## Contents

1. Get Started with the Oracle Visual Builder Add-in for Excel
2. Install the Oracle Visual Builder Add-in for Excel
3. Create a Table Layout in an Excel Workbook
4. Create a Form-over-Table Layout in an Excel Workbook
5. Limit the Data Downloaded by the Oracle Visual Builder Add-in for Excel
6. Use Lists of Values in an Excel Workbook
7. Use Macros in an Integrated Excel Workbook
8. Change the REST Service Endpoint Connected to by Your Excel Workbook
9. Publish an Excel Workbook that Accesses REST Services
10. View and Edit Data Using an Excel Workbook
<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>11</td>
<td>Internationalization</td>
</tr>
<tr>
<td>12</td>
<td>Security Best Practices</td>
</tr>
<tr>
<td>13</td>
<td>Troubleshoot Excel Workbooks Integrated with the Oracle Visual Builder Add-in for Excel</td>
</tr>
<tr>
<td>14</td>
<td>Known Limitations of the Oracle Visual Builder Add-in for Excel</td>
</tr>
<tr>
<td>15</td>
<td>Data Types Supported by the Oracle Visual Builder Add-in for Excel</td>
</tr>
<tr>
<td>16</td>
<td>Third Party License</td>
</tr>
</tbody>
</table>
Preface

Using the Oracle Visual Builder Add-in for Excel in Oracle Integration describes how to develop Excel workbooks that can retrieve and modify data exposed by a REST service and can also send back modified data to the REST service.

Topics:
• Audience
• Related Resources
• Documentation Accessibility
• Conventions

Audience

Using the Oracle Visual Builder Add-in for Excel in Oracle Integration is intended for Oracle Visual Builder users who want to create and publish Excel workbooks that integrate with the enterprise applications that they develop. This guide is also helpful to users who need to work with published Excel workbooks.

Related Resources

For more information, see these Oracle resources:
• Oracle Public Cloud
  http://cloud.oracle.com
• About Oracle Visual Builder in Developing Applications with Oracle Visual Builder in Oracle Integration

Documentation Accessibility

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

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit http://www.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Conventions

The following text conventions are used in this document:
<table>
<thead>
<tr>
<th>Convention</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>boldface</strong></td>
<td>Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.</td>
</tr>
<tr>
<td><em>italic</em></td>
<td>Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.</td>
</tr>
<tr>
<td><strong>monospace</strong></td>
<td>Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.</td>
</tr>
</tbody>
</table>
Get Started with the Oracle Visual Builder Add-in for Excel

The Oracle Visual Builder Add-in for Excel integrates Microsoft Excel spreadsheets with REST services to retrieve, analyze, and edit data from REST resource collections.

After you install the add-in, a new ribbon tab (Oracle Visual Builder) appears in Microsoft Excel. This ribbon tab exposes buttons to configure a worksheet to integrate with a REST service to download data to a data table that you create in the worksheet. Once the data table has been created and populated with data, you can review, modify and create data before uploading changes to the REST service.

Once you complete the integration of the Excel worksheet with the REST service, you can optionally publish the Excel workbook that contains the integrated worksheet. Publication prepares the worksheet for the users who will use it for data entry tasks. Among other changes, the publication process removes buttons from the Oracle Visual Builder ribbon tab that are not required by users of the integrated worksheet.

The following image shows a published worksheet that is integrated with a REST resource collection that manages employees. The user of this worksheet has updated one row and created a new row with employee data. These changes have been successfully uploaded to the REST resource collection, as indicated by messages in the Status column. The user has also updated data in another row that has yet to be uploaded, as indicated by the Update message in the Change column.

The following image shows the Table layout, which is one of the two types of layout that the add-in can create in an Excel worksheet. The second type is the Form-over-Table layout which you can configure for REST services where a parent-child relationship exists between a parent resource and child resources in the REST resource collection. You can create one type of layout per worksheet in your Excel workbook. That is, each worksheet in the Excel workbook can include a layout.

Subsequent sections in this guide describe how you install, configure, publish and use an Excel worksheet that is integrated with a REST service.
<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
<th>F</th>
<th>G</th>
<th>H</th>
<th>I</th>
<th>J</th>
<th>K</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Change</td>
<td>Status</td>
<td>Id*</td>
<td>First Name</td>
<td>Last Name*</td>
<td>Email*</td>
<td>Phone #</td>
<td>Hire Date*</td>
<td>Job Title*</td>
<td>Salary</td>
</tr>
<tr>
<td>2</td>
<td>Update</td>
<td>Succeeded</td>
<td>100 Stephen</td>
<td>King</td>
<td>SKING</td>
<td>513.123.4567</td>
<td>6/17/2003 12:00 AM President</td>
<td>24,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>Create</td>
<td>Succeeded</td>
<td>211 Walter</td>
<td>Moreno</td>
<td>wmmoreno</td>
<td>513.123.4567</td>
<td>7/9/2019 12:00 AM Administration Assistant</td>
<td>12,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>Update</td>
<td></td>
<td>101 Nina</td>
<td>Kochhar</td>
<td>NKOCCHAR</td>
<td>513.123.4568</td>
<td>6/21/2005 12:00 AM Administration Vice President</td>
<td>17,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>5</td>
<td></td>
<td></td>
<td>102 Leo</td>
<td>De Haan</td>
<td>LDEHAAN</td>
<td>513.123.4569</td>
<td>1/13/2001 12:00 AM Administration Vice President</td>
<td>17,000.00</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td></td>
<td></td>
<td>103 Alexander</td>
<td>Hunold</td>
<td>AHUOLD</td>
<td>590.423.4567</td>
<td>1/3/2006 12:00 AM Programmer</td>
<td>9,000.00</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Install the Oracle Visual Builder Add-in for Excel

Install the Oracle Visual Builder Add-in for Excel using the installer that you download from when you click the **Edit Data in Excel** tile in your visual application's Data Manager page.

You can install the Oracle Visual Builder Add-in for Excel in the 32-bit version of Microsoft Excel 2016 on a computer using the Windows 7 or Windows 10 operating system. For more details, see *Supported Platforms for the Visual Builder Add-in for Excel* that you retrieve from My Oracle Support (https://support.oracle.com) if you search for Doc ID 2474783.1.

Close all open instances of Excel before you double-click the **vbcs-excel-addin-installer.exe** installer file to launch the installation wizard. The following image shows the start screen of the installation wizard. Once you click **Install**, the installation wizard proceeds to install the add-in. It also verifies that required software is installed on the computer where you want to install the add-in. Required software includes the Microsoft .NET Framework and Visual Studio Tools for Office Runtime. If this software is not present, the installer installs Microsoft .NET Framework and Visual Studio Tools for Office Runtime in that order.
**Note:**

The user must have Administrator privileges to successfully install Microsoft .NET Framework and Visual Studio Tools for Office Runtime. No administrator privileges are needed to install the add-in. The add-in is installed for the current Windows user only.

On completion of these steps a screen appears where you click **Finish** to close the installation wizard.

The add-in installer generates a log file in the following location:

%TEMP%\vbcs\vbcs-installer-log.txt

The add-in is enabled by default when you install it. You can disable and re-enable it using the **Oracle Visual Builder Add-in for Excel** check box in the COM Add-ins dialog that you access by clicking Excel's **File > Options > Add-Ins** menu, as shown in the following image. To verify that the add-in is correctly installed, you can download and run the Visual Builder Add-in for Excel - Client Health Check Tool. For more details, see *How to use Visual Builder Add-in for Excel - Client Health Check Tool* that you retrieve from My Oracle Support (https://support.oracle.com) if you search for Doc ID 2477792.1.
The Oracle Visual Builder Add-in for Excel installer also supports optional command-line switches that you can specify with the installer executable file.

Table 2-1  Oracle Visual Builder Add-in for Excel Installer Command-Line Switches

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>/help</td>
<td>Displays a list of supported switches with description.</td>
</tr>
<tr>
<td>/quiet</td>
<td>Suppresses the interactive mode of the installer and does not install any missing prerequisite software.</td>
</tr>
<tr>
<td>/log &lt;path&gt;</td>
<td>Runs the installer and directs the log output to the specified log file. The default log file location is <code>%TEMP%\vbcsvbcs-installer-log.txt</code>.</td>
</tr>
<tr>
<td>Switch</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/roaming &lt;0</td>
<td>1&gt;</td>
</tr>
<tr>
<td></td>
<td>• 0 to install the add-in to the local application data folder (%localappdata%\Oracle\Oracle Visual Builder Add-in for Excel). Use /roaming 0 to install to the local application data folder during an upgrade from a prior installation that was installed to the roaming application data folder.</td>
</tr>
<tr>
<td></td>
<td>• 1 to install the add-in to the end user’s roaming application data folder (%appdata%\Oracle\Oracle Visual Builder Add-in for Excel). This is the default installation location.</td>
</tr>
</tbody>
</table>
Create a Table Layout in an Excel Workbook

Create a Table layout in the Excel worksheet when you want to render the data that the add-in downloads from the REST service to the Excel workbook in a tabular format.

Run Excel and create a blank workbook using the standard Excel workbook file format type (.xlsx) or the macro-enabled workbook type (.xlam). The Oracle Visual Builder Add-in for Excel does not support other Excel formats (.xls, and so on).

1. In the Excel ribbon, select the **Oracle Visual Builder** tab.

2. Click on the cell where you want to locate the table.

3. In the Oracle Visual Builder tab, click **Designer**.

4. When prompted, paste in the desired metadata API (or Catalog API) URL in the Metadata API dialog that appears, and then click **OK**.
Note:
If multiple business objects are found, the Available Business Objects dialog prompts you to choose from the available business objects.

The add-in creates a data table in the Excel workbook that renders Columns headers for the data, a placeholder data row, and the Layout Designer opens in the Excel Task Pane.

5. Click the Columns tab to add or remove columns, or change the order in which columns appear. Review the list of columns to verify that the add-in only creates the necessary columns.

6. Review the created table in the Excel workbook. Click the Designer ribbon tab if you want to make additional changes to the table.

Configuration is now complete and ready to be published. We recommend you test the table by performing various operations, such as download, update, and upload before you publish the workbook to distribute it to users. Alternatively, you can configure the workbook further so that the add-in limits the data that it downloads, as described in Limit the Data Downloaded by the Oracle Visual Builder Add-in for Excel. Also, you can change the REST service endpoint used by your Excel workbook after you complete configuration of the workbook, as described in Change the REST Service Endpoint Connected to by Your Excel Workbook.

If you want to select a different REST resource in this workbook, first delete the table by clicking the Delete Layout button. Then, click the Designer button again and repeat the steps above. Alternatively, create another Excel workbook to use as a source workbook.

REST service endpoints exposed through Oracle Rest Data Services (ORDS) may be used for creating Table layouts, provided the resources are exposed via AutoREST,
and you use ORDS version 18.3 or higher. The metadata API URL is determined by the ORDS configuration, but it must include the `open-api-catalog` segment, identify the resource (table), and return OpenAPI 2.0 metadata for the table. For example, for an endpoint with `myhost.example.com:8888` as the host and domain portion, and where the schema/application is `hr_demo`, and the table is `employees`, use:

http(s)://myhost.example.com:8888/ords/hr_demo/open-api-catalog/employees/

See Automatic Enabling of Schema Objects for REST Access (AutoREST) in Oracle REST Data Services Installation, Configuration, and Development Guide.
4

Create a Form-over-Table Layout in an Excel Workbook

Create a Form-over-Table layout in the Excel worksheet when a parent-child relationship exists in the data that the add-in downloads from the REST service to the Excel workbook.

The add-in displays a dialog where you can choose the Form-over-Table layout if resource (business object) you chose has one or more child resources with a one-to-many relationship between parent and child. If the add-in finds more than one such child resource, it prompts you to pick the resource you want to use in the Form-over-Table layout.

Run Excel and create a blank workbook using the standard Excel workbook file format type (.XLSX) or the macro-enabled workbook type (.XLSM). The Oracle Visual Builder Add-in for Excel does not support other Excel formats (.XLS, and so on).

1. In the Excel ribbon, select the Oracle Visual Builder tab.

2. Click on the cell where you want to locate the form and table.

3. In the Oracle Visual Builder tab, click Designer.

4. When prompted, paste in the desired metadata API (or Catalog API) URL in the Metadata API dialog that appears.
5. Choose a business object, such as **DepartmentsWithEmployees** as illustrated in the previous image, where a parent-child relationship exists, and choose **Form-over-Table Layout** in the New Layout Setup dialog that appears, and then click **OK**.

The add-in creates a Form-over-Table in the Excel worksheet and opens the Layout Designer that you use to modify the newly-inserted form and table, as illustrated in the following image.

6. Customize the form and table by modifying the automatically populated properties in the Layout Designer that appears in the Excel Task Pane.

If, for example, you do not specify a value for the Search field or Finder property, as described in Limit the Data Downloaded by the Oracle Visual Builder Add-in for Excel, the add-in downloads the first parent item it encounters in the REST service to the form, and the child items, if any, to the table.

The Form tab enables you to add or remove fields to the form. The Table tab performs a similar function for the table under the form.

7. Review the created table and form in the Excel workbook. Click the **Designer** ribbon tab if you want to make additional changes.
Configuration is now complete and you can publish the workbook. We recommend you test the workbook before you publish it to distribute it to users. Alternatively, you can configure the workbook further so that the add-in limits the data that it downloads, as described in Limit the Data Downloaded by the Oracle Visual Builder Add-in for Excel.

If you want to select a different REST resource in this workbook, first delete the layout by clicking the Delete Layout button. Then, click the Designer button again and repeat the steps above. Alternatively, create another Excel workbook to use as a source workbook.

**Note:**

REST service endpoints exposed via Oracle Rest Data Services (ORDS) do not provide the add-in with sufficient information to support Form-over-Table layouts.
Limit the Data Downloaded by the Oracle Visual Builder Add-in for Excel

Add search to your workbook so that the add-in automatically prompts users to enter search terms to limit the downloaded data when download occurs.

The add-in exposes two properties that you configure to limit the data that the add-in downloads to the workbook. Configure values for the Search property when you want to provide users with the option to enter search terms to filter the data that they download from the REST resource collection. Consider, for example, a data table that downloads data about employees. In this scenario, you can add the Department name business object field as a search field to the Search property, so that users can enter search criteria to download the employee records of those employees who belong to the department that matches the search criteria. You can enter multiple fields to the Search property so that users can enter multiple terms in their search.

Configure a value for the REST Finder property if the REST service you connect to supports finders. If you select a value for the REST Finder property, the add-in invokes it during download. A finder that a REST service supports may have parameters. If the finder that you choose for the REST Finder property specifies parameters, the add-in prompts users to provide parameter values during download. Refer to the following resources for more information about configuring the REST service that supports finders:

- About RESTful Web Services and ADF Business Components
- Filtering a Resource Collection with a Row Finder

You can configure one or both of these properties. If you configure values for both properties, the add-in prompts the user to enter the value(s) supported by the REST Finder property before it prompts the user to enter value(s) for the Search property. Once the user enters the requested input values, the add-in downloads data based on the values that the user input.

Use Search to Limit Downloaded Data

You can configure the workbook to enable a user to specify search values to limit the data that the add-in downloads to the workbook.

1. In the Excel ribbon, click Designer.
2. In the Query tab of the Layout Designer, click the Edit icon next to the Search property to open the Available Business Object Fields dialog.
3. Select the business object field that you want to enable users to enter search terms for. For example, select Department Name and click OK if you want to enable users to search on employees by department, as shown by the following image.
4. Click **Add Field** to add additional search fields.

5. Click **OK**.

**Use REST Finders to Limit Downloaded Data**

You can configure the workbook to enable a user to specify a value for a finder that the REST resource collection exposes.

1. In the Excel ribbon, click **Designer**.

2. In the Query tab of the Layout Designer, click the Edit icon next to the REST Finder property, as shown in the following image.

The REST service owner must configure the service to expose the options that appear in the Available Finders dialog.

3. Click **OK** to close the open dialogs.
Download Behavior in Workbooks that Use Search and REST Finders

If you configured a value for the REST Finder property, the behavior of the add-in depends on the configuration of the finder exposed by the REST service:

- If there are no parameters, there is no prompt.
- If there is exactly one parameter, a prompt appears that requests the user to specify a value for the parameter.
- If the finder has more than one parameter, a prompt appears that requests the user to specify values for each parameter.

**Note:**

REST service endpoints exposed via Oracle Rest Data Services (ORDS) do not support REST Finders.

If you added search fields to the Search property, the add-in prompts users who download employee data using the **Download Data** button to enter values in the search field, as illustrated in the following image.
Use Lists of Values in an Excel Workbook

The Oracle Visual Builder Add-in for Excel configures lists of values to appear in the Table and Form-over-Table layouts that you define in an Excel workbook when a REST resource item attribute has a list of values.

This release of the add-in only supports list of values of string type. Number and date list of values are not supported.

The list of values is the items that constitutes the valid set of values for a REST resource item attribute. Each item in the list of values has a display value (appears in the Excel workbook) and a target value, that is retrieved and posted to the REST resource item attribute. For example, a Job attribute for an Employee REST resource item might contain the following display and target values.

<table>
<thead>
<tr>
<th>Display value</th>
<th>Target value</th>
</tr>
</thead>
<tbody>
<tr>
<td>President</td>
<td>PRES</td>
</tr>
<tr>
<td>Finance Manager</td>
<td>FIN_MGR</td>
</tr>
<tr>
<td>Sales Manager</td>
<td>SAL_MGR</td>
</tr>
</tbody>
</table>

When the REST resource item attribute is exposed as a table column in an Excel worksheet, the add-in manages the mapping between the display and target values during the download and upload operations that the add-in performs to manage the data. The add-in caches the mapping of valid display and target values in the workbook.

When a user selects a cell in a table column that is bound to a REST resource item attribute, a combo box component appears that displays available values. A user can select or double-click one of the displayed values. In this case, the Change column displays Update. Alternatively, the user can type one or more of the starting characters for other values and then click the Search icon so that the combo box component filters the values it displays based on the user input. The following composite image displays both of the just-described scenarios. For the latter scenario, the user entered S to filter the combo box component to display values from the list that begin with S (Sales Manager, Sales Representative, and so on).
Users can also type in the new value, assuming the value they type is a valid value, such as Sales Manager, as shown in the previous image. If the new value is valid, the add-in will successfully upload the change and the Status column will display the Update Succeeded message. Where the new value is invalid, the add-in will not be able to upload the change. The Status column will display an Update Failed message and the Table Row Status task pane appears to display more detail about the cause of the failure to upload.

The add-in can also render a cascading list of values. That is, the value selected in one list determines the ranges of values that users can select from subsequent lists. For example, a table renders columns with lists of values for Countries, States, and Cities. The value that a user chooses in the Countries list determines the values that appear in the list for States, and so on.

To implement this type of list, the REST resource that the add-in connects to must meet the follow requirements:

- The endpoint must support REST framework version 5 or higher.
- The cascading LOV resource must be configured using static links, instead of row context links, with finders. For example, for Countries, States, and Cities, the item links for the parent resource must be configured as follows:

```
"item" : {
   "links" : [ {
      "rel" : "lov",
      "href" : "http://servicehost/rest/latest/Countries",
      "name" : "CountryView1",
      "kind" : "collection"
   },{ 
      "rel" : "lov",
```
Cascading list of values modelled with row context list of values are not supported.

- For the POST/PATCH request to process attributes in order, the attributes must have dependencies configured properly. For example, a cascading list of values that includes Countries, States, and Cities, State’s dependencies array should be ["Country"] and City’s dependencies array should be ["Country", "State"].

In the Excel worksheet, the add-in exposes the corresponding REST resource item attributes as table columns in the Excel worksheet (Countries, States, and Cities using our example). You can see and configure these columns like other columns. No extra configuration is required. Do not remove any column from the cascading list of values columns, if you want to implement a cascading list of values in your worksheet.

Do not paste data for multiple cascading list of value columns together. Paste data for one column at a time and follow the dependency order. For example, first country, then state, and finally city.

Note the following limitations that apply to list of values and cascading list of values:

- Fields in the form part of a Form-over-Table layout do not render a dropdown menu, regardless of the number of values in the list of values that the field displays.

- REST service endpoints exposed through Oracle Rest Data Services (ORDS) do not provide the add-in with sufficient information to support lists of values.

- The search functionality uses starts with operator regardless of the attribute’s data type. It does not work well for non-string data types.

- The add-in supports list of values that are defined with a single key attribute, and the first source attribute defined in the attributeMap metadata must be set to the key for the list of value resource.

- These are limitations with list of values in applications developed using Oracle Visual Builder:
  - The parent-child relationship to the list of value resource must have cardinality of "* to 1".
  - The title and the name of the business object that supplies the list of values must be the same.
  - The intended display attribute on the business object that supplies the list of values must contain "name" (for example, "name" or "displayName").

This requirement exists because the add-in inspects the REST metadata for the resource that provides the list of values to determine the display attribute.
If an attribute name contains the pattern “name”, the add-in uses that attribute to display the list of values. If no attributes contain the “name” pattern, the add-in uses the first non-reserved attribute it encounters. Reserved attributes include “id”, “lastUpdatedBy”, and so on. The add-in uses a default attribute if it does not find a non-reserved attribute to use.
Use Macros in an Integrated Excel Workbook

You can configure macros that the Oracle Visual Builder Add-in for Excel executes at specific points in the life cycle of an integrated Excel workbook.

Use of this functionality requires you to use the Excel macro-enabled workbook type (.XLSM) and create your macro(s) in a macro module. For more information about creating macros, see Microsoft’s documentation. Describing how to create macros in an Excel workbook is outside the scope of this guide.

Some companies block the usage of Excel macros because they do not think macros are sufficiently secure. Consider your intended user base before you add a macro. After creating a macro, take steps to protect that macro from malicious and/or accidental alterations that might produce unexpected or harmful results. You are responsible for the security risks involved in using macros. Ensure you research this topic before you deliver an integrated workbook to your customers. If a macro that you create results in changes that are incompatible with the add-in or results in undesirable behavior, change the macro to avoid this behavior.

The Layout Designer’s Advanced tab exposes two properties where you can specify macros to execute after download completes (Post-Download Macro) or before an upload begins (Pre-Upload Macro). The macro that you specify for the Post-Download Macro property is not invoked if the user cancels download, if the table or form is empty, or in the event of an unexpected exception. The macro that you specify for the Pre-Upload Macro property is invoked just prior to upload. If the macro returns any value other than true, the upload operation aborts and a message appears in the Status Viewer to notify the user. If the macro returns true, upload proceeds normally. To return a true or false value from a macro, define a Boolean Function. See Microsoft documentation for details.

When an error occurs during the execution of a macro, Excel displays a Microsoft Visual Basic window to the user. We recommend that you implement a robust error handling strategy so that the window displays a useful message to the user who encounters an error during macro execution. The following is a simplistic example. The
appropriate error handling strategy for a given macro depends on the logic in the macro.

Sub Refresh()
    On Error GoTo ErrHandler:
    ActiveWorkbook.RefreshAll
    Exit Sub

ErrHandler:
    Dim failureMessage As String
    failureMessage = Err.Description
    MsgBox "Unable to refresh. Details: " & failureMessage
    Exit Sub
End Sub
Tip:

The add-in creates and maintains named ranges for the data table. Your macros should never modify these named ranges. However, your macros can access the named range to locate the data table on a dynamic basis, as the following example illustrates:

```vba
Sub AddColor()
    On Error GoTo ErrHandler:
    Dim table As Range
    Set table = Sheets("Data").Range("TBL246043480")
    ' The named range, TBL246043480, is managed automatically by the add-in

    Dim cRows As Long
    cRows = table.Rows.Count
    Dim currentTableRow As Long
    Dim amount As Long
    For currentTableRow = 2 To cRows ' start with 2 to skip header row
        amount = table(currentTableRow, 6) ' Amount is sixth column in the table
        If amount < 0 Then
            Debug.Print "Found negative amount = "; amount
            table(currentTableRow, 6).Interior.ColorIndex = 22 ' a light red
        End If
    Next

    Exit Sub

ErrHandler:
    Dim failureMessage As String
    failureMessage = Err.Description
    MsgBox "Unable to finish adding color. Details: " & failureMessage

    Exit Sub
End Sub
```

Note:

Macro recording is incompatible with add-in features such as download and upload and is not supported. Do not attempt to record any add-in features. In some cases, you may see unexpected exceptions.

Do not leave the Excel Visual Basic editor’s break mode on when you invoke Download Data or Upload Changes. It can result in an unexpected exception. It is not supported.
Change the REST Service Endpoint Connected to by Your Excel Workbook

You can change the metadata API (REST service endpoint) used by your Excel workbook after you complete configuration of the workbook.

You may need to do this when you complete configuration of your Excel workbook using a development environment and are ready to distribute the workbook to users who connect to a production environment with a different value for the REST service endpoint.

You change the REST service endpoint in the General tab of the Layout Designer that you access when you click Designer in a pre-publication Excel workbook that has been configured with a Table or Form-over-Table layout. The General tab exposes two input fields (Host and Service Path) for the REST service that your Excel workbook connects to. Change the REST service endpoint using these fields. A change to either of these input fields triggers the following actions in the Excel workbook by the add-in:

> Note:

A change that you make to the value for the Host field in one Excel worksheet applies to other worksheets in the Excel workbook because the Excel workbook uses one service host for all worksheets.

A change to the Host value triggers the following actions in the Excel workbook by the add-in:

- Discards the cache of list of values
- Clears all data from the Table or Form-over-Table layout, including pending changes
- Logs you (the current user) out from the Host
- Re-renders all workbook layouts in their empty state (after logging into the new host)

A change to the Service Path value triggers the following actions in the Excel workbook by the add-in:

- Clears all data from the Table or Form-over-Table layout, including pending changes
- Re-renders the layout in its empty state

It is important that the new REST service endpoint that you specify has the same definition as the initial REST service endpoint that you used in the Excel workbook. Errors may occur if the endpoint definitions differ. For example, if your layout is configured to use a phone number attribute that is available in the initial REST service endpoint but is not defined in the new REST service endpoint, Excel workbook operations may fail once it uses the new REST service endpoint. You can change the
When you publish an Excel workbook, if you don't select the **Remove Service Host** checkbox, then you cannot later change the REST service endpoint. For those cases, perform this task in the pre-publication Excel workbook and then publish the Excel workbook. For the cases where you select the checkbox, then the end user will be prompted for the service host as described in Publish an Excel Workbook that Accesses REST Services.
Publish an Excel Workbook that Accesses REST Services

Once you complete configuration of an Excel workbook so that it accesses REST services and test it thoroughly, publish it for end users to do data entry work.

Before you click the Publish button, make a backup copy of the un-published (source) workbook in case you need to make any further configuration changes at a later date post-publication. Consider adding a filename suffix of _src for the source workbook. Then, remove the suffix in the published copy. Excel will not allow you to use the same filename for both workbooks. You should also click the Clear button to remove all downloaded data from the workbook before you publish it. Also, consider using the Document Inspector in Excel to remove personal information from the workbook before you click Publish to distribute it to users. See Microsoft's documentation for information about Excel's Document Inspector.

Once you are ready to distribute the workbook to users, click Publish. In the Publish Workbook dialog that appears, select the Remove the service host from the published workbook checkbox if you want users to enter the service host when they open the published workbook, and click OK.

Accept the default file name and directory location values for the published workbook or select alternative values in the subsequent dialog, then click Save. A confirmation dialog appears with a message to say that the workbook has been successfully published to the directory location with the file name that you specified. You can now distribute the published workbook to users.

When users open a published workbook where you removed the service host value by selecting the Remove the service host from the published workbook checkbox, a Workbook Setup dialog first appears that prompts users to enter their service host value.
Once they enter this value, the published workbook opens. If you published the workbook without removing the service host, the published workbook opens directly. There are a number of differences with the source workbook:

- The design tools do not appear in the Oracle Visual Builder tab of the Excel ribbon (Designer, Delete Layout, and Publish buttons), as shown in the following image.

- Excel's sheet protection is on. This mode enables true read-only behavior for cells that should be read-only. It also prevents the user from performing various other Excel actions that might disrupt the integration with the REST service.
View and Edit Data Using an Excel Workbook

In Microsoft Excel, select the Oracle Visual Builder tab and use the buttons it exposes to work with the data that the workbook accesses.

The data operations that users can perform in the workbook depend on the operations exposed by the REST service resource the workbook uses. For example, users will not be enable to create new rows in the workbook table if the REST service resource does not support create operations. Similarly, if the resource does not support update operations, users will not be able to edit the data they download to the workbook. The following sections describe the operations that users can perform in a data table, assuming the underlying resource supports the operation.

Users can remove all data from the workbook, including any pending changes that have not yet been uploaded to the REST service by clicking the Clear button. This button does not make any calls to the REST service and does not change data in the REST service.

Downloading Data to the Workbook Table

Users download data to the workbook using the Download Data button in the Oracle Visual Builder tab. The workbook prompts users for a user name and password the first time that they connect to the REST service that the workbook is configured to use. If you attempt to download more than 499 rows, the add-in retrieves the first 499 rows and then prompts you if you want to continue.

If you configured the workbook to allow users to filter the data that they download to the workbook based on search fields, such as a specific department, a dialog or dialogs appear where they specify the value(s) to search on, as shown in the following
example where data for the Sales department will be downloaded. Limit the Data Downloaded by the Oracle Visual Builder Add-in for Excel describes how you configure the workbook to allow users to filter the data that they download.

On successful connection to the REST service and completion of dialog prompts for input values to search data (if configured), the add-in renders the data table that the workbook author published, as shown in the following image.

Editing Downloaded Data in the Workbook

Users edit the editable cells that contain the downloaded data. The following image shows three examples where a user has updated data in the table. The Change column for the first row displays an Update message that indicates the user has updated this row with required and valid values. The Change column for the second row also displays an Update message, but its Status column displays an Invalid message and a red border appears around the Hire Date* column's cell to indicate that a value is required in that cell's field. Finally, in the third row, the user attempted to input Software in the field for Department. As Software is not a value in the underlying list of values that this cell displays in its list of choices, a red border appears around this cell.

For columns that are bound to a list of values, users can select a value in a combo box component that displays available values. A user can select or double-click one of the displayed values. In this case, the Change column displays Update. Alternatively, the user can type one or more of the starting characters for other values and then click the Search icon so that the combo box component filters the values it displays based on the user input. The following composite image displays both of the just-described scenarios. For the latter scenario, the user entered S to filter the combo box component to display values from the list that begin with S (Sales Manager, Sales Representative, and so on).
Creating New Data to Upload to the REST Service

Users can use one of the following methods to insert new data in the table for subsequent upload to the REST service:

1. Type in the empty row (placeholder row) that appears below the table header row before they download data. In the following image, users can enter data in the row cells for Name, Email*, Hire Date, and Department as these row cells accept data input. The other cells in the placeholder row are read-only. Once a user enters a value in a row that accepts data input, a Create message appears in the Change column. An Invalid message appears in the Status column if the row contains a cell where the user is required to enter a value and has yet to do so, or has entered an incorrect value, such as an unexpected data type. A red border also appears around the cell where a value is required or invalid.

Once the user enters data in the placeholder row, the row below the current placeholder row becomes a placeholder row where data values can be entered, and so on.
2. Click the **Download Data** button to download data from the REST service and insert a full Excel row into the table inside the current table boundaries. The following example shows where three new Excel rows have been inserted inside the current table boundaries. A message (Create) appears on the left of each new row to indicate this row is new and has not yet been uploaded to the web application that exposes the REST service. Edit the cells in these new rows with the data changes that you want to make. An **Invalid** message appears in the Status column rows until you enter required values or valid values, as indicated by the red border that surrounds a number of cells in Rows 5 and 6. Required values and valid values have been entered in the newly-inserted Row 4, so it does not display the **Invalid** message or the red border cells.

3. Select the row immediately below the last downloaded row in the table, right-click and choose **Insert** from the context menu that appears. The add-in inserts a new row in the table. A **Create** message appears in the Change column and an **Invalid** message will appear in the Status column if the newly-created row requires you to specify valid or required values.

4. Click the **Upload Changes** button to upload the newly-created data instances to the REST service.

**Deleting Resource Items from the REST Service**

Users can mark rows for deletion from the REST service using the **Row Changes** drop-down list’s **Mark for Delete** menu option.

To mark a row for deletion from the REST service:

1. Select the table row that you want to delete from the REST service. If you want to select a range of table rows to mark for deletion, hold down your keyboard’s Shift key and select the first and last row in the range of table rows that you want to delete.
2. Click the **Row Changes** drop-down list and click **Mark for Delete**.

A **Delete** message appears in the Change column and the add-in changes the style applied to the data in the table rows, as illustrated by the following image where the user marked selected the first three rows in the table for deletion.

3. Click the **Upload Changes** button.

4. In the Upload confirmation dialog that appears with a message that the table has pending deletions, click **Yes** to continue.

The add-in uploads the changed rows from the table, which includes the request to delete the rows that you marked for deletion. The table in the Excel workbook refreshes so that the deleted rows no longer appear, as illustrated in the following image.

Tip:

If you change your mind about deleting one or more rows among the rows that you have marked for deletion, select these rows and use the **Row Changes** drop-down list’s **Unmark Pending Changes** option so that the add-in does not include these rows in the rows that it sends to the REST service for deletion. Use the **Unmark Pending Changes** option before you upload changes from the Excel workbook. The Unmark Pending Changes option does not work after you upload changes.

Uploading Changes to the REST Service

Once you complete the edits that you want to make, be that updates to existing rows, creating new rows, or deleting existing rows, click the **Upload Changes** button to upload all the changes to the REST service. The Oracle Visual Builder Add-in for Excel uploads all the rows marked as changed (those rows with **Create** or **Update** in the Change column). The add-in also deletes the rows marked for deletion (rows with **Delete** in the Change column). Review the Status column to see which rows succeeded and failed. The Status column displays a **Create Failed** message or an **Update Failed** message in the cell of a row that the Oracle Visual Builder Add-in for Excel failed to upload and the Table Row Status appears in Excel’s task pane to provide additional information on the failure, as in the following example where the user did not enter a required value (Hire Date). In the following example, the user needs to enter a hire date and click the **Upload Changes** button to attempt to upload the modified data again. Successfully deleted rows are removed from the Excel worksheet while failed attempts to delete a row result in the row remaining in the Excel worksheet.
worksheet. You can inspect the reason for the failure to delete in the status in Excel's task pane.

Tip:

If you change your mind about uploading changes for a particular row to the REST service, select these rows and use the Row Changes drop-down list’s Unmark Pending Changes option so that the add-in does not include these rows in the rows that it send to the REST service for update, creation, or deletion.

Data Consistency

When the REST service is properly configured, the add-in enforces data consistency. For example, the add-in detects and reacts to the following scenario:

1. User A downloads information from a resource into a table in their integrated workbook.
2. User B downloads the same information into a table in their integrated workbook, edits it, and uploads changes.
3. User A then edits the same information (downloaded in Step 1) and uploads the changes.
4. The add-in provides the REST service with the necessary information (entity tags) to prevent User A's changes from overwriting those changes made by User B. Instead, when the server detects such a change, its response allows the add-in to display an error message similar to the following for any such rows in the table:

   This row has been modified by another user. Please download before editing.

The REST service must support data consistency verification using an entity tag (ETag) mechanism, as described by Data Consistency Tasks in Accessing Business Objects Using REST APIs.
11 Internationalization

Workbooks created using the Oracle Visual Builder Add-in for Excel support localization. The add-in is localized in the Oracle standard languages.

The date, date-time, and number formats used by the add-in are culture-sensitive. When using the add-in, you work with:

- Strings visible on the Oracle Visual Builder ribbon and in various dialogs displayed by the add-in. These strings, known as the **Add-in language**, are owned by the add-in and are localized.
- Strings visible as column headers and field labels are known as **Layout strings**. These are owned by the REST service.

The add-in sends the `accept-language` header to the service to request the language configured in Excel. The language setting specified for Excel is used for the add-in language, layout strings, and for requests to the REST service. When you save a workbook, the current language is noted and saved with the workbook. When the same workbook is opened again, the add-in checks the language setting for Excel and compares it to the language saved with the workbook.

A language change triggers the following actions in the Excel workbook by the add-in:

- Discards the cache of lists of values
- Clears all data from all workbook layouts, including pending changes
- Re-renders all workbook layouts in their empty state (after logging in) with the new language

**Note:**

Because layout strings are owned by the REST service, contact the REST service owner for any missing translations or languages.
Change the Add-in Language

You can select a different language for the add-in to the display language that your installation of Excel uses. Do this if you want to evaluate your workbook with different languages.

1. In Excel, click the Oracle Visual Builder tab.
2. Choose Select Language from the Advanced drop-down.
3. In the Add-in Language drop-down list that appears, select the language that you want to use. The drop-down list displays the languages that the add-in supports.
4. Click OK.
5. Restart Excel for the change you have just made to take effect.

The user-interface elements (Download Data and so on) for the add-in now use the language you selected. If the REST resource that the workbook uses supports the language you selected, layout strings in the workbook use the selected language. If the selected language is a language that renders right-to-left, such as Arabic or Hebrew, add-in user interface elements render right-to-left. The language that Excel uses remains unchanged, as does the format used for dates, times, and numbers. See Excel’s and/or Windows options to change Excel’s language and formats for dates, times, and numbers.

When you re-open a workbook that was rendered in a different language to the language that you have just selected, the add-in automatically clears the layout. If you do not want to clear the layout, close the workbook without saving changes. Change the add-in language back to the original language and restart Excel before you open the workbook again.

The language that you choose for the add-in language is stored in a local file in the Windows user profile and remains unchanged. You can select the Use Excel’s Language Setting option in the Add-in Language drop-down list to remove this setting for the current user.
Security Best Practices

Ensure that you manage data that you download to Excel workbooks securely.

You can configure security options for the REST endpoints exposed by the business objects in your visual application. For example, you can allow anonymous access or enable basic authentication. See About Allowing Access to the Catalog API. Secure Business Objects describes how to can secure the data stored by business objects.

Security Guidelines

Follow these best practices:

• Update the Oracle Visual Builder Add-in for Excel to the latest version available.
• Restrict access to Excel documents containing sensitive data.
• Consider adding passwords to workbooks to further reduce exposure.
• Always use HTTPS endpoints instead of HTTP.
• Do not use basic authentication.
• Ensure that the latest Windows updates and security patches have been applied to the client computers where you install the Oracle Visual Builder Add-in for Excel.
• Turn off older obsolete security protocols such as SSL.

Basic Authentication

The add-in supports basic authentication. When using REST service endpoints protected by basic authentication, end users are automatically prompted for credentials when the add-in connects to the endpoint. When used with HTTP, basic authentication is not secure. Basic authentication should only be used with HTTPS, and preferably only in non-production environments.

JSON Web Token

In addition to basic authentication, the add-in also supports authentication for REST services exposed by Fusion applications that use the JSON Web Token (JWT) relay servlet. No configuration is required by you. The add-in automatically detects whether the Fusion application’s REST service has the /anticsrf and /tokenrelay endpoints configured. The add-in then displays a popup browser window and navigates to the hosting web application’s login page. When the end user provides valid credentials, the popup browser window automatically closes and access to the REST services can proceed, using the token obtained during the login sequence.
Note:

In this release of the add-in, using **self-signed** certificates with the JWT relay servlet will not work. A valid certificate issued from a well-known root certificate authority should work fine with the JWT relay servlet.
Troubleshoot Excel Workbooks Integrated with the Oracle Visual Builder Add-in for Excel

The Oracle Visual Builder Add-in for Excel can generate a detailed log file and diagnostic report to help you identify and resolve issues in the Excel workbooks that you integrate using the add-in.

In addition, consider running the Client Health Check tool to determine if your environment is configured correctly after you install the add-in. You download this tool from My Oracle Support. See the How to use Visual Builder Add-in for Excel - Client Health Check Tool document that you retrieve from My Oracle Support (https://support.oracle.com) if you search for Doc ID 2477792.1.

Additional information about troubleshooting can be found in the Troubleshooting Guide for the Visual Builder Add-in for Excel that you retrieve from My Oracle Support (https://support.oracle.com) if you search for Doc ID 2485062.1.

Network Monitor

Use the Network Monitor window to inspect the content of the REST service calls between your Excel workbook and the REST service that it connects to if you encounter unexpected errors.

The Network Monitor window provides information such as the start time, the elapsed time, and response for each REST call that originates from the workbook. In addition, it provides the JSON payload that the Excel workbook and the REST service exchange.

The Network Monitor window generally goes to the background while you perform the steps of your use case. Bring the window forward to see the details of each request and response. You can select and copy information from the window. You may need this detail when troubleshooting problems with the service owner. The window shows up to 100 request-response events. Older events are discarded as new ones are added.

1. In Excel, click the Oracle Visual Builder tab.
2. Select Network Monitor.
   The Network Monitor window displays.
3. Repeat the steps that lead to the issue.
4. Review the details of each request and response.
5. Optionally, select and copy information from the window.

Logging

When reporting an issue about the Oracle Visual Builder Add-in for Excel, generate a detailed log file that captures the steps that lead to the problem you want to report.
1. In Excel, click the **Oracle Visual Builder** tab.

2. Select **Log Activity** from the Advanced drop-down list to specify a directory location and file name for the log file.
   
   This starts the logging session.

3. Repeat the steps that lead to the issue.

4. Exit Excel completely to stop the logging session and before you access the log file.

   **Note:**

   The next time you run Excel logging will no longer be enabled.

The log file that you generate captures information about steps during an Excel session.

**Logging Console**

The Logging console displays log messages based on the actions performed. If you encounter any issues, view the logging messages to troubleshoot and diagnose the issues.

1. In Excel, click the **Oracle Visual Builder** tab.

2. Select **Log Console**.
   
   The Logging Console window displays.

3. Repeat the steps that lead to the issue.

4. Review the logged messages.

5. Optionally, select **Set Level** and select a value from the drop-down list to specify the log level.
   
   By default, the log level is set to Off. The change to the log level is temporary. The log level resets when you exit Excel.

**Diagnostic Report**

The diagnostic report contains information that can help resolve issues. Please provide a diagnostic report when reporting a problem with the Oracle Visual Builder Add-in for Excel.

1. In Excel, click the **Oracle Visual Builder** tab.

2. Select **Diagnostic Report** from the Advanced drop-down.

3. Save the diagnostic report to a directory location with a file name of your choice.

4. Review the content of the diagnostic report to remove any sensitive information that you do not want to share before you use it to report an issue.
Known Limitations of the Oracle Visual Builder Add-in for Excel

The Oracle Visual Builder Add-in for Excel has a number of known limitations.

- The add-in does not support calculated fields from Oracle Visual Builder applications.
- Read-only fields from Oracle Visual Builder applications do not display as read-only in the Excel workbooks that you integrate using the add-in.
- Never edit the Key column of the table.
- Avoid using the following Excel features with the add-in. The following is a sample list of Excel features that do not work well with the add-in. Other Excel features not listed may also not work well with the add-in.
  - Do not use the Protect Sheet or Workbook features of Excel.
  - Do not attempt to re-arrange the layout of the integrated worksheet. For example, do no insert rows above the table or columns to the left of the table.
  - Do not use the Mark as Final command to make the Excel workbook read-only.
  - Do not delete anything from the integrated worksheet using Excel's delete features, including the Delete key.
- If a REST service owner makes significant changes to the service after the workbook is configured to integrate with the service, the integration may not function as expected. In such cases, delete and recreate the layout in the workbook.
- Here are limitations for working with Oracle REST Data Services (ORDS) version 18.3.
  - The add-in uses ORDS Open API 2.0. Only ORDS endpoints created using AutoREST are supported. REST services created from ORDS modules are not supported.
  - Only basic authentication is supported when working with ORDS services.
  - Here are a few limitations based on the service metadata provided by ORDS:
    * All number values are formatted in the table using the default number format (nnnn.nn), this includes values that might be expected to be formatted as integers, such Department ID.
    * Required fields are not marked with an asterisk in the table column headers.
    * Title property is not present for resource item attributes. Therefore, the attribute name, for example employee_id is used instead for the column header label.
Lists of value relationships, such as Department ID or Name in the Employee record are not available from the metadata. Users must directly enter the required values for Department ID instead of picking from a list.

- For some row-level errors, such as First Name is too long, the ORDS server does not provide a specific reason for the error.
- In some cases, the ORDS server returns Create Failed for rows, when in fact the Create operation was successful. Re-downloading rows into the table will show the created rows.

- Review Data Types Supported by the Oracle Visual Builder Add-in for Excel to understand limitations around data type support.
- Review Use Lists of Values in an Excel Workbook to understand limitations around using lists of values.
Data Types Supported by the Oracle Visual Builder Add-in for Excel

The Oracle Visual Builder Add-in for Excel supports a variety of data types exposed by business objects in web applications developed using Oracle Visual Builder and data types exposed by REST services.

The add-in ignores fields with unsupported data types when you create a Table layout or Form-over-Table layout in the Excel workbook. If, for example, a REST service that you invoke to retrieve data includes the attachment attribute data type, the add-in ignores it and does not create a column in the data table for this attribute type.

The following table lists the data types that the add-in supports from Oracle Visual Builder and from REST services. The add-in does not support the Time data type from Oracle Visual Builder.

<table>
<thead>
<tr>
<th>Supported data types from Oracle Visual Builder</th>
<th>Supported data types from REST services</th>
</tr>
</thead>
<tbody>
<tr>
<td>Boolean</td>
<td>boolean</td>
</tr>
<tr>
<td>Date</td>
<td>date</td>
</tr>
<tr>
<td>Datetime</td>
<td>datetime</td>
</tr>
<tr>
<td>Number</td>
<td>integer</td>
</tr>
<tr>
<td>String</td>
<td>number</td>
</tr>
<tr>
<td>Reference</td>
<td>string</td>
</tr>
</tbody>
</table>

The add-in supports ADF REST resources and requires a minimum of REST API framework version 4. The add-in uses version 6 of the REST API framework if available. See Framework Versions of Accessing Business Objects Using REST APIs.

Excel specifications and limits for numeric data are documented under the Calculation specifications and limits section of the Excel specifications and limits page in Microsoft's documentation.
Third Party License

The Oracle Visual Builder Add-in for Excel includes the following third-party software.

NewtonSoft.Json

The MIT License (MIT)

Copyright (c) 2007 James Newton-King

Permission is hereby granted, free of charge, to any person obtaining a copy of this software and associated documentation files (the "Software"), to deal in the Software without restriction, including without limitation the rights to use, copy, modify, merge, publish, distribute, sublicense, and/or sell copies of the Software, and to permit persons to whom the Software is furnished to do so, subject to the following conditions:

The above copyright notice and this permission notice shall be included in all copies or substantial portions of the Software.

THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL THE AUTHORS OR COPYRIGHT HOLDERS BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY, WHETHER IN AN ACTION OF CONTRACT, TORT OR OTHERWISE, ARISING FROM, OUT OF OR IN CONNECTION WITH THE SOFTWARE OR THE USE OR OTHER DEALINGS IN THE SOFTWARE.