# Table of Contents

Introduction ................................................................................................. 3
Overview ...................................................................................................... 4
Key Features ............................................................................................... 5
Latest Features .......................................................................................... 5
Installation .................................................................................................. 5
Pre-requisites ............................................................................................. 5
Instructions ................................................................................................. 5
Contacting Support .................................................................................. 6
Getting Started .......................................................................................... 7
General | Login .......................................................................................... 7
Connecting to Global Data Workstation ..................................................... 7
Connecting to Global Model Workstation .................................................. 8
How to create an XLSM File ...................................................................... 11
Configure Proxy ......................................................................................... 12
Reload and Load your saved selections ...................................................... 13
Downloading your online selection ............................................................ 14
Excel Tab Names Explained ....................................................................... 16
Data Series Group ..................................................................................... 16
Local model database connection ............................................................. 16
Refreshing Your Worksheet ...................................................................... 19
Chart Commands ...................................................................................... 20
Split on command ...................................................................................... 21
Reshape & Fold data .................................................................................. 21
Export worksheet data to E-Views ............................................................. 22
Website and Databank .............................................................................. 22
Latest Analysis ........................................................................................... 23
Scripted Commands .................................................................................. 24
Example Scripting Commands .................................................................. 24
Frequency = Annual ............................................................................... 24
Retrieve save selection by scripted commands ...................................... 25
Script Command directly from a chosen data-bank ................................. 26
Selecting columns from an active worksheet .......................................... 26
Introduction

This document describes the Oxford Economics Excel Data Workstation, an Excel add-in application used to connect to, explore and analyse information in the Global Data Workstation and Global Model Workstation.

Excel Data Workstation is one of three complementary platforms which make up the Oxford Economics Global Data Services. The other two platforms are Global Data Workstation and Global Model Workstation. Together they provide economic data and analysis functions covering 200 countries, 100 industries and over 3,000 cities and sub-regions.

Customers must have a subscription to one or more of the following products/services to take advantage of the facilities in Excel Data Workstation:

- Global Data Workstation
- Global Model Workstation

Global Data Workstation (GDW): Is an online web-based system currently serving 19 pre-packaged databanks covering several macro, industry, cities and labour economics domains. GDW provides a variety of data selection, heat map and chart visualisations, data downloads, data sharing and personalisation features. All databanks are updated periodically and the intention is to add more databanks over time.

Global Model Workstation (GMW): This personal desktop application provides a variety of user driven economic forecasting, scenario modelling, and visualisation capabilities supported by periodic updates of the underlying macro and industry model bases.

Excel Data Workstation Big Picture
Excel Data Workstation (EDW): Is a simple but powerful Excel add-in which allows the user to connect to GDW and GMW, and bring data selectively into Excel worksheets. Data from these different systems can be intermingled in the same worksheets, saved and then refreshed easily at a later time when the origin source data is periodically updated.

Overview

The pictures below explain how EDW fits into a data flow originating with data in GDW and GMW, and Excel worksheets themselves, finally ending with Excel tables and charts of your own custom analyses. Essentially, EDW bridges the gap between economic data sources and insightful results… right where you usually need it, in Excel.
Key Features

- 44 high quality charts in five dashboards of key economic forecast indicators.
- Dashboards available for both annual and quarterly data series.
- Dashboards cover all locations and 36 indicators in all measurements.
- Retrieves forecast data series from either Global Data Workstation or Global Model Workstation. On-demand refresh of data using built-in automation of Excel Data Workstation commands.

EDW v3.0 reproduces all the built-in dashboards of the new Global Model Workstation and Global Data Workstation:

- Forecast/Scenario Overview
- Forecast/Scenario Overview
- Inflation and its determinants
- Trade and balance of payments
- Labour market

Latest Features

- Direct model DB connectivity, including residuals, stacked quarters layout, and multiple measurements
- Direct export of sheet data to model DB and E-Views, and simple browser for latest analysis in the user’s subscription (currently RBs, DI's, and CEF's and can be extended to other products if needed)
- Latest reports can be downloaded and opened from within Excel
- Scripting which works with the new model DB connection
- High quality charts
- Zero installation – double click to run (We are looking to improve this to add desktop shortcuts)

To keep up to date with latest developments, tips and case studies using EDW, please visit Oxford Economics TechLabs Blog

Installation

You can download the Excel workbooks and get going immediately. Of course, you need the latest version of our Excel add-in but the dashboards themselves run OOTB - out-of-the box. We placed the big dashboard refresh button in a more obvious position and improved the styling of pivot charts, so they display properly across all supported Excel platforms, regardless of the number of data series being plotted.

Pre-requisites

Before you install the Excel Plug-in, your computer should meet the following requirements:

1. Microsoft Windows versions greater than Windows XP (i.e. will not work in Windows XP)
3. Microsoft .NET 4.5 (will be installed if not present)
4. An active internet connection to use the online Global Data Workstation

Instructions

To run the new Excel Plug-in, simply double click on the .XLL file and Excel will open with the Oxford Economics menu loaded. The plug-in will then automatically load the next time you open Excel.

*(We strongly recommend you uninstall your previous version of the Excel Plugin, or disable it in Excel’s COM Add-ins menu.)* If you require assistance, please do not hesitate to contact our Support team on support@oxfordeconomics.com

If you wish to run the current Mondrian Dashboard Workbooks (available for Macro, Industry, GSS, Brexit databanks/models), then please contact support@oxfordeconomics.com to find out more information.

The new version of the plugin can be uninstalled at any time by clicking on General > Uninstall Add-in. We recommend restarting Excel after doing this to complete the process.
Contacting Support

If you encounter any problems with Excel Data Workstation, please contact Oxford Economics support at support@oxfordeconomics.com.

Please ensure you attach a copy of your log file as this will really help the team diagnose the issue you are experiencing. You can find the Log file by clicking on General > View Log Copy the selected text and paste it into an email and attach it to your email.

Here you will also be able to access the User Guide for your reference.
Getting Started

Once installed, an Oxford Economics menu item will appear in Microsoft Excel. Click on this menu to show a ribbon holding a number of functions as shown below. We recommend you mouse over the controls to read the detailed tooltips provided. This is a good way to quickly get an idea of the features available.

The ribbon is divided into logically related groups of functionalities, e.g. General, Connection, Refresh, and Data Series. Each group has one or more controls used to perform actions, enter information, make selections, or indicate status information.

We describe the function of each control in detail below and show examples of applying some of the more advanced features.

The first time a control is mentioned, or if it is mentioned out-of-context, we’ll use the naming convention Group-Name | Control-Name to refer to it, otherwise we’ll simply call it Control-Name.

General | Login

We must ensure you have established a connection to both the Global Model Workstation and Global Data Workstation before we proceed.

You must establish a connection to at least one of these data sources in order to pull data into Excel Data Workstation.

Connecting to Global Data Workstation

A connection to Global Data Workstation requires online login authentication, so you must firstly have access to the internet from Excel Data Workstation to do this.

Having confirmed internet connectivity, click General | Login. In the pop-up dialogue box enter your credentials which are the same username and password you normally use to log into My Oxford (the subscription content area in Oxford Economics Website) and online databank (Global Data Workstation).
You will have to log in the first time after installing Excel Data Workstation. Thereafter, your credentials will be remembered between different Excel sessions. If your credentials change or you log out deliberately, then you'll have to log in again to use EDW with GDW as a data source.

**Connecting to Global Model Workstation**

To establish a connection with the Global Model Workstation, the application must be started and directed to the View and Download Data window. To do this from the main menu, select the databank you would like to load and click View Data.

If you start GMW after Excel, then you need to perform a Selections | Reload to force a connection attempt, as described below. Do the same again if you happen to close and then reopen GMW mid-session, and you wish to re-connect EDW. Please ensure you are also logged into GMW.
You can check this by clicking on Help > Login from GMW homepage.

After successfully logging into Global Data Workstation and/or detecting connectivity to Global Model Workstation, additional functions are revealed. You should also notice their relevant connection statuses will be updated from Red to Green as shown below.

In the latest version the plug-in automatically detects when the desktop source is currently available and handles erroneous calls appropriately.

If you log out of Global Data Workstation by clicking General | Logout, the connection with it will be terminated and you will of course have to log back in to work with it. You should notice the connection status of GDW will be updated from Red to Green. If you are using the latest version, you will be prompted to log back into GDW when you click on Global Data Workstation.
In the latest version the plugin automatically detects when the desktop source is currently available and handles erroneous calls appropriately.

If you log out of Global Data Workstation by clicking General | Logout, the connection will be terminated and you will of course have to log back in to work with it. If you are using the latest version you will be prompted to log back into GDW when you click on Global Data Workstation.
How to create an XLSM File

Extensive automation of Excel Data Workstation ribbon commands is available through a Visual Basic for Applications (VBA) interoperability interface, aka Automation API. You will read about the Automation API later in this user guide and take a look at the provided classes, modules and example programs.

EDW is equipped with a built-in VBA macro-enabled template file (.xltm) from which you can create your own macro-enabled workbooks (.xlsm). This workbook has everything you need, including example programs mentioned above, for invoking the Automation API to control EDW ribbon commands.

A new macro-enabled workbook is created by clicking on General | Arrow to open the About box and then choosing Create XLSM File. These actions are shown in the image below and will produce a new worksheet, tentatively called Excel Data Workstation VBA InteropN, which will be saved as soon as you’re ready.

After closing the open dialog boxes, take some time to do the following:

1. Click Alt-F11 to open Visual Basic for Applications development tools.
2. Select and open the module AcropolisAutomationExamples which contains example macro programs to automate Excel Data Workstation by performing various commands normally available through its ribbon buttons.
3. All ribbon commands can still be used manually, as normal.
4. When you decide to save the new workbook, you must do so as a macro-enabled Excel file, with an .xlsm file extension.
5. When you open your file, you will be prompted as usual to Enable Content, i.e. to enable macros (unless you had saved it earlier in a ‘trusted location’).
Configure Proxy

If you're on a corporate network which is protected through a ‘proxy server’ (something which is quite common in medium to large enterprises), you may need to configure EDW’s proxy settings. Click on General > Configure Proxy to open a dialogue box as shown below.

Here you will find several proxy mode settings which you can choose and test individually for connectivity. If you find one that works, close out the windows and return to EDW. If you still can’t connect, please contact your in-house IT support team for assistance with the custom proxy settings.
Reload and Load your saved selections

To be able to use the Load command from the Excel Plug-In you must have already saved selections under your Global Data Workstation account. If you do not currently have any saved selections, please refer to our Global Model Workstation User Guide, Under View data > Save selections and settings.

Once you have saved the selections you require, you can now begin to pull the data from the Excel Plug-In by clicking on Reload then Load. Please note, if you delete, change or rename saved selections at source, please remember to perform a Reload in EDW.

Clicking the Load menu (down arrow) shows a merged list of saved selections from both GDW and GMW. Plain icons denote the online (GDW) saved selections, whilst icons with a yellow mark denote local desktop (GMW) saved selections (the latter are also appended with ‘local’ in parentheses). However, we still recommend you use a clear naming convention so as not to confuse local and online selections with each other.

The Excel Plug-In automatically monitor connections, so Reload serves another important purpose which is to manually check and re-establish connectivity with available data sources. This comes handy when you wish to force a connection with Global Model Workstation in case it was started after Excel was opened.
Downloading your online selection

Clicking on **Load** will display your list of recently saved selections. As mentioned above plain icons denote the online (GDW) saved selections, whilst icons with a yellow mark denote local desktop (GMW) saved selections.

Clicking on an online saved selection will prompt you to complete the saved selection details before you download your data.
In the configure download pop up you are given the opportunity to change both the format and layout of your data.

<table>
<thead>
<tr>
<th>Configuration Option</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Colour Data Series Values</strong></td>
<td>This option allows to add a colour to distinguish to series easily at the top of your spreadsheet</td>
<td><img src="image1" alt="Example Image" /></td>
</tr>
<tr>
<td><strong>Use Default Template</strong></td>
<td>Selecting this option removed any colours that can separate your data</td>
<td><img src="image2" alt="Example Image" /></td>
</tr>
<tr>
<td><strong>Transpose Data Series Values</strong></td>
<td>Transpose creates a new data file in which the rows and columns in the original data file are transposed so that cases (rows) become variables and variables (columns) become cases</td>
<td><img src="image3" alt="Example Image" /></td>
</tr>
<tr>
<td><strong>Fold Data Series Values</strong></td>
<td>Converts a horizontal data series layout to a vertical (columnar) data series layout and vice versa</td>
<td><img src="image4" alt="Example Image" /></td>
</tr>
</tbody>
</table>
Excel Tab Names Explained

The name given to the tab when you load a saved selection. Technically, it should have the same name as the saved selection. For instance, in the examples earlier the saved selection is called ‘GBR, USA / Key Indicators (A)’, but the tab’s name is ‘GBR, USA _ Key Indicators (A)’. Since ‘/’ character is not allowed in Excel tab names, Excel Data Workstation has replaced with the underscore ‘_’ character.

This is not a major problem, just something you must be aware of. Similarly, ‘[’, ‘]’ characters will be replaced by Excel Data Workstation with ‘_’.

Another issue is Excel’s 31-character limit for tab names. If your saved selection name is greater than 31 characters, Excel Data Workstation first truncates it to 28 characters and then appends ‘…’ to the end of it. When worksheets are treated as data sources they are referenced by their literal names. Therefore, it is important to use the name as it appears in the tab. If you have many tabs open or a narrow screen, the name in the tab will be truncated further by Excel, and this name becomes the new name of the tab and its new literal reference! See below image as an example:

Data Series Group

We’ve upgraded the user interface, so that you can keep the various add series panels open simultaneously!

Here you can make ad-hoc selections of data series from any connected data source. As seen in Figure 19 you can choose individual databanks, manually type or choose location and indicator mnemonics, apply period and frequency settings, and select what measurements you require.

This part of the ribbon allows you to make ad-hoc selections of data series from both a connected data source and a local model database file. You can choose individual databanks, manually type or choose location and indicator mnemonics, apply period and frequency settings, and select what measurements you require.

Within reason, Excel Data Workstation will do its best to accommodate slight variations in column headers found in different databanks and model databases from different economic domains. It will also sensibly fit data retrieved with different period and frequency settings into the same table.

New ad-hoc data series downloads will be appended to current active worksheet, including worksheets containing previously retrieved saved selection data. This additive behaviour allows you to combine data series from different data sources and from different databanks in the same worksheet.

Local model database connection

Here you can import data directly from Oxford Economics’ proprietary .db files with all the same formatting options as is available with the Global Data and Global Model Workstations.
Each databank has a unique set of locations and indicators. You can enter location and indicator mnemonic codes, or you can use the browse buttons to make selections in a tree holding less terse names for the same concepts.

*Please note, these typically do not match the location and indicator codes used by the Global Data Workstation or Global Model Workstation.*

Near the bottom of the Add Data Series pane is a checkbox indicating whether or not your base is in your desired model directory. Check or uncheck this parameter as is necessary and select the model directory as is appropriate.
After you've selected the required frequency, date interval and measurement, click **Add Data Series**.

You will now see the results you selected in the Local model database connection. You can also make your data selection using the same process for both Global Data Workstation and Global Model Workstation loaded in the same sheet.
Refreshing Your Worksheet

If you are working on a worksheet that is already populated with data from a connected data source, i.e. a downloaded saved selection or from using a data series command, you can easily refresh this data by simply clicking on Current Worksheet.

You must ensure that you are:

1. Connected to the original data source(s)
2. The required saved selections exist on your chosen data sources
3. You must have a command script in your worksheet.

If you have numerous worksheets within the same workbook, you can also easily refresh all the data by clicking on All Worksheets.

Worksheets, charts, workbooks and applications that are 'refreshable' will stay in sync with the latest refreshed data. This will also depend on the formulas, calculations and calculation options specified in your workbooks.

Please note that the refresh action has no affect if you are working from a workbook that has no script commands. If you would like to add script commands, please go to Scripted Commands.
Chart Commands

To use chart commands in any of your worksheets, you must ensure that you have Location and Indicator columns to the left and your numerical data displayed horizontally.

Once you have your data in the correct format, you can now highlight your data range and click on Lines when you are ready.

Click on the chart to make the changes you require. For example, chart styles, chart elements and chart filters.
Split on command

This command will work with any data table provided it has a header in row 1 of the active worksheet. It is not specific to data worksheets created in Excel Data Workstation.

A typical scenario is to download a manageable chunk of data from GDW or GMW using Selections | Load commands and then splitting-out the worksheet either by Location or Indicator. Excel Data Workstation takes a general approach and will split-out the worksheet by the data values of the column in which you have made a cell selection.

<table>
<thead>
<tr>
<th>Steps</th>
<th>Example</th>
<th>Explanation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select where you would like your data to be split. For example, by indicator.</td>
<td></td>
<td>Selecting where you would like to split your data and clicking on Split on Column will prompt the plug-in begin to separate your selection into different sheets.</td>
</tr>
<tr>
<td>Once you have clicked on Split on Column you will notice the plug-in beginning to split your data by adding sheets named depending on your data split selection</td>
<td></td>
<td>The plug-in will separate your data into different worksheets depending on your selection.</td>
</tr>
<tr>
<td>You can also refresh your 'Split data worksheets' by simply clicking on Current Worksheet from your original worksheet</td>
<td></td>
<td>Clicking on Current Worksheet will not only refresh your original worksheet but also your split data worksheets.</td>
</tr>
</tbody>
</table>

Reshape & Fold data

This command is effective on tabulated data series having several explicit years (or year/quarter) columns, e.g. 2016, 2017, or 201801, 201802, 201803 etc. You can very quickly convert your data from horizontal data series to vertical data series by a click of a button.

It is a very complicated command for our programmes and machines to manipulate. For example, each year is a dimension (column header) whose measures (column data) are the data series values, which is a conflation of concepts. In fact, for data pivoting and other analytical processing procedures, having a unique Year dimension and unique Data dimension with consistent measures (data values) is preferable.

The purpose of Fold is collecting all the year columns into one column called Year and gather all the data series values into another column called Data. These new columns can now be considered as first-class dimensions with explicit measure data. The previously implicit nature of dimensions and measures has been eliminated in favour of something more explicit.

The effect on the shape of the table is to take it from being a wide table of many year columns to a narrow table of one-year column and one data column. The folded table becomes much longer too. It also has more data overall as the other columns' data needs to be repeated for each year value. To preserve the uniqueness of the rows, Excel Data Workstation will generate a unique ID for each row (a 1-to-1 relationship) and each series (a 1-to-Many relationship).
Simply click on **Toggle Fold** and the plug-in will automatically begin to fold your data.

If you wish to undo your data fold, simply click on **Toggle Fold** again and your data will revert to its original worksheet layout.

### Export worksheet data to E-Views

Export worksheet data to E-Views, E-Views-compatible worksheets, and to Oxford Economics database files (note: series that are to be exported to a .db file must be imported from an existing .db file).

![Excel Export Worksheet](image)

### Website and Databank

1. Click on **My Oxford** this will take you directly to the website at a click of a button.

2. Click on **Databank** is context sensitive and will take you directly to the Edit Selections page in Global Data Workstation for the databank associated with the active worksheet, provided it is unambiguous. Otherwise this command will take you to the home page of GDW.
Latest Analysis

Get the latest Oxford Economic reports from the website directly within excel.

Clicking on the Latest Analysis tab you will be prompted to log into your account using your usual Oxford Economic credentials you use to log into the website.

If you are having trouble logging into your account, please do not hesitate to contact our Support Team as they will be able to provide you with your credentials.

From here, you can further extend the list of available reports by clicking the more button. You can filter this list by report type using the drop down or by keyword search. This search checks text entered against title, author, region, as well as the abstract.
Scripted Commands

These libraries allow users to work with Oxford Economics data in a convenient, automated way. This includes most Excel Add-in functionality. When a worksheet is generated using the built-in commands from the Selections and Data Series groups, Excel Data Workstation records a script representation of the commands used and permanently associates them with the worksheet. These scripts capture the command and its specific parameter settings.

The benefit of scripts is you can read, edit and run them using the Scripting | Edit window. You can save script files and share them with others who also use Excel Data Workstation. Saved script files can, of course, be opened then edited and run thus affording a flexible workflow around the data you're interested in.

Scripts are saved with the Excel worksheet, so they're not lost when the workbook is closed. When you reopen the workbook later, you can replay the script to refresh and replace the data in the active worksheet, which is equivalent to clicking Refresh | Current Worksheet.

In addition to replacing the data in the active worksheet, you can append it to the current worksheet or replace a worksheet. To cycle through each worksheet's commands set, use the dropdown bar located at the top-left corner of the scripts window. You can also import and export scripts to and from file as well as save scripts to sheet without having to run them.

Example Scripting Commands

Scripts give the ability to treat any Excel Data Workstation worksheet as a data source (alongside Global Data Workstation and Global Model Workstation data sources).

Create a new saved selection of your choice depending on the subscriptions you have. For example,

Databank = Global Economics Locations = USA, GBR

Indicators = All Key Variables

Measurement = Level Values

Frequency = Annual

Period = 2017 to 2027
Ensure you **Reload** your saved selections in the plug-in before you continue. Now you can click on your recently saved selection from the dropdown list under **Load**.

*Please note that excel will slightly change your saved selection tab name.*

Retrieve save selection by scripted commands

You can specify the name of a saved selection to be used as a source of data. For example,

```
[get from ss.[Example Selection] with Template = true , ColorSheet = true , Format = Default , Transpose = false]
```

You can reuse this script command in other sheets, even blank scripts. You **must** ensure you specify the prefix `ss`, which indicates the source is a data-bank saved selection. If the name has spaces or special characters, you must enter it between square brackets. The name must exactly match the name of the saved selection on the chosen data-bank.
Script Command directly from a chosen data-bank

When querying individual data series using a script, specify the name of a databank (as opposed to a saved selection) to be used as the source of data:

```plaintext
get @level from db.[Global Economics] where Location in 'GBR', 'USA' and Indicator in 'CPI', 'GC', 'C' and Year from 2013 to 2023 with Frequency = 'Annual'
```

Specifying db, indicates the source is a databank. If the name has spaces or special characters, enter it between square brackets. The name must exactly match the name of the chosen databank.

Specify the locations and indicators to be retrieved, and the year range. The frequency is either Annual or Quarterly.

Selecting columns from an active worksheet

Selecting columns from an active worksheet

The select command is used to retrieve data from the active worksheet. In other words, a worksheet can serve as the data source.

The select command can specify which columns should be retrieved. So, this command copies only Location and Indicator columns, and implicitly all year data:

```plaintext
select Location, Indicator
```

If no column is specified, all the columns are retrieved. So, this command effectively copies the entire active worksheet:

```plaintext
select
```

Filter a specific row

Filter data using the where clause:

```plaintext
select Location, Indicator where Year = 2018
```

Specify a range like below:

```plaintext
select Location, Indicator where Year from 2018 to 2027
```

Or, specify many values:

```plaintext
select Location, Indicator where Year in 2018, 2019, 2022
```

Multiply scripted commands

Enter and run multiple commands separated by semicolons:

```plaintext
select Location, Indicator where Year = 2018; select Location, Indicator where Year = 2019
```

Individual commands are not required to be on separate lines so long as they are separated by a semicolon from each other.
Get Data from Named Worksheet

To query a specific worksheet and use it as source of data, specify the literal name of the worksheet exactly as it appears in the Excel tab. Use this syntax to query several worksheets at once, and can’t rely on the default behaviour of assuming the current active worksheet is the data source.

```sql
select Location, Indicator from ws.[GBR, USA _ Key Indicators (A)]
```

You must specify the prefix `ws`, indicating the source is a worksheet. If the name has spaces or special characters, enter it between square brackets. The name must exactly match the name of the chosen worksheet (Excel tab name).

Differencing Data Series

Use the `diff` operator to join two select commands. The ‘Year’ column / dimension must be included in the select sentence.

```sql
select Indicator, Year where Location = "United States" and [Indicator code] = "GDP$" diff select Indicator, Year where Location = "United Kingdom" and [Indicator code] = "GDP$"
```

If more than one data series (i.e. a set) is returned by any of the select clauses on either side of the `diff` operator, the data series’ year column values will be aggregated before `diff` is applied. The default aggregation mechanism is summation.

If this aggregation is not desirable, then supply additional filter criteria to whittle down the selection set to a single series which won’t require aggregation prior to applying `diff`.

Take care which column selectors (aka projections) you choose as they will be aggregated (or combined) where possible.

Experimenting a little with simple select commands using various projection combinations and where filter clauses helps a lot to build up bigger scripts.

Plotting Data Series

Two simple plot command options are available, namely lines and bars. The plot command will chart the result set produced from applying select.

```sql
select Location, Indicator, Year where Location = "United States"; plot as lines; plot as bars
```

Intermediate select results can be stored away in a variable with into. This variable can subsequently be handed to plot for charting.

```sql
select Location, Indicator, Year from ws.[GBR, USA _ Key Indicators (A)] where Indicator = "GDP, real, LCU" and [Location code] = 'GBR' into UKGDP; select Location, Indicator, Year from ws.[GBR, USA _ Key Indicators (A)] where Indicator = "GDP, real, LCU" and [Location code] = 'USA' into USGDP; plot UKGDP as lines; plot USGDP as bars;
```
Applying built in functions

You can specify a function to apply to retrieved data also. select Location, Indicator, @level from [My Worksheet]; select Location, Indicator, @diff from [My Worksheet]; select Location, Indicator, @pch from [My Worksheet]

Preserving user-defined columns

While the Excel Data Workstation reserves the right to clear the entire data worksheet on refresh, it is still possible to establish a set of user-defined columns that will be preserved between such commands. Currently, the cells of these columns must be formulas (i.e. fixed-value columns will not be copied over).

`preserve 'first column name', 'second column name';`

To build more complex tables, the preferred method is to use Power Query to construct a pivot table on top of the raw Oxford Economics data worksheet. Any calculations or custom formulas can be built off of the pivot table and calculated automatically on refresh. A simple VBA macro can then be used to automate the refresh of both the data sheet and Power Query table.

The functions corresponding to Excel Data Workstation measure settings are:

- `@level`: the original value (quarterly and annual data)
- `@diff`: difference between consecutive values in time (quarterly and annual data)
- `@pch`: % change between consecutive values in time (quarterly and annual data)
- `@pchy`: % change between y-to-y values in time (quarterly data only)
- `@pach`: % annualised change between y-to-y values in time, i.e. growth (quarterly data only)
- `@max`: maximum value
- `@min`: minimum value
- `@avg`: average value
- `@sum`: sum of values (default function in scripting command)
- `@data`: no summing (i.e. no aggregation) of values
- `@now`: when used instead of a literal year value, it will use the current year. Simple expressions like `@now-1` and `@now+4` are admissible.

For example, the script below will return (historical + forecast) data in the period commencing the previous year and four years of ahead of the current year:

get @level from db.[Global Economics] where Location = 'GBR' and Indicator = 'CPI' and Year from @now-1 to @now+4 with Frequency = 'Annual'

Excel Data Workstation Automation

EDW is equipped with a built-in VBA macro-enabled template file (.xltm) from which you can create your own macro-enabled workbooks (.xlsm). This workbook has everything you need to program the Automation API to control EDW ribbon commands. A new macro-enabled workbook is created by clicking on General | Arrow to open the About box and then choosing Create XLSM File (see the notes earlier in this document for more details).

This file is ready for you to use right away! Create a VBA macro with the following snippet of code which is the simplest example of programmatically refreshing a named worksheet "TEST SELECTION", which had previously been populated with data using Excel Data Workstation ribbon commands.
Sub Acropolis_Eample_RefreshWorksheet()

' MUST INITIALISE ACROPOLIS (False if using a Debug release)
Acropolis = InitAcropolis(True)

' SWITCH ON POP-UP MESSAGES
IsShowingMessages = True

' # EXAMPLE: REFRESH NAMED WORKSHEET
SavedSelection = "TEST SELECTION"
Acropolis.RefreshWorksheet SavedSelection, IsShowingMessages

End Sub

VBA Project Structure

The built-in macro-enabled workbook contains various modules and classes (namely Acropolis* files) which can be imported into your own VBA projects.

A typical VBA project structure for EDW automation is shown below.

Project Classes

AcropolisDataSeriesConfig.cls – Defines an object used to configure individual data series downloads. There is also a utility method which generates a JSON string representation of the configuration object, so it can be passed easily to the compatible automation interface.
Acropolis Automation Examples Module

**AcropolisAutomationExamples.cls**

' Contains sample VBA programs exercising the complete Excel Data Workstation automation interface.

Private Sub Acropolis_Example_IsConnected()
' TO SWITCH OFF POP-UP MESSAGES...
' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
' (Where not supplied, IsShowingMessages = True is the default.)
    IsShowingMessages = True

' # EXAMPLE: CHECKING CONNECTIONS
    OnlineConnected = Application.Run("Acropolis_IsConnected", "OnlineDataWorkstation")
    DesktopConnected = Application.Run("Acropolis_IsConnected", "DesktopModelWorkstation")

' # EXAMPLE: LOG MESSAGE
' Log messages are added in Acropolis' log window
    OnlineMessage = "Online Data Workstation Connected: " & OnlineConnected
    DesktopMessage = "Desktop Model Workstation Connected: " & DesktopConnected
    Application.Run "Acropolis_Log", OnlineMessage
    Application.Run "Acropolis_Log", DesktopMessage

' # EXAMPLE: ALERT POP-UP MESSAGE
    Application.Run "Acropolis_ShowAlert", OnlineMessage, IsShowingMessages
    Application.Run "Acropolis_ShowAlert", DesktopMessage, IsShowingMessages

End Sub

Private Sub Acropolis_Example_GetSelectionsList()
' TO SWITCH OFF POP-UP MESSAGES...
' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
' (Where not supplied, IsShowingMessages = True is the default.)
    IsShowingMessages = True

' # EXAMPLE: LIST OF SAVED SELECTIONS
    Dim SS() As Variant
    SS = Application.Run("Acropolis_GetSelectionsList")

End Sub

Private Sub Acropolis_Example_Reload()
' TO SWITCH OFF POP-UP MESSAGES...
' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
' (Where not supplied, IsShowingMessages = True is the default.)
    IsShowingMessages = True

' # EXAMPLE: RELOAD SELECTIONS
' Try adding/removing saved selections in GDW/GMW before executing next statement to verify this updates the SS list
    Application.Run "Acropolis_Reload"
    Dim SS() As Variant
    SS = Application.Run("Acropolis_GetSelectionsList")
    Application.Run "Acropolis_ShowAlert", "RELOAD Saved Selections: " & Join(SS, ", "), IsShowingMessages

End Sub

Private Sub Acropolis_Example_DownloadSelection()
' TO SWITCH OFF POP-UP MESSAGES...
' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
' (Where not supplied, IsShowingMessages = True is the default.)
IsShowingMessages = True

' # EXAMPLE: DOWNLOADING A SAVED SELECTION
SavedSelection = "TEST SELECTION"
Application.Run "Acropolis_DownloadSelection", SavedSelection

End Sub

Private Sub Acropolis_Example_RefreshWorksheet()
' TO SWITCH OFF POP-UP MESSAGES...
' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
' (Where not supplied, IsShowingMessages = True is the default.)
IsShowingMessages = True

' # EXAMPLE: REFRESH WORKSHEET
Application.Run "Acropolis_ShowAlert", "About to run REFRESH WORKSHEET (Null)", IsShowingMessages

' Null will refresh the active worksheet
Application.Run "Acropolis_RefreshWorksheet", Null

Application.Run "Acropolis_ShowAlert", "About to run REFRESH WORKSHEET (TEST SELECTION)", IsShowingMessages

' Refreshing a named worksheet
SavedSelection = "TEST SELECTION"
Application.Run "Acropolis_Log", "Refreshing " & SavedSelection
Application.Run "Acropolis_RefreshWorksheet", SavedSelection

Application.Run "Acropolis_ShowAlert", "About to run REFRESH ALL WORKSHEETS", IsShowingMessages

' # EXAMPLE: REFRESH ALL WORKSHEETS
Application.Run "Acropolis_Log", "Refreshing All Worksheets"
Application.Run "Acropolis_RefreshAllWorksheets"

End Sub

Private Sub Acropolis_Example_FoldWorksheet()
' TO SWITCH OFF POP-UP MESSAGES...
' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
' (Where not supplied, IsShowingMessages = True is the default.)
IsShowingMessages = True

SavedSelection = "TEST SELECTION"

' # EXAMPLE: FOLD / UNFOLD
Application.Run "Acropolis_ShowAlert", "About to run FOLD", IsShowingMessages
Worksheets(SavedSelection).Activate
Application.Run "Acropolis_FoldWorksheet"

Application.Run "Acropolis_ShowAlert", "About to run UNFOLD", IsShowingMessages
Worksheets(SavedSelection).Activate
Application.Run "Acropolis_FoldWorksheet"

End Sub

Private Sub Acropolis_Example_SplitWorksheet()
' TO SWITCH OFF POP-UP MESSAGES...
' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
' (Where not supplied, IsShowingMessages = True is the default.)
IsShowingMessages = True

' # EXAMPLE: SPLIT
Application.Run "Acropolis_ShowAlert", "About to run SPLIT ON COLUMN NUMBER", IsShowingMessages

Application.Run "Acropolis_SplitWorksheetOnColumnNumber", Int(2)

Application.Run "Acropolis_ShowAlert", "About to run SPLIT ON COLUMN NAME", IsShowingMessages

Application.Run "Acropolis_SplitWorksheetOnColumnName", "Location"

End Sub

' It's much simpler to run a script (see Acropolis_Example_RunScript)
Private Sub Acropolis_Example_RunScript()

    ' TO SWITCH OFF POP-UP MESSAGES...
    ' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
    ' (Where not supplied, IsShowingMessages = True is the default.)
    IsShowingMessages = True

    ' # EXAMPLE: RUN SCRIPT

    Script = "get @level from db.[Global Economics] where Location in 'GBR', 'USA' and Indicator = 'GDP' and Year from 2013 to 2023 with Frequency = 'Annual"
    Application.Run "Acropolis_Log", "Running script: " & Script
    Application.Run "Acropolis_ShowAlert", "About to run RUN SCRIPT (Replace) & vbNewLine & Script, IsShowingMessages

End Sub

Private Sub Acropolis_Example_DownloadDataSeries()

    ' TO SWITCH OFF POP-UP MESSAGES...
    ' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
    ' (Where not supplied, IsShowingMessages = True is the default.)
    IsShowingMessages = True

    ' # EXAMPLE: DOWNLOAD INDIVIDUAL DATA SERIES

    Dim Config As AcropolisDataSeriesConfig
    Set Config = New AcropolisDataSeriesConfig

    Config.Databank = "Global Economics"
    Config.Locations = "USA, GBR, AUS"
    Config.Indicators = "GDP"
    Config.Measurement = MeasurementOptions.Level
    Config.Frequency = FrequencyOptions.Annual
    Config.FromYear = 2013
    Config.ToYear = 2019

    jsonConfig = Config.Stringify

    Application.Run "Acropolis_ShowAlert", "About to run DOWNLOAD DATA SERIES (Replace)", IsShowingMessages

    SheetName = Application.Run("Acropolis_DownloadDataSeries", jsonConfig, "ReplaceSheet")
    Application.Run "Acropolis_Log", "Downloaded " & jsonConfig & " into " & SheetName

    Application.Run "Acropolis_ShowAlert", "About to run DOWNLOAD DATA SERIES (Append)", IsShowingMessages

    SheetName = Application.Run("Acropolis_DownloadDataSeries", jsonConfig)
    Application.Run "Acropolis_Log", "Downloaded " & jsonConfig & " into " & SheetName

    Application.Run "Acropolis_ShowAlert", "About to run DOWNLOAD DATA SERIES (New)", IsShowingMessages

    SheetName = Application.Run("Acropolis_DownloadDataSeries", jsonConfig, DataSeriesInsertBehaviourOptions.NewSheet, IsShowingMessages, "NEW SHEET")
    Application.Run "Acropolis_Log", "Downloaded " & jsonConfig & " into " & SheetName

End Sub

Private Sub Acropolis_Example_RunScript()

    ' TO SWITCH OFF POP-UP MESSAGES...
    ' ADD THIS FLAG AS THE FINAL ARGUMENT TO ACROPOLIS COMMANDS
    ' (Where not supplied, IsShowingMessages = True is the default.)
    IsShowingMessages = True

    ' # EXAMPLE: RUN SCRIPT

    Script = "get @level from db.[Global Economics] where Location in 'GBR', 'USA' and Indicator = 'GDP' and Year from 2013 to 2023 with Frequency = 'Annual"
    Application.Run "Acropolis_Log", "Running script: " & Script
    Application.Run "Acropolis_ShowAlert", "About to run RUN SCRIPT (Replace) & vbNewLine & Script, IsShowingMessages
TrueFalse = Application.Run("Acropolis_RunScript", Script, "ReplaceSheet")
Application.Run "Acropolis_Log", "RUN SCRIPT (Replace) | " & TrueFalse & " | " & Script
Application.Run "Acropolis_ShowAlert", "About to run RUN SCRIPT (Append)" & vbNewLine & Script, IsShowingMessages
TrueFalse = Application.Run("Acropolis_RunScript", Script, "AppendToSheet")
Application.Run "Acropolis_Log", "RUN SCRIPT (Append) | " & TrueFalse & " | " & Script
Application.Run "Acropolis_ShowAlert", "About to run RUN SCRIPT (New)" & vbNewLine & Script, IsShowingMessages
TrueFalse = Application.Run("Acropolis_RunScript", Script, "NewSheet", IsShowingMessages, "NEW SHEET")
Application.Run "Acropolis_Log", "RUN SCRIPT (New) | " & TrueFalse & " | " & Script

End Sub

Private Sub Acropolis_Example_AutomationHelpers()

' !! IMPORTANT - CHOOSE PRODUCTION OR DEBUG MODE !!

'Acropolis_ProductionMode
Acropolis_DebugMode

S = AcropolisHelpers_DownloadSelection("TEST SELECTION")
TF = AcropolisHelpers_ResfreshWorksheet("TEST SELECTION")

Acropolis_ResfreshAllWorksheets

' This script will default to Annual Frequency and T-1 to T+5 Years Period
TF = AcropolisHelpers_RunScript("get @level from db.[Global Economics] where Location = 'GBR' and Indicator = 'GDP'")
TF = AcropolisHelpers_RunScript("get @level from db.[Global Economics] where Location = 'GBR' and Indicator = 'GDP'", MyAcropolisObject)

Dim SS() As Variant
SS = AcropolisHelpers_GetSelectionsList
SS = AcropolisHelpers_Reload

TF = AcropolisHelpers_IsConnected("OnlineDataWorkstation")

End Sub

Acropolis Automation Helpers Module

**AcropolisAutomationHelpers.cls**

' Automation helpers are provided for the convenience of external programs to invoke (e.g. external .NET programs).
' These external programs wouldn't have an instance of Acropolis so the helper functions will create one.
' However, if an Acropolis instance is supplied, it will be used instead.

Private plsProduction As Boolean
Public Property Get IsProduction() As Boolean
    IsProduction = plsProduction
End Property
Private Property Let IsProduction(Value As Boolean)
    plsProduction = Value
End Property

Public Sub Acropolis_DebugMode()
    IsProduction = False
End Sub

Public Sub Acropolis_ProductionMode()
    IsProduction = True
End Sub
Example use:
'S = Acropolis_DownloadSelection("TEST SELECTION")
Public Function AcropolisHelpers_DownloadSelection(SavedSelection As String) As String
    If AcropolisHelpers_IsAcropolisAvailable(IsProduction) = False Then
        Exit Function
    End If

    Acropolis_DownloadSelection = Application.Run("Acropolis_DownloadSelection", SavedSelection)
End Function

Example use:
'TF = Acropolis_RefreshWorksheet("TEST SELECTION")
Public Function AcropolisHelpers_RefreshWorksheet(SheetName As String) As Boolean
    If AcropolisHelpers_IsAcropolisAvailable(IsProduction) = False Then
        Exit Function
    End If

    Application.Run "Acropolis_Log", "Refreshing (" & SheetName & ")"
    TrueFalse = Application.Run("Acropolis_RefreshWorksheet", SheetName)

    Acropolis_RefreshWorksheet = TrueFalse
End Function

Example use:
'Acropolis_RefreshAllWorksheets
Public Sub AcropolisHelpers_RefreshAllWorksheets()
    If AcropolisHelpers_IsAcropolisAvailable(IsProduction) = False Then
        Exit Function
    End If

    Application.Run "Acropolis_Log", "Refreshing All Worksheets" & SheetName

    Application.Run "Acropolis_RefreshAllWorksheets"
End Sub

Example use:
'TF = Acropolis_RunScript("get @level from db.[Global Economics] where ..ETC.. with Frequency = 'Annual'")
'TF = Acropolis_RunScript("get @level from db.[Global Economics] where ..ETC.. with Frequency = 'Annual'", MyAcropolisObject)
Public Function AcropolisHelpers_RunScript(Script As String) As Boolean
    If AcropolisHelpers_IsAcropolisAvailable(IsProduction) = False Then
        Exit Function
    End If

    Application.Run "Acropolis_Log", "Running with script: " & Script
    TrueFalse = Application.Run("Acropolis_RunScript", Script, "ReplaceSheet")

    Application.Run "Acropolis_Log", "RUN SCRIPT (Replace) | " & TrueFalse & " | " & Script

    Acropolis_RunScript = TrueFalse
End Function

Example use:
'Dim SS() As String
Public Function AcropolisHelpers_GetSelectionsList() As Variant()
    If AcropolisHelpers_IsAcropolisAvailable(IsProduction) = False Then
        Exit Function
    End If

    Dim SS() As Variant
    SS = Application.Run("Acropolis_GetSelectionsList")
    Acropolis_GetSelectionsList = SS
End Function
Public Function AcropolisHelpers_Reload() As String
    If AcropolisHelpers_IsAcropolisAvailable(IsProduction) = False Then
        Exit Function
    End If

    Application.Run "Acropolis_Reload"
    Dim SS() As Variant
    SS = Application.Run("Acropolis_GetSelectionsList")
    Acropolis_Reload = SS
End Function

Public Function AcropolisHelpers_IsConnected(DataSource As String) As Boolean
    If AcropolisHelpers_IsAcropolisAvailable(IsProduction) = False Then
        Exit Function
    End If

    IsConnected = Application.Run("Acropolis_IsConnected", DataSource)
    Dim Message As String
    Select Case DataSource
        Case "OnlineDataWorkstation"
            Message = "Online Data Workstation Connected: " & IsConnected
        Case Else
            Message = "Desktop Model Workstation Connected: " & IsConnected
    End Select
    Application.Run "Acropolis_Log", Message
    Acropolis_IsConnected = IsConnected
End Function

Public Function AcropolisHelpers_IsAcropolisAvailable(IsProduction As Boolean) As Boolean
    Dim PreferredAvailable As Boolean
    PreferredAvailable = IsPreferredAvailable(IsProduction)
    If PreferredAvailable = False Then
        PreferredAvailable = IsPreferredAvailable(Not IsProduction)
    If PreferredAvailable = False Then
        MsgBox "The Oxford Economics Excel Data Workstation (Excel Add-in) is not available. Please install and ensure it is visible in the menu tabs." & vbNewLine & vbCrLf & "For more information, please visit our website: http://www.oxfordeconomics.com/excel-data-workstation"
        AcropolisHelpers_IsAcropolisAvailable = False
    Else
        MsgBox "The add-in requested is not active. We will use the alternative add-in found." & vbCrLf & vbCrLf & "Ensure both the requested and installed add-in states match." & vbCrLf & vbCrLf & "PRODUCTION add-in info: Requested = " & IsProduction & vbCrLf & " | Installed = " & IsInstalled_COMAddInProd & vbCrLf & vbCrLf & "DEBUG add-in info: Requested = " & IsDebug & vbCrLf & " | Installed = " & IsInstalled_COMAddInDev
        AcropolisHelpers_IsAcropolisAvailable = True
    End If
    Else
        AcropolisHelpers_IsAcropolisAvailable = True
    End If
End Function

Private Function IsPreferredAvailable(IsProduction As Boolean) As Boolean
    If IsInstalled_COMAddInDev And Not IsProduction Then
        IsPreferredAvailable = True
        Exit Function
    ElseIf IsInstalled_COMAddInProd And IsProduction Then
        IsPreferredAvailable = True
        Exit Function
End If
IsPreferredAvailable = False
End Function

' ==== Helper Functions ====

Private Function IsInstalled_COMAddInDev() As Boolean
Dim Description As String
For i = 1 To Application.COMAddIns.Count
    Description = Application.COMAddIns.Item(i).Description
    IsCOMAddInDev = InStr(1, Description, "Acropolis.ExcelAddIn", vbTextCompare)
    If IsCOMAddInDev Then
        IsInstalled_COMAddInDev = True
        Exit Function
    End If
Next i
IsInstalled_COMAddInDev = False
End Function

Private Function IsInstalled_COMAddInProd() As Boolean
Dim Description As String
For i = 1 To Application.COMAddIns.Count
    Description = Application.COMAddIns.Item(i).Description
    IsCOMAddInProd = InStr(1, Description, "Oxford Economics Excel Add-in", vbTextCompare)
    If IsCOMAddInProd Then
        IsInstalled_COMAddInProd = True
        Exit Function
    End If
Next i
IsInstalled_COMAddInProd = False
End Function

Acropolis Constants Module

**AcropolisConstants.cls**

' Constants which make it easier and more declarative to parameterise automation interface methods.

Enum DataSourceOptions
    OnlineDataWorkstation = 1
    DesktopModelWorkstation = 2
    ExcelWorksheet = 3
End Enum

Enum FrequencyOptions
    Quarterly = 2
    Annual = 2
    Both = 3
End Enum

Enum MeasurementOptions
    Level = 1
    Pch = 2
    Diff = 3
End Enum

Enum DataSeriesInsertBehaviourOptions
    NewSheet = 1
    ReplaceSheet = 2
    AppendToSheet = 3
End Enum
Acropolis Data Series Configuration Class

**AcropolisDataSeriesConfig.cls**

' Defines an object used to configure individual data series downloads.
' A utility method generates a JSON string representation of the configuration object
' so it can be passed easily to the compatible automation interface.

Private pDatabank As String
Private pLocations As String
Private pIndicators As String
' Private pMeasurement As MeasurementOptions
Private pMeasurement As String
Private pFrequency As FrequencyOptions
Private pFromYear As Integer
Private pToYear As Integer

Public Property Get Databank() As String
    Databank = pDatabank
End Property
Public Property Let Databank(Value As String)
    pDatabank = Value
End Property

Public Property Get Locations() As String
    Locations = pLocations
End Property
Public Property Let Locations(Value As String)
    pLocations = Value
End Property

Public Property Get Indicators() As String
    Indicators = pIndicators
End Property
Public Property Let Indicators(Value As String)
    pIndicators = Value
End Property
' Public Property Get Measurement() As MeasurementOptions
Public Property Get Measurement() As String
    Measurement = pMeasurement
End Property
' Public Property Let Measurement(Value As MeasurementOptions)
Public Property Let Measurement(Value As String)
    pMeasurement = Value
End Property

Public Property Get Frequency() As FrequencyOptions
    Frequency = pFrequency
End Property
Public Property Let Frequency(Value As FrequencyOptions)
    pFrequency = Value
End Property

Public Property Get FromYear() As Integer
    FromYear = pFromYear
End Property
Public Property Let FromYear(Value As Integer)
    pFromYear = Value
End Property
Public Property Get ToYear() As Integer
    ToYear = pToYear
End Property

Public Property Let ToYear(Value As Integer)
    pToYear = Value
End Property

Public Function Stringify() As String
    Dim json As String
    json = "{ "Databank": ", "Locations": ", "Indicators": ", "Measurement": ", "Frequency": ", "FromYear": ", "ToYear": " }"
    If Me.Databank = vbNullString Or Me.Locations = vbNullString Or Me.Indicators = vbNullString Or Me.Measurement = vbNullString Or Me.Frequency = 0 Or Me.FromYear = 0 Or Me.ToYear = 0 Then
        MsgBox "All properties must have a valid value: " & vbCrLf & json, vbOKOnly, "AcropolisDataSeriesConfig Error"
        Stringify = vbNullString
        Exit Function
    End If
    Stringify = json
End Function

If you would like to be provided with more examples, please contact Oxford Economics Support and they will be able to assist you further.

END OF DOCUMENT