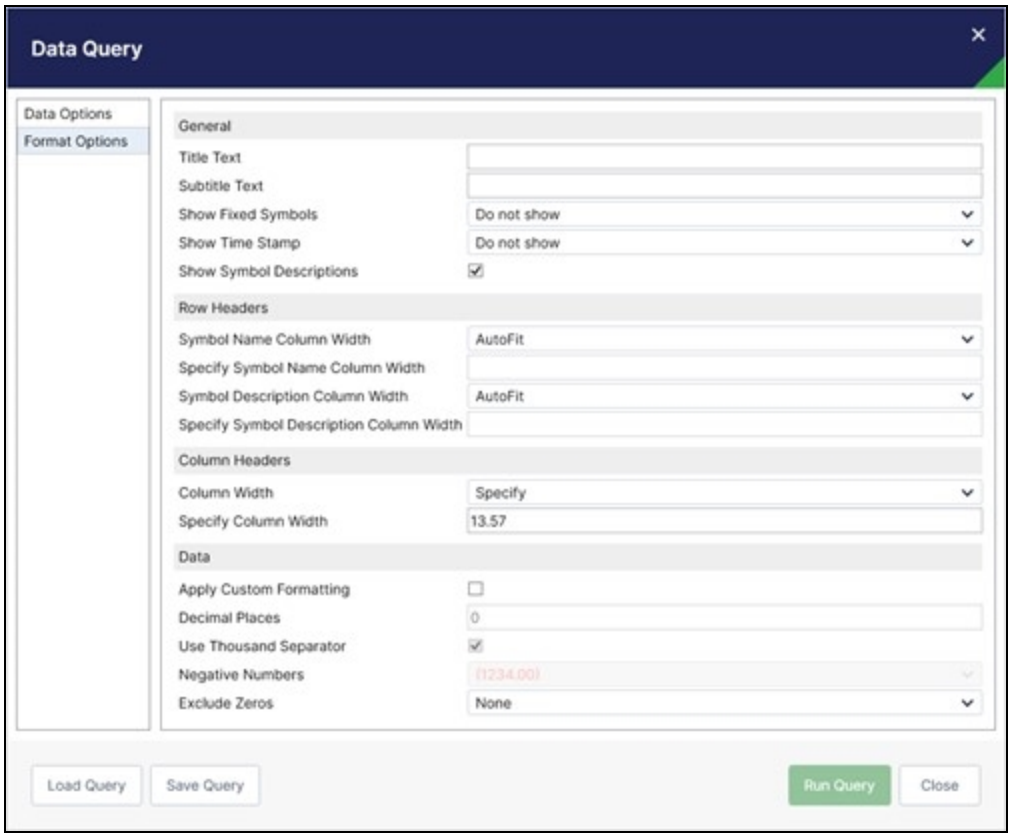
Field	Description
Query Type	Specify whether Longview Formulas are output and whether data that has been adjusted by journal entries displays in the data query results, using one of the following options from the drop-down list: ■ Values, Adjusted — To output data values only, and display data that has been adjusted by journal entries. The time stamp for the query is output as a hard-coded string value. ■ Values, Unadjusted – To output data values only, and display data that has not been adjusted by journal entries. The time stamp for the query is output as a hard-coded string value. ■ Formulas, Adjusted, with Cell References – To create an LVCELL function for each data intersection in the query results, an LVDESC function for each symbol description in the Row, Column, and Fixed dimensions, and an LVREFRESHDATETIME function for the time stamp. Symbol names in each of the LVCELL and LVDESC functions are references to the cells that contain the required values for the data intersection. Note that the cell references also include absolute cell references (for example, A1), where the address of the cell remains the same, regardless of the position of the cell that contains the function. For more information, see LVCELL, LVDESC, and LVREFRESHDATETIME. Data that has been adjusted by journal entries displays in the data query results. **Note:** If As is set to Formulas, Adjusted, with Cell References, and Show Fixed Symbols is set to Do not show in the Format Options page, cell references are output for symbols in the Row and Column dimensions only. All other symbol names in each of the LVCELL and LVDESC functions are hard coded as the required values for the data intersection. For more information on Format Options, see “Specifying format options for data queries”. ■ Formulas, Adjusted, without Cell References -To create an LVCELL function for each data intersection in the query results, an LVDESC function for each symbol description in the Row, Column, and Fixed dimensions, and an LVREFRESHDATETIME function for the time stamp. Symbol names in each of the LVCELL and LVDESC functions are hard coded as the required values for the data intersection. For more information, see LVCELL, LVDESC, and LVREFRESHDATETIME. Data that has been adjusted by journal entries displays in the data query results. The default value for this field is Values, Adjusted.
Include comments	Select this field to include comments in the data query results. Comments retrieved from the database will be displayed as Excel cell comments on the corresponding cells.

Specifying format options for data queries

Before you run a data query, you can specify how the results of the query are formatted in the Microsoft Excel worksheet. For more information on the layout of the query results, see Understanding data query output.

To specify format options:

1. Click Format Options in the left pane of the Data Query dialog. The Format Options page opens.



2. In the General section, complete the following fields:

Field	Description
Title Text	Specify the text for the data query title. If you leave this field blank, no title row is included in the data query results.
Subtitle Text	Specify the text for the data query subtitle. If you leave this field blank, no subtitle row is included in the data query results.
Show Fixed Symbols	Specify whether the Fixed dimension symbols display in the data query results, using one of the following options from the drop-down list: **Note:** The functionality for this option includes the Worksheet dimension symbol that is included in the Query Information section. The Worksheet dimension symbols in the worksheet tabs are unaffected. Do not show – Fixed dimension symbols do not display in the data query results. **Note:** If Show Fixed Symbols is set to Do not show, and the data query is set up to output Longview Formulas with cell references, cell references are output for symbols in the Row and Column dimensions only. All other symbol names in each of the LVCELL and LVDESC functions are hard coded as the required values for the data intersection. Top – Fixed dimension symbols display in the Query Information section above the data query results. The default value for this field is Do not show.
Show Time Stamp	Specify whether the time stamp displays in the data query results, using one of the following options from the drop-down list: ■ Do not show — The time stamp does not display in the data query results. ■ Top — The time stamp displays in the Query Information section above the data query results. The default value for this field is Do not show.
Show Symbol Descriptions	Select this field to show symbol descriptions for symbols in all dimensions in the data query results. Clear this field to show only symbol names for symbols in all dimensions in the data query results. This field is selected by default.

3. In the Row Headers section, complete the following fields:

Field	Description
Symbol Name Column Width	Specify how the width of the Symbol Name column (the column containing the Row dimension symbol names) is determined, using one of the following options from the drop-down list: ■ AutoFit — To automatically fit the width for the Symbol Name column to the longest symbol name. This does not include the Worksheet and Fixed dimension symbol names. ■ Specify — To specify a custom width for the Symbol Name column. The default value for this field is AutoFit.
Specify Symbol Name Column Width	This field is available only if Symbol Name Column Width is set to Specify. Type a number for the width of the Symbol Name column in characters. Type a number from 0 through 255, including decimals. The default value for this field is 22.71.
Symbol Description Column Width	Specify how the width of the Symbol Description column (the column containing the Row dimension symbol descriptions) is determined, using one of the following options from the drop-down list: ■ AutoFit — To automatically fit the width for the Symbol Description column to the longest symbol description. This does not include the Worksheet and Fixed dimension symbol descriptions. ■ Specify — To specify a custom width for the Symbol Description column. The default value for this field is AutoFit.
Specify Symbol Description Column Width	This field is available only if Symbol Description Column Width is set to Specify. Type a number for the width of the Symbol Description column in characters. Type a number from 0 through 255, including decimals.

4. In the Column Headers section, complete the following fields:

Field	Description
Column Width	Specify how the width of the Data columns (the columns containing the Column dimension symbol names and descriptions) is determined, using one of the following options from the drop-down list: ■ AutoFit — To automatically fit the width for the Data columns to the symbol name, symbol description, or data value, whichever is longer. ■ Specify — To specify a custom width for the Data columns. The default value for this field is Specify.

Field	Description
Specify Column Width	This field is available only if Column Width is set to Specify. Type a number for the width of the Data columns in characters. Type a number from 0 through 255, including decimals. The default value for this field is 13.57.

5. In the Data section, complete the following fields:

> **Note:** Custom formatting as specified in the data section overrides the formatting applied by Longview cell styles. For more information, see Formatting query results.

Field	Description
Apply Custom Formatting	Select this field to apply custom formatting to the data values in the data query results. This field is cleared by default.
Decimal Places	This field is available only if Apply Custom Formatting is selected. Type an integer from 0 through 9 for the number of decimal places to include in the symbol data. The default value for this field is 0.
Use Thousand Separator	This field is available only if Apply Custom Formatting is selected. Select this field to use your regional separator (as set in your system settings) as the thousands separator in the symbol data. For example, if your region is set to United States and your regional separator is a comma, the symbol data appears as follows: “1,000,000”. Clear this field to display the symbol data without thousands separators; for example, “1000000”. This field is selected by default.

Field	Description
Negative Numbers	This field is available only if Apply Custom Formatting is selected. Specify how negative numbers in the symbol data display in the data query results using one of the following options from the drop-down list: ■ -1234.00 — Negative numbers are preceded by a minus sign. ■ 1234.00 — Negative numbers display in red. ■ -1234.00 — Negative numbers are preceded by a minus sign and display in red. ■ (1234.00) — Negative numbers are enclosed in parentheses. ■ (1234.00) — Negative numbers are enclosed in parentheses and display in red. The default value for this field is (1234.00)
Exclude Zeros	Specify whether zero data will be suppressed, using one of the following options from the drop-down list: ■ None — No rows or columns are suppressed. ■ Rows — Rows where all the data returned are values of “zero”. ■ Columns — Columns where all the data returned are values of “zero”. ■ Rows and Columns — Rows and Columns where all the data returned are values of “zero”. The default value for this field is None.

6. Proceed to Running a data query.

Running a data query

When you are finished specifying the data and format options for the query, you can run the data query.

To run a data query:

1. Do one of the following:

   - If you are querying data in a workbook that does not contain existing data queries, click Run Query. The results of the data query appear in Microsoft Excel, with your specified orientation, symbols, and formats. Proceed to step 3.

   - If you are querying data in a workbook that contains existing data queries, and any symbols that you specified for the Worksheet dimension are duplicated in the Worksheet dimension of the existing data queries, click Run Query, and proceed to the next step.

2. In the dialog that appears, do one of the following:
   - Click Create to create a new worksheet or worksheets for the data query results.
   - Click Reuse to reuse the existing worksheet or worksheets for the new data query results. The existing data query results are overwritten.

     Note: Any existing data and formatting (font, borders, or alignment for example) is retained when you reuse the existing worksheet or worksheets. This includes formatting automatically applied to the existing data query results.

3. Click the File tab.
4. Click Save.
5. If you are saving the workbook for the first time, the Save As dialog opens. Enter a name for the workbook and click Save.

Rerunning a data query

You can use the Longview Add-In for Office to rerun an existing data query. Rerunning a data query refreshes data values in the query and allows you to modify the query as necessary.

For information on refreshing queries that contain Formulas, see Refreshing data queries.

When you open a workbook containing data queries, the associated values are not updated to show the latest values in the Longview data server repository. The data will be exactly the same as when you last saved and closed the workbook. To see the most up-to-date data from the data server repository, you must rerun your data queries.

If you create a new worksheet or worksheets for the rerun data query results, and symbols in the Worksheet dimension of the rerun data query are duplicates of the symbols in the Worksheet dimension of the existing data query, symbol names in the worksheet tabs of the rerun data query results are appended with (N), where N is a unique number corresponding to the number of duplicates created.

For example, if two new worksheets are created (for two duplicate symbols), the symbol names in the worksheet tabs appear as follows:

- SymName
- SymName (2)
- SymName (3)

To rerun a data query:

1. Right-click any cell in the results of the data query that you want to rerun and select Rerun Query. The Data Query dialog opens, with the Data Options page displayed.

2. Make any changes to the data query as necessary.

> **Note:** If you specify more than one symbol for the Row, Column, or Worksheet dimensions, you cannot change the orientation of that dimension to a fixed dimension unless you delete the additional symbols.

3. Clear or select the 'Clear previous results' checkbox.

• Selected (default): Removes all existing query results and formatting before displaying the new results.

• Cleared: Retains existing query results and formatting, overlaying the new results on top of the existing data

4. Click Run Query.

If you did not make any changes to the data query, or if any symbols in the Worksheet dimension of the modified data query are duplicates of the symbols in the Worksheet dimension of the existing data query, a dialog appears prompting you to create or reuse existing worksheet(s).

> **Note:** The 'Clear previous results' checkbox only applies when you choose reuse existing worksheets in this dialog.

- Click Create to create a new worksheet or worksheets for the rerun data query results.
- Click Reuse to reuse the existing worksheet or worksheets for the rerun data query results. The existing data query results are overwritten based on your Clear previous results selection.

> **Note:** Any existing data and formatting (font, borders, or alignment for example) is retained when you reuse the existing worksheet or worksheets with the Clear previous results check box not selected. This includes formatting automatically applied to the existing data query results.

Refreshing data queries

When you open a workbook containing a data query, the associated values are the same as when you last saved and closed the workbook. To see the most up-to-date data from the data server repository, you can manually refresh your data query.

Refreshing data queries in a worksheet

To see the most current data server repository information for a data query in a worksheet, you must manually refresh that data query. You can also choose to refresh all data query data in the open workbook.

For more information, see Refreshing data queries in a workbook.

To refresh a data query in a worksheet:

1. Open the worksheet containing the data query to refresh.
2. Click the Longview tab.

3. In the Data group, click Refresh Worksheet. All cells containing query data in the worksheet are refreshed.

Refreshing data queries in a workbook

In Microsoft Excel, a workbook consists of one or more worksheets. Each worksheet appears as a tab at the bottom of the workbook. A Microsoft Excel workbook corresponds to a single .xls or .xlsx file.

To see the most current data server repository information in all workbook cells containing data queries, you must manually refresh those queries. You can also choose to refresh data queries in the active worksheet only.

For more information, see Refreshing data queries in a worksheet.

This functionality is available for data queries only if one of the Formulas options was specified for the As field when you ran the data query. If one of the Values options was specified for the As field, you can refresh the data by rerunning the data query.

For more information, see Querying base data and Rerunning a data query.

To refresh all data queries in a workbook:

1. Open the workbook containing the data queries to refresh.
2. Click the Longview tab.
3. In the Data group, click the arrow beside Refresh Workbook, and click one of the following options:

Option	Description
By Cycling Through All Worksheets	If you select this option, the system refreshes the workbook by making a call to the data server for each worksheet in the workbook. **Note:** Longview recommends that you use this option if you are refreshing a large amount of data or using a slower network.
With a Single Database Retrieval	If you select this option, the system refreshes the Database Retrieval workbook by making a single call to the data server for the entire workbook.

All cells containing data queries in the workbook are refreshed.

Saving and loading data query selections

You may save the query selections that you have selected in the Data Options and Format Options tabs of the Data Query dialog. This will enable you to quickly prepopulate the Data Query dialog in the existing session or within a future session.

The query selections are saved to a file. You can name and save the file in a local or network drive. This enables you to give the query a meaningful name and the ability to share your query with others.

Saving data query selections

To save your query selections, you must first specify the selections you wish to save in the Data Query dialog. For more information, see Querying base data.

To save data query selections:

1. Complete the fields in the Data Query dialog.
2. Click Save Query.
3. The Save As dialog opens. Enter a name for the query and click Save.

Loading data query selections

You may quickly pre-populate the Data Query dialog with selections from a saved query.

To load data query selections:

1. Click the Longview tab.
2. In the Insert group, click Data Query. The Data Query dialog opens with the Data Options page in view.

   Note: You must be connected to the data server to run a data query.

3. Click Load Query.
4. The Open dialog appears.
5. Select the query file that you wish to load and click Open.
6. The fields of the Data Query dialog are populated with the query selections from the selected file.
7. You may modify these selections before running the query.

Querying Symbols

You can use the Longview Add-In for Office to query symbol data such as the balance type, weight, and priority of a specified symbol or symbols from the data server repository.

Understanding symbol query output

A symbol is a plane of data within the database. Symbols identify specific cells within a hierarchy and are used to store information in the Longview database. When you run a symbol query, the results of the query are output to a Microsoft Excel workbook. The symbol query results appear in the following layout:

	A	B	C
1	Hierarchy	ACCOUNTS_Root	ACCOUNTS_Symbol
2	**TRIALBAL - Trial Balance**	**TRIALBAL**	**TRIALBAL**
3	**BALSHEET - Balance Sheet**	**TRIALBAL**	**BALSHEET**
4	**+TOTASSET - Total Assets**	**TRIALBAL**	**TOTASSET**
5	***+CURAST - Current Assets Total***	**TRIALBAL**	**CURAST**
6	**+CASHT - Cash Total**	**TRIALBAL**	**CASHT**
7	+CASH1 - Cash Account 1	TRIALBAL	CASH1
8	+CASH2 - Cash Account 2	TRIALBAL	CASH2
9	+CASH3 - Cash Account 3	TRIALBAL	CASH3

The following colors and character formats are applied to symbols in the Hierarchy column in the symbol query results:

Colors and character formats	Description
Red	Red characters indicate that the symbol is a static symbol. For more information, see [DimName]_SymbolType in the following table.
Blue	Blue characters indicate that the symbol is a carry forward symbol. For more information, see [DimName]_SymbolType in the following table.
Black	Black characters indicate that the symbol is a standard symbol. For more information, see [DimName]_SymbolType in the following table.
Italic	Italic formatting indicates that the symbol is a virtual symbol. For more information, see [DimName]_Virtual in the following table.
Bold	Bold formatting indicates that the symbol is a parent symbol. **Note:** Bold formatting is applied to parent symbols in all columns in the symbol query results.
Thick bottom border	If multiple root symbols were specified in the symbol query, a thick bottom border separates the lists of symbols for each root symbol in the query results.

Symbol queries display the following columns in the query results:

Heading	Description
Hierarchy	The hierarchy for your queried symbols appears in this column, including the symbol name and description as specified for Hierarchy in step 3. Symbols are indented to show their relation to other symbols in the hierarchy. The symbol weight appears to the left of the symbol name: ■ A plus (+) indicates that the symbol weight is 1. ■ A minus (-) indicates that the symbol weight is -1. ■ If there is no sign to the left of the symbol name, the symbol weight is 0. For more information, see [DimName]_Weight.
[DimName]_Root	The root symbol of the specified symbol.
[DimName]_Symbol	The symbol name of the specified symbol.
[DimName]_DescE	The English language description of the specified symbol.
[DimName]_DescA	The alternate language description of the specified symbol.
[DimName]_SymbolType	The symbol type of the specified symbol. Possible values for SymbolType include the following: ■ Standard — Standard symbols roll up normally (for example, months adding to quarters adding to years). Most symbols are standard. ■ Carry Forward — Carry forward symbols roll up the last child time period symbol's value to its parent. This symbol type is used only for account and time period symbols to identify rollup behavior of business accounts in the time periods dimension. ■ Static — Static symbols do not roll up to the parent symbol in any dimension. They override the symbol types of any symbol that they intersect with in other dimensions.
[DimName]_ChildSort	Indicates how child symbols of the specified symbol are sorted in the hierarchy. Possible values for ChildSort include the following: ■ Manual — Child symbols are sorted manually by symbol priority. ■ Ascending — Child symbols are sorted by name in alphabetical order. ■ Descending — Child symbols are sorted by name in reverse-alphabetical order.

Heading	Description
[DimName]_BalanceT ype	The balance type of the specified symbol. Possible values for BalanceType include the following: ■ Debit — The symbol is a debit symbol. ■ Credit — The symbol is a credit symbol. ■ Neither — The symbol is not a credit or a debit symbol. **Note:** Debit and Credit balance types apply to symbols in the ACCOUNTS dimension only. Symbols in all other dimensions have a balance type of Neither.
[DimName]_Parent	The parent symbol of the specified symbol.
[DimName]_Weight	Indicates the effect that the specified symbol has on its parent. Possible values for Weight include the following: ■ 1 — The symbol is added to its parent. ■ -1 — The symbol is subtracted from its parent. ■ 0 — The symbol has no mathematical effect on its parent.
[DimName]_Priority	The specified symbol's priority. Priority is a number that designates a symbol's position in the hierarchy relative to its parent. Symbols are listed in order of ascending priority, with zeroes falling at the bottom of the list. A symbol can have multiple parents and a different priority relative to each parent.
[DimName]_Level	The number of levels the specified symbol is below the root symbol of the current symbol query.
[DimName]_ NumOfChildren	The number of child symbols that the specified symbol has.
[DimName]_Virtual	Indicates whether the specified symbol is a virtual symbol. A virtual symbol is a parent symbol whose data is not stored in the data server repository like other symbols. Instead, the value for the parent is calculated on the server side when it is queried. The data for virtual parent symbols is calculated only when specifically requested. Symbols for quarters in the TIMEPER dimension are often virtual parent symbols. Possible values for Virtual include the following: ■ TRUE — The symbol is a virtual symbol. ■ FALSE — The symbol is not a virtual symbol.

Heading	Description
[DimName]_AcceptRollup	Indicates whether the specified symbol receives rollup data from its child symbols. Possible values for AcceptRollup include the following: ■ TRUE — The symbol receives rollup data from its child symbols. ■ FALSE — The symbol does not receive rollup data from its child symbols.
[DimName]_Created	The date and time that the symbol was created.
[DimName]_Modified	The date and time that the symbol was last modified.
[DimName]_ TotalDescendants	The total number of parent and child symbols below the symbol in the hierarchy.

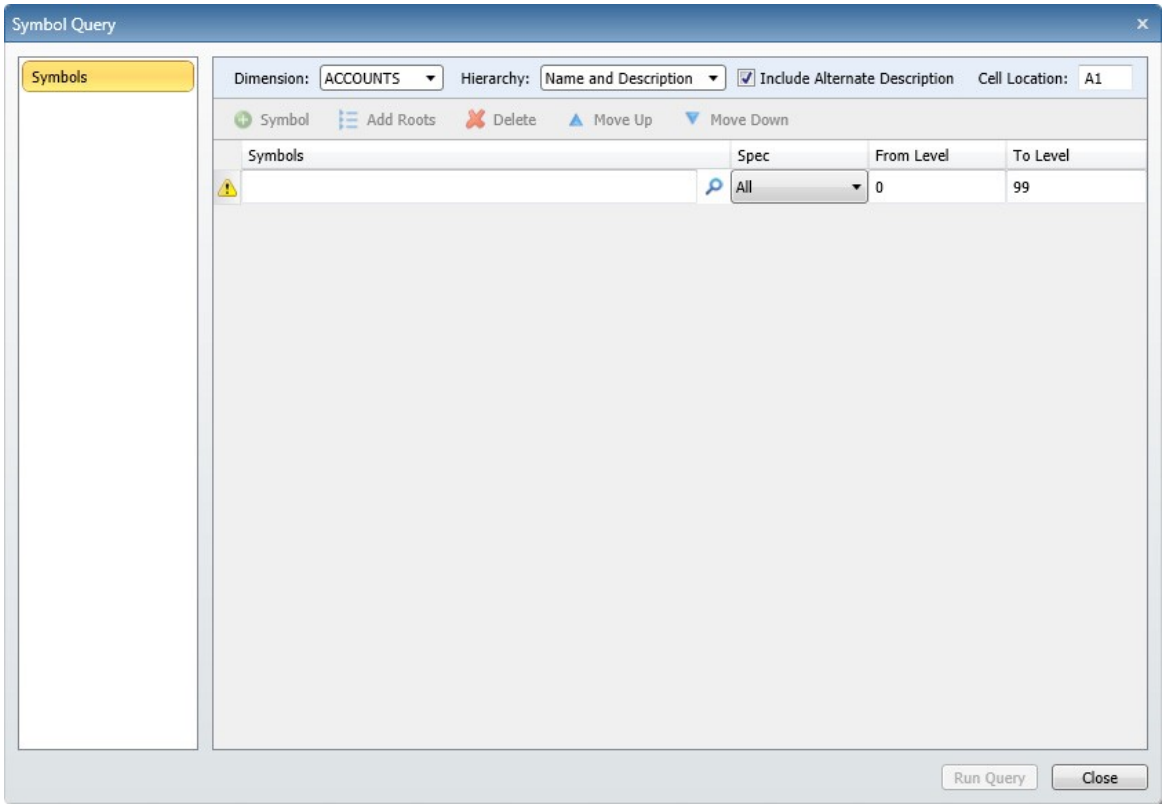
Querying symbol information

You can query a list of symbols into a Microsoft Excel worksheet that shows the details for each of those symbols.

Longview recommends that you place one query per worksheet in Microsoft Excel. However, if you have to run multiple queries and place them in a single Microsoft Excel worksheet, ensure that the query results do not overlap.

To query symbols:

1. Click the Longview tab.
2. In the Insert group, click Symbol Query. The Symbol Query dialog opens.



3. Complete the following fields:

Field	Description
Dimension	Select the dimension containing the symbol or symbols that you want to query.
Hierarchy	Select the symbol data to display in the query results, using one of the following options: ■ Name and Description — To display the symbol name and description. ■ Name — To display the symbol name only. ■ Description — To display the symbol description only. The default value for this field is Name and Description.
Include Alternate Description	Select this field to include the alternate language description of the symbol in the symbol query results.
Cell Location	Type the location of the top left cell of the query results. Keep in mind that the query results overwrite any data in the worksheet cells.

4. To select the symbol data to appear in your Microsoft Excel worksheet, complete these fields:

Field	Description
Symbols	Type the name of a symbol that you want to query or click the symbol selector button to search the hierarchy for the selected dimension. For information, see Using the Symbol Selector.

Field	Description
Spec	Select the type of symbol to query, using one of the following options: ■ All — To query all symbol types. ■ Leaf — To query only leaf symbols. ■ Parent — To query only parent symbols. ■ Root and Parent — To query only root and parent symbols. The default value for this field is All.
From Level	Type a number that specifies the start of the list of symbols to appear in the query results. The number refers to a number of levels of descendants below the symbol selected for Symbols in the hierarchy. For example, if you type 0, the list of symbols in the query results starts from the selected symbol. If you type 1, the list of symbols in the query results starts one level below the selected symbol.
To Level	Type a number that specifies the end of the list of symbols to appear in the query results. The number refers to a number of levels of descendants below the symbol selected for Symbols in the hierarchy. For example, if you type 3, the list of symbols in the query results ends three levels below the selected symbol. **Note:** The number specified for To Level must be equal to or higher than the number specified for From Level. If you specify 99 levels for To Level, all levels below the level specified for From Level are included in the results.

5. To add all root symbols for the specified dimension to the symbol query, click Add Roots.

   Note: Root symbols are added to the symbol query in alphabetical order.

6. If you want to query data for additional symbols, complete the following steps:
   a. Click Symbol. A new row appears.
   b. For the new symbol, complete the following fields:

Field	Description
Symbols	Type the name of a symbol that you want to query or click the symbol selector button to search the hierarchy for the selected dimension. For information, see Using the Symbol Selector.

Field	Description
Spec	Select the type of symbol to query, using one of the following options: ■ All — To query all symbol types. ■ Leaf — To query only leaf symbols. ■ Parent — To query only parent symbols. ■ Root and Parent — To query only root and parent symbols. The default value for this field is All
From Level	Type a number that specifies the start of the list of symbols to appear in the query results. The number refers to a number of levels of descendants below the symbol selected for Symbols in the hierarchy. For example, if you type 0, the list of symbols in the query results starts from the selected symbol. If you type 1, the list of symbols in the query results starts one level below the selected symbol.
To Level	Type a number that specifies the end of the list of symbols to appear in the query results. The number refers to the number of levels of descendants below the symbol selected for Symbols in the hierarchy. For example, if you type 3, the list of symbols in the query results ends three levels below the selected symbol. **Note:** The number specified for To Level must be equal to or higher than the number specified for From Level. If you specify 99 levels for To Level, all levels below the level specified for From Level are included in the results.

c. Repeat step a and step b for each additional symbol that you want to include in your symbol query.

Note: To delete additional symbols that you have added, select the row containing the symbol, and click Delete. You can also change the order in which the symbols appear in the query results by selecting the row containing the symbol and clicking Move Up or Move Down.

7. Click Run Query. The results of the symbol query appear in the worksheet.
8. Click the File tab.
9. Click Save.
10. If you are saving the workbook for the first time, the Save As dialog opens. Enter a name for the workbook and click Save.

Rerunning a symbol query

You can use the Longview Add-In for Office to rerun an existing symbol query. Rerunning a symbol query refreshes data values in the query and also allows you to modify the query as necessary.

When you open a workbook containing symbol queries, the associated values are not updated to show the latest values in the Longview data server repository. The data will be exactly the same as when you last saved and closed the workbook. To see the most up-to-date data from the data server repository, you must rerun your symbol queries.

To rerun a symbol query:

1. Right-click any cell in the results of the symbol query that you want to rerun and select Rerun Query. The Symbol Query dialog opens.
2. Make any changes to the symbol query as necessary.
3. Clear or select the 'Clear previous results' checkbox.

   • Selected (default): Removes all existing query results and formatting before displaying the new results.

   • Cleared: Retains existing query results and formatting, overlaying the new results on top of the existing data.
4. Click Run Query. The results of the symbol query appear in Microsoft Excel, with refreshed data values and any specified changes.

> **Note:** Any formatting (font, borders, or alignment for example) in the worksheet is retained when you rerun a symbol query with Clear previous results check box not selected. This includes any colors and character formats automatically applied to existing symbol query results.

Saving and loading symbol query selections

You may save the query selections that you have selected in the Symbol Query dialog. This will enable you to quickly pre-populate the Symbol Query dialog in the existing session or within a future session.

The query selections are saved to a file. You can name and save the file in a local or network drive. This enables you to give the query a meaningful name and the ability to share your query with others.

Saving symbol query selections

To save your query selections, you must first specify the selections you wish to save in the Symbol Query dialog. For more information, see Querying symbol information.

To save symbol query selections:

1. Complete the fields in the Symbol Query dialog.
2. Click Save Query.
3. The Save As dialog opens.
4. Enter a name for the query and click Save.

Loading symbol query selections

You may quickly pre-populate the Symbol Query dialog with selections from a saved query.

To load data query selections:

1. Click the Longview tab.
2. In the Insert group, click Symbol Query. The Symbol Query dialog opens.

   Note: You must be connected to the data server to run a symbol query.

3. Click Load Query.
4. The Open dialog appears.
5. Select the query file that you wish to load and click Open.
6. The fields of the Symbol Query dialog are populated with the query selections from the selected file.

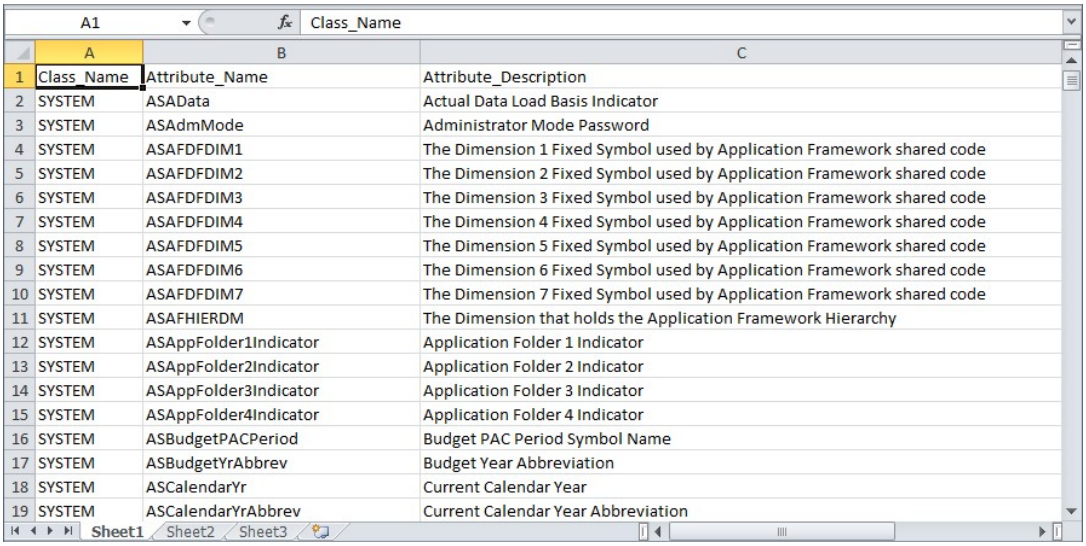
You may modify these selections before running the query.

Querying Attributes

You can use the Longview Add-In for Office to query attribute data from the data server repository.

Understanding attribute query output

An attribute describes the characteristics of an object in the application. You can use the Longview Add-In for Office to query System, User, or Symbol attributes from the Longview data server repository. When you run an attribute query, the results of the query are output to a Microsoft Excel workbook. The attribute query results appear in the following layout:



A1 | Class_Name

	A	B	C
1	Class_Name	Attribute_Name	Attribute_Description
2	SYSTEM	ASAData	Actual Data Load Basis Indicator
3	SYSTEM	ASAdmMode	Administrator Mode Password
4	SYSTEM	ASAFDFDIM1	The Dimension 1 Fixed Symbol used by Application Framework shared code
5	SYSTEM	ASAFDFDIM2	The Dimension 2 Fixed Symbol used by Application Framework shared code
6	SYSTEM	ASAFDFDIM3	The Dimension 3 Fixed Symbol used by Application Framework shared code
7	SYSTEM	ASAFDFDIM4	The Dimension 4 Fixed Symbol used by Application Framework shared code
8	SYSTEM	ASAFDFDIM5	The Dimension 5 Fixed Symbol used by Application Framework shared code
9	SYSTEM	ASAFDFDIM6	The Dimension 6 Fixed Symbol used by Application Framework shared code
10	SYSTEM	ASAFDFDIM7	The Dimension 7 Fixed Symbol used by Application Framework shared code
11	SYSTEM	ASAFHIERDM	The Dimension that holds the Application Framework Hierarchy
12	SYSTEM	ASAppFolder1Indicator	Application Folder 1 Indicator
13	SYSTEM	ASAppFolder2Indicator	Application Folder 2 Indicator
14	SYSTEM	ASAppFolder3Indicator	Application Folder 3 Indicator
15	SYSTEM	ASAppFolder4Indicator	Application Folder 4 Indicator
16	SYSTEM	ASBudgetPACPeriod	Budget PAC Period Symbol Name
17	SYSTEM	ASBudgetYrAbbrev	Budget Year Abbreviation
18	SYSTEM	ASCalendarYr	Current Calendar Year
19	SYSTEM	ASCalendarYrAbbrev	Current Calendar Year Abbreviation

Sheet1 Sheet2 Sheet3

The following format is applied to attributes in the query results:

- The first row for a different User or Symbol attribute displays with a gray background.
- Multiple attributes are sorted by the attribute name in the query results, in alphabetical ascending order. Since attribute query results for User or Symbol attributes can return multiple attribute values for each User or Symbol object (with each attribute value displayed on a separate row), this provides you with a visual indicator to distinguish between separate attributes.
- In addition, the columns displayed in the attribute query results vary depending on the option selected for Retrieve.
- If Values was selected, see Understanding attribute values query results.
- If Definitions was selected, see Understanding attribute definitions query results.

Understanding attribute values query results

Attribute queries that are set to retrieve attribute values display the following columns in the query results:

Column	Description
Class_ Name	The attribute class. Possible values for Class_Name include the following: ■ SYSTEM — System attributes specify system-wide characteristics. ■ USER — User attributes describe the attributes of a particular user. ■ SYMBOL — Symbol attributes describe the characteristics of individual symbols.
Attribute_ Name	The name of the attribute.
Attribute_ Description	The description of the attribute.
Attribute_ Type	The type of data value that can be specified for the attribute, and whether the attribute will accept a single value or a list of values (in which case the attribute type is appended with List). Possible values for Attribute_Type include the following: Date, Date List, Double, Double List, Integer, Integer List, String, String List, Symbol, and Symbol List.
Object_ Name	The name of the attribute object. ■ If the attribute class is SYSTEM, DBDEFAULT displays in the column. ■ If no attribute value is explicitly set for the attribute, [Default] displays in the column.
Attribute_ Value	The numeric or string value that the attribute is set to. If no value is set for the attribute, the column is left empty.
Same_as_ Default	Indicates whether the value the attribute is set to is the same as the default value for the attribute. Possible values for Same_as_Default include the following: ■ Y — The attribute value is the same as the default value. ■ N — The attribute value is different than the default value. ■ N/A — The attribute does not have a default value.

Understanding attribute definitions query results

Attribute queries that are set to retrieve attribute definitions display only the following columns in the query results:

Heading	Description
Class_Name	The attribute class. Possible values for Class_Name include the following: ■ SYSTEM — System attributes specify system-wide characteristics. ■ USER — User attributes describe the attributes of a particular user. ■ SYMBOL — Symbol attributes describe the characteristics of individual symbols.
Attribute_ Name	The name of the attribute.
Attribute_ Description	The description of the attribute.
Attribute_ Type	The type of data value that can be specified for the attribute, and whether the attribute will accept a single value or a list of values (in which case the attribute type is appended with List). Possible values for Attribute_Type include the following: Date, Date List, Double, Double List, Integer, Integer List, String, String List, Symbol, and Symbol List.
Attribute_ AccessType	Indicates whether users can set values for the attribute. Possible values for Attribute_ AccessType include the following: ■ Read — The attribute is read-only. ■ Write — Users can set the attribute value.
Attribute_ Value	The numeric or string value that the attribute is set to. If no value is set for the attribute, the column is left empty.

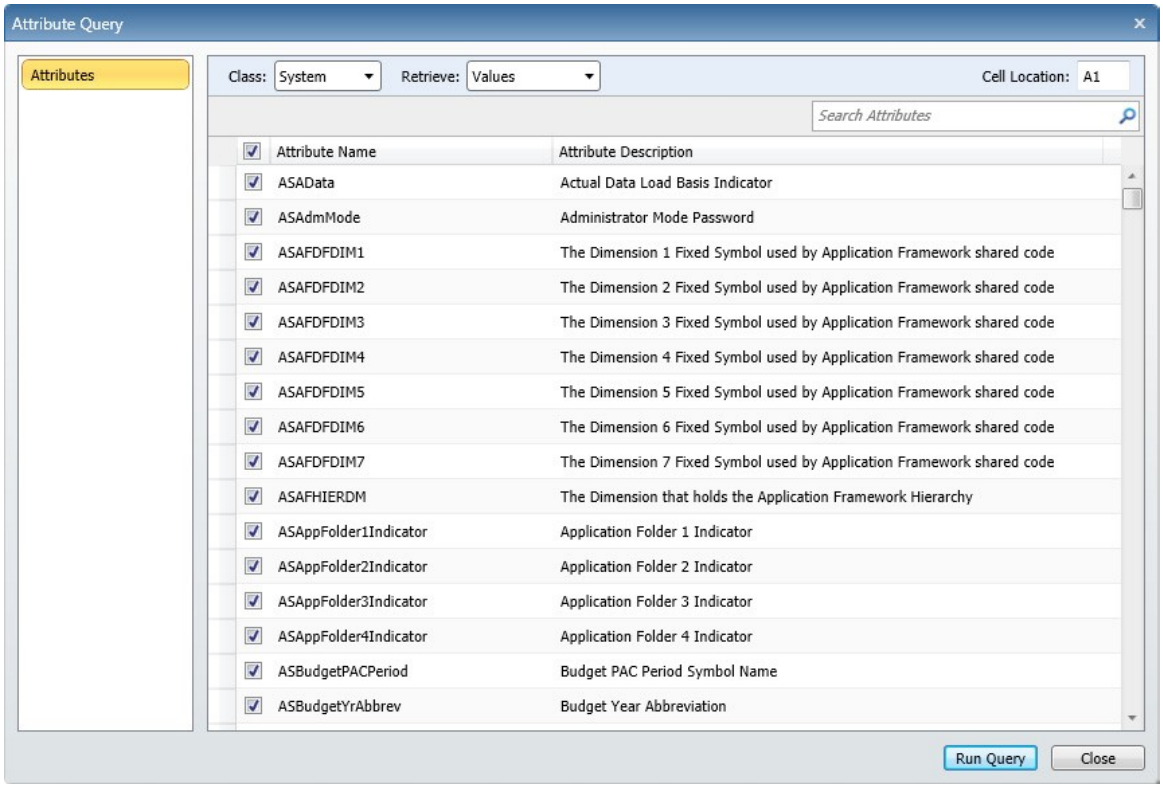
Querying attribute information

When querying attribute data, you have the option of querying an entire attribute class, multiple attributes, or a single attribute.

Longview recommends that you place one query per worksheet in Microsoft Excel. However, if you must run multiple queries and place them in a single Microsoft Excel worksheet, ensure that the query results do not overlap.

To query attributes:

1. Click the Longview tab.
2. In the Insert group, click Attribute Query. The Attribute Query dialog opens.



3. Complete the following fields:

Field	Description
Class	Select the class of attributes that you want to query, using one of the following options from the drop-down list: ■ System — To query System attributes. System attributes specify system-wide characteristics. There is only one object in the System attribute class — the application itself. ■ User — To query User attributes. User attributes describe the attributes of a particular user. Each user is an object in the User attribute class. ■ Symbol — To query Symbol attributes. Symbol attributes describe the characteristics of individual symbols. Each symbol is an object in the Symbol attribute class.

Field	Description
Retrieve	Select the type of attribute data that you want to query, using one of the following options from the drop-down list: ■ Values — To query attribute descriptions and values. ■ Definitions — To query attribute definitions only. The default value for this field is Values.
Cell Location	Type the location of the top left cell of the query results. Keep in mind that the query results overwrite any existing data in the worksheet cells.

4. By default, all attributes in the specified class are selected. If you do not want to query all attributes, click the Select All check box in the header row to deselect all attributes, and select the check box adjacent to the attributes that you want to query.

> **Note:** To search for specific attributes to query, type the attribute name or description in the Search Attributes field. The first matching attribute is selected. To find the next matching attribute, click Next.

5. Click Run Query. The results of the attribute query appear in the worksheet.
6. Click the File tab.
7. Click Save.
8. If you are saving the workbook for the first time, the Save As dialog opens. Enter a name for the workbook and click Save.

Rerunning an attribute query

You can use the Longview Add-In for Office to rerun an existing attribute query. Rerunning an attribute query refreshes data values in the query and allows you to modify the query as necessary.

When you open a workbook containing attribute queries, the associated values are not updated to show the latest values in the Longview data server repository. The data will be exactly the same as when you last saved and closed the workbook. To see the most up-to-date data from the data server repository, you must rerun your attribute queries.

To rerun an attribute query:

1. Right-click any cell in the results of the attribute query that you want to rerun and select Rerun Query. The Attribute Query dialog opens.
2. Make any changes to the attribute query as necessary.
3. Clear or select the 'Clear previous results' checkbox.

   • Selected (default): Removes all existing query results and formatting before displaying the new results.

   • Cleared: Retains existing query results and formatting, overlaying the new results on top of the existing data.

4. Click Run Query. The results of the attribute query appear in Microsoft Excel, with refreshed data values and any specified changes.

> **Note:** Any formatting (font, borders, or alignment for example) in the worksheet is retained when you rerun an attribute query with the Clear previous results checkbox not selected. This includes the formatting automatically applied to existing attribute query results.

Saving and loading attribute query selections

You may save the query selections that you have selected in the Attribute Query dialog. This will enable you to quickly pre-populate the Attribute Query dialog in the existing session or within a future session.

The query selections are saved to a file. You can name and save the file in a local or network drive. This enables you to give the query a meaningful name and the ability to share your query with others.

Saving attribute query selections

To save your query selections, you must first specify the selections you wish to save in the Symbol Query dialog. For more information, see Querying attribute information.

To save symbol query selections:

1. Complete the fields in the Attribute Query dialog.
2. Click Save Query.
3. The Save As dialog opens.
4. Enter a name for the query and click Save.

Loading attribute query selections

You may quickly pre-populate the Attribute Query dialog with selections from a saved query.

To load attribute query selections:

1. Click the Longview tab.
2. In the Insert group, click Attribute Query. The Attribute Query dialog opens.

   > **Note:** You must be connected to the data server to run an attribute query.

3. Click Load Query. The Open dialog appears.
4. Select the query file that you wish to load and click Open.
5. The fields of the Attribute Query dialog are populated with the query selections from the selected file.

You may modify these selections before running the query.

Extracting Data

You can use the Longview Add-In for Office to extract data from the data server repository and output it to a Microsoft Excel spreadsheet. Output data appears as a flat table where each data value is represented as a row and each dimension is represented as an intersecting column. This output format can be conveniently used in a Microsoft Excel Table or a PivotTable, which can further organize the data.

You can configure the data extract to your preferred settings in the Data Extract dialog. You can also save your preferred settings and use them to re-run the data extract.

Once you are connected to the Longview data server, you are ready to work with the data contained within it. This chapter explains how to use the Longview Add-In for Office to extract data.

Analyzing Data Query Results

You can further investigate the results of a data query by reorienting them or drilling into them multiple times in succession. In addition, you can revert to previous reorientations and data drills.

Reorienting data query results

You can use the Longview Add-In for Office to reorient data query results to investigate any data discrepancies for a specified row and fixed dimension in a selected worksheet.

The first time you reorient data query results, a new data query opens in a new workbook in Microsoft Excel so that the integrity of the original workbook is maintained.

The new data query is reoriented so that data for the symbol and one level of symbols below the symbol in the selected fixed dimension displays in rows on the left side of the worksheet. The symbol that was in the Row dimension (corresponding to the data intersection that was right-clicked) displays in a fixed dimension in the Query Information section.

For more information on the Query Information section, see Understanding data query output.

The Worksheet dimension symbol for the worksheet selected in step 2 is the only symbol selected for the Worksheet dimension in the new data query. All other selected symbols and settings are the same as the data query that was reoriented.

If you want, you can reorient the new data query. However, in this case, a new data query does not open in a new workbook in Microsoft Excel. Instead, the new data query replaces the worksheet containing the previous data query.

For example, consider the following data query:

	A	B	C	D	E	F	G
1	E0110T	Los Angeles Total					
2	DIM3SET	Details Default					
3	USD	US Dollar					
4	SEGMENTST	Total Segments					
5	GrsCY_ASC	Current Year					
6	DIM7SET	Controls Default					
7	Query Time Stamp	4/4/2014 11:02 AM					
8							
9			A10Q1	A10Q2	A10Q3	A10Q4	AYR10
10			Qtr 1 10	Qtr 2 10	Qtr 3 10	Qtr 4 10	2010
11	A41110	Net product revenue - ext	1,037,594	65,846	19,461	83,541	**1,206,442**
12	A41002	41002 - Product revenue - I/C	24,967	95,662	87,598	54,021	**262,248**
13	**A41010**	**Product revenue**	**1,062,561**	**161,508**	**107,059**	**137,562**	**1,468,690**
14							

GrsCY_ASC Sheet1 Sheet2 Sheet3

To investigate data values for the A41110 — Net product revenue - ext symbol intersecting with the E0110T — Los Angeles Total symbol in the ENTITIES dimension, you could right-click any data intersection in the A41110 row, point to Reorient, and then click ENTITIES. A new workbook opens, containing a new data query.

	A	B	C	D	E	F	G
1	A41110	Net product revenue - ext					
2	DIM3SET	Details Default					
3	USD	US Dollar					
4	SEGMENTST	Total Segments					
5	GrsCY_ASC	Current Year					
6	DIM7SET	Controls Default					
7	Query Time Stamp	4/4/2014 1:20 PM					
8							
9			A10Q1	A10Q2	A10Q3	A10Q4	AYR10
10			Qtr 1 10	Qtr 2 10	Qtr 3 10	Qtr 4 10	2010
11	110	Los Angeles - E0110	28,456	38,164	45,972	28,456	**141,048**
12	E0130P1	ABC US LLP (110)	1,012,854	29,451	29,836	15,283	**1,087,424**
13	E0140P1	ABC NA LLP (110)	21,251	93,893	31,251	93,823	**240,218**
14	**E0110T**	**Los Angeles Total**	**1,062,561**	**161,508**	**107,059**	**137,562**	**1,468,690**
15							

GrsCY_ASC | Sheet1 | Sheet2 | Sheet3

In the new data query, data displays for the E0110T — Los Angeles Total symbol and one level of symbols below the E0110T — Los Angeles Total symbol (now in the Row dimension), and the A41110 — Net product revenue - ext symbol displays in the Query Information section (now in a fixed dimension).

To reorient data query results:

1. Open the workbook containing data query results that you want to reorient.
2. Select the worksheet for the Worksheet dimension symbol that you want to investigate.
3. Right-click any data intersection in a row for the Row dimension symbol that you want to investigate, point to Reorient, and then click DimName, where DimName is the name of the fixed dimension that you want to display data for. A new data query opens in a new workbook in Microsoft Excel.

Drilling into data query results

You can use the Longview Add-In for Office to drill into data query results to investigate data values for child symbols in the Row or Column dimensions.

The first time you drill into data query results, a new data query opens in a new workbook in Microsoft Excel so that the integrity of the original workbook is maintained.

The new data query displays data for all child symbols of the parent symbol that you selected, and the parent symbol itself.

The Worksheet dimension symbol for the worksheet selected in step 2 is the only symbol selected for the Worksheet dimension in the new data query. All other selected symbols and settings are exactly the same as the data query that was drilled into.

If you want, you can drill into parent symbols in the new data query. However, in this case, a new data query does not open in a new workbook in Microsoft Excel. Instead, the new data query replaces the worksheet containing the previous data query.

For example, consider the following data query:

	A	B	C	D	E	F	G
1			A10Q1	A10Q2	A10Q3	A10Q4	AYR10
2			Qtr 1 10	Qtr 2 10	Qtr 3 10	Qtr 4 10	2010
3	A41010	Product revenue	9,230	1,653	2,013	1,165	14,061
4	A41200	Service revenue	241	125	136	562	1,064
5	A41300	41300 - Other (non-product) revenue	479	145	625	132	1,381
6	A41000	Total revenue	9,950	1,923	2,774	1,859	16,506
7							

To investigate data values for the A41010 — Product revenue symbol, you could right-click either the symbol name or the symbol description (cell A3 or B3) and select Data Drill. A new workbook opens, containing a new data query.

	A	B	C	D	E	F	G
1			A10Q1	A10Q2	A10Q3	A10Q4	AYR10
2			Qtr 1 10	Qtr 2 10	Qtr 3 10	Qtr 4 10	2010
3	A41110	Net product revenue - ext	1,150	926	1,356	492	3,924
4	A41002	41002 - Product revenue - I/C	8,080	727	657	673	10,137
5	A41010	Product revenue	9,230	1,653	2,013	1,165	14,061
6							

In the new data query, data displays for the two child symbols of A41010 (A41110 — Net product revenue - ext and A41002 — 41002 - Product revenue - I/C) and A41010 itself in the Row dimension.

To drill into data query results:

1. Open the workbook containing data query results that you want to drill into.
2. Select the worksheet for the Worksheet dimension symbol that you want to investigate.
3. Right-click the symbol name or symbol description cell for any parent symbol in the Row or Column dimensions that you want to investigate and select Data Drill.

Note: Data Drill functionality is unavailable for parent symbols whose descendants are already displayed in the data query results.

A new data query opens in a new workbook in Microsoft Excel.

Reverting orientation and data drill

If you reorient or drill into data query results multiple times in succession, you can use the Longview Add-In for Office to revert to previous data drills and reorientations.

- To revert to the previous data drill or reoriented data query results, right-click any data intersection in the results, point to Go, and then click Back.
- To revert to the data query results after your first data drill or reorientation, right-click any data intersection in the results, point to Go, and then click Home.

Analyzing LVCELL functions

You can use the Longview Add-In for Office to run a data query from an existing LVCELL function. This allows you to investigate and analyze data values for child symbols of the symbols as specified for the

LVCELL function. This functionality is useful when you want to analyze data in a worksheet that contains LVCELL functions and does not contain any existing data queries.

Note: The Analyze option is available for worksheet cells containing one LVCELL function with valid syntax.

For more information on working with LVCELL functions, see LVCELL.

To analyze an LVCELL function:

1. Right-click the worksheet cell containing the LVCELL function and select Analyze. The Data Query dialog opens, with the Data Options page in view.
   a. The Data Options fields are prepopulated with the following values:

      Note: For more information on the Data Options fields, see Querying base data.

Field	Value
Show	All Data
As	Formulas, Adjusted, with Cell References
Cell Location	A1

   b. The symbols in the Data Options page are prepopulated with the following values:

Symbol field	Prepopulated with...	Spec	Level
Row dimension symbol	The first symbol as specified in the LVCELL function.	All	1
Column dimension symbol	The second symbol as specified in the LVCELL function.	All	1
Worksheet dimension symbol	The third symbol as specified in the LVCELL function.	All	0
Fixed dimension symbols (all other dimensions in your system)	All symbols following the third symbol as specified in the LVCELL function.	All	0

2. Make any changes to the Data Options as necessary.
3. Click Format Options in the left pane of the Data Query dialog. The Format Options page opens, populated with the default values.

   For more information, see Specifying format options for data queries.
4. Make any changes to the Format Options as necessary.
5. Click Run Query. The data query opens in a new workbook in Microsoft Excel with your specified orientation, symbols, and formats.

Formatting Query Results

When you run a data, symbol, or attribute query using the Longview Add-In for Office, default Longview formatting is applied to the query results. You can make changes to the query results in the Microsoft Excel workbook. However, depending on the type of change that you make, you may invalidate the query. If a query is invalidated, it cannot be rerun. In the case of data queries, it also cannot be analyzed, reoriented, or drilled into.

Note: You can change formatting for data query results without invalidating the query by modifying the format options or the Longview cell styles applied to the query results. For more information, see Specifying format options for data queries and Formatting data query results using Longview cell styles.

The following changes do not invalidate a query:

- Any formatting (font, borders, or alignment for example).
- Inserting or deleting a row or column outside of the range of the query results

The following changes invalidate a query:

- Any manual change to the values of a cell in the range of the query results.
- Inserting or deleting a row or column in the range of the query results.

If you want to query data from the data from the data server repository and you want to include additional commentary, images, columns, or rows in your worksheet, Longview recommends that you use Longview Functions.

For more information, see Using Longview Functions.

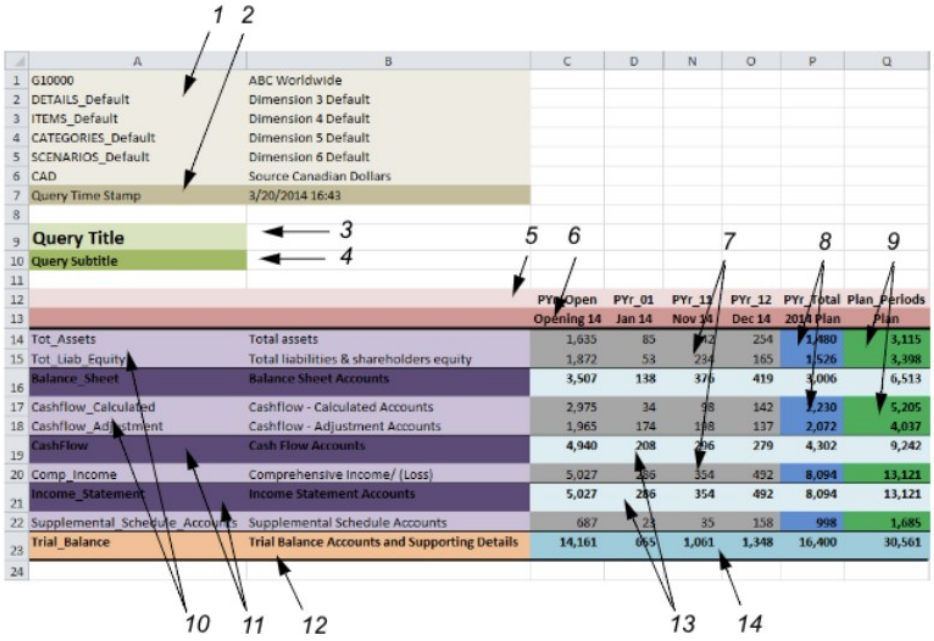
Formatting data query results using Longview cell styles

When you run a data query, preconfigured Longview cell styles are automatically created and applied to the data query results. You can apply Longview cell styles (available on the Home tab, in the Styles group) to content in other Microsoft Excel workbooks. In addition, you can modify these cell styles as necessary, and merge them with any custom cell styles that you have created.

For more information on working with cell styles, see the Microsoft Excel Help.

The Longview cell styles are applied to the following sections of the data query results:

Note: In the following example, colored shading has been added to the data query results to better illustrate the separate sections. In addition, columns E to M have been hidden to display all relevant sections of the data query results.



#	Style name	Description	Additional notes
1	LV_FixedSymbol	Fixed Symbols	
2	LV_QueryInfo	Time Stamp	
3	LV_Title	Report Title	
4	LV_Subtitle	Report Subtitle	
5	LV_CHeader	Column Name Headers	
6	LV_ CHeaderLastRow	Column Description Headers	
7	LV_DataDefault	Data Default	This cell style is applied to data that is not for a root symbol (total) or parent symbol (subtotal) in the data query.
8	LV_ CDataSubTotal	Column Data Subtotals (Parents)	
9	LV_CDataTotal	Column Data Totals (Roots)	

#	Style name	Description	Additional notes
10	LV_RHeader	Row Header Default	This cell style is applied to child symbols, and any parent symbols that are at the level of symbol detail specified in the data query.
11	LV_ RHeaderSubTotal	Row Header Subtotals (Parents)	This cell style is applied to parent symbols that are not at the level of symbol detail specified in the data query. For example, if you set Level to 2 for a symbol in the Row dimension of your data query (2 levels of symbol detail below the selected symbol is included in the query), the cell style is applied to parent symbols at the first level of symbol detail, but not to parent symbols at the second level of symbol detail.
12	LV_RHeaderTotal	Row Header Totals (Roots)	This cell style is applied to root symbols only.
13	LV_ RDataSubTotal	Row Data Subtotals (Parents)	This cell style is applied to parent symbols that are not at the level of symbol detail specified in the data query. For example, if you set Level to 2 for a symbol in the Row dimension of your data query (2 levels of symbol detail below the selected symbol is included in the query), the cell style is applied to parent symbols at the first level of symbol detail, but not to parent symbols at the second level of symbol detail.
14	LV_RDataTotal	Row Data Totals (Roots)	This cell style is applied to root symbols only.

Without the added shading, the data query results example appears as follows:

	A	B	C	D	N	O	P	Q
1	G10000	ABC Worldwide						
2	DETAILS_Default	Dimension 3 Default						
3	ITEMS_Default	Dimension 4 Default						
4	CATEGORIES_Default	Dimension 5 Default						
5	SCENARIOS_Default	Dimension 6 Default						
6	CAD	Source Canadian Dollars						
7	Query Time Stamp	3/20/2014 16:43						
8								
9	**Query Title**							
10	**Query Subtitle**							
11								
12			PYr_Open	PYr_01	PYr_11	PYr_12	PYr_Total	Plan_Periods
13			Opening 14	Jan 14	Nov 14	Dec 14	2014 Plan	Plan
14	Tot_Assets	Total assets	1,635	85	142	254	1,480	3,115
15	Tot_Liab_Equity	Total liabilities & shareholders equity	1,872	53	234	165	1,526	3,398
16	**Balance_Sheet**	**Balance Sheet Accounts**	**3,507**	**138**	**376**	**419**	**3,006**	**6,513**
17	Cashflow_Calculated	Cashflow - Calculated Accounts	2,975	34	98	142	2,230	5,205
18	Cashflow_Adjustment	Cashflow - Adjustment Accounts	1,965	174	198	137	2,072	4,037
19	**CashFlow**	**Cash Flow Accounts**	**4,940**	**208**	**296**	**279**	**4,302**	**9,242**
20	Comp_Income	Comprehensive Income/ (Loss)	5,027	286	354	492	8,094	13,121
21	**Income_Statement**	**Income Statement Accounts**	**5,027**	**286**	**354**	**492**	**8,094**	**13,121**
22	Supplemental_Schedule_Accounts	Supplemental Schedule Accounts	687	23	35	158	998	1,685
23	**Trial_Balance**	**Trial Balance Accounts and Supporting Details**	**14,161**	**655**	**1,061**	**1,348**	**16,400**	**30,561**
24								

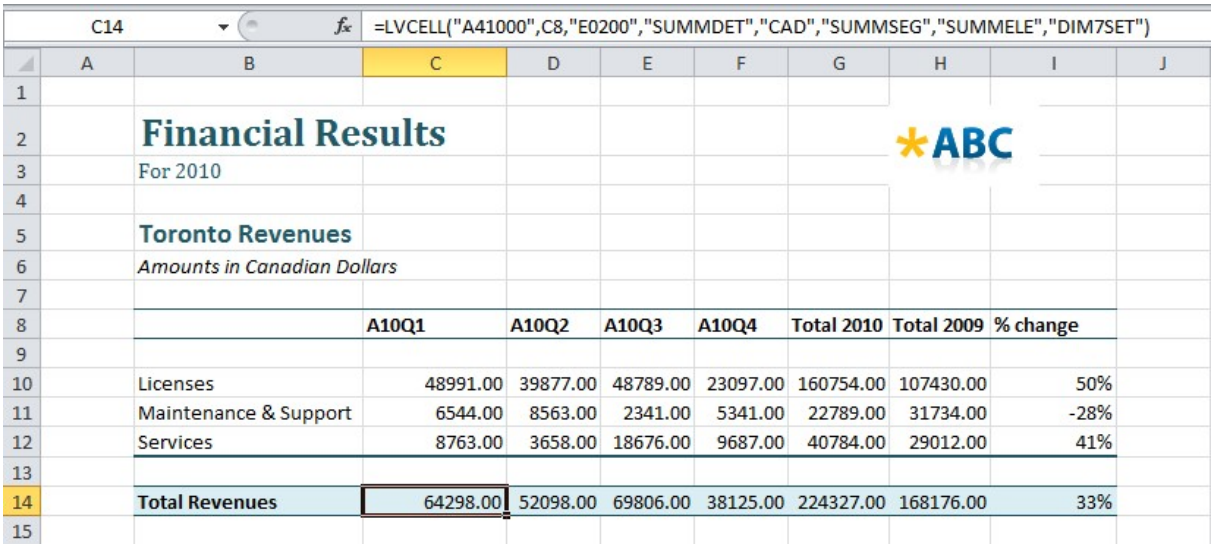
Working With Data Using Longview Functions

The Longview Add-In for Office includes Longview Functions that allow you to query different types of data into worksheet cells and submit data from worksheet cells to the Longview data server repository.

Using Longview Functions

You can use Longview Functions in the Longview Add-In for Office to query different types of data from the data server repository. Longview Functions are useful when you want to query a variety of specific items from the data server repository (data for a specific attribute or symbol, for example), and you want to include titles, additional rows or columns, inserted graphics, or other customizations in your Microsoft Excel worksheet.

Consider the following example:



	A10Q1	A10Q2	A10Q3	A10Q4	Total 2010	Total 2009	% change
Licenses	48991.00	39877.00	48789.00	23097.00	160754.00	107430.00	50%
Maintenance & Support	6544.00	8563.00	2341.00	5341.00	22789.00	31734.00	-28%
Services	8763.00	3658.00	18676.00	9687.00	40784.00	29012.00	41%
Total Revenues	64298.00	52098.00	69806.00	38125.00	224327.00	168176.00	33%

In the worksheet, LVCELL functions are used to query data for specific symbols, such as Total Revenues. In addition, as visible in the Formula Bar, the LVCELL functions reference other cells in the worksheet (in this case cell C8, which contains the time period). Additional customizations can be made to the worksheet without affecting the validity of the data returned from the data server repository.

For more information on the LVCELL function, see LVCELL.

Using multiple Longview Functions in a single cell

When you are working with Longview Functions in the Longview Add-In for Office, you can use certain Longview Functions multiple times in the same worksheet cell. Multiple Longview Functions can be used in the same cell in the following scenarios:

- Multiple Longview Functions that return numeric values can be used in formulas within a cell.

  For example:

```
=SUM(LVCELL("Sym_dim1", "Sym_dim2", ...),LVCELL("Sym_dim1", "Sym_
dim2", ...))
```

- Longview Functions that return numeric values can be used in place of function parameters that require values within other Longview Functions.

  For example:

  ```
  =LVSUBMIT("Sym_dim1", "Sym_dim2", ...,LVCELL("Sym_dim1", "Sym_dim2",
  ...))
  ```

  where the LVCELL function provides the value to be submitted to the data server repository for the LVSUBMIT function.

- Longview Functions that return symbol names can be used in place of function parameters that require symbol names within other Longview Functions.

  For example:

  ```
  =LVCELL("Sym_dim1", "Sym_dim2",LVCHILD("Dim", "Sym_name"), ...)
  ```

  where the LVCHILD function provides the symbol name required for the third dimension in the LVCELL function.

  Note: In any of the above examples, you can also use a reference to another cell location that contains the required Longview Function, instead of entering the additional Longview Functions directly in the formula.

The following Longview Functions cannot be used multiple times in the same cell, or used in place of parameters within other Longview Functions:

- LVLOCK
- LVSUBMIT

Using multiple LVCELL functions in a formula

When you are working with LVCELL functions in the Longview Add-In for Office, Longview recommends that you use one LVCELL function per worksheet cell. If you have a formula that requires multiple LVCELL functions, Longview recommends that you place each LVCELL function in a separate worksheet cell, and then use cell references to the cells that contain the LVCELL functions in the formula.

For example, if you are working with an IF formula that requires the following:

```
IF(LVCELLa<>0, LVCELLb, LVCELLc)
```

You can place LVCELLa in cell C2, LVCELLb in cell C3, and LVCELLc in cell C4, and use the following syntax for the IF formula:

```
IF(C2<>0, C3, C4)
```

Longview recommends that you do not use multiple LVCELL functions within the same cell.

If you use multiple LVCELL functions (for example,

```
=LVCELL(LVCELL(LVCELL(...))) or =IF(LVCELL(),LVCELL(),LVCELL())
```

you may need to refresh the worksheet multiple times to view the query data. Other unexpected behavior can occur.

Refreshing data from cells containing Longview Functions

You can refresh Longview Functions to view the most up-to-date data from the data server repository. You should refresh your worksheet or workbook in any of the following scenarios:

- When you open a workbook containing Longview Functions, the associated values are the same as when you last saved and closed the workbook. If you want to see the most up-to-date data from the data server repository, you can refresh the workbook.
- If you have had a worksheet containing function data open for a while, it is possible that another user has made changes in the Longview data server repository in the interim. You can refresh the worksheet to reflect any changes that other users have made to the data.
- When you enter an LVCELL or LVREFRESHDATETIME function into a worksheet cell, you must refresh the worksheet or workbook before you can see the query data in the worksheet cell.

Note: Longview recommends that you use one LVCELL function per work-sheet cell. If you use multiple Longview Functions within the same cell (for example, =LVCELL(LVCELL (LVCELL(...))) or =IF(LVCELL(),LVCELL(),LVCELL()), you may need to refresh the worksheet or workbook multiple times to view the query data.
For more information, see Using multiple LVCELL functions in a formula.

Refreshing function data in a worksheet

To see the most current data server repository information in worksheet cells containing Longview Functions, you must manually refresh those functions. You can also choose to refresh all function data in the open workbook.

Note: You can use the following procedure to refresh values in data query results if one of the Formulas options was specified for the As field when you ran the data query. For more information, see Querying base data.

To refresh all function data in a worksheet:

1. Open the worksheet containing function data to refresh.
2. Click the Longview tab.
3. In the Data group, click Refresh Worksheet. All cells containing functions in the worksheet are refreshed.

Refreshing function data in a workbook

In Microsoft Excel, a workbook consists of one or more worksheets. Each worksheet appears as a tab at the bottom of the workbook. A Microsoft Excel workbook corresponds to a single .xls or .xlsx file.

To see the most current data server repository information in all workbook cells containing Longview Functions, you must manually refresh those functions. You can also choose to refresh function data in the active worksheet only.

Note: You can use the following procedure to refresh values in data query results if one of the Formulas options was specified for the As field when you ran the data query. For more information, see Querying base data.

To refresh all function data in a workbook:

1. Open the workbook containing function data to refresh.
2. Click the Longview tab.
3. In the Data group, click the arrow beside Refresh Workbook, and click one of the following options:

Field	Description
By Cycling Through All Worksheets	If you select this option, the system refreshes the workbook by making a call to the data server for each worksheet in the workbook. **Note:** Longview recommends that you use this option if you are refreshing a large amount of data or using a slower network.
With a Single Database Retrieval	If you select this option, the system refreshes the workbook by making a single call to the data server for the entire workbook.

All cells containing functions in the workbook are refreshed.

Submitting data using the Longview Add-In for Office

You can use the Longview Add-In for Office to submit data from worksheet cells to the Longview dataserver repository. The LVLOCK and LVSUBMIT functions are used as a part of this process.

To apply and release locks in the Longview Add-In for Office, you must have Modify Data authorization. If you do not have Modify Data authorization, the Submit button is unavailable. In addition, to submit data in the Longview Add-In for Office, you must have Submit Data authorization for the Longview Add-In for Office. If you do not have Submit Data authorization, the Submit options are unavailable. These authorizations are set in Longview Application Administrator.

For more information, see the Longview Application Administrator Guide or contact your System Administrator.

Submitting data using the Longview Add-In for Office consists of the following main steps:

1. Applying Locks
2. Submitting Data
3. Releasing Locks

> **Note:** If an error occurs at any point in the process, the system stops at that point and an error message displays. The system will attempt to release any locks that were applied in the first step. For example, if an error occurs when locks are being applied, the system returns an error message and does not proceed to submit data. Any locks that were successfully applied are released.

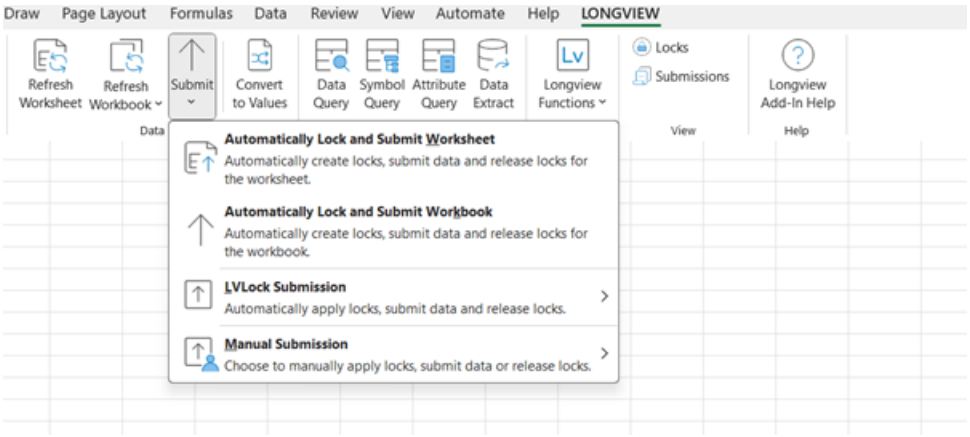
> **Note:** You can use the Data Locks tool to view and, if authorized, remove locks applied by other users, and the User Submissions tool to view the status of submitted data batches. For more information, see Accessing the Longview Add-In for Office.

The Submit menu provides different options to handle these steps, from fully automated to completely manual.

Accessing the Submit Options

To access the submission options:

1. Open the required Excel workbook.
2. Select the Longview tab from the Excel ribbon.
3. In the Data group, click Submit to view the available submission options.



The available submission options are:

1. Automatically Lock and Submit Worksheet
2. Automatically Lock and Submit Workbook

3. LVLock Submission
4. Manual Submission

Automatically Lock and Submit Worksheet

Use this option to submit data from the active worksheet with automatic lock management. This option will:

- Identify all LVSUBMIT, LVSSUBMIT and LVSUBMITCOMMENT functions in the active worksheet and automatically create the required locks to all referenced symbols.
- Submit the data to the Longview database for the active worksheet.
- Release the locks once the submission is complete.

This option eliminates the need for LVLOCK functions and is recommended when submitting data from a single worksheet.

Automatically Lock and Submit Workbook

Use this option to submit data from the entire workbook with automatic lock management.

This option will:

- Identify all LVSUBMIT, LVSSUBMIT and LVSUBMITCOMMENT functions in the active workbook and automatically create the required locks to all referenced symbols.
- Submit the data to the Longview database for the entire workbook.
- Release the locks once the submission is complete.

This option is recommended when submitting data from multiple worksheets within a single workbook.

LVLock Submission

The LVLock Submission option uses LVLOCK functions to define locks during the submission process. This option:

- Applies locks as defined by each LVLOCK function in the active worksheet or workbook.
- Submits data for each LVSUBMIT function in the active worksheet or workbook.
- Releases the locks defined by each LVLOCK function once the submission is complete.

This option provides two sub-options:

a. Submit Worksheet - Processes the active worksheet only.

b. Submit Workbook - Processes the entire workbook.

Manual Submission

Manual Submission provides complete control over each step of the submission process. This option includes separate commands for:

- Manually applying locks
- Manually submitting data
- Manually releasing locks

Manually applying locks

Before you can manually submit data using the Longview Add-In for Office, you must apply locks for the sections of the data server repository that you plan to submit data to.

To apply locks manually:

1. Open the workbook containing the LVLOCK functions that define the sections of the Longview data server repository to be locked and the LVSUBMIT functions that define the data values to be submitted to the Longview data server repository. For more information on the functions, see LVLOCK and LVSUBMIT.
   - LVLOCK: Ready to lock appears in each cell that contains an LVLOCK function,
   - LVSUBMIT: Ready to submit (Value) appears in each cell that contains an LVSUBMIT function, where Value is the value to be submitted to the data server repository.

   Note: You can use the Data Locks tool to view and (if you have authorization) delete any locks that other users may have applied to the data server repository. For more information, see Accessing the Longview Add-In for Office.
2. Click the Longview tab.
3. In the Data group, click Submit, point to Manual Submission, and then click one of the following options:

Option	Description
Apply Locks (Worksheet)	If you select this option, the system applies a lock as defined by each LVLOCK function in the active worksheet.
Apply Locks (Workbook)	If you select this option, the system applies a lock as defined by each LVLOCK function in the active workbook.

4. The Results dialog opens.
5. Review the information for the number of LVLOCK functions for which a lock was successfully acquired out of the total number of LVLOCK functions in the active worksheet or workbook (depending on the option selected in step 3).

   Note: The total number of LVLOCK functions in the active worksheet or workbook does not include LVLOCK functions that contain invalid parameters, or an invalid number of parameters.

6. Click OK.
    - LVLOCK: Created lock LockID appears in each cell that contains an LVLOCK function for which a lock was acquired, where LockID is the system-assigned ID of the lock.
7. Proceed to Manually submitting data.

Manually submitting data

After you have applied locks for the sections of the data server repository that you plan to submit data to, you can use the Longview Add-In for Office to manually submit data to the data server.

To submit data manually:

1. Open the workbook containing the LVLOCK functions that define the sections of the Longview data server repository to be locked and the LVSUBMIT functions that define the data values to be submitted to the Longview data server repository. For more information on the functions, see LVLOCK and LVSUBMIT.
2. Click the Longview tab.
3. If you have not already done so, apply the locks to the data server as described in Manually applying locks.
4. In the Data group, click Submit, point to Manual Submission, and then click one of the following options:

Option	Description
Submit Data (Worksheet)	If you select this option, the system submits data for each LVSUBMIT function in the active worksheet.
Submit Data (Workbook)	If you select this option, the system submits data for each LVSUBMIT function in the active workbook.

5. The Results dialog opens.
6. Review the information for the total number of values submitted by LVSUBMIT functions in the active worksheet or workbook (depending on the option selected in step 4).

    Note: The total number of LVSUBMIT functions in the active worksheet or workbook does not include LVSUBMIT functions that contain invalid parameters, or an invalid number of parameters.

7. Click OK.
    - LVSUBMIT: Submitted (Value) appears in each cell that contains an LVSUBMIT function for which data was submitted successfully, where Value is the value that was submitted to the data server repository.

        Note: You can use the User Submissions tool to view the status of the batches created for your data submission process. For more information, see Accessing the Longview Add-In for Office.

8. Proceed to Manually releasing locks.

Manually releasing locks

After you have manually submitted data to the data server, you can use the Longview Add-In for Office to release the locks on the sections of the data server repository so that other users can work with the data.

To release locks manually:

1. Open the workbook containing the LVLOCK functions that define the sections of the Longview data server repository to be locked and the LVSUBMIT functions that define the data values to be submitted to the Longview data server repository. For more information on the functions, see LVLOCK and LVSUBMIT.
    - LVLOCK: Created lock LockID appears in each cell that contains an LVLOCK function for which a lock was acquired Manually applying locks, where LockID is the system-assigned ID of the lock.
2. Click the Longview tab.
3. In the Data group, click Submit, point to Manual Submission, and then click one of the following options:

Option	Description
Release Locks (Worksheet)	If you select this option, the system releases the lock defined by each LVLOCK function in the active worksheet.
Release Locks (Workbook)	If you select this option, the system releases the lock defined by each LVLOCK function in the active workbook.

4. LVLOCK: Ready to lock appears in each cell that contains an LVLOCK function for which a lock was acquired successfully.

Converting Longview Functions to static values

You can use the Longview Add-In for Office to convert all cells in the active workbook containing Longview Functions to static data values. When you convert cells in the workbook to data values, you effectively remove all Longview Functions, References, and features from the workbook. You do not have to be connected to a Longview data server to convert Longview Functions to static values.

This functionality is useful if you want to provide a copy of the workbook to users who do not have the Longview Add-In for Office installed, or if you want to create a workbook that displays the data values in the Longview data server repository at a specific date and time.

Caution: If you want to continue working with Longview Functions or other Longview Add-In for Office features in the active workbook, make sure that you save the workbook before proceeding. This procedure cannot be undone.

To convert Longview Functions to static values:

1. Click the Longview Tab.
2. In the Data group, click Convert to Values. A confirmation dialog appears.
3. Click Convert. The active workbook now has the following properties:
   - All cells that contained Longview Functions are converted to static data values. This includes any Longview Functions created in data queries. For more information, see Querying base data.
   - Any queries in the workbook can no longer be rerun. In addition, data queries can no longer be reoriented or drilled into.
   - Cells that contained any functions other than Longview Functions are unchanged. However, worksheet cells that contained Longview Functions used in conjunction with other functions in the same worksheet cell are also converted to static data values. For example:

Formula	Converted to...
=AVERAGE(2,4,8)	Unchanged
=LVCELL("Sym_dim1", "Sym_dim2", ...)	Static data value
=AVERAGE(2,4,LVCELL("Sym_dim1", "Sym_dim2", ...))	Static data value
=AVERAGE(2,4)&LVCELL("Sym_dim1", "Sym_dim2", ...)	Static data value
=LVDIMNAME(AVERAGE(2,4))	Static data value

4. Save the workbook containing static data values with a new file name.

Longview Function Syntax

When you are working with Longview Functions, keep the following general syntax rule in mind. For each function parameter, you can either specify the required value, or specify a cell location that contains the required value. If you specify a value, enclose the parameter in double quotation marks ("Parameter"). If you specify a cell location, do not enclose the parameter in double quotation marks. An exception to this rule is function parameters that require a numeric value. If you specify a numeric value, do not enclose the parameter in double quotation marks.

Note: Excel expects function parameters in the language of the Operating System. If your workbooks will be used in Excel with different Language settings, you may find it useful to use a non-zero value to represent TRUE and a zero to represent FALSE for function parameters of Boolean type.

Longview Functions include the following:

- LVATTRIBUTE
- LVCELL
- LVCHILD
- LVDESC

- LVDIMDESC
- LVDIMNAME
- LVGROUP
- LVGROUPDESC
- LVIDENTIFIER
- LVLOCK
- LVPARENT
- LVREFRESHDATETIME
- LVSCELL
- LVSESSION
- LVSUBMIT
- LVSYMBOLSELECTOR
- LVUSER
- LVUSERDESC

Longview Functions include the following:

- LVATTRIBUTE
- LVCELL
- LVCHILD
- LVDESC
- LVDIMDESC
- LVDIMNAME
- LVGROUP
- LVGROUPDESC
- LVIDENTIFIER
- LVLOCK
- LVPARENT

- LVREFRESHDATETIME
- LVSCELL
- LVSESSION
- LVSUBMIT
- LVSYMBOLSELECTOR
- LVUSER
- LVUSERDESC

LVATTRIBUTE

Using the LVATTRIBUTE function, you can query Longview attribute data. You can use the LVATTRIBUTE function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVATTRIBUTE syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying attribute data using the Function Arguments dialog.

LVATTRIBUTE syntax

In the worksheet cell, type:

```
=LVATTRIBUTE("Attribute_class", "Attribute_name", ["object"])
```

where:

- Attribute_class is the attribute class. Specify one of the following values:

Value	Description
SYSTEM	Describes the entire Longview system at the highest level. Attributes of this class specify system-wide characteristics. There is only one object in the SYSTEM attribute class – the Longview application itself.
USER	Describes the attributes of a particular user. Each user is an object in the USER attribute class.
SYMBOL	Describes the characteristics of individual symbols. Each symbol is an object in the SYMBOL attribute class.

- Attribute_name is the name of the attribute. To find a list of attribute names in your data server repository, see Querying attribute information.

- object is the object whose attribute value you want to query. The object is dependent on the value specified for Attribute_class.
  - If you specified SYSTEM for Attribute_class, specify DBDEFAULT for object, or leave the parameter blank.
  - If you specified USER for Attribute_class, specify a username for object.
  - If you specified SYMBOL for Attribute_class, specify a symbol name for object.

For example:

```
=LVATTRIBUTE("SYMBOL","ZGPNativeCurrency","Toronto")
```

Querying attribute data using the Function Arguments dialog

You can query attribute data by completing the function parameters in the Function Arguments dialog.

To query attribute data:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVATTRIBUTE. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Attribute_class	Specify the attribute class.
Attribute_name	Specify the name of the attribute.
<object>	Optionally, specify the object whose attribute value you want to query.

   For more information, see LVATTRIBUTE syntax.
4. Click OK. The query data appears in the worksheet.

LVCELL

Using the LVCELL function, you can query single cell values from the Longview data server repository. This function is useful when you want to populate your Microsoft Excel reports using direct references to the data server repository. These references can be refreshed to show the latest data server repository values.

Note: You can also run a data query from a worksheet cell that contains an LVCELL function. For more information, see Analyzing LVCELL functions.

You can use the LVCELL function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVCELL syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying cell values using the Function Arguments dialog.

LVCELL syntax

In the worksheet cell, type:

```
=LVCELL("Sym_dim1", "Sym_dim2", "Sym_dim3",…, "Sym_dimN")
```

where:

- Sym_dimN is the name of the symbol in dimension N that serves as a coordinate for the specific data server repository cell. Specify one symbol name per dimension.

The Longview data server repository may contain a maximum of sixteen dimensions. Include as many symbols as there are dimensions. For example, if there are six dimensions in the Longview data server repository, include six symbols.

Always type the symbols in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, type the name of an account for the first symbol. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

For example:

```
=LVCELL("TOTASSET","FINYEAR","CHICAGO","CUSD","DIM4SET")
```

Querying cell values using the Function Arguments dialog

You can query a single cell value from the Longview data server repository by completing the function parameters in the Function Arguments dialog.

To query a cell value:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVCELL. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Sym_dim1... Sym_dimN	Specify a symbol name for each dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.

   For more information, see LVCELL syntax .
4. Click OK. [Refresh required] appears in the worksheet cell.
5. Refresh the worksheet. For more information, see Refreshing data from cells containing Longview Functions. The query data appears in the worksheet.

LVCHILD

Using the LVCHILD function, you can query the name of the child of a parent symbol to a worksheet cell.

You can use the LVCHILD function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVCHILD syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying child symbols using the Function Arguments dialog.

LVCHILD syntax

In the worksheet cell, type:

```
=LVCHILD("Dim", "Sym_name", [number])
```

where:

- Dim is the name, description, or number of the dimension containing the child symbol. If you enter a dimension description, the parameter is not case-sensitive.

  Note: The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

- Sym_name is the name of the parent symbol.
- number is an optional parameter that represents the numeric position of the child symbol relative to the other child symbols at the same level in the hierarchy. If you do not specify a value for number, a default value of 1 is used.

For example:

```
=LVCHILD("TIMEPER","2014",1)
```

Querying child symbols using the Function Arguments dialog

You can query a child symbol by completing the function parameters in the Function Arguments dialog.

To query a child symbol:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVCHILD. The Function Arguments dialog opens.

3. Complete the following fields:

Field	Description
Dim	Specify the dimension containing the child symbol.
Sym_name	Specify the name of the parent symbol. You may use the Symbol Selector control or manually type in the symbol name.
<number>	Optionally, specify a number indicating the relative position of the child symbol.

   For more information, see LVCHILD syntax.

4. Click OK. The query data appears in the worksheet.

LVDESC

Using the LVDESC function, you can query the description of a symbol to a worksheet cell.

You can use the LVDESC function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVDESC syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying symbol descriptions using the Function Arguments dialog.

LVDESC syntax

In the worksheet cell, type:

```
=LVDESC("Dim", "Sym_name", ["language"])
```

where:

- Dim is the name, description, or number of the dimension containing the symbol. If you enter a dimension description, the parameter is not case-sensitive.

  Note: The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

- Sym_name is the name of the symbol.
- language is an optional parameter for the language to use. Specify one of the following values:

Value	Description
"EN"	The symbol description appears in English.
"ALT"	The symbol description appears in the alternate language as set for the system.

  If you do not specify a value for language, the symbol description appears in English.

For example:

```
=LVDESC("ACCOUNTS","IncStmt","EN")
```

Querying symbol descriptions using the Function Arguments dialog

You can query a symbol description by completing the function parameters in the Function Arguments dialog.

To query a symbol description:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVDESC. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension containing the symbol.
Sym_name	Specify the symbol name. You may use the Symbol Selector control or manually type in the symbol name.
<language>	Optionally, specify the language to use for the symbol description.

   For more information, see LVDESC syntax.

4. Click OK. The query data appears in the worksheet.

LVDIMDESC

Using the LVDIMDESC function, you can query the description of a dimension to a worksheet cell.

You can use the LVDIMDESC function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVDIMDESC syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying dimension descriptions using the Function Arguments dialog.

LVDIMDESC syntax

In the worksheet cell, type:

```
=LVDIMDESC(Dim)
```

where:

- Dim is the name or number of the dimension.

> **Note:** The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

For example:

```
=LVDIMDESC("ACCOUNTS")
```

Querying dimension descriptions using the Function Arguments dialog

You can query a dimension description by completing the function parameters in the Function Arguments dialog.

To query a dimension description:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVDIMDESC. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension to query.

   For more information, see LVDIMDESC syntax.

4. Click OK. The query data appears in the worksheet.

LVDIMNAME

Using the LVDIMNAME function, you can query the name of a dimension to a worksheet cell.

You can use the LVDIMNAME function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVDIMNAME syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying dimension names using the Function Arguments dialog.

LVDIMNAME syntax

In the worksheet cell, type:

```
=LVDIMNAME(Dim)
```

where:

- Dim is the dimension number. The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository.

  For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6.

For example:

```
=LVDIMNAME(1)
```

Querying dimension names using the Function Arguments dialog

You can query a dimension name by completing the function parameters in the Function Arguments dialog.

To query a dimension name:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVDIMNAME. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension number. LVDIMNAME syntax

   For more information, see LVDIMNAME syntax.

4. Click OK. The query data appears in the worksheet.

LVGROUP

Using the LVGROUP function, you can query the name of your current user group to a worksheet cell.

You can use the LVGROUP function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVGROUP syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying user group names using the Function Arguments dialog.

LVGROUP syntax

In the worksheet cell, type:

```
=LVGROUP()
```

Querying user group names using the Function Arguments dialog

You can query the name of your current user group by completing the function parameters in the Function Arguments dialog.

To query the current user group name:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVGROUP. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

LVGROUPDESC

Using the LVGROUPDESC function, you can query the description of your current user group to a worksheet cell.

You can use the LVGROUPDESC function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVGROUPDESC syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying user group descriptions using the Function Arguments dialog.

LVGROUPDESC syntax

In the worksheet cell, type:

```
=LVGROUPDESC()
```

Querying user group descriptions using the Function Arguments dialog

You can query the description of your current user group by completing the function parameters in the Function Arguments dialog.

To query the current user group description:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVGROUPDESC. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

LVIDENTIFIER

Using the LVIDENTIFIER function, you can query the Longview Identifier (LID) for the system to which you are currently connected to a worksheet cell.

You can use the LVIDENTIFIER function in the following ways:

Type the function syntax into a worksheet cell. For more information, see LVIDENTIFIER syntax.

Complete the function parameters in the Function Arguments dialog. For more information, see Querying the Longview Identifier using the Function Arguments dialog.

LVIDENTIFIER syntax

In the worksheet cell, type:

```
=LVIDENTIFIER()
```

Querying the Longview Identifier using the Function Arguments dialog

You can query the current Longview Identifier (LID) by completing the function parameters in the Function Arguments dialog.

To query the current Longview Identifier:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVIDENTIFIER. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

LVLOCK

Using the LVLOCK function, you can define a section of the Longview data server repository to lock. Before you can submit data, you must lock the section of the data server repository that you plan to submit data to. This prevents conflicts with other users who are working with the server data.

Locks that you define with LVLOCK functions are not applied to the data server repository until you select one of the Apply Locks options from the Submit button in the ribbon.

For more information on applying locks, see Submitting data using the Longview Add-In for Office.

Use the LVLOCK function in conjunction with the LVSUBMIT function in the data submission process. For more information, see LVSUBMIT.

You can use the LVLOCK function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVLOCK syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Defining data to be locked using the Function Arguments dialog.

LVLOCK syntax

In the worksheet cell, type:

```
=LVLOCK("SymSpecs_dim1", "SymSpecs_dim2", "SymSpecs_dim3",…, "SymSpecs_
dimN", "Comment")
```

where:

- SymSpecs_dimN is the symbol specification for the symbol or symbols in dimension N to include in the locked section of the Longview data server repository. Include at least one

symbol specification for each dimension in the Longview data server repository.

For each symbol specification, specify a symbol name and the level of symbols below that symbol in the hierarchy to include in the locked section of the Longview data server repository. You can use the following:

Symbol specification	Description
Specification	Meaning
symbol###	all leaf symbols under symbol
symbol#n	all symbols under symbol n levels down, including symbol
symbol##n	all parent symbols under symbol n levels down
symbol##+n	all parent symbols under symbol n levels down, including symbol

If you specify multiple symbol specifications for a dimension, they must be delimited by commas (,).

Always type the symbol specifications in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, enter an account for the first symbol specification. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

Comment is a parameter that specifies a brief description of the lock. If you do not specify a comment, the lock description is left blank.

For example:

```
=LVLOCK("TOTASSET#99","FINYEAR#99","CHICAGOT","CUSD","DIM4SET#99","Longview
Total Assets Lock")
```

Defining data to be locked using the Function Arguments dialog

You can define a section of the Longview data server repository to lock by completing the function parameters in the Function Arguments dialog.

To define data to be locked:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVLOCK. The Function Arguments dialog opens.
3. Complete the following fields, where N is the total number of dimensions in your system:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.

Field	Description
Sym_dim1... Sym_dimN	Specify at least one symbol specification for each dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.
Comment	Enter a brief description of the lock.

For more information, see LVLOCK syntax.

4. Click OK. LVLOCK: Ready to lock appears in the worksheet cell.
5. Proceed to apply the lock to the data server repository. For more information, see Submitting data using the Longview Add-In for Office.

LVPARENT

Using the LVPARENT function, you can query the name of a parent symbol to a worksheet cell.

You can use the LVPARENT function in the following ways:

1. Type the function syntax into a worksheet cell. For more information, see LVPARENT syntax.
2. Complete the function parameters in the Function Arguments dialog. For more information, see Querying parent symbols using the Function Arguments dialog.

LVPARENT syntax

In the worksheet cell, type:

=LVPARENT("Dim", "Sym_name", ["sym_root"])

where:

- Dim is the name, description, or number of the dimension containing the parent symbol. If you enter a dimension description, the parameter is not case sensitive.

  Note: The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

- Sym_name is the name of the child symbol.
- sym_root is an optional parameter for the name of the root symbol of the hierarchy containing the parent symbol. Specify a root symbol if the child symbol specified for Sym_name belongs to multiple parent symbols. If you do not specify a value for sym_root, the first parent symbol in the hierarchy is returned.

For example:

```
=LVPARENT("ENTITIES","TORONTO","ABCOPERATIONS")
```

Querying parent symbols using the Function Arguments dialog

You can query a parent symbol by completing the function parameters in the Function Arguments dialog.

To query a parent symbol:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVPARENT. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension containing the parent symbol.
Sym_name	Specify the name of the child symbol. You may use the Symbol Selector control or manually type in the symbol name.
<sym_root>	Optionally, specify the name of the root symbol of the hierarchy containing the parent symbol.

   For more information, see LVPARENT syntax.

4. Click OK. The query data appears in the worksheet.

LVREFRESHDATETIME

Using the LVREFRESHDATETIME function, you can query the date and time when the cell location was last refreshed.

You can use the LVREFRESHDATETIME function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVREFRESHDATETIME syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying the refresh date and time using the Function Arguments dialog.

LVREFRESHDATETIME syntax

In the worksheet cell, type:

```
=LVREFRESHDATETIME([long_format])
```

where:

- long_format is an optional parameter for the date and time format. Specify one of the following values:

Values	Description
TRUE	The refresh date and time appear in the long format, as specified in your system.
FALSE	The refresh date and time appear in the short format, as specified in your system.

If you do not specify a value for long_format, the refresh date and time appear in the short format, as specified in your system.

For example:

```
=LVREFRESHDATETIME(TRUE)
```

Querying the refresh date and time using the Function Arguments dialog

You can query the refresh date and time by completing the function parameters in the Function Arguments dialog.

To query the refresh date and time:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVREFRESHDATETIME. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
<long_format>	Optionally, specify the date and time format.

   For more information, see LVREFRESHDATETIME syntax.

4. Click OK. [Refresh required] appears in the worksheet cell.
5. Refresh the worksheet. For more information, see Refreshing data from cells containing Longview Functions. The query data appears in the worksheet.

LVSCELL

Using the LVSCELL function, you can query single schedule values from the Longview data server repository. This function is useful when you want to populate your Microsoft Excel reports using direct references to the data server repository. These references can be refreshed to show the latest data server repository values.

You can use the LVSCELL function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVSCell syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying schedule cell values using the Function Arguments dialog.

LVSCell syntax

In the worksheet cell, type:

```
=LVSCELL(“ScheduleName”, "Sym_dim1", "Sym_dim2", "Sym_dim3",…, "Sym_dimN",
“ESym_dimN”)
```

where:

- ScheduleName is the name of the schedule.
- Sym_dimN is the name of the symbol in dimension N that serves as a coordinate for the specific data server repository cell. Specify one symbol name per dimension.
- ESym_dimN is the name of the symbol in schedule dimension N that serves as a schedule coordinate for the specific data server repository cell. Specify one symbol name per schedule dimension.

The Longview data server repository may contain a maximum of sixteen dimensions and eight schedule dimensions. Include as many symbols as there are dimensions. For example, if there are six base dimensions and 1 schedule dimension in the schedule in the Longview data server repository, include six symbols and 1 schedule symbol.

Always type the symbols in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, type the name of an account for the first symbol.

For example:

```
=LVSCELL("ICStandard","11300", "A17_09_YTD", "E11120", "CONS_IC", "PRODUCTS_
TB", "DEPARTMENTS_TB","SCENARIOS_A001", "USD", "E13100")
```

Querying schedule cell values using the Function Arguments dialog

You can query a single schedule cell value from the Longview data server repository by completing the function parameters in the Function Arguments dialog.

To query a schedule cell value:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSCELL. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Schedule	Specify the name of the schedule.
Sym_dim1... Sym_dimN	Specify a symbol name for each dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.

Field	Description
Sym_schedule Dim1... Sym_ scheduleDimN	Specify a symbol name for each schedule dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.

For more information, see LVSCell syntax.

4. Click OK. [Refresh required] appears in the worksheet cell.
5. Refresh the worksheet. For more information, see Refreshing data from cells containing Longview Functions. The query data appears in the worksheet.

LVSESSION

Using the LVSESSION function, you can query various session properties for the system to which you are currently connected to a worksheet cell. You can use the LVSESSION function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVSESSION syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying Session Properties using the Function Arguments dialog.

LVSESSION syntax

In the worksheet cell, type:

```
=LVSESSION("property")
```

where:

- property is the property you wish to retrieve. Specify one of the following values:

Field	Description
Group	The current user group is returned.
Host	The current name of the host application server is returned.
HttpProtocol	The http protocol for the current system is returned.
Identifier	The current Identifier for the system is returned.
IsLongviewSession	Returns a boolean indicating whether the connection is a Longview connection to a Longview data source.
IsTidemarkSession	Returns a boolean indicating whether the connection is a Tidemark connection to a Tidemark data source.
IsValid	Returns a boolean indicating that the session object has been initialized.
Port	The current port for the application server is returned.
RDBMS	The current RDBMS for the system is returned.
SessionID	The current session id is returned.
UserID	The current user id is returned.
WebBridge	The web bridge for the current system is returned.
WebBridgePath	The web bridge path for the current system is returned.
WebServer	The web server for the current system is returned.

For example:

```
=LVSESSION("SessionID")
```

Querying Session Properties using the Function Arguments dialog

You can query the current Session Properties by completing the function parameters in the Function Arguments dialog.

To query the current Session Properties:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSESSION. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
property	Specify the property to retrieve.

   For more information, see LVSESSION syntax.

4. Click OK. The query data appears in the worksheet.

LVSUBMIT

Using the LVSUBMIT function, you can define a data value to be submitted to the Longview data server repository. Before you submit data, you must lock the section of the data server repository that you plan to submit data to. For more information, see LVLOCK.

Data values that you define with LVSUBMIT functions are not submitted to the data server repository until you select one of the Submit options from the Submit button in the ribbon. For more information, see Submitting data using the Longview Add-In for Office.

Keep in mind the following when using this function:

- You can submit one value at a time for each cell containing an LVSUBMIT function.
- You can submit data only to leaf cells or static parents in the Longview data server repository.
- You can submit base data only. You cannot submit schedule data.

Note: Longview recommends that you avoid using two or more LVSUBMIT functions that specify the exact same coordinates for a data server repository cell but different data values, in the same worksheet. If you use LVSUBMIT functions in this way, unexpected behavior can occur.

You can use the LVSUBMIT function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVSUBMIT syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Defining data to be submitted using the Function Arguments dialog.

LVSUBMIT syntax

In the worksheet cell, type:

```
=LVSUBMIT("Sym_dim1", "Sym_dim2", "Sym_dim3",…, "Sym_dimN", Value)
```

where:

- Sym_dimN is the name of the symbol in dimension N that serves as a coordinate for the specific data server repository cell. Specify only one symbol name per dimension.

The Longview data server repository may contain a maximum of sixteen dimensions. Include as many symbols as there are dimensions. For example, if there are six dimensions in the Longview data server repository, include six symbols.

Always type the symbols in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, type the name of an account for the first symbol. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

Value is the value to be submitted to the data server repository. If the value to submit is a character string, enclose it in double quotation marks. If the value to submit is numeric, do not enclose it in double quotation marks.

For example:

```
=LVSUBMIT("HEADCOUNT","FINYEAR","CHICAGO","CUSD","DIM4SET",100)
```

Defining data to be submitted using the Function Arguments dialog

You can define a data value to be submitted to the Longview data server repository by completing the function parameters in the Function Arguments dialog.

To define data to be submitted:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSUBMIT. The Function Arguments dialog opens.
3. Complete the following fields, where N is the total number of dimensions in your system:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Sym_dim1... Sym_dimN	Specify one symbol specification for each dimension in your system.
value	Enter the value to be submitted to the data server repository.

   For more information, see LVSUBMIT syntax.
4. Click OK. LVSUBMIT: Ready to submit (Value) appears in the worksheet cell, where Value is the value entered in the Value field.
5. Proceed to submit data from the LVSUBMIT cell to the data server. For more information, see Submitting data using the Longview Add-In for Office.

LVSYMBOLSELECTOR

Using the LVSYMBOLSELECTOR function, you can insert a Symbol Selector into any cell in a worksheet. Users can double-click a cell containing an LVSYMBOLSELECTOR function to open the Symbol Selector dialog, and select one or more symbols from the symbol hierarchy. The symbol names of the selected symbol or symbols appear in the worksheet cell.

Note: If multiple symbols are selected, the symbol names that appear in the cell are delimited with semicolons (;).

Note: Excel expects function parameters in the language of the Operating System. If your workbooks will be used in Excel with different Language settings, you may find it useful to use a non-zero value to represent TRUE and a zero to represent FALSE for function parameters of Boolean type.

A cell that contains an LVSYMBOLSELECTOR function is indicated by a yellow background.

For more information on the Symbol Selector, see Using the Symbol Selector.

You can use the LVSYMBOLSELECTOR function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVSYMBOLSELECTOR syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Inserting a Symbol Selector using the Function Arguments dialog.

LVSYMBOLSELECTOR syntax

In the worksheet cell, type:

```
=LVSYMBOLSELECTOR("Dim", "Sym_spec", "Initial_sym", Allow_leaf, "Allowable_
parents", Allow_readonly, Allow_multiple, "Attribute_filter")
```

where:

- Dim is the name or number of the dimension whose hierarchy you want to appear in the Symbol Selector dialog.

  Note: The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

- Sym_spec is the level of symbols to display in the Symbol Selector hierarchy, in relation to a specified symbol name.

  The following specifications are possible for Sym_spec:

Specification	Meaning
symbol###	all leaf symbols under symbol
symbol#n	all symbols under symbol n levels down, including symbol
symbol##n	all parent symbols under symbol n levels down
symbol##+n	all parent symbols under symbol n levels down, including symbol
symbol#*	all roots of symbol
symbol#-n	all symbols exactly n levels down from symbol
symbol##^n	all ancestors of symbol that are n levels above symbol, including symbol

Specification	Meaning
symbol##^n:root	all symbols above symbol n levels up, under root, including symbol, where root is the root symbol (e.g., BALSHEET##^2:TRIALBAL)

If you do not specify a value for Sym_spec, all symbols in the dimension specified for Dim display in the Symbol Selector hierarchy.

- Initial_sym is the name of the symbol that is automatically selected when the Symbol Selector dialog opens.

  The symbol name also appears in the cell that contains the LVSYMBOLSELECTOR function. If you do not specify a value for Initial_sym, [Double-click to select symbol] appears in the cell.

- Allow_leaf specifies whether users can select leaf symbols in the Symbol Selector dialog. Specify one of the following values:

Value	Description
TRUE	Users can select leaf symbols in the Symbol Selector dialog.
FALSE	Users cannot select leaf symbols in the Symbol Selector dialog.

If you do not specify a value for Allow_leaf, users cannot select leaf symbols in the Symbol Selector dialog.

- Allowable_parents specifies the parent symbols that users can select in the Symbol Selector dialog. Specify one of the following values:

Value	Description
ALL	Users can select all parent symbols in the Symbol Selector dialog.
NONE	Users cannot select any parent symbols in the Symbol Selector dialog.
STATIC	Users can select static parent symbols only in the Symbol Selector dialog.

- Allow_readonly specifies whether users can select read-only symbols in the Symbol Selector dialog. Specify one of the following values:

Value	Description
TRUE	Users can select read-only symbols in the Symbol Selector dialog.
FALSE	Users cannot select read-only symbols in the Symbol Selector dialog.

If you do not specify a value for Allow_readonly, users cannot select read-only symbols in the Symbol Selector dialog.

- Allow_multiple specifies whether users can select multiple symbols in the Symbol Selector dialog. Specify one of the following values:

Value	Description
TRUE	Users can select multiple symbols in the Symbol Selector dialog.

Value	Description
FALSE	Users can select only one symbol in the Symbol Selector dialog.

If you do not specify a value for Allow_multiple, users can select only one symbol in the Symbol Selector dialog.

- Attribute_filter specifies an attribute filter, or attribute filters. Attribute filters restrict the symbols that appear in the Symbol Selector hierarchy to symbols that meet the filter condition.

  You can specify up to two attribute filters linked by AND or OR, enclosed in double quotations marks, using the following syntax:

  ```
  FilterType{AttrName{Operation{Expression
  ```

  where:

Syntax	Description
FilterType	Specifies the method to use to search the hierarchy for symbols matching the filter criteria. Specify one of the following values: ■ ALL — To find only the symbols whose attributes match the filter criteria, with no descendants. ■ PARENT — Starting from the top of the hierarchy, to find the symbols whose attributes match the filter criteria, including all descendants. This only checks symbols that are parents to other symbols. ■ LEAF — Starting from the bottom of the hierarchy, to find the symbols whose attributes match the filter criteria, including all ancestors. This only checks symbols that are leaf symbols; that is, that have no descendants. ■ ROOT — Searching only the top of the hierarchy to find root symbols whose attributes match the filter criteria, including all descendants. If you specify two attribute filters, FilterType must be the same for both filters.
AttrName	The name of a Symbol attribute. You can use an attribute query to learn the names of all Symbol attributes in your system. For more information, see Querying attribute information.
Operation	Specifies the operation to filter the attributes by. Specify one of the following values: ■ EQ — Filters for attributes that are exactly equal to (exactly match) the string specified for Expression. ■ NE — Filters for attributes that are not equal to (do not exactly match) the string specified for Expression.

Syntax	Description
Expression	A string. If the expression contains spaces, enclose the expression in double quotation marks preceded with a backslash (\"expression with spaces\"). If the expression is a list, separate multiple items with a pipe (|). For Non-List Attributes, the filter behaves as follows: ■ AttrName{EQ{Expression — Matches only if the attribute is an exact match of the expression. ■ AttrName{NE{Expression — Matches if the attribute is not an exact match of the expression. For List Attributes, the filter behaves as follows: ■ AttrName{EQ{Expression — Matches if the attribute is an exact match of the expression, or is a list of values, any one of which exactly matches the expression. ■ AttrName{NE{Expression — Matches if the attribute is empty or a list of values, none of which exactly matches the expression.

If you specify two attribute filters, enclose each filter in parentheses, for example " (attributefilter1) AND (attributefilter2)".

If you don't include an attribute filter, enter two double quotation marks

("").

For example:

```
=LVSYMBOLSELECTOR
("ENTITIES","TENTITIES#99","CHICAGO",TRUE,"ALL",TRUE,FALSE,"ALL

{ZGPNativeCurrency{EQ{CUSD")

=LVSYMBOLSELECTOR("ACCOUNTS","TRIALBAL#99","CASHT",TRUE,"ALL",,,"")

=LVSYMBOLSELECTOR("ACCOUNTS",,,,"ALL",,,"")
```

After a symbol is selected for a cell that contains an LVSYMBOLSELECTOR function, that cell can be used as a cell location containing a symbol name for parameters in other Longview Functions that require a symbol name.

Consider the following example:

If an LVSYMBOLSELECTOR function, specified as follows:

```
=LVSYMBOLSELECTOR("ACCOUNTS",,,,"ALL",,FALSE,"")
```

...is used to select the symbol 11100 - Total cash in cell A1, then cell A1 can be used as a cell location containing a symbol name in the following Longview Function:

```
=LVCELL($A$1,"FINYEAR","CHICAGO","CUSD","DIM4SET")
```

For more information, see Longview Function syntax.

Note: This functionality applies to Symbol Selectors when one symbol is selected.

Inserting a Symbol Selector using the Function Arguments dialog

You can insert a Symbol Selector into a worksheet cell by completing the function parameters in the Function Arguments dialog.

To insert a Symbol Selector:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSYMBOLSELECTOR. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension.
<sym_spec>	Specify the symbol and the level of symbols to display in the Symbol Selector hierarchy. You may use the Symbol Spec Selector control or manually type in the symbol name.
<initial_sym>	Specify the name of the symbol that is automatically selected when the Symbol Selector dialog opens. You may use the Symbol Selector control or manually type in the symbol name.
<allow_leaf>	Specify whether users can select leaf symbols in the Symbol Selector dialog.
Allowable_parents	Specify the type of parent symbols that users can select in the Symbol Selector dialog.
<allow_readonly>	Specify whether users can select read-only symbols in the Symbol Selector dialog.
<allow_multiple>	Specify whether users can select multiple symbols in the Symbol Selector dialog.
<attribute_filter>	Specify an attribute filter, or attribute filters. Attribute filters restrict the symbols that appear in the Symbol Selector hierarchy to symbols that meet the filter condition.

   For more information, see LVSYMBOLSELECTOR syntax.

4. Click OK. The symbol selector appears in the worksheet cell, with a yellow background applied. If you specified a symbol for Initial_sym, that symbol name appears in the cell. Otherwise, [Double-click to select symbol] appears in the cell.

LVUSER

Using the LVUSER function, you can query your current username to a worksheet cell.

You can use the LVUSER function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVUSER syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying the current username using the Function Arguments dialog.

LVUSER syntax

In the worksheet cell, type:

```
=LVUSER()
```

Querying the current username using the Function Arguments dialog

You can query your current username by completing the function parameters in the Function Arguments dialog.

To query the current username:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVUSER. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

LVUSERDESC

Using the LVUSERDESC function, you can query your current user description to a worksheet cell.

You can use the LVUSERDESC function in the following ways:

- Type the function syntax into a worksheet cell. For more information, see LVUSERDESC syntax.
- Complete the function parameters in the Function Arguments dialog. For more information, see Querying the current user description using the Function Arguments dialog.

LVUSERDESC syntax

In the worksheet cell, type:

```
=LVUSERDESC()
```

Querying the current user description using the Function Arguments dialog

You can query your current user description by completing the function parameters in the Function Arguments dialog.

To query the current user description:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVUSERDESC. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

LVATTRIBUTE Function

Using the LVATTRIBUTE function, you can query Longview attribute data. You can use the LVATTRIBUTE function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVATTRIBUTE syntax

In the worksheet cell, type:

```
=LVATTRIBUTE("Attribute_class", "Attribute_name", ["object"])
```

where:

- Attribute_class is the attribute class. Specify one of the following values:

Value	Description
SYSTEM	Describes the entire Longview system at the highest level. Attributes of this class specify system-wide characteristics. There is only one object in the SYSTEM attribute class – the Longview application itself.
USER	Describes the attributes of a particular user. Each user is an object in the USER attribute class.
SYMBOL	Describes the characteristics of individual symbols. Each symbol is an object in the SYMBOL attribute class.

- Attribute_name is the name of the attribute. To find a list of attribute names in your data server repository, see Querying attribute information.
- object is the object whose attribute value you want to query. The object is dependent on the value specified for Attribute_class.
  - If you specified SYSTEM for Attribute_class, specify DBDEFAULT for object, or leave the parameter blank.
  - If you specified USER for Attribute_class, specify a username for object.
  - If you specified SYMBOL for Attribute_class, specify a symbol name for object.

For example:

```
=LVATTRIBUTE("SYMBOL","ZGPNativeCurrency","Toronto")
```

Querying attribute data using the Function Arguments dialog

You can query attribute data by completing the function parameters in the Function Arguments dialog.

To query attribute data:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVATTRIBUTE. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Attribute_class	Specify the attribute class.
Attribute_name	Specify the name of the attribute.
<object>	Optionally, specify the object whose attribute value you want to query.

4. Click OK. The query data appears in the worksheet.

LVCELL Function

Using the LVCELL function, you can query single cell values from the Longview data server repository. This function is useful when you want to populate your Microsoft Excel reports using direct references to the data server repository. These references can be refreshed to show the latest data server repository values.

Note: You can also run a data query from a worksheet cell that contains an LVCELL function. For more information, see Analyzing LVCELL functions.

You can use the LVCELL function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVCELL syntax

In the worksheet cell, type:

```
=LVCELL("Sym_dim1", "Sym_dim2", "Sym_dim3",…, "Sym_dimN",["include_
comments"])
```

where:

- Sym_dimN is the name of the symbol in dimension N that serves as a coordinate for the specific data server repository cell. Specify one symbol name per dimension.

The Longview data server repository may contain a maximum of sixteen dimensions. Include as many symbols as there are dimensions. For example, if there are six dimensions in the Longview data server repository, include six symbols.

Always type the symbols in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, type the name of an account for the first symbol. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

- 'include_comments' is an optional parameter that specifies whether comments are retrieved from the Longview database. Specify one of the following:

Value	Description
FALSE	No comments are retrieved from the database. This is the default value.
TRUE	Comments are retrieved from the database for the LVCELL intersection and displayed as an Excel cell comment. Comments refresh automatically when the LVCELL function is refreshed.

For example:

```
=LVCELL("TOTASSET","FINYEAR","CHICAGO","CUSD","DIM4SET",FALSE)
```

Querying cell values using the Function Arguments dialog

You can query a single cell value from the Longview data server repository by completing the function parameters in the Function Arguments dialog.

To query a cell value:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVCELL. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Sym_dim1... Sym_dimN	Specify a symbol name for each dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.

Field	Description
<include_ comments>	Optionally, specify whether comments are retrieved from the Longview database. ■ By default, it is FALSE, and LVCELL displays no comments. ■ If set to TRUE, LVCELL shows the associated comment as an Excel cell comment.

4. Click OK. [Refresh required] appears in the worksheet cell.
5. Refresh the worksheet. For more information, see Refreshing data from cells containing Longview Functions. The query data appears in the worksheet.

Viewing Audit Data

The 'View Audit Data' feature enables users to quickly access audit trail information for a specific data intersection directly from an LVCELL. By using the right-click context menu, users can launch the existing Audit Data application with relevant audit records automatically filtered to the selected intersection. This streamlines access to audit information and eliminates the need for manual searches.

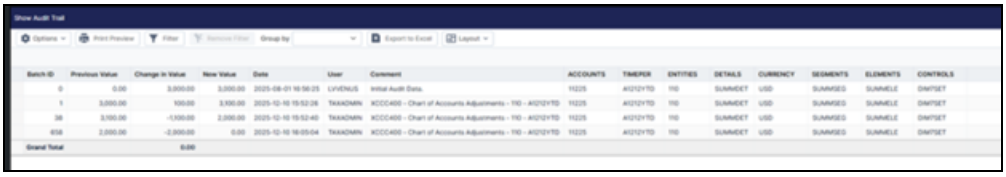
Prerequisites :

- Audit Trail must be enabled for the system.
- The user must have the appropriate permissions to access the Audit Data application.

Note: If Audit Trail is disabled, the View Audit Data option is hidden or unavailable in the LVCELL right-click context menu.

Steps to View Audit Data for an LVCELL:

1. Navigate to the worksheet or grid containing the target LVCELL.
2. Right-click the LVCELL for which you want to review audit information.
3. From the context menu, select **View Audit Data**.
4. The **Audit Data** application opens in a new view.
5. The audit data is automatically filtered to display records related to the selected LVCELL intersection, including dimensions such as account, time period, entity, and other applicable dimensions.



The **Audit Data** application displays historical changes for the selected intersection, including previous values, updated values, timestamps, and user information. The information displayed is based on system configuration and user permissions.

LVCHILD Function

Using the LVCHILD function, you can query the name of the child of a parent symbol to a worksheet cell.

You can use the LVCHILD function in the following ways:

- Type the function syntax into a worksheet cell. For more information,
- Complete the function parameters in the Function Arguments dialog.

LVCHILD syntax

In the worksheet cell, type:

```
=LVCHILD("Dim", "Sym_name", [number])
```

where:

- Dim is the name, description, or number of the dimension containing the child symbol. If you enter a dimension description, the parameter is not case-sensitive.

  Note: The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

- Sym_name is the name of the parent symbol.
- number is an optional parameter that represents the numeric position of the child symbol relative to the other child symbols at the same level in the hierarchy. If you do not specify a value for number, a default value of 1 is used.

For example:

```
=LVCHILD("TIMEPER","2014",1)
```

Querying child symbols using the Function Arguments dialog

You can query a child symbol by completing the function parameters in the Function Arguments dialog.

To query a child symbol:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVCHILD. The Function Arguments dialog opens.

3. Complete the following fields:

Field	Description
Dim	Specify the dimension containing the child symbol.
Sym_name	Specify the name of the parent symbol. You may use the Symbol Selector control or manually type in the symbol name.
<number>	Optionally, specify a number indicating the relative position of the child symbol.

4. Click OK. The query data appears in the worksheet.

LVCOMMENT Function

Using the LVCOMMENT function, you can query comments associated with specific data intersections from the Longview data server repository. This function is useful when you want to display detailed information that users have provided for a specific data intersection in your Microsoft Excel reports. These references can be refreshed to display the latest comments from the data server repository.

You can use the LVCOMMENT function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVCOMMENT syntax

In the worksheet cell, type:

```
=LVCOMMENT("Sym_dim1", "Sym_dim2", "Sym_dim3",..., "Sym_dimN")
```

where:

- Sym_dimN is the name of the symbol in dimension N that serves as a coordinate for the specific data server repository cell. Specify one symbol name per dimension.

The Longview data server repository may contain a maximum of sixteen dimensions. Include as many symbols as there are dimensions. For example, if there are six dimensions in the Longview data server repository, include six symbols.

Always type the symbols in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, type the name of an account for the first symbol. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

For example:

```
=LVCOMMENT("TOTASSET","FINYEAR","CHICAGO","CUSD","DIM4SET")
```

Querying Comments using the Function Arguments dialog

You can query a comment from the Longview data server repository by completing the function parameters in the Function Arguments dialog.

To query a comment:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVCOMMENT. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Sym_dim1... Sym_dimN	Specify a symbol name for each dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.

4. Click OK. [Refresh required] appears in the worksheet cell.
5. Refresh the worksheet. For more information, see Refreshing data from cells containing Longview Functions. The query data appears in the worksheet.

LVDESC Function

Using the LVDESC function, you can query the description of a symbol to a worksheet cell.

You can use the LVDESC function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVDESC syntax

In the worksheet cell, type:

```
=LVDESC("Dim", "Sym_name", ["language"])
```

where:

- Dim is the name, description, or number of the dimension containing the symbol. If you enter a dimension description, the parameter is not case-sensitive.

> **Note:** The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

- Sym_name is the name of the symbol.
- language is an optional parameter for the language to use. Specify one of the following values:

Value	Description
"EN"	The symbol description appears in English.
"ALT"	The symbol description appears in the alternate language as set for the system.

If you do not specify a value for language, the symbol description appears in English.

For example:

```
=LVDESC("ACCOUNTS","IncStmt","EN")
```

Querying symbol descriptions using the Function Arguments dialog

You can query a symbol description by completing the function parameters in the Function Arguments dialog.

To query a symbol description:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVDESC. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension containing the symbol.
Sym_name	Specify the symbol name. You may use the Symbol Selector control or manually type in the symbol name.
<language>	Optionally, specify the language to use for the symbol description.

4. Click OK. The query data appears in the worksheet.

LVDIMDESC Function

Using the LVDIMDESC function, you can query the description of a dimension to a worksheet cell.

You can use the LVDIMDESC function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVDIMDESC syntax

In the worksheet cell, type:

```
=LVDIMDESC(Dim)
```

where:

- Dim is the name or number of the dimension.

> **Note:** The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

For example:

```
=LVDIMDESC("ACCOUNTS")
```

Querying dimension descriptions using the Function Arguments dialog

You can query a dimension description by completing the function parameters in the Function Arguments dialog.

To query a dimension description:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVDIMDESC. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension to query.

4. Click OK. The query data appears in the worksheet.

LVDIMNAME Function

Using the LVDIMNAME function, you can query the name of a dimension to a worksheet cell.

You can use the LVDIMNAME function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVDIMNAME syntax

In the worksheet cell, type:

```
=LVDIMNAME(Dim)
```

where:

- Dim is the dimension number. The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository.

  For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6.

For example:

```
=LVDIMNAME(1)
```

Querying dimension names using the Function Arguments dialog

You can query a dimension name by completing the function parameters in the Function Arguments dialog.

To query a dimension name:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVDIMNAME. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension number. LVDIMNAME syntax

4. Click OK. The query data appears in the worksheet.

LVGROUP Function

Using the LVGROUP function, you can query the name of your current user group to a worksheet cell.

You can use the LVGROUP function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVGROUP syntax

In the worksheet cell, type:

```
=LVGROUP()
```

Querying user group names using the Function Arguments dialog

You can query the name of your current user group by completing the function parameters in the Function Arguments dialog.

To query the current user group name:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVGROUP. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

LVGROUPDESC Function

Using the LVGROUPDESC function, you can query the description of your current user group to a worksheet cell.

You can use the LVGROUPDESC function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVGROUPDESC syntax

In the worksheet cell, type:

```
=LVGROUPDESC()
```

Querying user group descriptions using the Function Arguments dialog

You can query the description of your current user group by completing the function parameters in the Function Arguments dialog.

To query the current user group description:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVGROUPDESC. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

LVIDENTIFIER Function

Using the LVIDENTIFIER function, you can query the Longview Identifier (LID) for the system to which you are currently connected to a worksheet cell.

You can use the LVIDENTIFIER function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVIDENTIFIER syntax

In the worksheet cell, type:

```
=LVIDENTIFIER()
```

Querying the Longview Identifier using the Function Arguments dialog

You can query the current Longview Identifier (LID) by completing the function parameters in the Function Arguments dialog.

To query the current Longview Identifier:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVIDENTIFIER. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

LVLOCK Function

Using the LVLOCK function, you can define a section of the Longview data server repository to lock. Before you can submit data, you must lock the section of the data server repository that you plan to submit data to. This prevents conflicts with other users who are working with the server data.

Locks that you define with LVLOCK functions are not applied to the data server repository until you select one of the Apply Locks options from the Submit button in the ribbon.

For more information on applying locks, see Submitting data using the Longview Add-In for Office.

Use the LVLOCK function in conjunction with the LVSUBMIT function in the data submission process. For more information, see LVSUBMIT.

You can use the LVLOCK function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVLOCK syntax

In the worksheet cell, type:

```
=LVLOCK("SymSpecs_dim1", "SymSpecs_dim2", "SymSpecs_dim3",…, "SymSpecs_
dimN", "Comment")
```

where:

- SymSpecs_dimN is the symbol specification for the symbol or symbols in dimension N to include in the locked section of the Longview data server repository. Include at least one symbol specification for each dimension in the Longview data server repository.

For each symbol specification, specify a symbol name and the level of symbols below that symbol in the hierarchy to include in the locked section of the Longview data server repository. You can use the following:

Symbol specification	Description
Specification	Meaning
symbol###	all leaf symbols under symbol
symbol#n	all symbols under symbol n levels down, including symbol
symbol##n	all parent symbols under symbol n levels down
symbol##+n	all parent symbols under symbol n levels down, including symbol

If you specify multiple symbol specifications for a dimension, they must be delimited by commas (,).

Always type the symbol specifications in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, enter an account for the first symbol specification. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

Comment is a parameter that specifies a brief description of the lock. If you do not specify a comment, the lock description is left blank.

For example:

```
=LVLOCK("TOTASSET#99","FINYEAR#99","CHICAGOT","CUSD","DIM4SET#99","Longview
Total Assets Lock")
```

Defining data to be locked using the Function Arguments dialog

You can define a section of the Longview data server repository to lock by completing the function parameters in the Function Arguments dialog.

To define data to be locked:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVLOCK. The Function Arguments dialog opens.
3. Complete the following fields, where N is the total number of dimensions in your system:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Sym_dim1... Sym_dimN	Specify at least one symbol specification for each dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.
Comment	Enter a brief description of the lock.

4. Click OK. LVLOCK: Ready to lock appears in the worksheet cell.
5. Proceed to apply the lock to the data server repository. For more information, see Submitting data using the Longview Add-In for Office.

LVMIDESC Function

Using the LVDIMDESC function, you can query the description of a dimension to a worksheet cell.

You can use the LVDIMDESC function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVDIMDESC syntax

In the worksheet cell, type:

```
=LVDIMDESC(Dim)
```

where:

- Dim is the name or number of the dimension.

> **Note:** The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

For example:

```
=LVDIMDESC("ACCOUNTS")
```

Querying dimension descriptions using the Function Arguments dialog

You can query a dimension description by completing the function parameters in the Function Arguments dialog.

To query a dimension description:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVDIMDESC. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension to query.

4. Click OK. The query data appears in the worksheet.

LVPARENT Function

Using the LVPARENT function, you can query the name of a parent symbol to a worksheet cell.

You can use the LVPARENT function in the following ways:

1. Type the function syntax into a worksheet cell.
2. Complete the function parameters in the Function Arguments dialog.

LVPARENT syntax

In the worksheet cell, type:

=LVPARENT("Dim", "Sym_name", ["sym_root"])

where:

- Dim is the name, description, or number of the dimension containing the parent symbol. If you enter a dimension description, the parameter is not case sensitive.

  Note: The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

- Sym_name is the name of the child symbol.
- sym_root is an optional parameter for the name of the root symbol of the hierarchy containing the parent symbol. Specify a root symbol if the child symbol specified for Sym_name belongs to multiple parent symbols. If you do not specify a value for sym_root, the first parent symbol in the hierarchy is returned.

For example:

```
=LVPARENT("ENTITIES","TORONTO","ABCOPERATIONS")
```

Querying parent symbols using the Function Arguments dialog

You can query a parent symbol by completing the function parameters in the Function Arguments dialog.

To query a parent symbol:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVPARENT. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension containing the parent symbol.
Sym_name	Specify the name of the child symbol. You may use the Symbol Selector control or manually type in the symbol name.
<sym_root>	Optionally, specify the name of the root symbol of the hierarchy containing the parent symbol.

4. Click OK. The query data appears in the worksheet.

LVREFRESHDATETIME Function

Using the LVREFRESHDATETIME function, you can query the date and time when the cell location was last refreshed.

You can use the LVREFRESHDATETIME function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVREFRESHDATETIME syntax

In the worksheet cell, type:

```
=LVREFRESHDATETIME([long_format])
```

where:

- long_format is an optional parameter for the date and time format. Specify one of the following values:

Values	Description
TRUE	The refresh date and time appear in the long format, as specified in your system.
FALSE	The refresh date and time appear in the short format, as specified in your system.

If you do not specify a value for long_format, the refresh date and time appear in the short format, as specified in your system.

For example:

```
=LVREFRESHDATETIME(TRUE)
```

Querying the refresh date and time using the Function Arguments dialog

You can query the refresh date and time by completing the function parameters in the Function Arguments dialog.

To query the refresh date and time:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVREFRESHDATETIME. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
<long_format>	Optionally, specify the date and time format.

4. Click OK. [Refresh required] appears in the worksheet cell.
5. Refresh the worksheet. For more information, see Refreshing data from cells containing Longview Functions. The query data appears in the worksheet.

LVSCELL Function

Using the LVSCELL function, you can query single schedule values from the Longview data server repository. This function is useful when you want to populate your Microsoft Excel reports using direct references to the data server repository. These references can be refreshed to show the latest data server repository values.

You can use the LVSCELL function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVSCell syntax

In the worksheet cell, type:

```
=LVSCELL(“ScheduleName”, "Sym_dim1", "Sym_dim2", "Sym_dim3",…, "Sym_dimN",
“ESym_dimN”)
```

where:

- ScheduleName is the name of the schedule.
- Sym_dimN is the name of the symbol in dimension N that serves as a coordinate for the specific data server repository cell. Specify one symbol name per dimension.
- ESym_dimN is the name of the symbol in schedule dimension N that serves as a schedule coordinate for the specific data server repository cell. Specify one symbol name per schedule dimension.

The Longview data server repository may contain a maximum of sixteen dimensions and eight schedule dimensions. Include as many symbols as there are dimensions. For example, if there are six base dimensions and 1 schedule dimension in the schedule in the Longview data server repository, include six symbols and 1 schedule symbol.

Always type the symbols in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, type the name of an account for the first symbol.

For example:

```
=LVSCELL("ICStandard","11300", "A17_09_YTD", "E11120", "CONS_IC", "PRODUCTS_
TB", "DEPARTMENTS_TB","SCENARIOS_A001", "USD", "E13100")
```

Querying schedule cell values using the Function Arguments dialog

You can query a single schedule cell value from the Longview data server repository by completing the function parameters in the Function Arguments dialog.

To query a schedule cell value:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSCELL. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Schedule	Specify the name of the schedule.
Sym_dim1... Sym_dimN	Specify a symbol name for each dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.

Field	Description
Sym_schedule Dim1... Sym_ scheduleDimN	Specify a symbol name for each schedule dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.

4. Click OK. [Refresh required] appears in the worksheet cell.
5. Refresh the worksheet. For more information, see Refreshing data from cells containing Longview Functions. The query data appears in the worksheet.

LVSESSION Function

Using the LVSESSION function, you can query various session properties for the system to which you are currently connected to a worksheet cell. You can use the LVSESSION function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVSESSION syntax

In the worksheet cell, type:

```
=LVSESSION(“property”)
```

where:

- property is the property you wish to retrieve. Specify one of the following values:

Field	Description
Group	The current user group is returned.
Host	The current name of the host application server is returned.
HttpProtocol	The http protocol for the current system is returned.
Identifier	The current Identifier for the system is returned.
IsLongviewSession	Returns a boolean indicating whether the connection is a Longview connection to a Longview data source.
IsTidemarkSession	Returns a boolean indicating whether the connection is a Tidemark connection to a Tidemark data source.
IsValid	Returns a boolean indicating that the session object has been initialized.
Port	The current port for the application server is returned.
RDBMS	The current RDBMS for the system is returned.
SessionID	The current session id is returned.
UserID	The current user id is returned.
WebBridge	The web bridge for the current system is returned.
WebBridgePath	The web bridge path for the current system is returned.
WebServer	The web server for the current system is returned.

For example:

```
=LVSESSION("SessionID")
```

Querying Session Properties using the Function Arguments dialog

You can query the current Session Properties by completing the function parameters in the Function Arguments dialog.

To query the current Session Properties:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSESSION. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
property	Specify the property to retrieve.

4. Click OK. The query data appears in the worksheet.

LVSLOCK Function

Using the LVSLOCK function, you can lock schedule-based data for a defined section of the Longview Data Server repository. Before you can submit data, you must lock the section of the data server repository that you plan to submit data to. This prevents conflicts with other users who are working with the server data.

Locks that you define with LVSLOCK functions are not applied to the data server repository until you select one of the Apply Locks options from the Submit button in the ribbon.

For more information on applying locks, see Submitting data using the Longview Add-In for Office.

Use the LVSLOCK function in conjunction with the LVSSUBMIT function in the data submission process. For more information, see LVSSUBMIT.

You can use the LVSLOCK function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVSLOCK syntax

In the worksheet cell, type:

```
=LVSLOCK(“Schedule”, "SymSpecs_dim1", "SymSpecs_dim2", "SymSpecs_dim3",…,
"SymSpecs_dimN", ["comment"])
```

where:

- Schedule is the name of the schedule.
- SymSpecs_dimN is the symbol specification for the symbol or symbols in dimension N to include in the locked section of the Longview data server repository. Include at least one symbol specification for each dimension in the Longview data server repository.

For each symbol specification, specify a symbol name and the level of symbols below that symbol in the hierarchy to include in the locked section of the Longview data server repository. You can use the following:

Symbol specification	Description
Specification	Meaning
symbol###	all leaf symbols under symbol
symbol#n	all symbols under symbol n levels down, including symbol
symbol##n	all parent symbols under symbol n levels down
symbol##+n	all parent symbols under symbol n levels down, including symbol

If you specify multiple symbol specifications for a dimension, they must be delimited by commas (,).

Always type the symbol specifications in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server

repository, enter an account for the first symbol specification. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

- Comment is an optional parameter that specifies a brief description of the lock. If the comment field is left blank, the system automatically generates a default lock description using the workbook and schedule names.

For example:

```
=LVSLOCK
("DataAttributes","TOTASSET#99","FINYEAR#99","CHICAGOT","CUSD","DIM4SET#99",
"Lock for Schedule DataAttributes")
```

Defining data to be locked using the Function Arguments dialog

You can define a section of the Longview data server repository to lock schedule-based data by completing the function parameters in the Function Arguments dialog.

To define data to be locked:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSLOCK. The Function Arguments dialog opens.
3. Complete the following fields, where N is the total number of dimensions in your system:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Schedule	Specify the name of the schedule.
Sym_dim1... Sym_dimN	Specify at least one symbol specification for each dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.
<comment>	Enter a brief description of the lock. ■ If the comment field is left blank, a default lock description is generated using the workbook and schedule names.

4. Click OK. LVSLOCK: Ready to create Schedule Lock appears in the worksheet cell.
5. Proceed to apply the lock to the data server repository.Then,the status changes to created Schedule lock. For more information, see Submitting data using the Longview Add-In for Office.

LVSSUBMIT Function

Using the LVSSUBMIT function, you can define a schedule data value to be submitted to the Longview data server repository. Before you submit schedule data, you must lock the section of the data server repository that you plan to submit schedule based data to. For more information, see LVSLOCK.

Data values that you define with LVSSUBMIT functions are not submitted to the data server repository until you select one of the Submit options from the Submit button in the ribbon. For more information, see Submitting data using the Longview Add-In for Office.

Keep in mind the following when using this function:

- You can submit one value at a time for each cell containing an LVSSUBMIT function.
- You can submit data only to leaf cells or static parents in the Longview data server repository.

Note: Longview recommends that you avoid using two or more LVSSUBMIT functions that specify the exact same coordinates for a data server repository cell but different data values, in the same worksheet. If you use LVSSUBMIT functions in this way, unexpected behavior can occur.

You can use the LVSSUBMIT function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVSSUBMIT syntax

In the worksheet cell, type:

```
=LVSSUBMIT("Schedule","Sym_dim1","Sym_dim2","Sym_dim3",…,"Sym_dimN","Value")
```

where:

- Schedule is the name of the schedule.
- Sym_dimN is the name of the symbol in dimension N that serves as a coordinate for the specific data server repository cell. Specify only one symbol name per dimension.

The Longview data server repository may contain a maximum of sixteen dimensions. Include as many symbols as there are dimensions. For example, if there are six dimensions in the Longview data server repository, include six symbols.

Always type the symbols in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, type the name of an account for the first symbol. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

- Value is the value to be submitted to the data server repository. If the value to submit is a character string, enclose it in double quotation marks. If the value to submit is numeric, do not enclose it in double quotation marks.

For example:

```
LVSSUBMIT("DataAttributes","A11101","A1001YTD","E0100","DETAILS_
Default","USD","SEGMENTS_Default","ELEMENTS_Default","CONTROLS_
Default","DataAttributeFlag",1000)
```

Defining data to be submitted using the Function Arguments dialog

You can define a data value to be submitted to the Longview data server repository by completing the function parameters in the Function Arguments dialog.

To define data to be submitted:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSSUBMIT. The Function Arguments dialog opens.
3. Complete the following fields, where N is the total number of dimensions in your system:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Schedule	Specify the name of the schedule.
Sym_dim1... Sym_dimN	Specify one symbol specification for each dimension in your system. You may use the Symbol Selector control or manually type in each symbol name.
Value	Enter the value to be submitted to the data server repository.

4. Click OK. LVSSUBMIT: Ready to submit (Value) appears in the worksheet cell, where Value is the value entered in the Value field.
5. Proceed to submit data from the LVSSUBMIT cell to the data server. For more information, see Submitting data using the Longview Add-In for Office.

LVSUBMIT Function

Using the LVSUBMIT function, you can define a data value to be submitted to the Longview data server repository. Before you submit data, you must lock the section of the data server repository that you plan to submit data to. For more information, see LVLOCK.

Data values that you define with LVSUBMIT functions are not submitted to the data server repository until you select one of the Submit options from the Submit button in the ribbon. For more information, see Submitting data using the Longview Add-In for Office.

Keep in mind the following when using this function:

- You can submit one value at a time for each cell containing an LVSUBMIT function.
- You can submit data only to leaf cells or static parents in the Longview data server repository.

Note: Longview recommends that you avoid using two or more LVSUBMIT functions that specify the exact same coordinates for a data server repository cell but different data values, in the same worksheet. If you use LVSUBMIT functions in this way, unexpected behavior can occur.

You can use the LVSUBMIT function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVSUBMIT syntax

In the worksheet cell, type:

```
=LVSUBMIT("Sym_dim1", "Sym_dim2", "Sym_dim3",…, "Sym_dimN", Value)
```

where:

- Sym_dimN is the name of the symbol in dimension N that serves as a coordinate for the specific data server repository cell. Specify only one symbol name per dimension.

The Longview data server repository may contain a maximum of sixteen dimensions. Include as many symbols as there are dimensions. For example, if there are six dimensions in the Longview data server repository, include six symbols.

Always type the symbols in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, type the name of an account for the first symbol. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

Value is the value to be submitted to the data server repository. If the value to submit is a character string, enclose it in double quotation marks. If the value to submit is numeric, do not enclose it in double quotation marks.

For example:

```
=LVSUBMIT("HEADCOUNT","FINYEAR","CHICAGO","CUSD","DIM4SET",100)
```

Defining data to be submitted using the Function Arguments dialog

You can define a data value to be submitted to the Longview data server repository by completing the function parameters in the Function Arguments dialog.

To define data to be submitted:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSUBMIT. The Function Arguments dialog opens.
3. Complete the following fields, where N is the total number of dimensions in your system:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Sym_dim1... Sym_dimN	Specify one symbol specification for each dimension in your system.
value	Enter the value to be submitted to the data server repository.

4. Click OK. LVSUBMIT: Ready to submit (Value) appears in the worksheet cell, where Value is the value entered in the Value field.
5. Proceed to submit data from the LVSUBMIT cell to the data server. For more information, see Submitting data using the Longview Add-In for Office.

LVSUBMITCOMMENT Function

Using the LVSUBMITComment function, you can define a comments to be submitted to the Longview data server repository. Before you submit comments to base data cells, you must lock the section of the data server repository that you plan to submit data to. For more information, see LVLOCK.

Comments that you define with LVSUBMITCOMMENT function are not submitted to the data server repository until you select one of the Submit options from the Submit button in the ribbon. For more information, see Submitting data using the Longview Add-In for Office.

Keep in mind the following when using this function:

- You can submit one Comment at a time for each cell containing an LVSUBMITCOMMENT function.

Note: Longview recommends that you avoid using two or more LVSUBMITCOMMENT functions that specify the exact same coordinates for a data server repository cell but different data values, in the same worksheet. If you use LVSUBMITCOMMENT functions in this way, unexpected behavior can occur.

You can use the LVSUBMITCOMMENT function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVSUBMITCOMMENT syntax

In the worksheet cell, type:

```
=LVSUBMITCOMMENT("Sym_dim1", "Sym_dim2", "Sym_dim3",..., "Sym_dimN",
"Comment")
```

where:

- Sym_dimN is the name of the symbol in dimension N that serves as a coordinate for the specific data server repository cell. Specify only one symbol name per dimension.

The Longview data server repository may contain a maximum of sixteen dimensions. Include as many symbols as there are dimensions. For example, if there are six dimensions in the Longview data server repository, include six symbols.

Always type the symbols in the same sequence as the dimensions in the Longview data server repository. For example, if ACCOUNTS is the first dimension in the Longview data server repository, type the name of an account for the first symbol. To find out the order of the dimensions in the Longview data server repository, use the LVDIMNAME function. For more information, see LVDIMNAME.

- Comment specifies the text of the comment to be submitted.

For example:

```
=LVSUBMITCOMMENT("HEADCOUNT","FINYEAR","CHICAGO","CUSD","DIM4SET","Longview
comment")
```

Defining data to be submitted using the Function Arguments dialog

You can define a comment to be submitted to the Longview data server repository by completing the function parameters in the Function Arguments dialog.

To define data to be submitted:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSUBMITCOMMENT. The Function Arguments dialog opens.
3. Complete the following fields, where N is the total number of dimensions in your system:

Field	Description
Defaults	If you have user defaults defined in the UGPDNInputDefault or UGPDNQueryDefault attributes, you may select from this drop list to quickly populate the symbol fields below.
Sym_dim1... Sym_dimN	Specify one symbol specification for each dimension in your system.
Comment	Enter the comment to be submitted to the data server repository.

4. Click OK. LVSUBMITCOMMENT: Ready to submit (Comment) appears in the worksheet cell, where Comment is the text entered in the **Comment** field.

5. Proceed to submit data from the LVSUBMITCOMMENT cell to the data server.For more information, see Submitting data using the Longview Add-In for Office.

LVSYMBOLSELECTOR Function

Using the LVSYMBOLSELECTOR function, you can insert a Symbol Selector into any cell in a worksheet. Users can double-click a cell containing an LVSYMBOLSELECTOR function to open the Symbol Selector dialog, and select one or more symbols from the symbol hierarchy. The symbol names of the selected symbol or symbols appear in the worksheet cell.

Note: If multiple symbols are selected, the symbol names that appear in the cell are delimited with semicolons (;).

Note: Excel expects function parameters in the language of the Operating System. If your workbooks will be used in Excel with different Language settings, you may find it useful to use a non-zero value to represent TRUE and a zero to represent FALSE for function parameters of Boolean type.

A cell that contains an LVSYMBOLSELECTOR function is indicated by a yellow background.

For more information on the Symbol Selector, see Using the Symbol Selector.

You can use the LVSYMBOLSELECTOR function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVSYMBOLSELECTOR syntax

In the worksheet cell, type:

```
=LVSYMBOLSELECTOR("Dim", "Sym_spec", "Initial_sym", Allow_leaf, "Allowable_
parents", Allow_readonly, Allow_multiple, "Attribute_filter")
```

where:

- Dim is the name or number of the dimension whose hierarchy you want to appear in the Symbol Selector dialog.

  Note: The dimension number is an integer corresponding to the order of the dimension as it appears in the Longview data server repository. For example, if you have six dimensions in your system, the dimensions are numbered from 1 to 6. If you specify a dimension number for Dim, do not enclose it in double quotation marks.

- Sym_spec is the level of symbols to display in the Symbol Selector hierarchy, in relation to a specified symbol name.

  The following specifications are possible for Sym_spec:

Specification	Meaning
symbol###	all leaf symbols under symbol
symbol#n	all symbols under symbol n levels down, including symbol
symbol##n	all parent symbols under symbol n levels down
symbol##+n	all parent symbols under symbol n levels down, including symbol
symbol#*	all roots of symbol
symbol#-n	all symbols exactly n levels down from symbol
symbol##^n	all ancestors of symbol that are n levels above symbol, including symbol
symbol##^n:root	all symbols above symbol n levels up, under root, including symbol, where root is the root symbol (e.g., BALSHEET##^2:TRIALBAL)

If you do not specify a value for Sym_spec, all symbols in the dimension specified for Dim display in the Symbol Selector hierarchy.

- Initial_sym is the name of the symbol that is automatically selected when the Symbol Selector dialog opens.

  The symbol name also appears in the cell that contains the LVSYMBOLSELECTOR function. If you do not specify a value for Initial_sym, [Double-click to select symbol] appears in the cell.

- Allow_leaf specifies whether users can select leaf symbols in the Symbol Selector dialog. Specify one of the following values:

Value	Description
TRUE	Users can select leaf symbols in the Symbol Selector dialog.
FALSE	Users cannot select leaf symbols in the Symbol Selector dialog.

If you do not specify a value for Allow_leaf, users cannot select leaf symbols in the Symbol Selector dialog.

- Allowable_parents specifies the parent symbols that users can select in the Symbol Selector dialog. Specify one of the following values:

Value	Description
ALL	Users can select all parent symbols in the Symbol Selector dialog.
NONE	Users cannot select any parent symbols in the Symbol Selector dialog.
STATIC	Users can select static parent symbols only in the Symbol Selector dialog.

- Allow_readonly specifies whether users can select read-only symbols in the Symbol Selector dialog. Specify one of the following values:

Value	Description
TRUE	Users can select read-only symbols in the Symbol Selector dialog.
FALSE	Users cannot select read-only symbols in the Symbol Selector dialog.

If you do not specify a value for Allow_readonly, users cannot select read-only symbols in the Symbol Selector dialog.

- Allow_multiple specifies whether users can select multiple symbols in the Symbol Selector dialog. Specify one of the following values:

Value	Description
TRUE	Users can select multiple symbols in the Symbol Selector dialog.
FALSE	Users can select only one symbol in the Symbol Selector dialog.

If you do not specify a value for Allow_multiple, users can select only one symbol in the Symbol Selector dialog.

- Attribute_filter specifies an attribute filter, or attribute filters. Attribute filters restrict the symbols that appear in the Symbol Selector hierarchy to symbols that meet the filter condition.

  You can specify up to two attribute filters linked by AND or OR, enclosed in double quotations marks, using the following syntax:

```
FilterType{AttrName{Operation{Expression
```

  where:

Syntax	Description
FilterType	Specifies the method to use to search the hierarchy for symbols matching the filter criteria. Specify one of the following values: ■ ALL — To find only the symbols whose attributes match the filter criteria, with no descendants. ■ PARENT — Starting from the top of the hierarchy, to find the symbols whose attributes match the filter criteria, including all descendants. This only checks symbols that are parents to other symbols. ■ LEAF — Starting from the bottom of the hierarchy, to find the symbols whose attributes match the filter criteria, including all ancestors. This only checks symbols that are leaf symbols; that is, that have no descendants. ■ ROOT — Searching only the top of the hierarchy to find root symbols whose attributes match the filter criteria, including all descendants. If you specify two attribute filters, FilterType must be the same for both filters.
AttrName	The name of a Symbol attribute. You can use an attribute query to learn the names of all Symbol attributes in your system. For more information, see Querying attribute information.

Syntax	Description
Operation	Specifies the operation to filter the attributes by. Specify one of the following values: ■ EQ — Filters for attributes that are exactly equal to (exactly match) the string specified for Expression. ■ NE — Filters for attributes that are not equal to (do not exactly match) the string specified for Expression.
Expression	A string. If the expression contains spaces, enclose the expression in double quotation marks preceded with a backslash (\"expression with spaces\"). If the expression is a list, separate multiple items with a pipe (|). For Non-List Attributes, the filter behaves as follows: ■ AttrName{EQ{Expression — Matches only if the attribute is an exact match of the expression. ■ AttrName{NE{Expression — Matches if the attribute is not an exact match of the expression. For List Attributes, the filter behaves as follows: ■ AttrName{EQ{Expression — Matches if the attribute is an exact match of the expression, or is a list of values, any one of which exactly matches the expression. ■ AttrName{NE{Expression — Matches if the attribute is empty or a list of values, none of which exactly matches the expression.

If you specify two attribute filters, enclose each filter in parentheses, for example " (attributefilter1) AND (attributefilter2)".

If you don't include an attribute filter, enter two double quotation marks

("").

For example:

```
=LVSYMBOLSELECTOR
("ENTITIES","TENTITIES#99","CHICAGO",TRUE,"ALL",TRUE,FALSE,"ALL

{ZGPNativeCurrency{EQ{CUSD")

=LVSYMBOLSELECTOR("ACCOUNTS","TRIALBAL#99","CASHT",TRUE,"ALL",,,"")

=LVSYMBOLSELECTOR("ACCOUNTS",,,,"ALL",,,"")
```

After a symbol is selected for a cell that contains an LVSYMBOLSELECTOR function, that cell can be used as a cell location containing a symbol name for parameters in other Longview Functions that require a symbol name.

Consider the following example:

If an LVSYMBOLSELECTOR function, specified as follows:

```
=LVSYMBOLSELECTOR("ACCOUNTS",,,,"ALL",,FALSE,"")
```

...is used to select the symbol 11100 - Total cash in cell A1, then cell A1 can be used as a cell location containing a symbol name in the following Longview Function:

```
=LVCELL($A$1,"FINYEAR","CHICAGO","CUSD","DIM4SET")
```

For more information, see Longview Function syntax.

Note: This functionality applies to Symbol Selectors when one symbol is selected.

Inserting a Symbol Selector using the Function Arguments dialog

You can insert a Symbol Selector into a worksheet cell by completing the function parameters in the Function Arguments dialog.

To insert a Symbol Selector:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVSYMBOLSELECTOR. The Function Arguments dialog opens.
3. Complete the following fields:

Field	Description
Dim	Specify the dimension.
<sym_ spec>	Specify the symbol and the level of symbols to display in the Symbol Selector hierarchy. You may use the Symbol Spec Selector control or manually type in the symbol name.
<initial_ sym>	Specify the name of the symbol that is automatically selected when the Symbol Selector dialog opens. You may use the Symbol Selector control or manually type in the symbol name.
<allow_ leaf>	Specify whether users can select leaf symbols in the Symbol Selector dialog.
Allowable_ parents	Specify the type of parent symbols that users can select in the Symbol Selector dialog.
<allow_ readonly>	Specify whether users can select read-only symbols in the Symbol Selector dialog.
<allow_ multiple>	Specify whether users can select multiple symbols in the Symbol Selector dialog.
<attribute_ filter>	Specify an attribute filter, or attribute filters. Attribute filters restrict the symbols that appear in the Symbol Selector hierarchy to symbols that meet the filter condition.

4. Click OK. The symbol selector appears in the worksheet cell, with a yellow background applied. If you specified a symbol for Initial_sym, that symbol name appears in the cell. Otherwise, [Double-click to select symbol] appears in the cell.

LVUSER Function

Using the LVUSER function, you can query your current username to a worksheet cell.

You can use the LVUSER function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVUSER syntax

In the worksheet cell, type:

```
=LVUSER()
```

Querying the current username using the Function Arguments dialog

You can query your current username by completing the function parameters in the Function Arguments dialog.

To query the current username:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVUSER. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

LVUSERDESC Function

Using the LVUSERDESC function, you can query your current user description to a worksheet cell.

You can use the LVUSERDESC function in the following ways:

- Type the function syntax into a worksheet cell.
- Complete the function parameters in the Function Arguments dialog.

LVUSERDESC syntax

In the worksheet cell, type:

```
=LVUSERDESC()
```

Querying the current user description using the Function Arguments dialog

You can query your current user description by completing the function parameters in the Function Arguments dialog.

To query the current user description:

1. Click the Longview tab.
2. In the Formulas group, click Longview Functions, and click LVUSERDESC. The Function Arguments dialog opens.
3. Click OK. The query data appears in the worksheet.

Working with Longview Add-In for Office APIs

Automation APIs are available in the Longview Add-In for Office for automating common operations such as refreshes and submissions and orchestrating those processes across many sheets in a workbook. You can use VBA to call these Automation APIs.

Caution: The Longview Add-In for Office APIs are supported for Longview connections only. APIs are not supported for Tidemark connections. Calling the Longview Add-In for Office APIs when connected to a Tidemark environment may yield unexpected results and/or behavior.

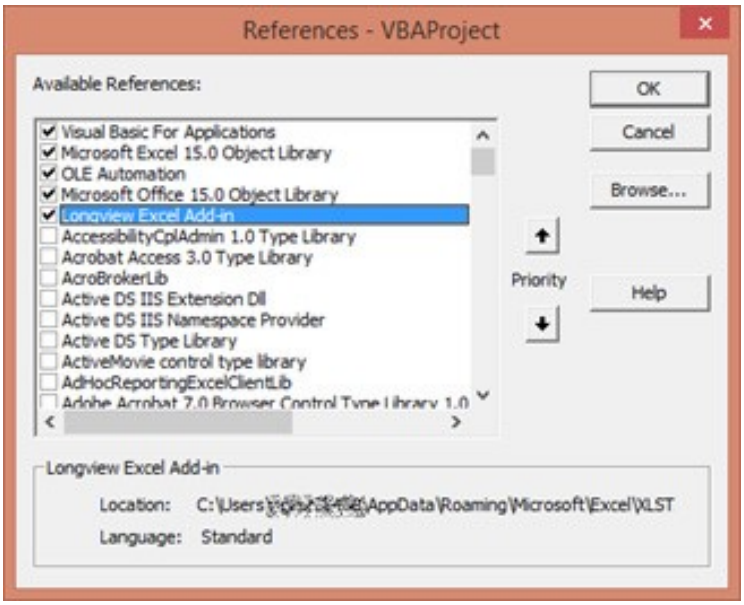
In general, the Automation APIs simulate the actions available in the Longview Add-In for Office toolbar:

- Connect
- Disconnect
- Refresh Worksheet
- Refresh Workbook
- Data Query
- Symbol Query
- Data Extract

There are also several other functions that you may use to help automate your workbooks using VBA.

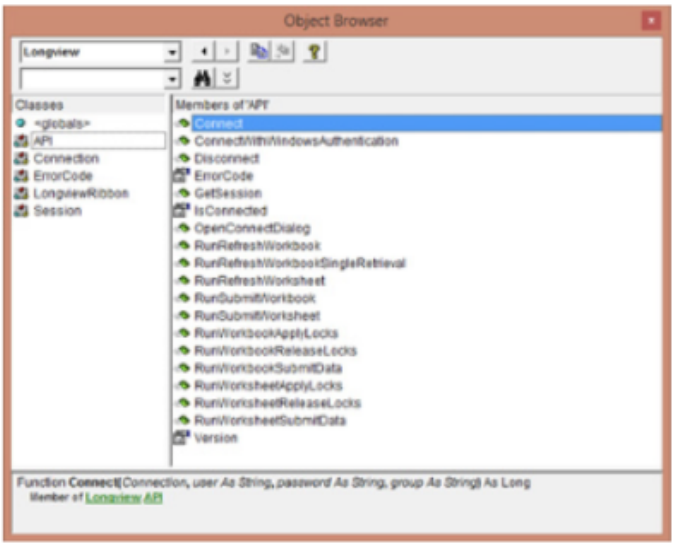
Getting Started with Automation APIs

Before you can use the Automation APIs in your workbooks, you will need to ensure that the Longview Excel Add-In Reference is added to your VBA Project.



Note: The library name may appear as Longview or Longview Excel Add-In.

Once in VBA, you may use the Object Browser to browse the objects in the API.



API Class

The API class is the main class used to implement the Automation APIs. This section describes the Functions and Properties of the API class.

API.Connect

The API.Connect function connects to the Longview data server with specified parameters.

Parameters	Description
connection	Type: string or Type: connection This parameter is the name of a defined Connection or a Connection object.
user	Type: string
password	Type: string
group	Type: string

The API.Connect function returns the following:

Return Code	Description
Long	API.ErrorCode.NoError = Successful Otherwise, an error has been encountered

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Set LVConnection = New Longview.Connection
Dim rc As Long: rc = LVAPI.ErrorCode.NoError
' Set connection parameters With LVConnection .HostOrProxy = "MyServer"
.Identifier = "DEMO101"
.Port = "29010"
End With
' Connect
rc = LVAPI.Connect(LVConnection, "JSmith", "xxx", "Administrators")
' Handle connection return code
Select Case rc
Case LVAPI.ErrorCode.NoError: MsgBox "Connected", vbOKOnly + vbInformation,
"Connection Successful"
Case LVAPI.ErrorCode.InvalidParameter: MsgBox "Invalid parameter specified"
& vbLf & "Return code " & rc, vbOKOnly + vbExclamation, "Connection Error"
Case LVAPI.ErrorCode.ConnectServerFailed: MsgBox "Server connection failed"
& vbLf & "Return code " & rc, vbOKOnly + vbExclamation, "Connection Error"
Case LVAPI.ErrorCode.AlreadyConnected: MsgBox "Already connected" & vbLf &
"Return code " & rc, vbOKOnly + vbExclamation, "Already Connected"
Case Else: MsgBox "A connection error has occurred" & vbLf & "Return code "
& rc, vbOKOnly + vbExclamation, "Connection Error"
End Select
```

API.Disconnect

The API.Disconnect function disconnects the user from the current Longview data server.

Parameters	Description
None	

The API.Disconnect function returns the following:

Return Code	Description
Long	API.ErrorCode.NoError = Successful Otherwise, an error has been encountered

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim rc As Long: rc = LVAPI.ErrorCode.NoError
If LVAPI.IsConnected Then     rc = LVAPI.Disconnect()
    If rc <> LVAPI.ErrorCode.NoError Then
MsgBox "Disconnect Failed" & vbLf & "Return code " & rc, vbOKOnly +
vbExclamation, "Disconnect Failed"
    Else
MsgBox "Disconnected", vbOKOnly + vbInformation, " Successfully
Disconnected"
    End If
End If
```

API.GetSession

The API.GetSession function retrieves information about the current session.

Parameters	Description
session	Type: session

The API.GetSession function returns the following:

Return Code	Description
long	API.ErrorCode.NoError = Successful Otherwise, an error has been encountered

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim rc As Long: rc = LVAPI.ErrorCode.NoError
Dim session As New session
rc = LVAPI.GetSession(session)
Select Case rc
Case LVAPI.ErrorCode.NoError: MsgBox session.Host & " " & session.Identifier
& " " &
session.UserId
Case LVAPI.ErrorCode.NotConnected: MsgBox "Not connected" & vbLf & "Return
code " & rc, vbOKOnly + vbExclamation, "Not Connected"
Case Else: MsgBox "Error encountered" & vbLf & "Return code " & rc, vbOKOnly
+ vbExclamation, "Error encountered"
End Select
```

API.IsSymbolExists

The API.IsSymbolExists function returns a boolean value indicating whether a specified symbol exists (in the dimension if the dimension name is specified).

Parameters	Description
symbolName	Type: String
dimensionName (optional)	Type: String (optional) Optional parameter specifying the name of the dimension containing symbolName.

The API.IsSymbolExists function returns the following:

Return Code	Description
boolean	TRUE: Indicates symbol exists FALSE: Indicates symbol does not exist.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API Dim result As Variant
result = LVAPI.IsSymbolExists(“ABC123”) Range("I19").Value = result
result = LVAPI.IsSymbolExists(Range("H20").Value)
Range("I20").Value = result
```

API.OpenConnectDialog

The API.OpenConnectDialog function simulates running the menu option: Connect. This function launches the Connection dialog.

Parameters	Description
None	

The API.OpenConnect function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Call LVAPI.OpenConnectDialog
If LVAPI.IsConnected Then
MsgBox "Connected", vbOKOnly + vbInformation, "Connection Successful"
Else
MsgBox "Not connected" & vbLf & "Return code " & rc, vbOKOnly +
vbExclamation, "Not Connected"
End If
```

API.RetrieveChildSymbol

The API.RetrieveChildSymbol function retrieves the name of the child of a parent symbol.

Parameters	Description
symbolName	Type: String
dimensionName (optional)	Type: String (optional) Optional parameter specifying the name of the dimension containing symbolName.

The API.RetrieveChildSymbol function returns the following:

Return Code	Description
boolean	TRUE: Indicates symbol exists FALSE: Indicates symbol does not exist.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim result As Variant
result = LVAPI.IsSymbolExists(“ABC123”)
Range("I19").Value = result
result = LVAPI.IsSymbolExists(Range("H20").Value)
Range("I20").Value = result
```

API.RetrieveChildSymbolToWorksheet

The API.RetrieveChildSymbolToWorksheet function retrieves the name of the child of a parent symbol to a worksheet cell.

Parameters	Description
symbolname	Type: String Specifies the parent symbol whose child you wish to retrieve.
nthChild (optional)	Type: Long (default = 1) Optional parameter that represents the numeric position of the child symbol relative to the other child symbols at the same level in the hierarchy. If you do not specify a value for number, a default value of 1 is used.
worksheetName (optional)	Type: String Optional string representing the name of the worksheet to place the results. If the worksheet does not exist, it will be created. If the worksheetName parameter is not specified, the current worksheet will be used.
cell (optional)	Type: String Optional string representing the cell location to begin the symbol query. If the cell parameter is not specified, the default cell location A1 will be used.

Example

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
LVAPI.RetrieveChildSymbolToWorksheet "AYR2020", 2, "C1"
```

API.RetrieveDimDesc

The API.RetrieveDimDesc function retrieves the description of a dimension.

Parameters	Description
dimensionName	Type: String Specifies the name or number of the dimension.

The API.RetrieveDimDesc function returns the following:

Return Code	Description
String	Returns the description of the dimension specified.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim result As Variant

' Dimension description
result = LVAPI.RetrieveDimDesc("COST_CENTRES") Range("G2").Value = result
```

API.RetrieveDimDescToWorksheet

The API.RetrieveDimDescToWorksheet function retrieves the description of a dimension to a worksheet cell.

Parameters	Description
dimensionName	Type: String Specifies the name or number of the dimension.
worksheetName (optional)	Type: String Optional string representing the name of the worksheet to place the results. If the worksheet does not exist, it will be created. If the worksheetName parameter is not specified, the current worksheet will be used.
cell(optional)	Type: StringOptional string representing the cell location to begin the symbol query. If the cell parameter is not specified, the default cell location A1 will be used.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
LVAPI.RetrieveDimDescToWorksheet "ENTITIES", "H1"
```

API.RetrieveDimRoots

The API.RetrieveDimRoots function retrieves the root symbols of a dimension.

Parameters	Description
dimensionName	Type: String Specifies the name or number of the dimension.

The API.RetrieveDimRoots function returns the following:

Return Code	Description
String()	Returns the root symbols of the specified dimension.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim result As Variant
Dim x As Long

Dim rootNames() As String
rootNames = LVAPI.RetrieveDimRoots("ENTITIES")
For x = 0 To UBound(rootNames)
Range("A7").Offset(x).Value = rootNames(x)
Next x
```

API.RetrieveDimRootsToWorksheet

The API.RetrieveDimRootsToWorksheet function retrieves the root symbols of a dimension to a worksheet.

Parameters	Description
dimensionName	Type: String Specifies the name of the dimension.
worksheetName (optional)	Type: String Optional string representing the name of the worksheet to place the results. If the worksheet does not exist, it will be created. If the worksheetName parameter is not specified, the current worksheet will be used.
cell (optional)	Type: String Optional string representing the cell location to begin the symbol query. If the cell parameter is not specified, the default cell location A1 will be used.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim result As Variant
' Dimension roots
LVAPI.RetrieveDimRootsToWorksheet "ENTITIES", "F1", "Retrieve DimRoots to
WS"
```

API.RetrieveSymbolInfo

The API.RetrieveSymbolInfo function retrieves various metadata information for a specified symbol. Metadata information includes information such as: the description, balance type, number of children, weight.

Parameters	Description
dimensionName	Type: String Specifies the name of the dimension.
symbolName	Type: String Specifies the symbol whose information you wish to retrieve.
root (optional)	Type: String Optional parameter for the name of the root symbol of the hierarchy containing the symbolname. Specify a root symbol if the symbol name specified for symbolName belongs to multiple parent symbols.

The API.RetrieveSymbolInfo function returns the following:

Return Code	Description
Symbol	Contains the symbol information for the specified symbol.

Example:

```
im LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim symbol As Longview.symbol
Set symbol= LVAPI.RetrieveSymbolInfo("TIMEPERIODS", "A2020")
Range("I8").Value = symbol.Name
Range("J8").Value = symbol.Description
Range("K8").Value = symbol.BalanceType
Range("L8").Value = symbol.TotalDescendants
Range("M8").Value = symbol.Weight
```

```
Range("N8").Value = symbol.SymbolType
```

API.RetrieveSymbolList

The API.RetrieveSymbolList function retrieves a list of symbols for a specified dimension.

Parameters	Description
dimensionName	Type: String Specifies the name of the dimension.

The API.RetrieveSymbolList function returns the following:

Return Code	Description
String()	Returns a list of symbol names for the specified dimension.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim result As Variant
Dim symbols() As String symbols = LVAPI.RetrieveSymbolList("TIMEPERIODS")
Dim x As Long
For x = 0 To UBound(symbols)
        Range("U5").Offset(x).Value = symbols(x)
Next x
```

API.RetrieveSymbolListToWorksheet

The API.RetrieveSymbolListToWorksheet function retrieves a list of symbols for a specified dimension to a worksheet.

ParametUers	Description
dimensionName	Type: String Specifies the name of the dimension
worksheetName (optional)	Type: String Optional string representing the name of the worksheet to place the results. If the worksheet does not exist, it will be created. If the worksheetName parameter is not specified, the current worksheet will be used.

ParametUers	Description
cell (optional)	Type: String Optional string representing the cell location to begin the symbol query. If the cell parameter is not specified, the default cell location A1 will be used.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
LVAPI.RetrieveSymbolListToWorksheet "ENTITIES", "F1", "ENTITY Symbols"
```

API.RunRefreshWorkbook

The API.RunRefreshWorkbook function simulates running the menu option: Refresh Workbook >By Cycling Through All Worksheets.

This function refreshes all LV* functions in the workbook. The system refreshes the workbook by making a call to the data server for each worksheet in the workbook.

Parameters	Description
workbook (optional)	Type: workbook (optional) If workbook is not specified, then the active workbook will be refreshed. The workbook specified becomes the active workbook.

The API.RunRefreshWorkbook function returns the following:

Return Code	Notes
None	

Example:

```
Dim LVAPI As Longview.API
Dim WBExampleas Workbook
Set WBExample = Workbooks("Example1.xlsx")
'Refreshes the currently-active
workbook Call
LVAPI.RunRefreshWorkbook
'Refreshes the currently-active workbook
Call LVAPI.RunRefreshWorkbook(ActiveWorkbook)
'Refreshes the workbook containing this code
```

```
Call LVAPI.RunRefreshWorkbook(ThisWorkbook)
'Refreshes the workbook as set in variable WBExample
Call LVAPI.RunRefreshWorkbook(WBExample)
'Refreshes workbook "Example2.xlsx"
Call LVAPI.RunRefreshWorkbook(Workbooks("Example2.xlsx")
```

API.RunRefreshWorkbookSingleRetrieval

The API.RunRefreshWorkbookSingleRetrieval function simulates running the menu option: Refresh Workbook >With a Single Database Retrieval.

This function refreshes all LV* functions in the workbook. The system refreshes the workbook by making a single call to the data server for the entire workbook.

Parameters	Description
workbook (optional)	Type: workbook (optional) If workbook is not specified, then the active workbook will be refreshed. The workbook specified becomes the active workbook.

The API.RunRefreshWorkbookSingleRetrieval function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim WBExampleas Workbook
Set WBExample = Workbooks("Example1.xlsx")
'Refreshes the currently-active workbook
Call LVAPI.RunRefreshWorkbookSingleRetrieval
'Refreshes the currently-active workbook
Call LVAPI.RunRefreshWorkbookSingleRetrieval(ActiveWorkbook)
'Refreshes the workbook containing this code
Call LVAPI.RunRefreshWorkbookSingleRetrieval(ThisWorkbook)
'Refreshes the workbook as set in variable WBExample
Call LVAPI.RunRefreshWorkbookSingleRetrieval(WBExample)
'Refreshes workbook "Example2.xlsx"
```

```
Call LVAPI.RunRefreshWorkbookSingleRetrieval(Workbooks("Example2.xlsx")
```

API.RunRefreshWorksheet

The API.RunRefreshWorksheet function simulates running the menu option: Refresh Worksheet. This function refreshes all LV* functions in the requested worksheet.

Parameters	Description
worksheet (optional)	Type: worksheet (optional) If worksheet is not specified, then the active worksheet will be refreshed. The worksheet specified becomes the active worksheet in the active workbook.

The API.RunRefreshWorksheet function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
'Refresh active sheet by explicitly passing active sheet Call
LVAPI.RunRefreshWorksheet(ActiveSheet)
'Refresh active sheet by not specifying worksheet
Call LVAPI.RunRefreshWorksheet()
'Refresh specifically named worksheet
Call LVAPI.RunRefreshWorksheet(Worksheets(“Sheet1”))
'Refresh a specific worksheet
Dim WSTest as Worksheet
Set WSTest = Worksheets(“NameOfAWorksheet”)
Call LVAPI.RunRefreshWorksheet(WSTest)
```

API.RunSubmitWorkbook

The API.RunSubmitWorkbook function simulates running the menu option:

Submit > LVLOCK Submission >Submit Workbook.

This function invokes the following steps, in order:

1. Applies the lock(s) as defined by each LVLOCK and LVSLOCK function in the active workbook.
2. Submits data for each LVSUBMIT, LVSSUBMIT and LVSUBMITCOMMENT function in the active

workbook.

3. Releases the lock(s) defined by each LVLOCK and LVSLOCK function in the active workbook.

Parameters	Description
workbook	Type: workbook (optional) If workbook is not specified, then the active workbook will be submitted.The workbook specified becomes the active workbook.

The API.RunSubmitWorkbook function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim WBExampleas Workbook
Set WBExample = Workbooks("Example1.xlsx")
'Refreshes the currently-active
workbook Call
LVAPI.RunSubmitWorkbook
'Refreshes the currently-active workbook
Call LVAPI.RunSubmitWorkbook(ActiveWorkbook)
'Refreshes the workbook containing this code
Call LVAPI.RunSubmitWorkbook(ThisWorkbook)
'Refreshes the workbook as set in variable WBExample
Call LVAPI.RunSubmitWorkbook(WBExample)
'Refreshes workbook "Example2.xlsx"
Call LVAPI.RunSubmitWorkbook(Workbooks("Example2.xlsx")
```

API.RunSubmitWorksheet

The API.RunSubmitWorksheet function simulates running the menu option:

Submit > LVLOCK Submission > Submit Worksheet.

This function invokes the following steps, in order:

1. Applies the lock(s) as defined by each LVLOCK and LVSLOCK function in the specific worksheet.
2. Submits data for each LVSUBMIT, LVSSUBMIT and LVSUBMITCOMMENT function in the

specified worksheet.

3. Releases the lock(s) defined by each LVLOCK and LVSLOCK function in the specified worksheet.

Parameters	Description
worksheet (optional)	Type: worksheet (optional) If worksheet is not specified, then the active worksheet will be submitted.The worksheet specified becomes the active worksheet in the active workbook.

The API.RunSubmitWorksheet function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
'Submit active sheet by explicitly passing active sheet
Call LVAPI.RunSubmitWorksheet(ActiveSheet)
'Submit active sheet by not specifying worksheet
Call LVAPI.RunSubmitWorksheet()
'Submit specifically named worksheet
Call LVAPI.RunSubmitWorksheet (Worksheets("Sheet1"))
'Submit a specific worksheet
Dim WSTest as Worksheet
Set WSTest = Worksheets("NameOfAWorksheet")
Call LVAPI.RunSubmitWorksheet(WSTest)
```

API.RunSubmitAutoLockWorkbook

The API.RunSubmitAutoLockWorkbook function simulates running the menu option:

Submit >Automatically Lock and Submit Workbook.

This function invokes the following steps, in order:

1. Identifies all the LVSUBMIT, LVSSUBMIT and LVSUBMITCOMMENT functions in the active workbook and automatically creates the required locks to all referenced symbols.
2. Submits data for each LVSUBMIT, LVSSUBMIT and LVSUBMITCOMMENT function in the active workbook.
3. Release the locks once the submission is complete.

Parameters	Description
workbook	Type: workbook (optional) If workbook is not specified, then the active workbook will be submitted.The workbook specified becomes the active workbook.

The API.RunSubmitAutoLockWorkbook function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim WBExampleas Workbook
Set WBExample = Workbooks("Example1.xlsx")
'Refreshes the currently-active
workbook Call
LVAPI.RunSubmitAutoLockWorkbook
'Refreshes the currently-active workbook
Call LVAPI.RunSubmitAutoLockWorkbook(ActiveWorkbook)
'Refreshes the workbook containing this code
Call LVAPI.RunSubmitAutoLockWorkbook(ThisWorkbook)
'Refreshes the workbook as set in variable WBExample
Call LVAPI.RunSubmitAutoLockWorkbook(WBExample)
'Refreshes workbook "Example2.xlsx"
Call LVAPI.RunSubmitAutoLockWorkbook(Workbooks("Example2.xlsx")
```

API.RunSubmitAutoLockWorksheet

The API.RunSubmitAutoLockWorksheet function simulates running the menu option:

Submit >Automatically Lock and Submit Worksheet.

This function invokes the following steps, in order:

1. Identifies all the LVSUBMIT, LVSSUBMIT and LVSUBMITCOMMENT functions in the active worksheet and automatically creates the required locks to all referenced symbols.
2. Submits data for each LVSUBMIT, LVSSUBMIT and LVSUBMITCOMMENT function in the active worksheet.
3. Release the locks once the submission is complete.

Parameters	Description
worksheet (optional)	Type: worksheet (optional) If worksheet is not specified, then the active worksheet will be submitted.The worksheet specified becomes the active worksheet in the active workbook.

The API.RunSubmitAutoLockWorksheet function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
'Submit active sheet by explicitly passing active sheet
Call LVAPI.RunSubmitAutoLockWorksheet(ActiveSheet)
'Submit active sheet by not specifying worksheet
Call LVAPI.RunSubmitAutoLockWorksheet()
'Submit specifically named worksheet
Call LVAPI.RunSubmitAutoLockWorksheet (Worksheets("Sheet1"))
'Submit a specific worksheet
Dim WSTest as Worksheet
Set WSTest = Worksheets("NameOfAWorksheet")
Call LVAPI.RunSubmitAutoLockWorksheet(WSTest)
```

API.RunWorkbookApplyLocks

The API.RunWorkbookApplyLocks function simulates running the menu option: Submit >Manual Submission >Workbook >Apply Locks. This function applies a lock as defined by each LVLOCK function in the active workbook.

Parameters	Description
workbook (optional)	Type: workbook (optional) If workbook is not specified, then the active workbook will be refreshed. The workbook specified becomes the active workbook.

The API.RunWorkbookApplyLocks function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim WBExampleas Workbook
Set WBExample = Workbooks("Example1.xlsx")
'Refreshes the currently-active workbook
Call LVAPI.RunWorkbookApplyLocks
'Refreshes the currently-active workbook
Call LVAPI.RunWorkbookApplyLocks(ActiveWorkbook)
'Refreshes the workbook containing this code
Call LVAPI.RunWorkbookApplyLocks(ThisWorkbook)
'Refreshes the workbook as set in variable WBExample
Call LVAPI.RunWorkbookApplyLocks(WBExample)
'Refreshes workbook "Example2.xlsx"
Call LVAPI.RunWorkbookApplyLocks(Workbooks("Example2.xlsx")
```

API.RunWorkbookReleaseLocks

The API.RunWorkbookReleaseLocks function simulates running the menu option: Submit > Manual Submission > Workbook > Release Locks. This function releases the lock defined by each LVLOCK function in the active workbook.

Parameters	Description
workbook (optional)	Type: workbook (optional) If workbook is not specified, then the active workbook will be refreshed. The workbook specified becomes the active workbook

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim WBExampleas Workbook
Set WBExample = Workbooks("Example1.xlsx")
'Refreshes the currently-active workbook
```

```
Call LVAPI.RunWorkbookReleaseLocks
'Refreshes the currently-active workbook
Call LVAPI.RunWorkbookReleaseLocks(ActiveWorkbook)
'Refreshes the workbook containing this code
Call LVAPI.RunWorkbookReleaseLocks(ThisWorkbook)
'Refreshes the workbook as set in variable WBExample
Call LVAPI.RunWorkbookReleaseLocks(WBExample)
'Refreshes workbook "Example2.xlsx"
Call LVAPI.RunWorkbookReleaseLocks(Workbooks("Example2.xlsx")
```

API.RunWorkbookSubmitData

The API.RunWorkbookSubmitData function simulates running the menu option: Submit >Manual Submission >Workbook >Submit Data. This function submits data for each LVSUBMIT function in the active workbook.

Parameters	Description
workbook (optional)	Type: workbook (optional) If workbook is not specified, then the active workbook will be refreshed. The workbook specified becomes the active workbook

The API.RunWorkbookSubmitData function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Call LVAPI.RunWorkbookSubmitData(ThisWorkbook)
```

API.RunWorksheetApplyLocks

The API.RunWorksheetApplyLocks function simulates running the menu option: Submit >Manual Submission >Worksheet >Apply Locks. This function applies a lock as defined by each LVLOCK function in the specified worksheet.

Parameters	Description
worksheet (optional)	Type: worksheet (optional) If worksheet is not specified, then the active worksheet will be refreshed. The worksheet specified becomes the active worksheet in the active workbook.

The API.RunWorksheetApplyLocks function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
'Apply locks on active sheet by explicitly passing active sheet
Call LVAPI.RunWorksheetApplyLocks (ActiveSheet)
'Apply locks on active sheet by not specifying worksheet
Call LVAPI.RunWorksheetApplyLocks ()
'Apply locks on specifically named worksheet
Call LVAPI.RunWorksheetApplyLocks (Worksheets("Sheet1"))
'Apply locks on a specific worksheet
Dim WSTest as Worksheet
Set WSTest = Worksheets("NameOfAWorksheet")
Call LVAPI.RunWorksheetApplyLocks(WSTest)
```

API.RunWorksheetReleaseLocks

The API.RunWorksheetReleaseLocks function simulates running the menu option: Submit >Manual Submission > Worksheet >Release Locks. This function releases the lock defined by each LVLOCK function in the specified worksheet.

Parameters	Description
worksheet (optional)	Type: worksheet (optional) If worksheet is not specified, then the active worksheet will be refreshed. The worksheet specified becomes the active worksheet in the active workbook.

The API.RunWorksheetReleaseLocks function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
'Submit active sheet by explicitly passing active
sheet Call LVAPI.RunWorksheetReleaseLocks (ActiveSheet)
'Submit active sheet by not specifying worksheet
Call LVAPI.RunWorksheetReleaseLocks ()
'Submit specifically named worksheet
Call LVAPI.RunWorksheetReleaseLocks (Worksheets("Sheet1"))
'Submit a specific worksheet
Dim WSTest as Worksheet
Set WSTest = Worksheets("NameOfAWorksheet")
Call LVAPI.RunWorksheetReleaseLocks (WSTest)
```

API.RunWorksheetSubmitData

The API.RunWorksheetSubmitData function simulates running the menu option: Submit >Manual Submission >Worksheet >Submit Data. This function submits data for each LVSUBMIT function in the specified worksheet.

Parameters	Description
worksheet (optional)	Type: worksheet (optional) If worksheet is not specified, then the active worksheet will be refreshed. The worksheet specified becomes the active worksheet in the active workbook.

The API.RunWorksheetSubmitData function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Call LVAPI.RunWorksheetSubmitData(ActiveSheet)
'Submit active sheet by explicitly passing active sheet
Call LVAPI.RunWorksheetSubmitData (ActiveSheet)
'Submit active sheet by not specifying worksheet
```

```
Call LVAPI.RunWorksheetSubmitData ()
'Submit specifically named worksheet
Call LVAPI.RunWorksheetSubmitData (Worksheets("Sheet1"))
'Submit a specific worksheet
Dim WSTest as Worksheet
Set WSTest = Worksheets("NameOfAWorksheet")
Call LVAPI.RunWorksheetSubmitData (WSTest)
```

API Properties

The following table lists the properties of the API class:

Name	Type	Description
ErrorCode	object	ErrorCode represents common error codes that may be returned by various functions.
IsConnected	boolean	IsConnected returns a Boolean value indicating whether the user is currently connected or not.
Version	string	Version is a string indicating the version of the Longview Add-In for Office.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim rc As Long: rc = LVAPI.ErrorCode.NoError
MsgBox "Longview Version " & LVAPI.Version, vbOKOnly + vbInformation,
"Version Information"


If LVAPI.IsConnected Then
      MsgBox "Connected", vbOKOnly + vbInformation, "Connection Successful"
Else
MsgBox "Not connected" & vbLf & "Return code " & rc, vbOKOnly +
vbExclamation, "Not Connected"
End If
```

Connection Class

The Connection class represents the connection parameters used to connect to the Longview server. Use the Connection class in conjunction with API.Connect to create a connection to the Longview system.

Connection Properties

The following table lists the properties of the Connection class:

Name	Type	Description
HostOrProxy	string	HostOrProxy specifies the host name or proxy of the data server to connect to.
Identifier	string	Identifier specifies the Longview Identifier (LID) for the data server to connect to.
Port	string (optional)	Port specifies the port number of the data server to connect to. This field is omitted if you specified a URL (starting with http/s) for an HTTP Proxy Server for the HostOrProxy field.

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Set LVConnection = New Longview.Connection
Dim rc As Long: rc = LVAPI.ErrorCode.NoError
' Set connection parameters With LVConnection
.Identifier = "DEMO101"
        .HostOrProxy = "MyServer"
.Port = "29010" End With rc = LVAPI.Connect(LVConnection, "JSmith", "xxx",
"Administrators")
'You may also set the Connection info using Connection.SetServerInfo Call
LVConnection.SetServerInfo ("DEMO101", "MyServer", "29010")
```

Connection.SetServerInfo

The Connection.SetServerInfo function sets the connection information. This function can be used in conjunction with the API.Connect function.

Parameters	Description
Identifier	Type: string Identifier specifies the Longview Identifier (LID) for the system to connect to.
HostOrProxy	Type: string HostOrProxy specifies the host name or proxy of the data server to connect to.
Port	Type: string (optional) Port specifies the port number of the data server to connect to. This field is not required if you specified a URL (starting with http) for an HTTP Proxy Server for the Host field.

The Connection.SetServerInfo function returns the following:

Return Code	Description
None	

Example:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Set LVConnection = New Longview.Connection

Dim rc As Long: rc = LVAPI.ErrorCode.NoError
Call LVConnection.SetServerInfo ("DEMO101", "MyServer", "29010") rc =
LVAPI.Connect(LVConnection, "JSmith", "xxx", "Administrators")
```

DataExtract Class

The DataExtract class represents a data extract query that is built from the menu option: Data Extract. The functions and properties of this class can be used to build a new data extract query, modify an existing data extract query and output the results to a worksheet.

Property Summary

- Property AdjustedDetail As AdjustedDetail
- Property DataMode As DataMode
- Property Schedule As String
- Property ShowSymbolDescription As Boolean
- Property SortOrder As SortOrder
- Property SuppressZero As Boolean
- Property VaryingDimension As String
- Property SeperateNumericAndStringValue As Boolean

Sub/Function Summary

- Sub AddSymbolSpec(symbolSpec As String)
- Sub ClearSpecs([dimensionName As String])
- Sub LoadQuery(fileName As String)
- Sub RunToWorksheet([worksheetName As String], [cell As String])

Properties

Parameter	Description
AdjustedDetail	Property AdjustedDetail As AdjustedDetail Specifies whether the data query will return adjusted or unadjusted data. ■ AdjustedDetail_Adjusted – return adjusted data. This is the default. ■ AdjustedDetail_Unadjusted – return unadjusted data.
DataMode	Property DataMode As DataMode Specifies the type of hierarchy data to return. ■ DataMode_All – Return all types of data. This is the default. ■ DataMode_CTA – Return Cumulative Translation Adjustment data only. ■ DataMode_Leaf – Return data for leaf cells only.
Schedule	Property Schedule As String Specifies the name of the schedule to include in the data query. If this property is not set, base data will be returned. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown if the specified schedule name is invalid.
ShowSymbolDescription	Property ShowSymbolDescription As String Specifies whether symbol descriptions are returned. Symbol descriptions are returned in an extra column. The default setting is FALSE.
SortOrder	Property SortOrder As SortOrder Sets the order in which symbols will be returned. ■ SortOrder_Alphabetically– Return symbols sorted in alphabetical order. ■ SortOrder_Hierarchically – Return symbols as defined by their hierarchy order. This is the default.
SuppressZero	Property SupressZero As Boolean Specifies whether rows with zero data will be returned. The default value is TRUE.

Parameter	Description
VaryingDimension	Property VaryingDimension As String If this property is set, symbols for the specified dimension will be displayed as columns.
SeperateNumericAndStringValue	Property SeperateNumericAndStringValue As Boolean Specifies whether the data that is returned should be displayed in separate numeric and string columns. The default value is FALSE.

Subs/Functions

Parameter	Description
AddSymbolSpec	Sub AddSymbolSpec(symbolSpec As String) Adds a symbol spec representing the hierarchy you wish to query. A symbol spec must be added for every base dimension. ■ Parameters: symbolSpec – String representing a single symbol spec. For example: TRIALBAL#99. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the symbol spec is invalid.
ClearSpecs	Sub ClearSpecs([dimensionName As String]) Removes all symbol specs for the specified dimension that have been added or loaded in the Data Query. ■ Parameters: dimensionName – Optional string representing the dimension in which to clear. If no dimension is specified, specs will be cleared for all dimensions. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the specified dimension name is invalid.
LoadQuery	Sub LoadQuery(fileName As String) Loads a data extract that has been previously built and saved from the menu option: Data Extract. ■ Parameters: filename – String representing the full path to the query file to load. For example: `C:\SymbolQuery.lvqsq`. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the file cannot be loaded.

Parameter	Description
RunToWorksheet	Sub RunToWorksheet([worksheetName As String], [cell As String]) Runs a symbol query and places the results into the specified worksheet and cell location. ■ Parameters: worksheetName - Optional string representing the name of the worksheet to place the results. If the worksheet does not exist, it will be created. If the worksheetName parameter is not specified, the current worksheet will be used. cell - Optional string representing the cell location to begin the symbol query. If the cell parameter is not specified, the default cell location A1 will be used. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the query is invalid or when the specified worksheet name or cell are invalid.

Sample Usage:

```
Sub RunDataExtract()
On Error GoTo ErrorHandler
Dim query As Longview.DataExtract Set query = New Longview. DataExtract
query.AdjustedDetail = AdjustedDetail_Unadjusted query.DataMode = DataMode_
CTA
query.LoadQuery "C:\MyDataExtract.lvqde"
query.ClearSpecs "Entities"
query.AddSymbolSpec "CANADA#99"
query.RunToWorksheet "MyDataExtract", "B1"
ErrorHandler:
If Err.Number <> 0 Then
MsgBox "Unable to run Data Extract (" & Err.Number & ") - " &
Err.Description
End If
End Sub
```

DataQuery Class

The DataQuery class represents a data query that is built from the menu option: Data Query. The functions and properties of this class can be used to build a new data query, modify an existing data query and output the results to a worksheet.

Note: A symbol spec must be added for every base dimension.

Property Summary

- Property AdjustedDetail As AdjustedDetail
- Property ApplyFormatting As Boolean
- Property ApplyNegativeColor As Boolean
- Property ColumnWidth As Double
- Property CreateNewWorksheet As Boolean
- Property DataMode As DataMode
- Property DataOutput As DataOutput
- Property DecimalPlaces As Long
- Property FormulasOutput As FormulasOutput
- Property NegativeFormat As String
- Property RowDescriptionWidth As Double
- Property RowNameWidth As Double
- Property ShowFixedSymbols As Boolean
- Property ShowSymbolDescription As Boolean
- Property ShowTimestamp As Boolean
- Property Subtitle As String
- Property Title As String
- Property UseThousandsSeparator As Boolean
- Property ExcludeZeros As ExcludeZeros

Sub/Function Summary

- Sub AddColumnSpec(symbolSpec As String)
- Sub AddRowSpec(symbolSpec As String)
- Sub AddWorksheetSpec(symbolSpec As String)
- Sub ClearSpecs([dimensionName As String])
- Sub LoadQuery(fileName As String)

- Sub RunToWorksheet([worksheetName As String], [cell As String])
- Sub SetFixedSymbol(symbolName as String)

Properties

Parameter	Description
AdjustedDetail	Property AdjustedDetail As AdjustedDetail Specifies whether the data query will return adjusted or unadjusted data. ■ AdjustedDetail_Adjusted – Returns adjusted data. This is the default. ■ AdjustedDetail_Unadjusted – Returns unadjusted data.
ApplyFormatting	Property ApplyFormatting As Boolean Specifies whether the custom number formatting is applied to numbers in the results. The default value is FALSE. If the value is set to TRUE, the values of the DecimalPlaces, UseThousandsSeparator, NegativeFormat and ApplyNegativeColor properties will be applied to numbers in the query.
ApplyNegativeColor	Property ApplyNegativeColor As Boolean Specifies whether negative colors will be rendered in Red. The default value is FALSE.
ColumnWidth	Property ColumnWidth As Double Specifies the default column width for every column in the query. The default value is 13.57 points.
CreateNewWorksheet	Property CreateNewWorksheet As Boolean By default, a data query returns resulting data in a separate worksheet for every symbol in the Worksheet Dimension. This property specifies whether to replace an existing worksheet if the workbook already contains a worksheet with the same name. The default value is FALSE.
DecimalPlaces	Property DecimalPlaces As Double Specifies the number of decimals to show for each numeric data cell. The default value is 0.

Parameter	Description
FormulasOutput	Property FormulasOutput As FormulasOutput When DataOutput is set to DataOutput_Formulas, this property can be used to modify LVCELL() functions to use hard coded symbol names or as cell references to a cell containing the symbol name. ■ FormulasOutput_NoCellReferences – Symbol names are hard coded. This is the default. ■ FormulasOutput_CellReferences – Symbol names are replaced with cell references to the column or row heading containing the symbol name.
NegativeFormat	Property NegativeFormat As String A number format string representing how negative numbers will be formatted. See Excel documentation on valid string formats.
RowDescriptionWidth	Property RowDescriptionWidth As Double Specifies the default column width for columns representing a symbol description. The default value is 38.14 points.
RowNameWidth	Property RowNameWidth As Double Specifies the default column width for column representing a symbol name. The default value is 22.71 points.
ShowFixedSymbols	Property ShowFixedSymbols As String Specifies whether a legend displaying the selections for the fixed dimensions will be displayed above the actual data. The default value is FALSE.
ShowSymbolDescription	Property ShowTimestamp As Boolean Specifies whether the current date/time will be displayed above the actual data. The default value is FALSE.
Subtitle	Property Subtitle As String Specifies a subtitle to be displayed above the data grid and below the title.
Title	Property Title As String Specifies a title to be displayed above the data grid.
UseThousandsSeparator	Property UseThousandsSeparator As Boolean Specifies whether thousands separators are used when formatting numbers.

Parameter	Description
ExcludeZeros	Property ExcludeZeros As ExcludeZeros This property is used to specify whether “zero” data will be supressed. ■ ExcludeZeros_None – No rows or columns are suppressed. This is the default. ■ ExcludeZeros_Rows – Rows where all the data returned are values of “zero” will be suppressed. ■ ExcludeZeros_Columns – Columns where all the data returned are values of “zero” will be suppressed. ■ ExcludeZeros_RowsAndColumns – Rows and columns where all the data returned are values of “zero” will be suppressed.

Sub/Functions

Parameter	Description
AddColumnSpec	Sub AddColumnSpec (symbolSpec As String) Adds a symbol spec to the columns dimension representing the hierarchy you wish to query. You may add multiple column specs but only if the symbols are from the same dimension. ■ Parameters: symbolSpec - String representing a single symbol spec. For example: TRIALBAL#99. ■ Errors: ErrorCode.INVALID_PARAMETER - This error is thrown when the symbol spec is invalid or if the symbol being added has already been added in a row, worksheet or fixed dimension.

Parameter	Description
AddRowSpec	Sub AddRowSpec (symbolSpec As String) Adds a symbol spec to the down dimension representing the hierarchy you wish to query. You may add multiple row specs but only if the symbols are from the same dimension. ■ Parameters: symbolSpec – String representing a single symbol spec. For example: TRIALBAL#99. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the symbol spec is invalid or if the symbol being added has already been added in a column, worksheet or fixed dimension.
AddWorksheetSpec	Sub AddWorksheetSpec (symbolSpec As String) Adds a symbol spec representing the hierarchy you want to query. Each symbol in the spec will be added as a separate worksheet in the current workbook. ■ Parameters: symbolSpec – String representing a single symbol spec. For example: TRIALBAL#99. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the symbol spec is invalid or if the symbol being added has already been added as row, column or fixed dimension.
ClearSpecs	Sub ClearSpecs([dimensionName As String]) Removes all symbol specs for the specified dimension that have been added or loaded in the Data Query. ■ Parameters: dimensionName – Optional string representing the dimension in which to clear. If no dimension is specified, specs will be cleared for all dimensions. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the specified dimension name is invalid.

Parameter	Description
LoadQuery	Sub LoadQuery(fileName As String) Loads a data query that has been previously built and saved from the menu option: Data Query. ■ Parameters: filename – String representing the full path to the query file to load. For example: C:\SymbolQuery.lvqsq. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the file cannot be loaded.
RunToWorksheet	Sub RunToWorksheet([worksheetName As String], [cell As String]) Runs a symbol query and places the results into the specified worksheet and cell location. ■ Parameters: worksheetName – Optional string representing the name of the worksheet to place the results. If the worksheet does not exist, it will be created. If the worksheetName parameter is not specified, the current worksheet will be used. cell – Optional string representing the cell location to begin the symbol query. If the cell parameter is not specified, the default cell location A1 will be used. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the query is invalid or when the specified worksheet name or cell are invalid.
SetFixedSymbol	Sub SetFixedSymbol (symbolName As String) Specifies the fixed symbol name to be used when running the data query. ■ Parameters: symbolName – String representing the symbol name. ■ Errors: ErrorCode.INVALID_PARAMETER – This error is thrown when the symbol name is invalid or if the symbol belongs to a dimension that has already been assigned to the Row, Column or Worksheet dimensions.

Sample Usage:

```
Sub RunDataQuery()
On Error GoTo ErrorHandler
Dim query As Longview.DataQuery
Set query = New Longview. DataQuery
query.AdjustedDetail = AdjustedDetail_Unadjusted
query.DataMode = DataMode_CTA
query.LoadQuery "C:\MyDataQuery.lvqde"
query.ClearSpecs "Entities"
query.AddRowSpec "CANADA#99"
query.SetFixedSymbol "DIM10SET"
query.RunToWorksheet "B1"
ErrorHandler:
If Err.Number <> 0 Then
MsgBox "Unable to run Data Query (" & Err.Number & ") - " &
Err.Description
End If
End Sub
```

ErrorCode Class

The ErrorCode class represents a set of common return values.

Use the ErrorCode class to create relevant error messages for common errors that may be encountered.

ErrorCode Return Values

The following table lists some of the common return values for the ErrorCode class:

Name	Type	Description
AlreadyConnected	long	Return value to indicate that the user is already connected.
ConnectServerFailed	long	Return value to indicate that connection to the server failed.
InvalidParameter	long	Return value to indicate that there is an invalid parameter.
NoError	long	Return value to indicate that function executed successfully.
NotConnected	long	Return value to indicate that the user is not connected.
NotFound	long	Return value to indicate that some object was not found.

HierarchyQuery Class

The HierarchyQuery class represents a hierarchy of symbols in a dimension that can be used in memory or output to a worksheet.

Property Summary

Property IncludeDescriptions As Boolean

Sub/Function Summary

- Sub AddSymbolSpec(symbolSpec As String, [root As String])
- Function Run() As QueryResult
- Sub RunToWorksheet([worksheetName As String], [cell As String])

Properties

Parameter	Description
IncludeDescriptions	Property IncludeDescriptions As Boolean Sets whether the query results will return a column for the symbol description and the parent description. By default, the value is FALSE.

Subs/Functions

Parameter	Description
AddSymbolSpec	Sub AddSymbolSpec(symbolSpec As String, [root As String]) Adds a symbol spec representing the hierarchy you wish to query. ■ Parameters: ◦ symbolSpec – String representing a single symbol spec. For example: TRIALBAL#99. ◦ root – Optional string representing the root symbol name for the specified symbol. ■ Errors: ErrorCode.INVALID_PARAMETER - This error is thrown when the symbol spec is invalid or when the root is not a valid root for the specified symbol.
Run	Function Run() As QueryResult Runs the hierarchy query and returns the results in a QueryResult class. ■ Errors: ErrorCode.INVALID_PARAMETER - This error is thrown when no symbol spec has been specified.

Parameter	Description
RunToWorksheet	Sub RunToWorksheet([worksheetName As String], [cell As String]) Runs a hierarchy query and places the results into the specified worksheet and cell location. ■ Parameters: ◦ worksheetName – optional string representing the name of the worksheet to place the results. If the worksheet does not exist, it will be created. If the worksheetName parameter is not specified, the current worksheet will be used. ◦ cell – optional string representing the cell location to begin the symbol query. If the cell parameter is not specified, the default cell location A1 will be used. ■ Errors:ErrorCode.INVALID_PARAMETER – This error is thrown when the query is invalid or when the specified worksheet name or cell are invalid.

Sample Usage:

```
Sub RunHierarchyQuery()
On Error GoTo ErrorHandler
Dim result As Longview.QueryResult
Dim query As Longview.HierarchyQuery
Set query = New Longview.HierarchyQuery
Set result = query.Run()
Dim row As Long
Dim column As Long
For row = 1 to result.RowCount
For column = 1 to result.ColumnCount
' Place results of symbol query into cells of the worksheet manually
Range("A1").Offset(row - 1, column - 1).Value = result.GetCellValue(row,
column)
Next column
Next row
ErrorHandler:
If Err.Number <> 0 Then
MsgBox "Unable to run Hierarchy Query (" & Err.Number & ") - " &
Err.Description
End If
End Sub
```

HierachyQuery

The QueryResult class contains the results to a HierarchyQuery. The results are structured as a table where each row represents a symbol entry, and the columns represent the properties for that symbol.

Property Summary

- Property ColumnCount As Long
- Property RowCount As Long

Sub/Function Summary

- Function GetCellValue(row As Long, column as Long) As String

Properties

Parameter	Description
ColumnCount	Property ColumnCount As Long Gets the number of columns in the query result. ■ If IncludeDescriptions = True, there will be 4 columns. ■ If IncludeDescriptions = False, there will be 2 columns. You can use this information to help you write your query to a worksheet or iterate through the columns to do some processing in your VBA code.
RowCount	Property RowCount As Long Gets the number of rows in the query result. Each row represents a symbol entry.

Subs/Functions

Parameter	Description
GetCellValue	Function GetCellValue(row As Long, column As Long) As String Returns the string value at the specified row/column. Each row represents a symbol entry while the columns are the properties of the symbol as follows: ■ IncludeDescriptions = True ◦ Column 1 – Symbol Name ◦ Column 2 – Symbol Description ◦ Column 3 – Parent Symbol Name ◦ Column 4 – Parent Symbol Description ■ IncludeDescriptions = False ◦ Column 1 – Symbol Name ◦ Column 2 – Parent Symbol Name ■ Parameters: ◦ row – Long representing row number co-ordinate ◦ column – Long representing the column number co-ordinate ■ Errors: ErrorCode.INVALID_PARAMETER - This error is thrown when the specified row or column is outside the range of valid values.

Sample Usage:

```
Sub RunHierarchyQuery()
On Error GoTo ErrorHandler
Dim result As Longview.QueryResult
Dim query As Longview.HierarchyQuery
Set query = New Longview.HierarchyQuery
Set result = query.Run()
Dim row As Long
Dim column As Long
For row = 1 to result.RowCount
For column = 1 to result.ColumnCount
' Place results of symbol query into cells of the worksheet manually
Range("A1").Offset(row - 1, column - 1).Value = result.GetCellValue(row,
column)
Next column
```

```
Next row
ErrorHandler:
If Err.Number <> 0 Then
    MsgBox "Unable to run Hierarchy Query (" & Err.Number & ") - " &
Err.Description
End If
End Sub
```

Session Class

The Session class represents various aspects of a session such as the user id, host, port, and identifier of the currently connected Longview system.

Use the Session class in conjunction with API.GetSession to retrieve information for the current session.

Session Properties

The following table lists the properties of the Session class

Name	Type	Description
Group	string	Group is the Longview Group for the user in the session.
Host	string	Host is the host name of the data server in the session.
HttpProtocol	string	HttpProtocol is the http protocol for the current system.
Identifier	string	Identifier is the Longview Identifier for the session.
IsLongviewSession	boolean	IsLongviewSession indicates whether the connection is a Longview connection to a Longview data source.
IsTidemarkSession	boolean	IsTidemarkSession indicates whether the connection is a Tidemark connection to a Tidemark data source.
IsValid	boolean	IsValid returns a boolean indicating that the session object has been initialized.
Port	long	Port is the port number of the data server. In cases where HTTP proxy has been used, the Port will return an empty string.
RDBMS	string	RDBMS is the current RDBMS for the system.
SessionID		SessionID is the current session ID.
UserId	string	UserId is the Longview User ID for the user in the session.
WebBridge	string	WebBridge is the web bridge for the current system.
WebBridgePath	string	WebBridgePath is the web bridge path for the current system.
WebServer	string	WebServer is the web server for the current system.

Sample Usage:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim rc As Long: rc = LVAPI.ErrorCode.NoError
Dim session As New session
rc = LVAPI.GetSession(session)

If rc = LVAPI.ErrorCode.NoError Then

        MsgBox session.Host & " " & session.Identifier & " " &
session.UserId

ElseIf rc = LVAPI.ErrorCode.NotConnected Then

        MsgBox "Not connected" & vbLf & "Return code " & rc, vbOKOnly +
vbExclamation, "Not Connected"

Else

        MsgBox "Error encountered" & vbLf & "Return code " & rc, vbOKOnly +
vbExclamation, "Error encountered"

End If
```

Symbol Class

The Symbol class contains the results to API.SymbolInfo.

Property Summary

- Property AcceptRollup As Boolean
- Property BalanceType As String
- Property Children As Symbol()
- Property ChildSort As String
- Property Description As String
- Property Index As Long
- Property IsLeaf As Boolean

- Property Level As Long
- Property Name As String
- Property NumOfChildren As Long
- Property Parent As Symbol
- Property Priority As Long
- Property SymbolType As String
- Property TotalDescendants As Long
- Property Virtual As Boolean
- Property Weight As String

Properties

Parameter	Description
AcceptRollup	Property AcceptRollup As Boolean Indicates whether the specified symbol receives rollup data from its child symbols. Possible values for AcceptRollup include the following: ■ TRUE — The symbol receives rollup data from its child symbols. ■ FALSE — The symbol does not receive rollup data from its child symbols.
BalanceType	Property BalanceType As String The balance type of the specified symbol. Possible values for BalanceType include the following: ■ Debit — The symbol is a debit symbol. ■ Credit — The symbol is a credit symbol. ■ Neither — The symbol is not a credit or a debit symbol. **Note:** Debit and Credit balance types apply to symbols in the ACCOUNTS dimension only. Symbols in all other dimensions have a balance type of Neither.
Children	Property Children As Symbol() The immediate child symbols for the specified symbol.

Parameter	Description
ChildSort	Property ChildSort As String Indicates how child symbols of the specified symbol are sorted in the hierarchy. Possible values for ChildSort include the following: ■ Manual — Child symbols are sorted manually by symbol priority. ■ Ascending — Child symbols are sorted by name in alphabetical order. ■ Descending — Child symbols are sorted by name in reverse-alphabetical order.
Description	Property Description As String The English language description of the specified symbol.
Index	Property IsLeaf As Boolean Indicates whether the specified symbol is a leaf symbol.
Level	Property Level As Long The number of levels the specified symbol is below the root symbol of the current symbol query.
Name	Property Name As String The symbol name of the specified symbol.
Parent	Property Parent As Symbol The parent symbol of the specified symbol.
Priority	Property Priority As Long The specified symbol’s priority. Priority is a number that designates a symbol's position in the hierarchy relative to its parent. Symbols are listed in order of ascending priority, with zeroes falling at the bottom of the list. A symbol can have multiple parents and a different priority relative to each parent.

Parameter	Description
SymbolType	Property SymbolType As String The symbol type of the specified symbol. Possible values for SymbolType include the following: ■ Standard — Standard symbols roll up normally (for example, months adding to quarters adding to years). Most symbols are standard. ■ Carry Forward — Carry forward symbols roll up the last child time period symbol's value to its parent. This symbol type is used only for account and time period symbols to identify rollup behavior of business accounts in the time periods dimension. ■ Static — Static symbols do not roll up to the parent symbol in any dimension. They override the symbol types of any symbol that they intersect with in other dimensions.
TotalDescendants	Property TotalDescendants As Long The total number of parent and child symbols below the symbol in the hierarchy.
Virtual	Property Virtual As Boolean Indicates whether the specified symbol is a virtual symbol. A virtual symbol is a parent symbol whose data is not stored in the data server repository like other symbols. Instead, the value for the parent is calculated on the server side when it is queried. The data for virtual parent symbols is calculated only when specifically requested. Symbols for quarters in the TIMEPER dimension are often virtual parent symbols. Possible values for Virtual include the following: ■ TRUE — The symbol is a virtual symbol. ■ FALSE — The symbol is not a virtual symbol.
Weight	Property Weight As String Indicates the effect that the specified symbol has on its parent. Possible values for Weight include the following: ■ 1 — The symbol is added to its parent. ■ -1 — The symbol is subtracted from its parent. ■ 0 — The symbol has no mathematical effect on its parent.

Sample Usage:

```
Dim LVAPI As Longview.API
Set LVAPI = New Longview.API
Dim symbol As Longview.symbol
```

```
Set symbol = New Longview.symbol
Dim result As Variant
result = LVAPI.RetrieveSymbolInfo("TIMEPERIODS", "A2020", symbol)
Range("I8").Value = symbol.Name
Range("J8").Value = symbol.Description
Range("K8").Value = symbol.BalanceType
Range("L8").Value = symbol.TotalDescendants
Range("M8").Value = symbol.Weight
Range("N8").Value = symbol.SymbolType
```

SymbolQuery Class

The SymbolQuery class represents a symbol query that is built from the menu option: Symbol Query. The functions and properties of this class can be used to build a new symbol query, modify an existing symbol query and output the results to a worksheet.

Property Summary

- Property SymbolOutput As SymbolOutput

Sub/Function Summary

- Sub AddSymbolSpec(symbolSpec As String, [root As String])
- Sub ClearSpecs()
- Sub LoadQuery(fileName As String)
- Sub RunToWorksheet([worksheetName As String], [cell As String])

Properties

Parameter	Description
SymbolOutput	Property SymbolOutput As SymbolOutput Specifies whether the query results include symbol name, symbol description or both. ■ SymbolOutput_Name – Query results display symbol name. This is the default. ■ SymbolOutput_Description – Query results display symbol description SymbolOutput_NameAndDescription - Query results display both name and description.

Subs/Functions

Parameter	Description
AddSymbolSpec	Sub AddSymbolSpec(symbolSpec As String, [root As String]) Adds a symbol spec representing the hierarchy you wish to query. ■ Parameters: ◦ symbolSpec – String representing a single symbol spec. For example: TRIALBAL#99. ◦ root – Optional string representing the root symbol name for the specified symbol. ■ Errors: ErrorCode.INVALID_PARAMETER - This error is thrown when the symbol spec is invalid or when the root is not a valid root for the specified symbol.
ClearSpecs	Sub ClearSpecs() Removes all symbol specs that have been added or loaded in the SymbolQuery.
LoadQuery	Sub LoadQuery(fileName As String) Loads a query spec that has been previously built and saved from the menu option: Symbol Query ■ Parameters: filename - String representing the full path to the query file to load. For example: `C:\SymbolQuery.lvqsq`. ■ Errors: ErrorCode.INVALID_PARAMETER - This error is thrown when the file cannot be loaded.

Parameter	Description
RunToWorksheet	Sub RunToWorksheet([worksheetName As String], [cell As String]) Runs a symbol query and places the results into the specified worksheet and cell location. ■ Parameters: ◦ worksheetName – Optional string representing the name of the worksheet to place the results. If the worksheet does not exist, it will be created. If the worksheetName parameter is not specified, the current worksheet will be used. ◦ cell – Optional string representing the cell location to begin the symbol query. If the cell parameter is not specified, the default cell location A1 will be used. ■ Errors: ErrorCode.INVALID_PARAMETER - This error is thrown when the query is invalid or when the specified worksheet name or cell are invalid.

Sample Usage:

```
Sub RunSymbolQuery()
On Error GoTo ErrorHandler
Dim query As Longview.SymbolQuery
Set query = New Longview.SymbolQuery
query.LoadQuery "C:\MySymbolQuery.lvqdq"
query.SymbolOutput = SymbolOutput_Name
query.RunToWorksheet "MySymbolQuery", "B1"
ErrorHandler:
If Err.Number <> 0 Then
MsgBox "Unable to run Symbol Query (" & Err.Number & ") - " &
Err.Description
End If
End Sub
```

AttributeQuery

The QueryResult class contains the results to an AttributeQuery. The results are structured as a table where each row represents an attribute entry. The columns represent the properties for that attribute.

Property Summary

- Property ColumnCount As Long
- Property RowCount As Long

Sub/Function Summary

- Function GetCellValue(row As Long, column as Long) As String

Properties

Parameter	Description
ColumnCount	Property ColumnCount As Long Gets the number of columns in the query result. ■ If querying attribute definitions, there are 6 columns. ■ If querying attribute values, there are 7 columns. You can use this information to help you write your query to a worksheet or iterate through the columns to do some processing in your VBA code.
RowCount	Property RowCount As Long Gets the number of rows in the query result. Each row represents an attribute entry.

Subs/Functions

Parameter	Description
GetCellValue	Function GetCellValue(row As Long, column As Long) As String Returns the string value at the specified row/column. Each row represents a attribute while the columns are the properties of the symbol as follows: ■ Attribute Definitions ◦ Column 1 – Attribute class ◦ Column 2 – Attribute name ◦ Column 3 – Attribute description ◦ Column 4 – Attribute type ◦ Column 5 – Attribute access (Read/Write) ◦ Column 6 – Attribute default value ■ Attribute values ◦ Column 1 – Attribute class ◦ Column 2 – Attribute name ◦ Column 3 – Attribute description ◦ Column 4 – Attribute type ◦ Column 5 – Attribute object (user name, symbol name) ◦ Column 6 – Attribute value ◦ Column 7 – N/A for default, Y/N whether value is the same as the default. ■ Parameters: ◦ row – Long representing row number co-ordinate ◦ column – Long representing the column number co-ordinate ■ Errors: ErrorCode.INVALID_PARAMETER - This error is thrown when the specified row or column is outside the range of valid values.

Sample Usage:

```
Sub RunAttributeQuery()
On Error GoTo ErrorHandler
Dim result As Longview.QueryResult
Dim query As Longview.AttributeQuery
Set query = New Longview.AttributeQuery
Set result = query.Run()
Dim row As Long
```

```
Dim column As Long
For row = 1 to result.RowCount
For column = 1 to result.ColumnCount
' Place results of symbol query into cells of the worksheet manually
Range("A1").Offset(row - 1, column - 1).Value = result.GetCellValue(row,
column)
Next column
Next row
ErrorHandler:
If Err.Number <> 0 Then
MsgBox "Unable to run Attribute Query (" & Err.Number & ") - " &
Err.Description
End If
End Sub
```

HierachyQuery

The QueryResult class contains the results to a HierarchyQuery. The results are structured as a table where each row represents a symbol entry, and the columns represent the properties for that symbol.

Property Summary

- Property ColumnCount As Long
- Property RowCount As Long

Sub/Function Summary

- Function GetCellValue(row As Long, column as Long) As String

Properties

Parameter	Description
ColumnCount	Property ColumnCount As Long Gets the number of columns in the query result. ■ If IncludeDescriptions = True, there will be 4 columns. ■ If IncludeDescriptions = False, there will be 2 columns. You can use this information to help you write your query to a worksheet or iterate through the columns to do some processing in your VBA code.

Parameter	Description
RowCount	Property RowCount As Long Gets the number of rows in the query result. Each row represents a symbol entry.

Subs/Functions

Parameter	Description
GetCellValue	Function GetCellValue(row As Long, column As Long) As String Returns the string value at the specified row/column. Each row represents a symbol entry while the columns are the properties of the symbol as follows: ■ IncludeDescriptions = True ◦ Column 1 – Symbol Name ◦ Column 2 – Symbol Description ◦ Column 3 – Parent Symbol Name ◦ Column 4 – Parent Symbol Description ■ IncludeDescriptions = False ◦ Column 1 – Symbol Name ◦ Column 2 – Parent Symbol Name ■ Parameters: ◦ row – Long representing row number co-ordinate ◦ column – Long representing the column number co-ordinate ■ Errors: ErrorCode.INVALID_PARAMETER - This error is thrown when the specified row or column is outside the range of valid values.

Sample Usage:

```
Sub RunHierarchyQuery()
On Error GoTo ErrorHandler
Dim result As Longview.QueryResult
Dim query As Longview.HierarchyQuery
Set query = New Longview.HierarchyQuery
Set result = query.Run()
Dim row As Long
Dim column As Long
For row = 1 to result.RowCount
```

```
For column = 1 to result.ColumnCount
' Place results of symbol query into cells of the worksheet manually
Range("A1").Offset(row - 1, column - 1).Value = result.GetCellValue(row,
column)
Next column
Next row
ErrorHandler:
If Err.Number <> 0 Then
    MsgBox "Unable to run Hierarchy Query (" & Err.Number & ") - " &
Err.Description
End If
End Sub
```