Oracle® Cloud

Using the Oracle Visual Builder Add-in for Excel in Oracle Integration

E94774-07
June 2020

Oracle Cloud Using the Oracle Visual Builder Add-in for Excel in Oracle Integration,

E94774-07

Primary Author: Oracle Corporation

Contents

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

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 use. 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

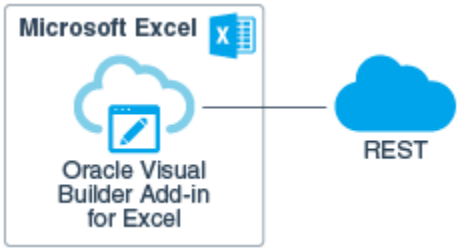
The following text conventions are used in this document.

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

1

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 the REST service.



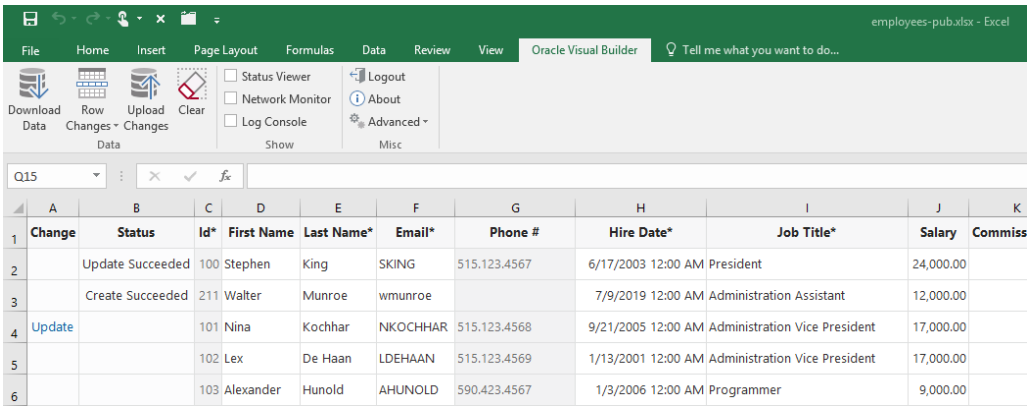
After you install the add-in, a new ribbon tab (**Oracle Visual Builder**) appears in Microsoft Excel. This ribbon tab exposes buttons to configure a worksheet to integrate with a REST service to download data to a data table that you create in the worksheet. Once the data table has been created and 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 service 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 service, 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 business object and child business objects in the REST service. 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. You can integrate Excel workbooks with REST services that provides a service description that supports the OpenAPI format. See REST Service Support.



	A	B	C	D	E	F	G	H	I	J	K
1	Change	Status	Id*	First Name	Last Name*	Email*	Phone #	Hire Date*	Job Title*	Salary	Commiss
2		Update Succeeded	100	Stephen	King	SKING	515.123.4567	6/17/2003 12:00 AM	President	24,000.00	
3		Create Succeeded	211	Walter	Munroe	wmunroe		7/9/2019 12:00 AM	Administration Assistant	12,000.00	
4	Update		101	Nina	Kochhar	NKOCHHAR	515.123.4568	9/21/2005 12:00 AM	Administration Vice President	17,000.00	
5			102	Lex	De Haan	LDEHAAN	515.123.4569	1/13/2001 12:00 AM	Administration Vice President	17,000.00	
6			103	Alexander	Hunold	AHUNOLD	590.423.4567	1/3/2006 12:00 AM	Programmer	9,000.00	

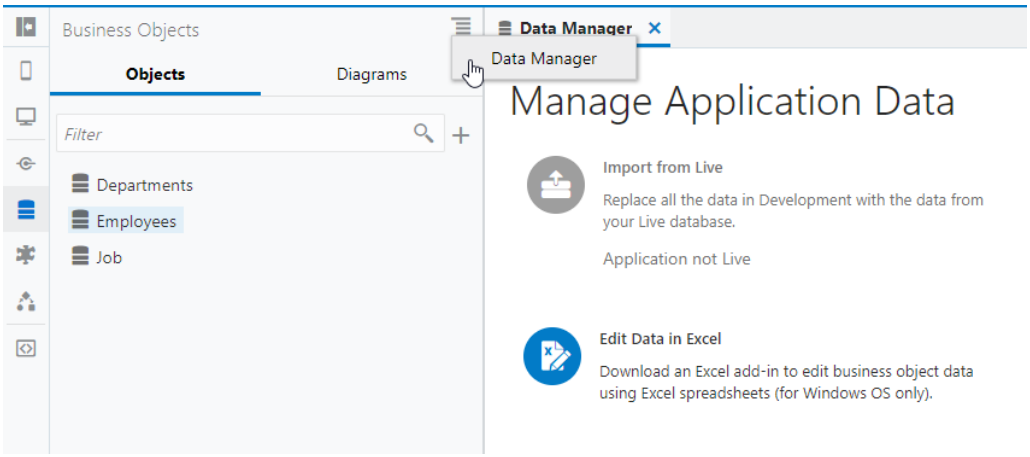
2

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.

> **Tip:**
>
> If you are upgrading from version 1.x, please review Migration first.

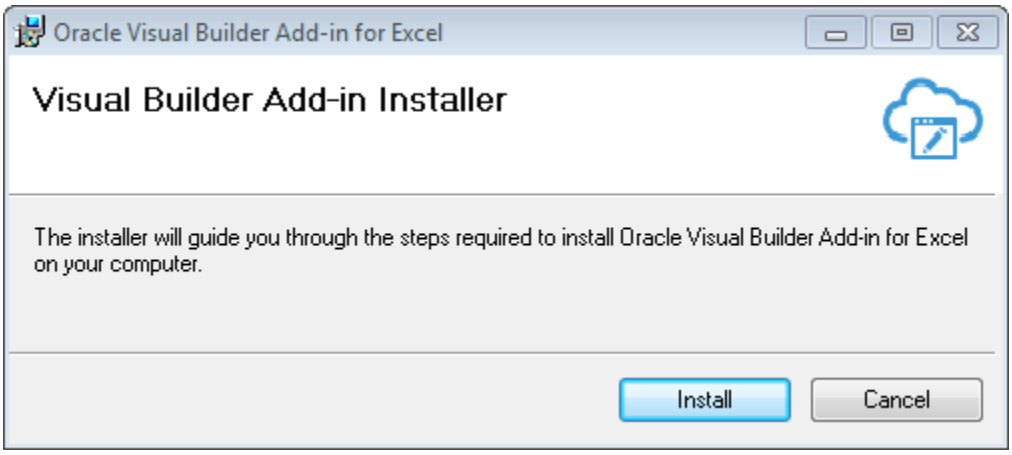
You can install the Oracle Visual Builder Add-in for Excel in the 32-bit version of Microsoft Excel 2016 on a computer using the Windows 10 operating system. For more details, see *Supported Platforms for the Visual Builder Add-in for Excel* that you retrieve from My Oracle Support (https://support.oracle.com)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



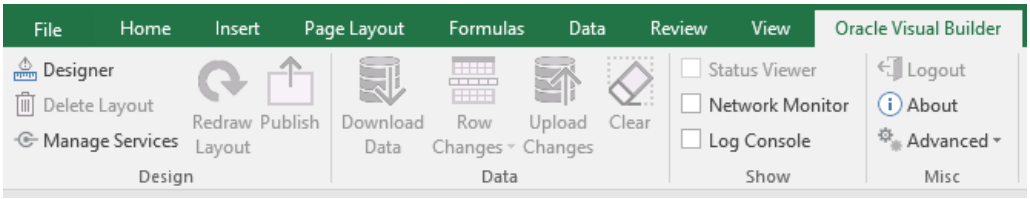
Close all open instances of Excel before you double-click the `vbafe-installer.exe` installer file to launch the installation wizard. The following image shows the start screen of the installation wizard. Once you click **Install**, the installation wizard proceeds to install the add-in. It also verifies 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.



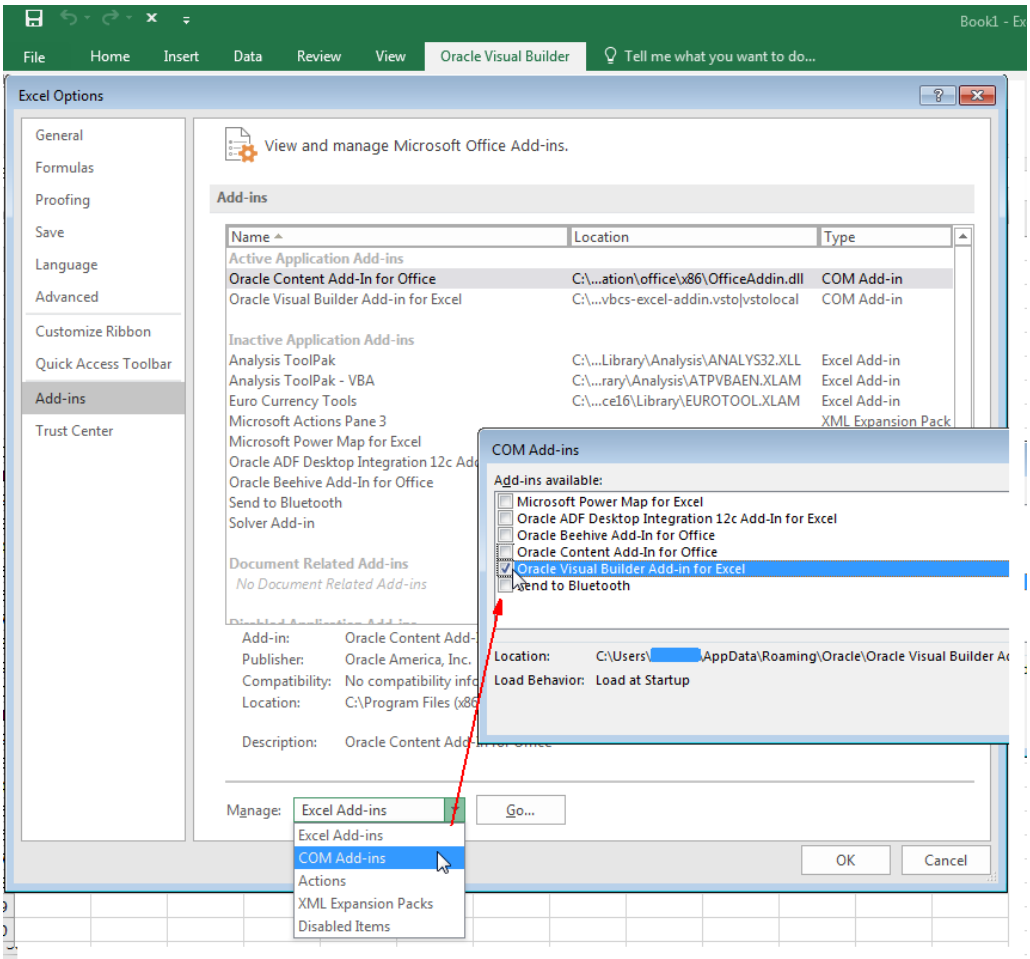
Once the installation of the add-in completes, start Microsoft Excel where you will see a new ribbon tab (**Oracle Visual Builder**), as shown in the following image. You use the commands in this ribbon tab to integrate the Excel spreadsheet with the REST service.



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

```
%TEMP%\vbafe\vbafe-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 window 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

Switch	Description
`/help`	Displays a list of supported switches with description.
`/quiet`	Suppresses the interactive mode of the installer and does not install any missing prerequisite software.
`/log <path>`	Runs the installer and directs the log output to the specified log file. The default log file location is `%TEMP%\vbcs\vbcs-installer-log.txt`.

Table 2-1 (Cont.) Oracle Visual Builder Add-in for Excel Installer Command-Line Switches

Switch	Description
`/roaming <0|1>`	Use this switch as follows: • `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. • `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.

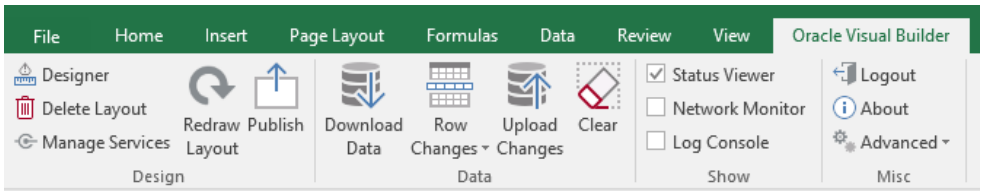
3

Create a Table Layout in an Excel Workbook

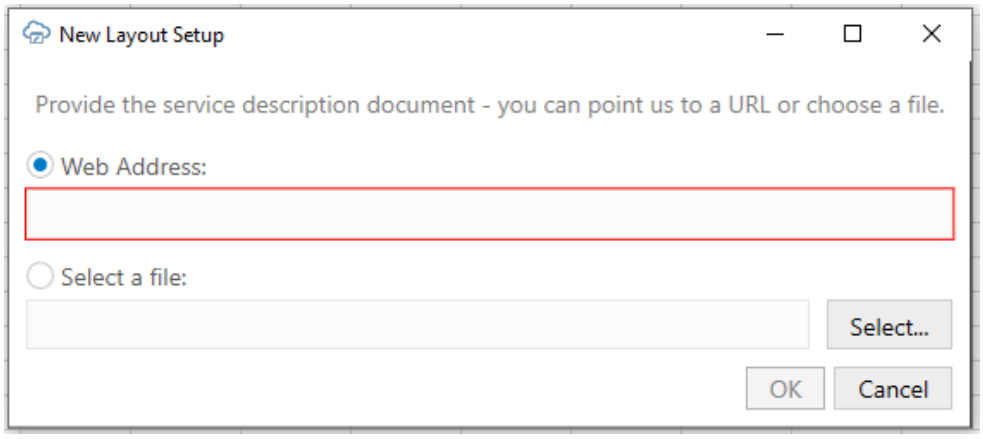
Create a Table layout in the Excel worksheet when you want to render the data that the add-in downloads from the REST service to the Excel workbook in a tabular format.

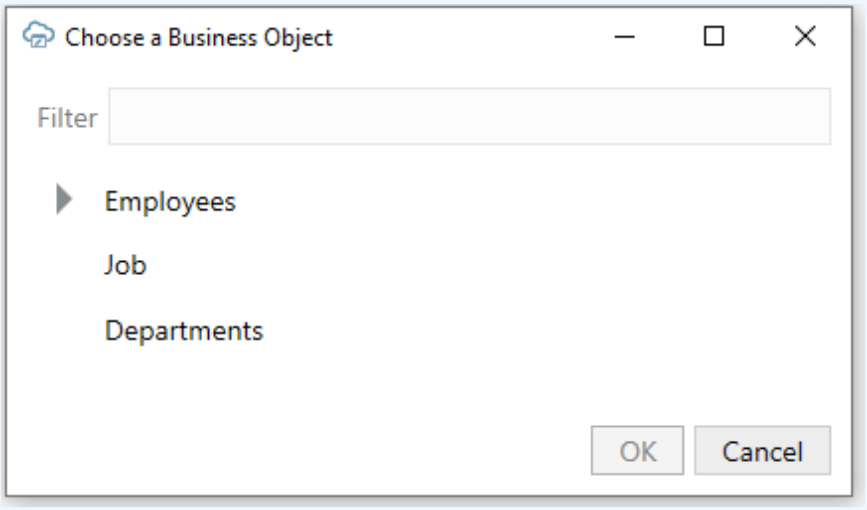
Run Excel and create a blank workbook using the standard Excel workbook file format type (`.XLSX`) or the macro-enabled workbook type (`.XLSM`). The Oracle Visual Builder Add-in for Excel does not support other Excel formats (`.XLS`, and so on).

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

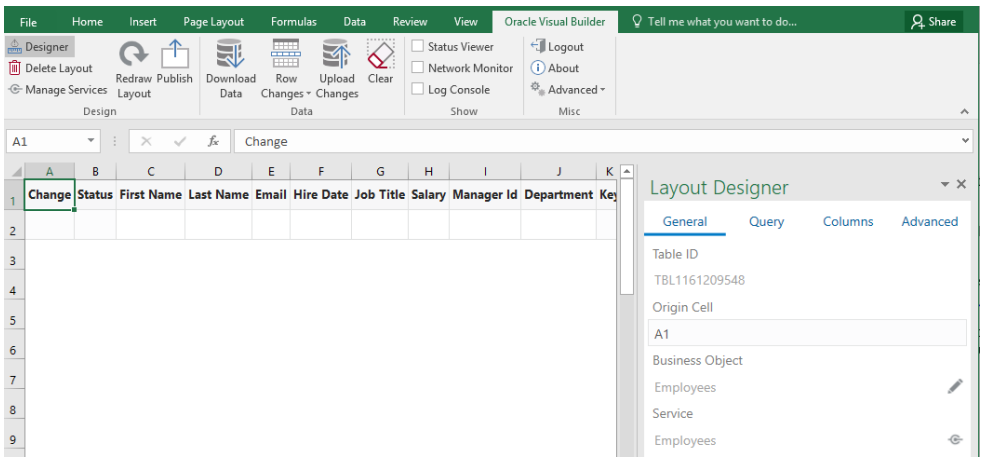


2. Click on the cell where you want to locate the table.
3. In the Oracle Visual Builder tab, click **Designer**.
4. When prompted, provide the service description document. Use the **Web Address** option (the default) if you access the service description from a URL. Use the **Select a file** option if the service description document is a local file on your computer.



> **Note:**
>
> If multiple business objects are found, the Choose a Business Object window prompts you to choose from the available business objects.
>
> 
>

The add-in creates a data table in the Excel workbook that renders Column 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 desired columns. After making changes in the Layout Designer, click **Redraw Layout** to see your changes reflected in the worksheet.

> **Note:**
>
> In the Columns tab of the Layout Designer, right-click a column to see choices for changing the order.

The workbook 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 Configure Search Options for Download.

Path Parameters

Some service paths include path parameters. The add-in includes support for that parameters that are defined in an OpenAPI service description document, and have a data type property. The add-in support path parameters that have string and integer data types. Other data types are not supported. For string-typed path parameters, values that users enter into the Download Path Parameter Editor are used verbatim in the resultant URL path that the REST request uses. For integer-typed values, certain culture-specific formatting is removed (for example, commas for thousands separators, parentheses for negative). In all cases the values used on the URL path are not URL-encoded, so the values entered must be acceptable by the REST service.

The following is an example of a service path with an embedded parameter:

```
/invoices/{invoices_Id}/child/invoiceLines
```

Note `{invoices_Id}` in the middle of the service path.

If you try to access this path as is, you get an error. Instead, you must substitute a proper value for `{invoices_Id}`, as in the following example:

```
/invoices/123456/child/invoiceLines
```

Accessing this service path should provide all the invoice lines for invoice number `123456`.

The add-in provides direct support for configuring a Table layout using a parameterized service path. The add-in extracts the path parameters automatically and prompts the user to provide the corresponding values at download time.

To configure this implementation in your Table layout, provide an OpenAPI-compliant service description. When prompted, choose a child business object or any parameterized path from the business object picker.

> **Tip:**
>
> When working with Oracle RAMP REST services, you should start with the web address to the parent BO description (and not the child address). In the example above, you would provide the address to the invoices description (not invoice lines). Oracle RAMP REST services cannot provide OpenAPI service descriptions for parameterized service paths.

Complete the layout configuration. When users click **Download** in the Oracle Visual Builder tab, the add-in displays the Download Path Parameter Editor where users provide the required path parameter values that enables the download of data to complete.

The Download Path Parameter Editor does not validate the value(s) that users enter. The value(s) that users provide must be valid. If the path includes multiple embedded parameters, the Download Path Parameter Editor prompts the user to provide a value for each embedded parameter.

4

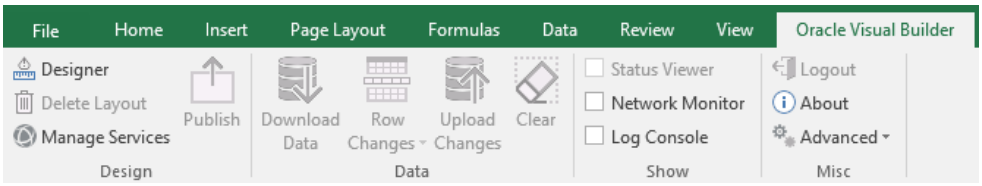
Create a Form-over-Table Layout in an Excel Workbook

You can create a Form-over-Table layout in an Excel worksheet when a parent-child relationship exists in the data that the add-in downloads from an Oracle RAMP REST Service.

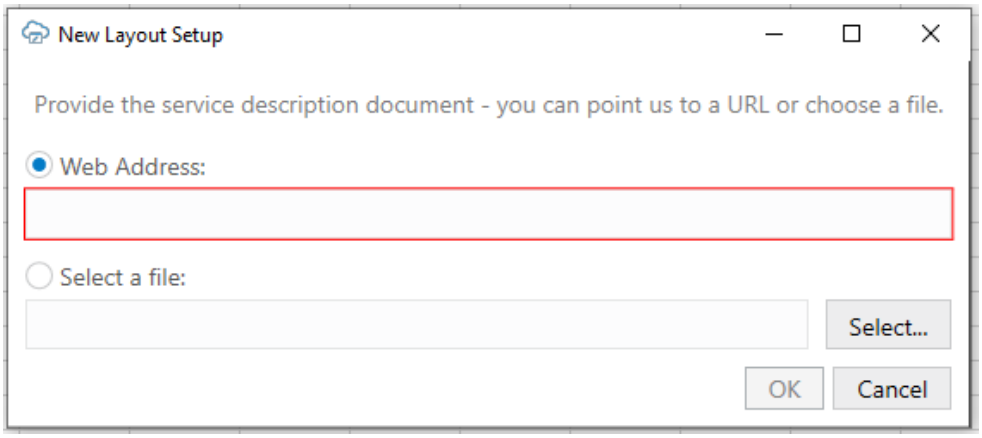
The add-in displays a window where you can choose the Form-over-Table layout if the business object you chose has one or more child business objects. If the add-in finds more than one such child business object, it prompts you to pick the business object you want to use in the Form-over-Table layout.

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.

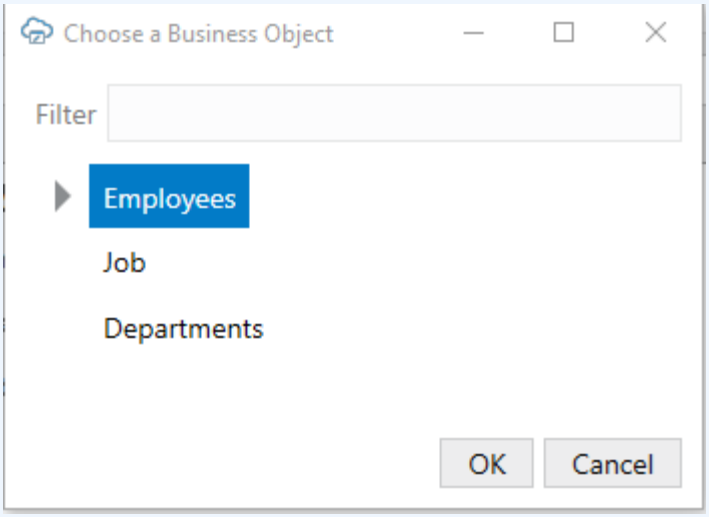


2. Click on the cell where you want to locate the form and table.
3. In the Oracle Visual Builder tab, click **Designer**.
4. When prompted, provide the service description document. Use the **Web Address** option (the default) if you access the service description from a URL. Use the **Select a file** option if the service description document is a local file on your computer.



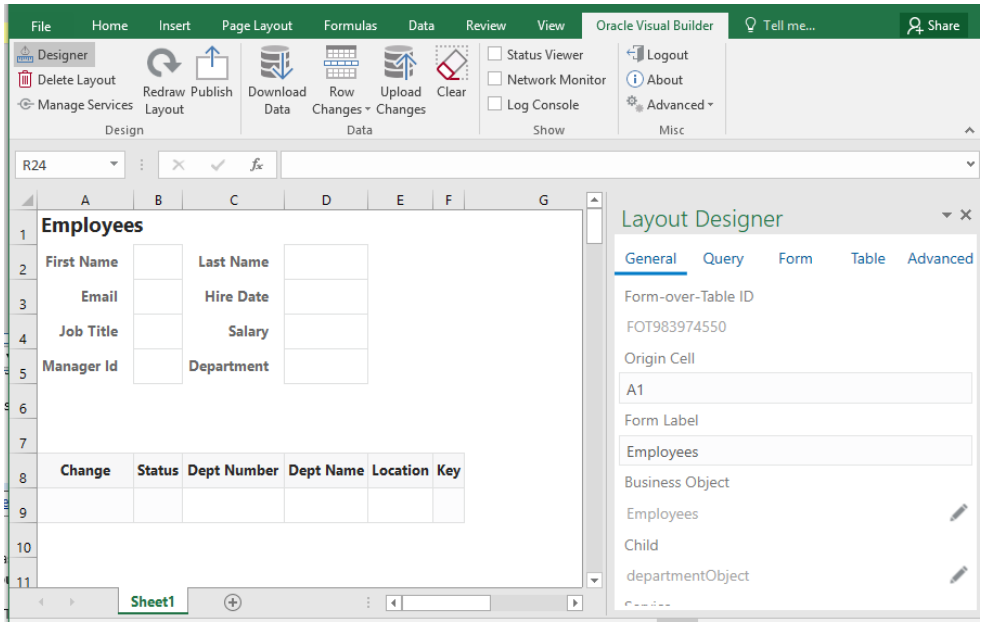
Note:

If your OpenAPI service description includes multiple business objects, a window similar to the following prompts you to choose from the available business objects.



5. Choose a business object, such as **Employees** as illustrated in the previous image, where a parent-child relationship exists, and choose **Form-over-Table Layout** in the New Layout Setup window 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. Click **Redraw Layout** to see changes you make in the Layout Designer reflected in the form and table.

If, for example, you do not specify a value for the Search field or Finder property, as described in Configure Search Options for Download, 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.

> **Tip:**
>
> In the Form or Table tab of the designer, right-click a field or column to see choices for changing the order.

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 Configure Search Options for Download.

5

Edit Service Descriptions and Business Objects

The Excel add-in provides editors where you can modify the service description that the workbook uses.

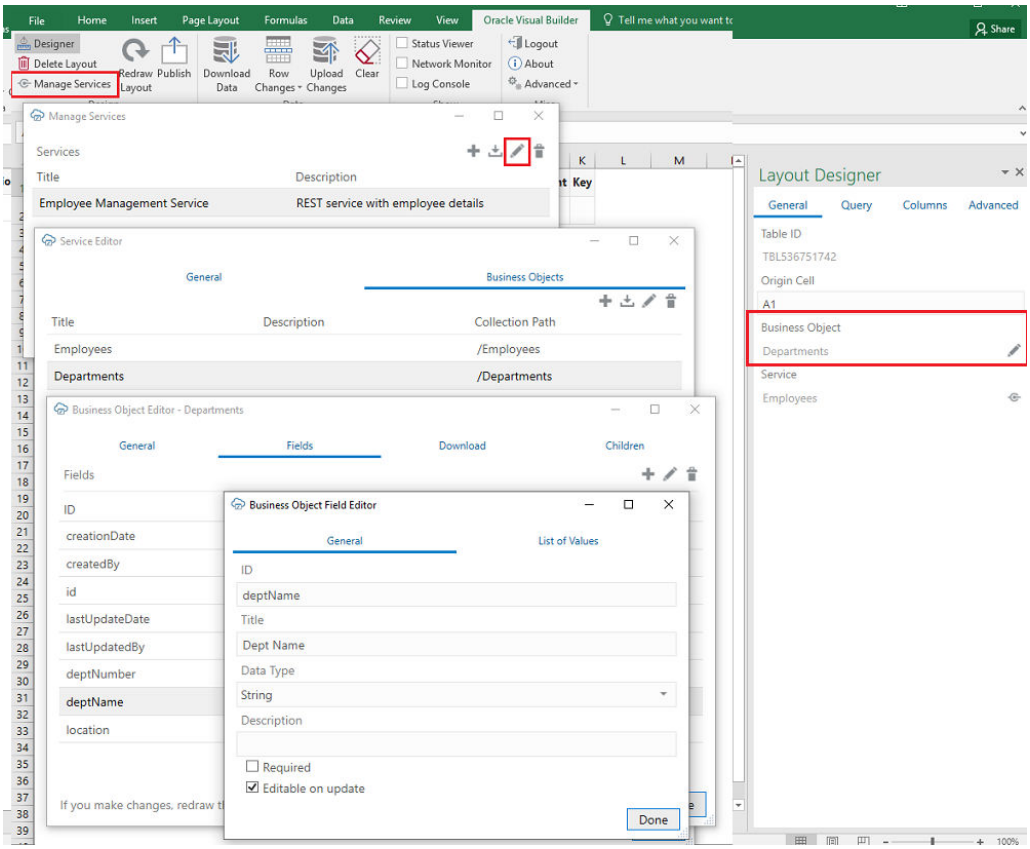
Using these service description editors, you can change the URL for the service host that provides access to the REST service your workbook uses, or you can add or remove fields that the business object exposed by the REST service supports. For the latter task, note that any changes that you make to a business object must be supported by the REST service that the workbook uses. If, for example, you add a field to a business object in the service description that the workbook uses, and the REST service does not support this field, errors occur when the workbook connects to the REST service.

Other examples of tasks that you can accomplish using these editors include:

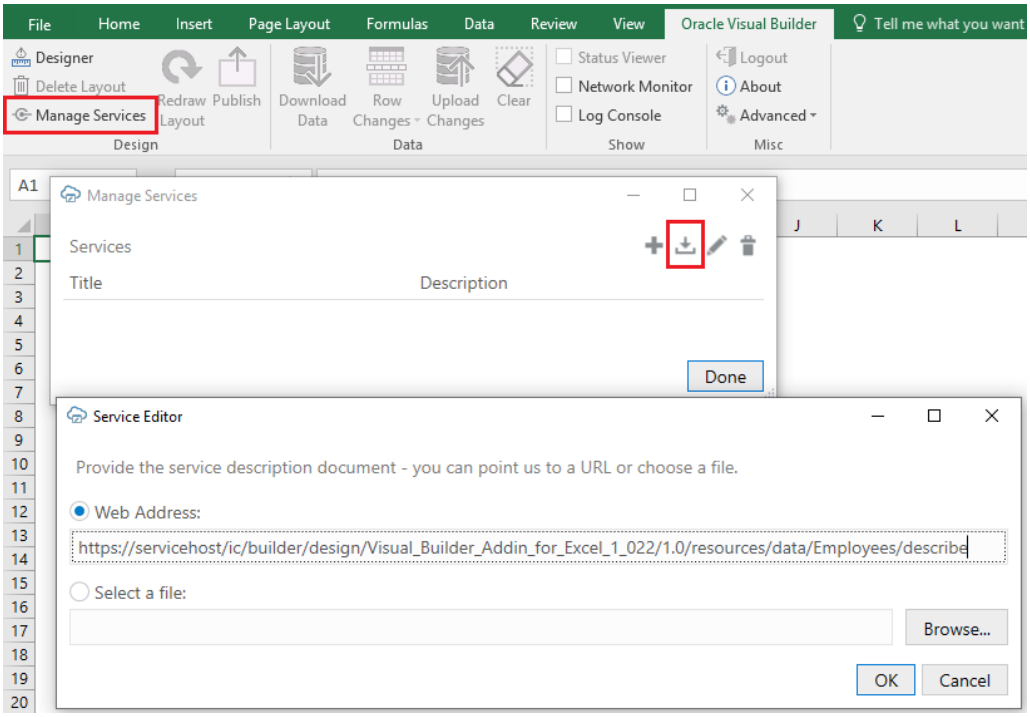
- Edit field titles
- Configure the list of values for a field
- Change where a field is editable
- Adjust the field data types (Advanced)

Using the service description editors, you can improve the service description in a variety of ways to enhance the overall user experience.

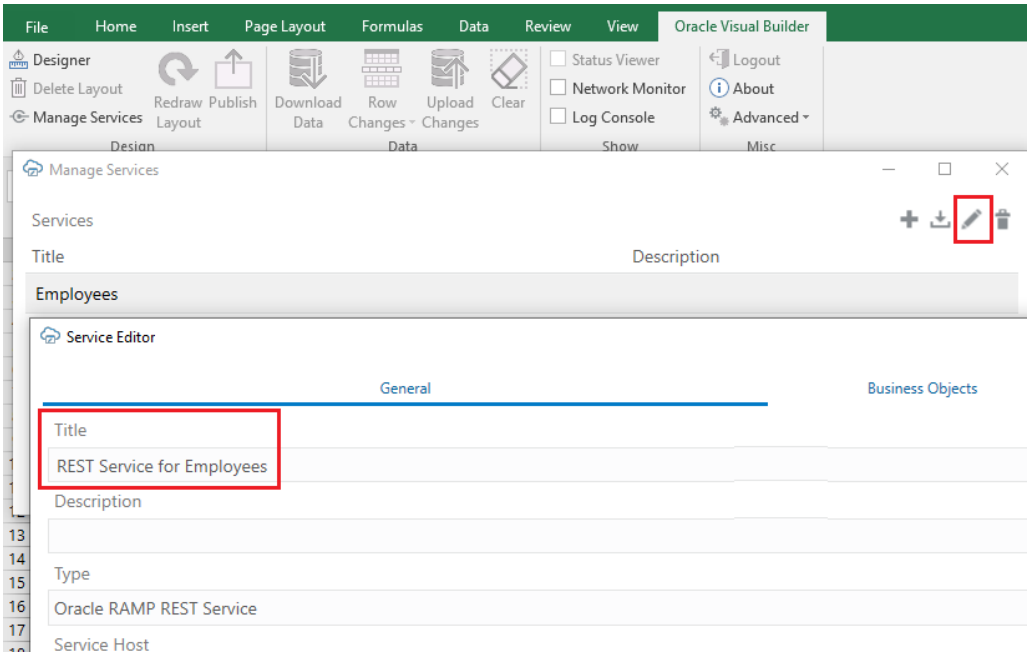
The add-in displays the editors progressively. That is, you access the editor to edit the REST service from the **Manage Services** button and you can then access the subsequent editors for the business objects and business objects fields that the service provides access to. You can also open the editor(s) for business objects and business object fields from the Layout Designer. Both options, using the **Manage Services** button and the Layout Designer, are illustrated in the following image.



When you click the **Manage Services** button in a blank Excel workbook that has not been integrated with a REST service using the Oracle Visual Builder add-in, the add-in prepares the workbook and displays a Manages Service window from where you can open a Services Editor to import the service description document.



After you import a service description document into the Excel workbook, consider editing it in the Services Editor so that the service has a descriptive name. Very often, the title that the add-in displays for the service in the Excel workbook is the name of one of the business objects that the service exposes. For example, the following image shows a service where the default value for the Title property is Employees. This service exposes an Employees business object, so to avoid confusion the user changes the title that the service uses in the Excel workbook to REST Service for Employees.



After making changes in the Service Editor, such as changing the service host or details of a business object, click **Redraw Layout** in the Oracle Visual Builder tab to see your changes reflected in the worksheet.

Create a Service Description in Oracle Visual Builder

You can create an OpenAPI-compliant service description in Oracle Visual Builder.

1. Sign in to your Visual Builder account in the Oracle Cloud.
2. Create a new visual application or open an existing visual application.
3. Create a service connection from an endpoint. See Create a Service Connection from an Endpoint.
4. Configure the service with the details that you will need in the Excel workbook integration.
5. In an Oracle Visual Builder web application, test the service connection by creating a sample page and table that retrieves data from the service.
6. In the source view of the visual application, locate and download the service.json file to your computer.

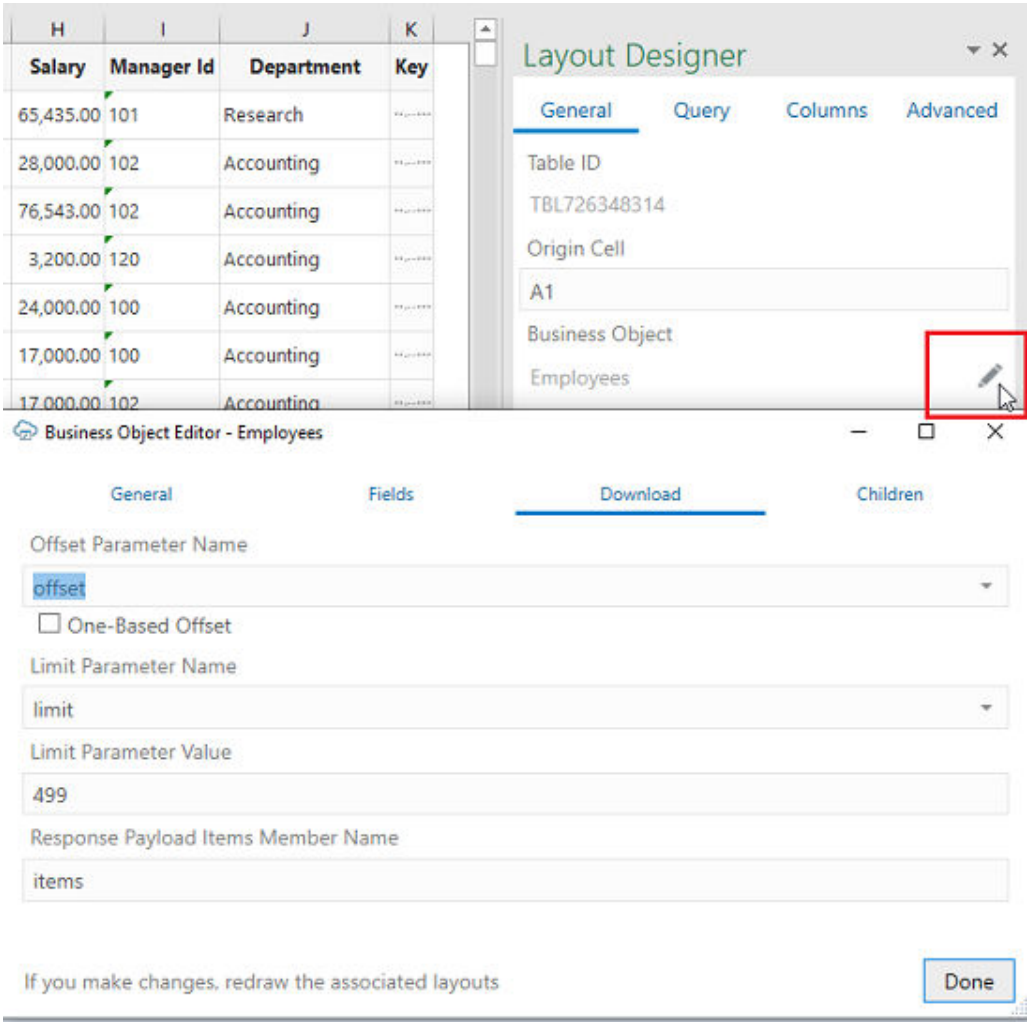
   The `service.json` file contains the OpenAPI service description for the REST service that you connected to from your Visual Builder visual application.

Configure Pagination for a Business Object

The add-in supports downloading pages of rows if the REST service supports pagination.

Imagine that you need to download 10,000 rows of data. Downloading one row at a time is too slow. Attempting to download all 10,000 rows in one request might time out. So, the solution is to download one page at a time where the page contains, for example, 500 rows.

You can configure the pagination behavior using the Download tab in the Business Object Editor.



The following list describes a number of the properties that the Download tab allows you configure values for:

- **Offset Parameter Name**: Controls where to start the next page. When fetching the first page, the add-in uses a value of zero. When fetching the second page, the add-in uses a value of 500, assuming the limit value is 499.
- **Limit Parameter Name**: Controls how many rows to fetch for each page.
- **Limit Parameter Value**: Controls the page size (number of rows that the add-in downloads).

For other service types, pagination may or may not be supported. If supported, the service may use parameter names like `offset` and `limit` or it may use other parameter names for the same purpose.

Consult the service API documentation to determine which parameters to use.

Notes

Certain OpenAPI document properties, such as Description, can contain formatting hints. The add-in displays the description text as is with no interpretation of such hints.

6

Configure Search Options for Download

Configure the search options for a given layout. The configured search options are used when the user clicks **Download** in the Oracle Visual Builder tab.

The add-in provides search options that you configure for each layout. The properties that you can configure depend on the type of service that you integrate your Excel workbook with. For example, all properties are available to use when your workbook integrates with an Oracle RAMP REST service while you can only configure values for the Search and Search Parameters properties when you use Oracle REST Data Services.

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 Oracle RAMP REST service or Oracle REST Data Services. 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 Oracle RAMP REST service you connect to supports finders. If you select a value for the REST Finder property, the add-in uses it during download. A finder that a Oracle RAMP 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.

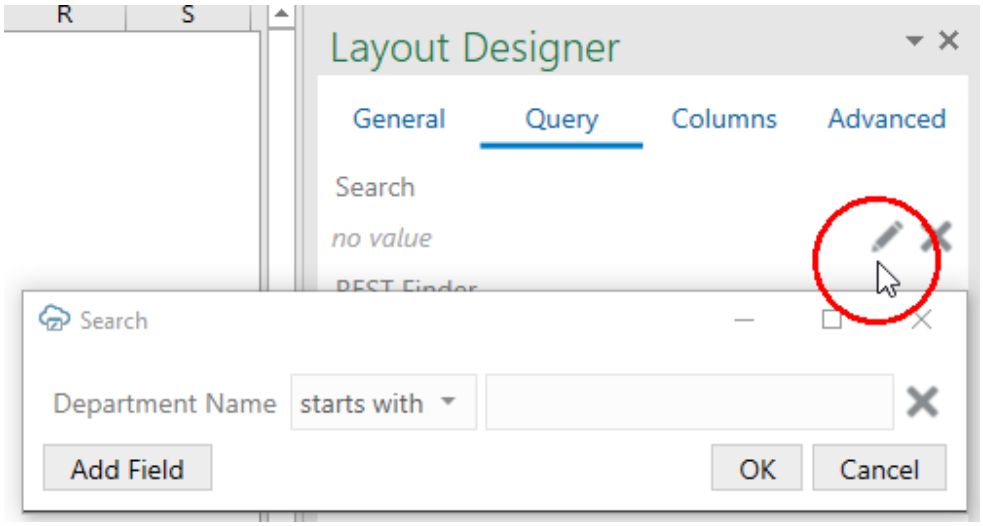
Add values for Search Parameters if the REST service that you download data from supports the addition of various parameters in the query portion string.

Depending on the service you use, you can configure one or more of these properties. If you configure values for all 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

For workbooks that integrate with Oracle RAMP REST service and Oracle REST Data Services, 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 window.
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.



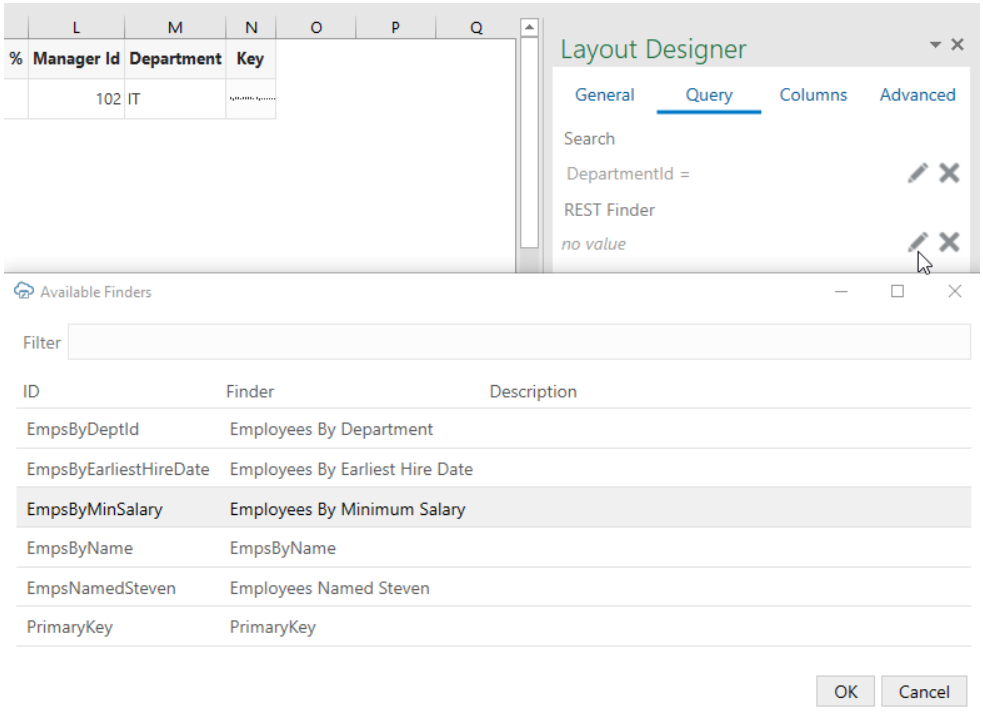
4. Click **Add Field** to add additional search fields.
5. Click **OK**.

Use REST Finders to Limit Downloaded Data

For workbooks that integrate with Oracle RAMP REST services, you may select one of the pre-defined REST Finders if any are associated with the layout's business object.

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 window.



3. Click **OK** to close the open windows.

Refer to the following resources for more information about configuring the Oracle RAMP REST service that supports finders:

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

Use Search Parameters to Limit Downloaded Data

You add search parameters to the workbook to include in the request to download data from the REST service.

Many REST services support the addition of various parameters in the query string portion of the URL, such as the following example.

```
GET…/orderReleaseLines?q=ID=1234
```

In this example:

- "q" is an optional parameter supported by the orderReleaseLines service
- "ID" is the name of a field that supports query
- 1234 is the search value

Each REST service defines which parameters can be used for search. Likewise, each REST service defines the required and supported syntax for the expression that appears on the right-hand side of the assignment operator (=).

The add-in does not know which parameters are useful for search. Likewise, the add-in does not know the proper syntax for the parameter values. So, you will need to consult the API documentation for the REST service you are using to identify whether to use "q" for the parameter name and how to formulate the search expression properly.

There is no validation in this editor or at download time. If you enter invalid information, you may get a bad request error.

When the user clicks the Download Data button, the search parameters are added to the appropriate URL along with the other search options (if applicable).

The add-in applies URL encoding to the parameter value at download time. So, you should not enter URL-encoded values. The search parameter name is not encoded.

If the REST service supports complex searches, you can create complex searches in the layout, as shown in the following example

```
q=((firstName LIKE '*es*') or ((hireDate< "2001-01-13") and (department =
10)))
```

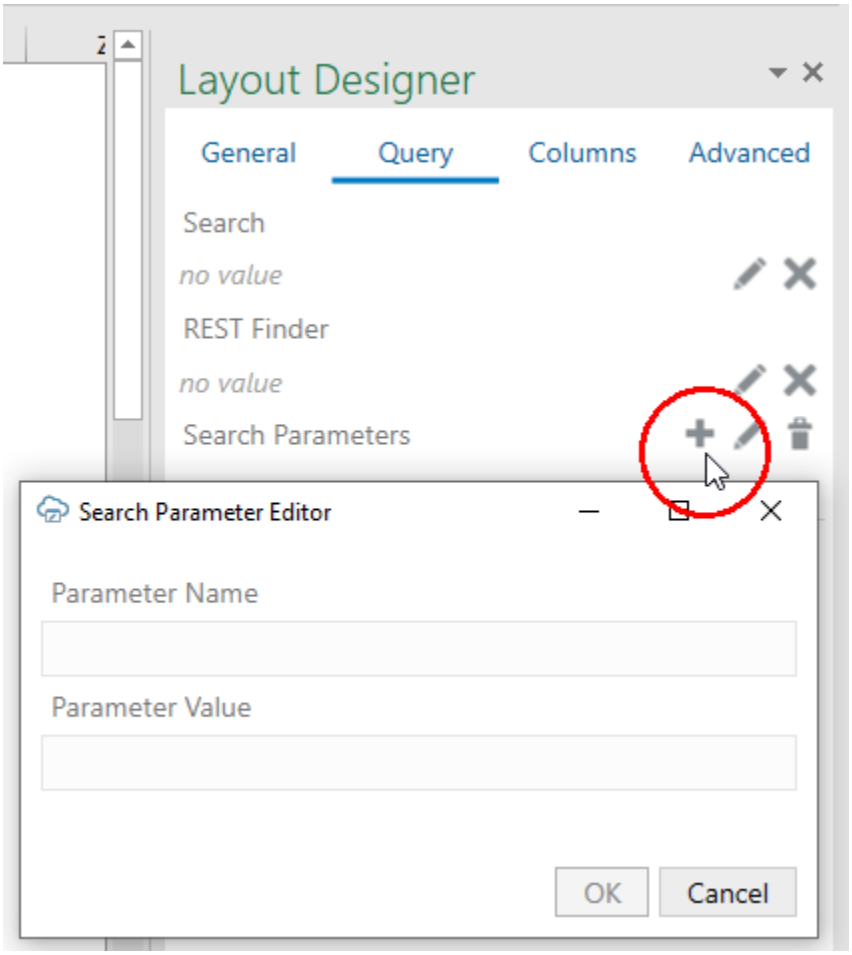
The Search Parameters property does not support "dynamic" search values. Using the example above, the value, 1234, is specific in the Layout Designer as a constant.

The Search Parameters property works in combination with the other two properties (where applicable). They are not mutually exclusive. However, some combinations may work where others may not. If you choose to configure multiple search options, you must ensure that the REST service supports that combination.

To add search parameters to your layout:

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

2. In the Query tab of the Layout Designer, click the add or edit search parameter icon next to the Search Parameters property to open the Search Parameter Editor where you add or edit search parameters.



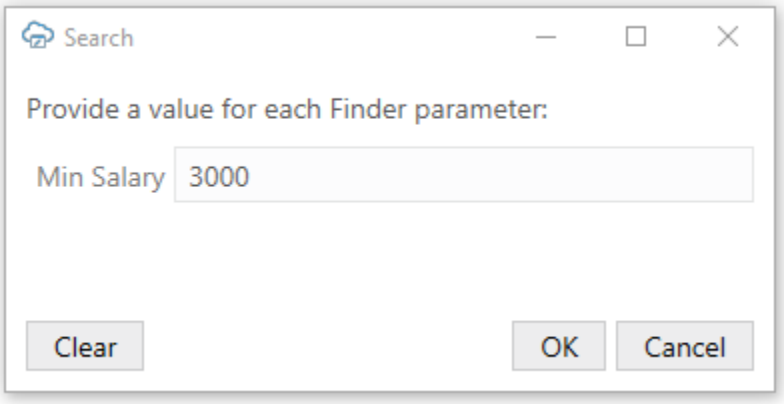
3. Click **OK** to close the open windows.

Download Behavior in Layouts that Use Search and REST Finders

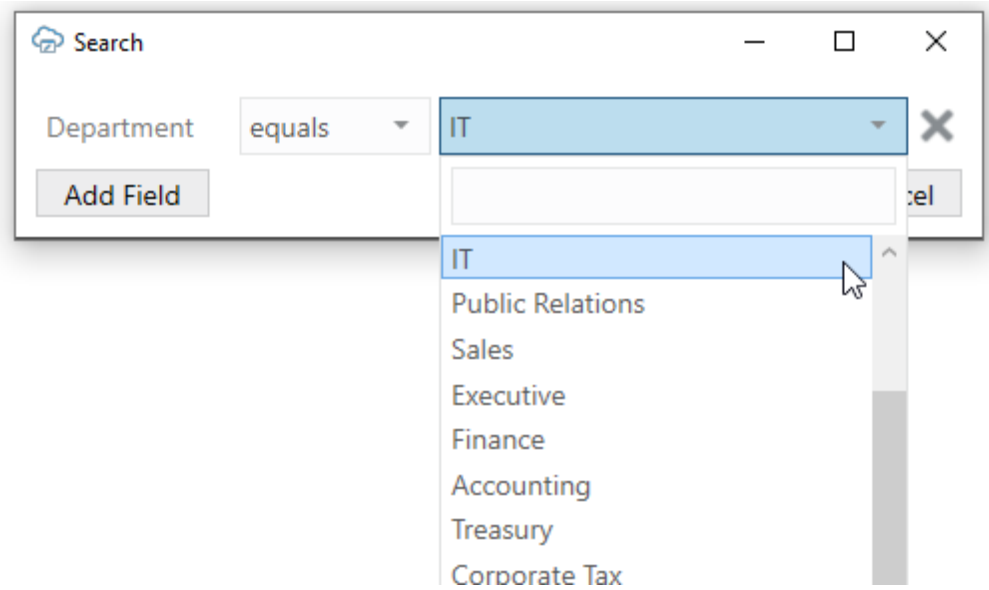
If you configured a value for the REST Finder property, the behavior of the add-in depends on the configuration of the finder exposed by the REST service:

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

> **Note:**
>
> Only Oracle RAMP REST services 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.



7

Use Lists of Values in an Excel Workbook

The Oracle Visual Builder Add-in for Excel provides support for a list of values for business object fields.

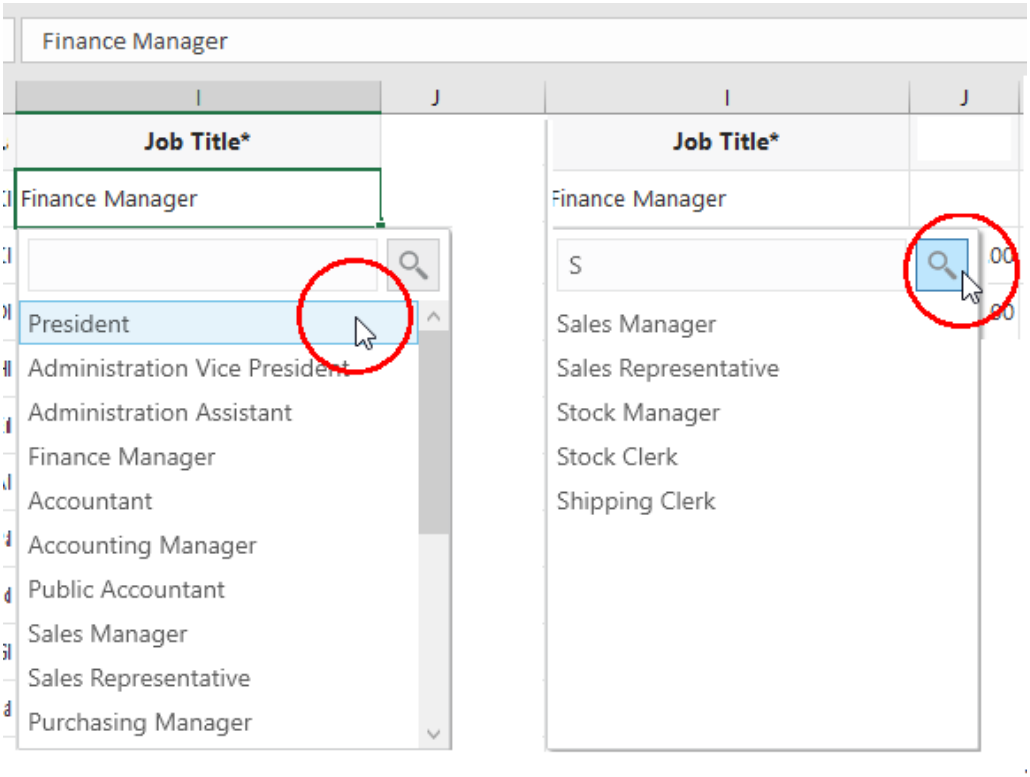
A *list of values* represents a relationship between a given field in one business object and a separate referenced business object. The list of values is the items that constitutes the valid set of values for a business object field. Each item in the list of values has a display value (appears in the Excel workbook) and an identity value, that is retrieved and posted to the business object field. For example, a Job attribute for an Employee business object might contain the following display and identity values.

For display values, the add-in supports lists of values of string. For identity values, the add-in supports integer or string. Decimal numbers, dates, and date-time values are not supported for lists of values.

Display value	Identity value
President	PRES
Finance Manager	FIN_MGR
Sales Manager	SAL_MGR

When a business object field has a properly configured list of values, you can expect the following behavior in Table and Form-over-Table layouts in your Excel workbook:

- On download, the identity values are replaced by the display values
- On upload, the display values are replaced by the identity values
- When a user selects a cell in a table column that is not read-only, a search box appears that displays available values. A user can select one of the displayed values. Alternatively, the user can type one or more of the starting characters for other values and then click the Search icon so that the search box 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 search box 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`.



The add-in supports multiple display fields for one list of values. In that case, values of all display fields are concatenated and the concatenated value is shown in a single Excel cell.

> **Note:**
>
> When using a list of values with multiple display fields, if you enter a value directly in the cell that has not yet been cached, the value will be considered invalid initially. So, use the search box to select the value instead. The default search is specific to the first display field. You can modify the search expression using the Business Object Field Editor.

Configure a List of Values for a Business Object Field

You can use the Business Object Field Editor to add or modify the list of values configuration for any eligible field.

You access this editor from the Edit icon beside the Business Object field in the Layout Designer, or by double-clicking a column or form field in the Layout Designer. Once you open the Business Object Editor for the business object, click the **Fields** tab and select the desired business object field. To open the Business Object Field Editor, click the Edit icon that appears in the upper right of the Business Object Editor, as shown in the following image where the user opened the Business Object Field Editor for the employee business object's Job Title field.

The List of Values tab in the Business Object Field Editor displays the following properties:

- **Enabled**: Selected if the business object field has a list of values

- **Referenced Business Object**: This business object provides the display values for the matching identity values
- **Identity Field**: The field in the referenced business object used to look up the display values for the identity values in the current field
- **Display Fields**: These fields come from the referenced business object and are shown instead of the identity values in the current field

To configure a list of values for a business object field that does not have one yet:

1. Select the **Enabled** checkbox in the List of Values tab of the Business Object Field Editor.
2. Select the appropriate business object using the Edit icon beside the **Referenced Business Object** field.

> **Tip:**
>
> If your service definition is missing the desired business object, you can add a new business object description to an existing service description. Click **Manage Services**. Open the desired service and click the **Import** button.

3. Choose the appropriate identity field from the referenced business object.
4. Choose the desired display field from the reference business object.

The properties that appear under the Simple Search Transform section determine how the add-in retrieves the values to appear in the search box next to the cell in the worksheet. The query parameter value is a string expression that may include `{$options.simpleSearch}`. The add-in replaces this expression with the input the user types in the search box. The add-in does not perform any validation on the values that you enter, so the add-in will make invalid requests that return errors if you enter invalid information.

Add values for the Name and Value properties based on what the REST service you use supports. Many REST services support a specific query parameter like `q` for search, as in the following example:

```
GET…/employees?q=LAST_NAME LIKE 'Jones*'
```

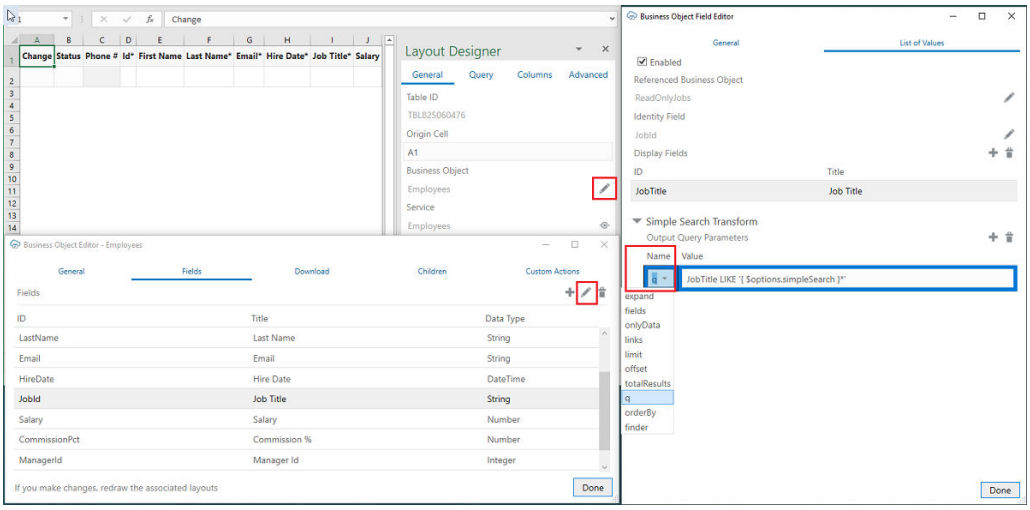
Select the appropriate query parameter in the Name dropdown list. In our example, this is: `q`. In the Value column provide a search expression that is appropriate for the REST service and include the `{$options.simpleSearch}` search text placeholder where appropriate. In our example, the Value is:

```
LAST_NAME LIKE '{$options.simpleSearch}*'
```

If the user types `Jo` in the search box and clicks the **Search** icon, the resulting request to the referenced business object will be something like:

```
GET.../employees?q=LAST_NAME LIKE 'Jo*'.
```

> **Note:**
>
> Consult the API documentation for the service you have chosen to use to determine the appropriate search syntax to use in the Value column. For example, if you use an Oracle RAMP REST service, you want to consult Understanding Framework Support for Query Syntax. Remember that the add-in uses REST framework version 6 when making requests to a RAMP service.



The add-in caches the data of list of values in the workbook. After you modify the configuration of any list of values, click **Redraw Layout** to redraw the layout and clear the cached data. Then, click the **Clear List of Values Cache** button that you access from the **Advanced** dropdown list in the Misc ribbon tab.

Edit Service Descriptions and Business Objects provides more detail about using the editors that the add-in provides to modify the service description that your workbook uses.

Configure a Cascading List of Values in a Layout

The add-in also includes support for cascading list of values if the REST service that the workbook uses is of type Oracle RAMP REST Service. 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 service that the add-in connects to must meet the follow requirements:

- The cascading list of values resource must be configured using static links with finders. For example, for Countries, States, and Cities, the item links for the parent business object must be configured as follows:

```
"links": {
  "CountryView1": {
    "operationRef": "http://servicehost/Countries/describe#/paths/
~1Countries/get",
```

```
      "x-lov": {
        "definedOn": "$request.body#/CountryId",
        "valueField": "Id",
        "displayFields": [
          "Name"
        ]
      }
    },
    "StateView1": {
      "operationRef": "http://servicehost/States/describe#/paths/
~1States/get",
      "x-lov": {
        "definedOn": "$request.body#/StateId",
        "valueField": "Id",
        "displayFields": [
          "Name"
        ]
      },
      "parameters": {
        "finder": "ByCountryFinder%3BCurrentCountry%3D{CountryId}"
      }
    },
    "CityView1": {
      "operationRef": "http://servicehost/Cities/describe#/paths/
~1Cities/get",
      "x-lov": {
        "definedOn": "$request.body#/CityId",
        "valueField": "Id",
        "displayFields": [
          "Name"
        ]
      },
      "parameters": {

"finder": "ByCountryAndStateFinder%3BCurrentCountry%3D{CountryId}
%2CCurrentState%3D{StateId}"
      }
    }
```

- 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 business object fields 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.

For any read-only field (column or form), the add-in still swaps identity values for display values and vice versa, as described above. However, the search box does not appear when the cell is selected.

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

- The add-in supports list of values that are defined with a single identity field.

- Cascading list of values is only supported for Oracle RAMP REST services.
- Cascading list of values behavior is not configurable in the Business Object Field Editor.
- For Oracle RAMP REST services, "row context" list of values are not supported.

8
Custom Actions

Workbooks that integrate with Oracle RAMP REST services can perfom custom actions.

Describe Custom Actions

If a given REST API supports custom actions, they are described in the OpenAPI v3 service description document generated by the Oracle RAMP REST service. For example, an `adjustSalary` custom action would appear in the `paths` collection:

```
// Note: some JSON content has been omitted for brevity/clarity

"/Employees/{Employees_Id}/action/adjustSalary": {
"parameters": [
{
  "$ref": "#/components/parameters/Employees_Id"
}
],
"post": {
"tags": [
  "Employees"
],
"summary": "adjustSalary",
"description": "adjustSalary",
"operationId": "do_adjustSalary_Employees",
"parameters": [
  {
    "$ref": "#/components/parameters/REST-Framework-Version"
  }
],
"responses": {
  "default": {
    "description": "The following table describes the default response
for this task.",
    "content": {
      "application/vnd.oracle.adf.actionresult+json": {
    "schema": {
      "type": "object",
      "properties": {
        "result": {
          "type": "string"
        }
      },
      "required": [
        "result"
      ],
      "additionalProperties": false
    }
      },
```

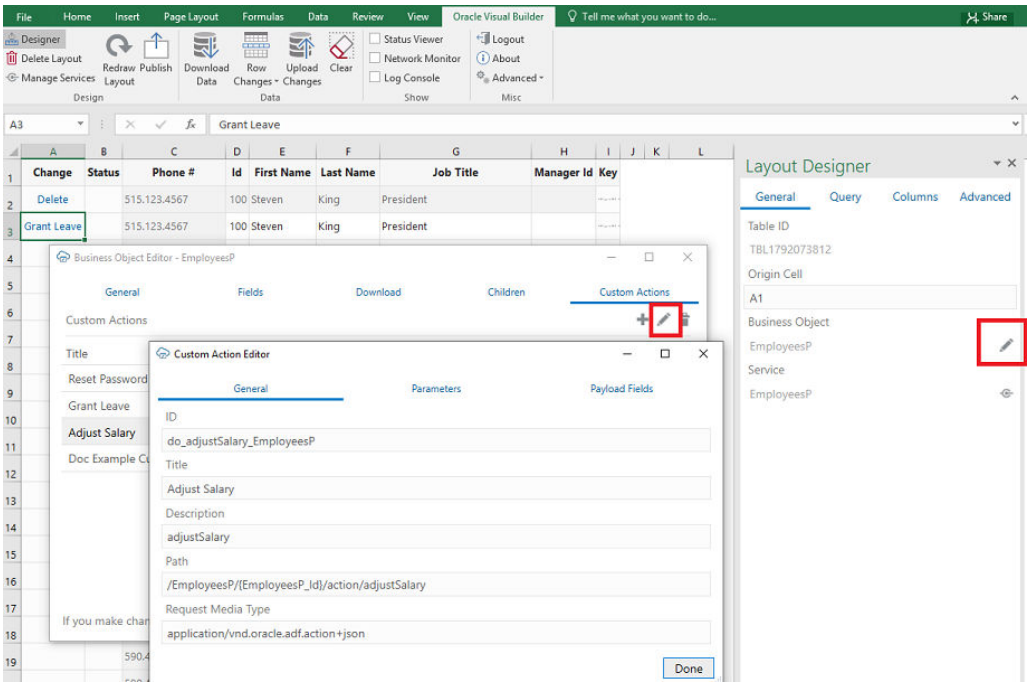
```
    }
  }
},
"requestBody": {
  "description": "The following table describes the body parameters in
the request for this task.",
  "content": {
    "application/vnd.oracle.adf.action+json": {
      "schema": {
    "type": "object",
    "properties": {
      "proposedSalary": {
        "type": "number",
        "nullable": true
      },
      "ajdustmentReason": {
        "type": "string",
        "nullable": true
      }
    },
    "additionalProperties": false
      }
    }
  }
}
}
}
```

Note the following:

- The path entry is a templated-string that includes the row/item ID. For example `/Employees/{Employees_Id}/action/adjustSalary`.
- The end of the path entry (`adjustSalary`) exactly matches the Java method name.
- The presence of a POST operation for the action path entry is required.
- The tags property under the post operation maps the custom action definitions to the corresponding business objects.
- In the `requestBody` schema, there are properties that match the parameters defined in the Java method signature. In this document, these properties are referred to as custom action Payload Fields.

Use Custom Actions

Custom actions exposed by the service that your workbook uses can be viewed and edited in the Custom Actions tab of the Business Object Editor. Properties such as Title and Description can edited as needed. The value of the Title property appears in the UI that the user of the workbook sees. Other properties of the custom action should generally be left as is. Payload fields can be edited by clicking the Edit button or double-clicking an entry in the list.



To pass argument values that the custom action needs, such as proposed salary, you create new columns in the Table Layout that map to the payload field of the custom action.

1. In the Layout Designer, click the **Columns** tab and click the **Add a column for a custom action payload field** button ( ).
2. In the Custom Actions dialog that appears, expand the custom action to select the payload field, and click **OK**.

The add-in adds a new column to the Table layout.

After downloading some existing rows, you perform the custom action as follows:

1. Select one or more rows in the Table Layout or in the table of a Form-over-Table layout.
2. Click the **Row Changes** drop-down list and click **Mark for Action**.

If more than one custom action is defined for the business object, a dialog appears that allows you to search for and select a custom action. After selecting the custom action, the selected table rows are updated in the Change column, and the Status Viewer indicates that the rows are pending the selected action. Up to this point, no REST requests have been made and the custom actions have not been performed. Data entry validation is performed at this point on custom action field columns for the current action.

You click **Upload Changes** to perform the custom actions. For each row that you marked for action, the add-in performs the following steps:

- Creates the payload by collecting the cell values for each custom action field column and adding the value to a simple JSON object (member name/value pairs) in the payload. The entire payload body follows this example format:

```
{
  "proposedSalary": 75123,
```

```
  "ajdustmentReason": "superior job rating"
}
```

  - There is no other content in the POST request body (no action name, no array of argument values).
  - Should any values from these columns be invalid (missing when required, incorrect data type, Excel formula error) the row will be omitted from the Upload operation and marked as failed
- Prepare the request
  - REST-Framework-Version header added (version 6)
  - Content-Type header added (`application/vnd.oracle.adf.action+json`)
- Make the request
  - Send the POST (POST is the only HTTP method supported for invoking custom actions)
  - See the note below about BATCH requests.
- Process the response
  - For 200 response status, the row is marked as succeeded
  - For 400 response status, the row is marked as failed, and the response payload is parsed for Oracle RAMP REST service error content, any error details can be seen in the Status Viewer pane.
  - A 412 response status indicates that the row was modified by some other agent or user after it was downloaded into the Excel table; such a status treated as a row-level error
  - Note: cell values in action rows are not refreshed. If the custom Java method logic has altered any values in the row, such changes will not be reflected in table row until the next Download operation.

For any rows marked for Update or Create, custom action field columns' values are reset to empty as part of Upload.

Rows marked for action can be combined with Update, Create, and Delete rows in the same Upload operation. For rows marked for Update, Create, or Delete, the cell values for all custom action field columns are ignored:

- They are not included in the request payload (eg. for Update (PATCH) or Create (POST))
- Invalid values in such cells do not prevent the Upload operation on that row from proceeding

Until a row is marked for action, cell values in custom action field columns do not participate in data entry validation. Likewise, for a row with an empty Change column, any edits to cells in custom action field columns do not convert the row to a pending Update row. Edits to custom action field columns' cells are ignored in rows marked as pending Create.

Limitations, Known Issues and Other Notes

Custom actions that correspond to view object methods (as opposed to view object row methods) are not currently supported.

Custom actions are supported for existing rows in the Table layout and the child table in a Form-over-Table layout. Performing actions on the parent form row in a Form-over-Table layout is not supported.

Custom actions defined in the OpenAPI 3 service description document that have request payload schema members that match business object fields are unlikely to function properly.

Invocation of custom actions in Batch request mode is not currently supported by the Oracle RAMP REST service. As such, when an Upload operation contains one or more custom action invocations, then the entire operation will be performed on a row-by-row basis (one `PATCH/POST/DELETE` request made for each of the participating rows). For best performance of bulk Updates, Creates, and Deletes, avoid mixing custom action invocations into those bulk operations.

Oracle RAMP REST service OpenAPI 3 service descriptions do not indicate which custom action request payload fields are required. You can use the Business Object Editor to adjust the **Required** property on payload fields.

9

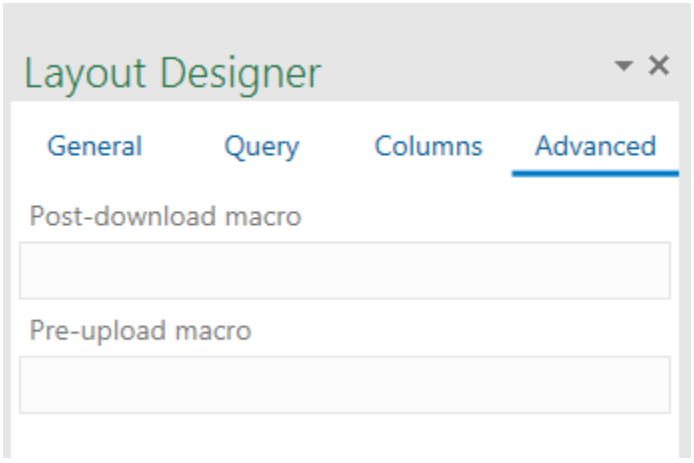
Use Macros in an Integrated Excel Workbook

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 used if the user cancels download, if the table or form is empty, or in the event of an unexpected error. The macro that you specify for the Pre-Upload Macro property is used 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:

```
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 use **Download Data** or **Upload Changes**. It can result in an unexpected exception. It is not supported.

10

Publish an Integrated Excel Workbook

Once you complete configuration of an Excel workbook, you can publish it for users to do data entry work. Publishing creates a copy of the workbook with the design tools hidden and worksheet protection turned on for each worksheet with a layout.

The recommended steps to take before you publish an Excel workbook are:

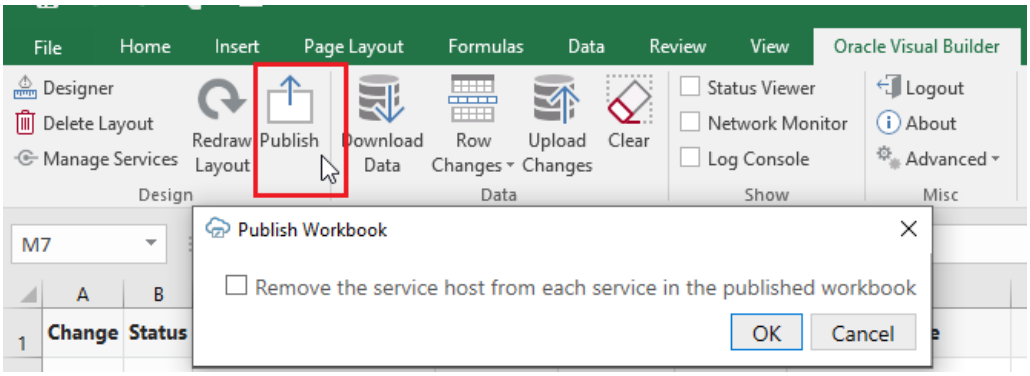
1. Complete configuration of the workbook.
2. Test the configuration thoroughly.
3. Use Excel's Inspect Workbook feature to review and remove personal information from the workbook.
4. In the Oracle Visual Builder tab, click **Clear** for each layout in the workbook.
5. Save this source version of the workbook.

> **Tip:**
>
> Publishing is optional. All the data editing features of an integrated workbook are available in both published and un-published copies of the workbook.

Once you complete configuration and testing of your workbook, 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 click the **Clear** button to remove all downloaded data from the workbook before you publish it.

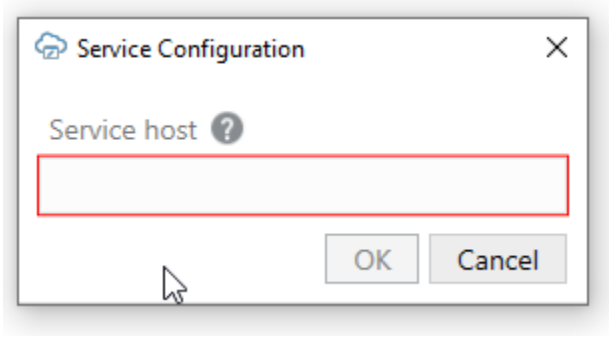
Use Excel's Inspect Workbook feature to review and remove your personal information from the workbook before you distribute it. You access the Inspect Workbook feature from Excel's File menu. Clear the checkbox beside the Hidden Worksheets entry in the Document Inspector where you choose the content to inspect and potentially remove. You must not remove hidden worksheets from the Excel workbook that you distribute. The Oracle Visual Builder Add-in uses hidden worksheets to integrate the Excel workbook and the REST service.

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



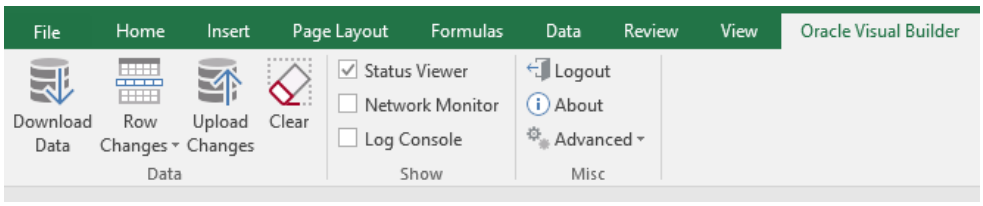
Accept the default file name and directory location values for the published workbook or select alternative values in the subsequent window, then click **Save**. A confirmation window appears with a message to say that the workbook has been successfully published to the directory location with the file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

When a user performs an action that requires access to the REST service in a published workbook where you removed the service host value, a Service Configuration window first appears that prompts the user to enter their service host value before the action can be executed. Actions that require access to the REST service include the Download Data and Upload Changes ribbon commands.



There are a number of differences with the source workbook:

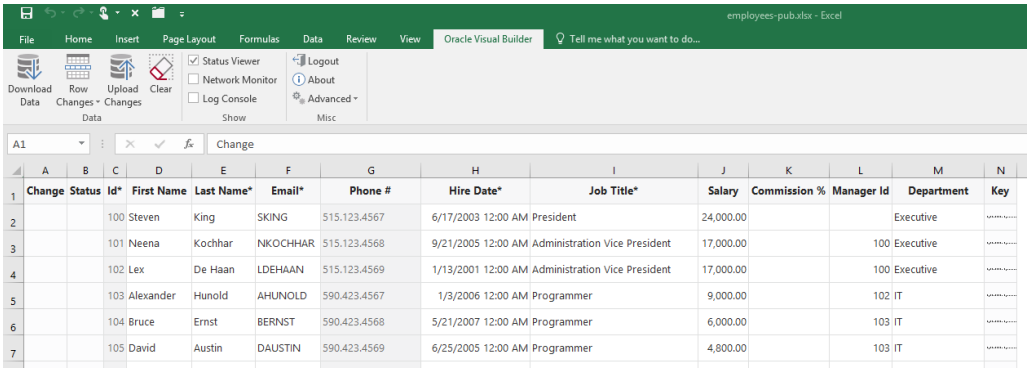
- The design tools do not appear in the Oracle Visual Builder tab of the Excel ribbon (Designer, Delete Layout, and Publish buttons), as shown in the following image.
- Excel's sheet protection is on. This mode enables true read-only behavior for cells that should be read-only. It also prevents the user from performing various other Excel actions that might disrupt the integration with the REST service.



11

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.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N
1	Change	Status	Id*	First Name	Last Name*	Email*	Phone #	Hire Date*	Job Title*	Salary	Commission %	Manager Id	Department	Key
2			100	Steven	King	SKING	515.123.4567	6/17/2003 12:00 AM	President	24,000.00			Executive	
3			101	Neena	Kochhar	NKOCHHAR	515.123.4568	9/21/2005 12:00 AM	Administration Vice President	17,000.00		100	Executive	
4			102	Lex	De Haan	LDEHAAN	515.123.4569	1/13/2001 12:00 AM	Administration Vice President	17,000.00		100	Executive	
5			103	Alexander	Hunold	AHUNOLD	590.423.4567	1/3/2006 12:00 AM	Programmer	9,000.00		102	IT	
6			104	Bruce	Ernst	BERNST	590.423.4568	5/21/2007 12:00 AM	Programmer	6,000.00		103	IT	
7			105	David	Austin	DAUSTIN	590.423.4569	6/25/2005 12:00 AM	Programmer	4,800.00		103	IT	

Users can perform the following operations in a layout if the REST service business object associated with the layout supports the operation:

- Edit existing rows
- Create new rows
- Delete existing rows

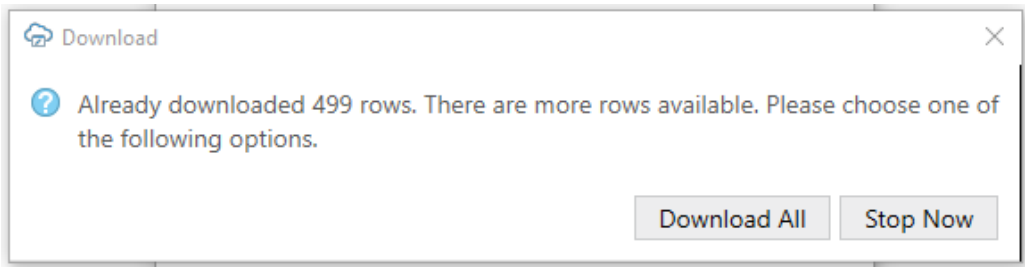
See REST Service Support for more detail about REST operations.

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.

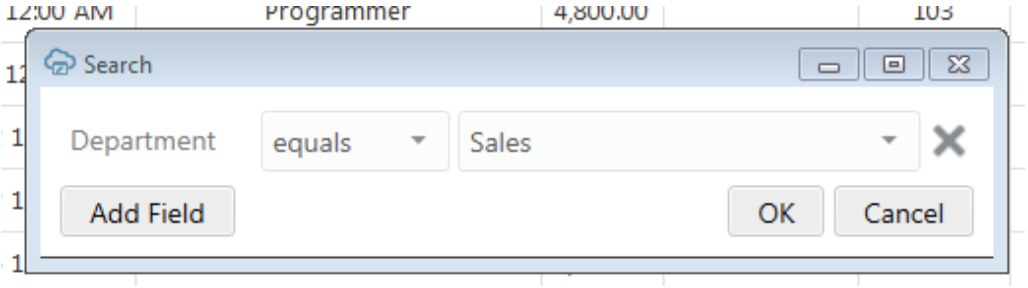
Downloading Data to the Workbook Table

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.

For REST services of type Oracle RAMP REST Service that support pagination, the add-in retrieves the first 499 rows by default and then prompts you if you want to continue to download more than 499 rows. You can view if the business object exposed by the Oracle RAMP REST Service supports pagination, and change the default download limit of 499, in the Download tab of the Business Object Editor.



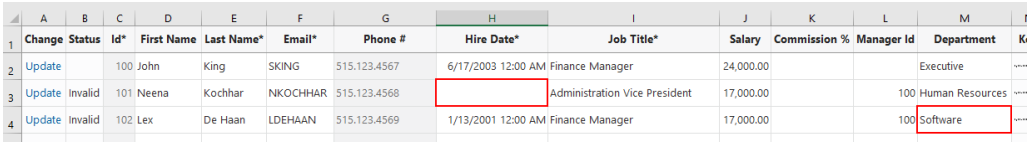
If you configured search options for download, a window or windows appear where users specify the value(s) to search on, as shown in the following example where data for the Sales department will be downloaded. Configure Search Options for Download describes how you configure search options for the workbook.



On successful connection to the REST service and completion of prompts for input values to search data (if configured), the add-in populates the table in the worksheet with data retrieved from the service.

Editing Downloaded Data in the Workbook

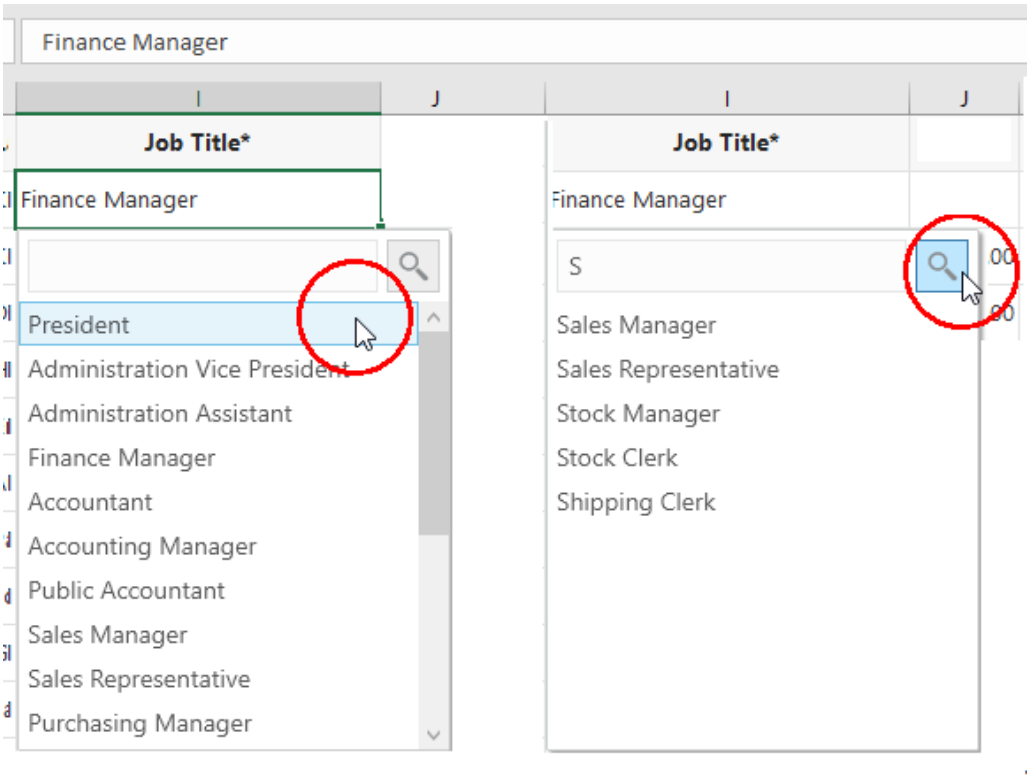
Users edit the editable cells that contain the downloaded data. The following image shows three examples where a user has edited data in the table. The Change column for the first row displays an `Update` message that indicates the user has updated this row with required and valid values. The Change 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.



Change	Status	Id*	First Name	Last Name*	Email*	Phone #	Hire Date*	Job Title*	Salary	Commission %	Manager Id	Department
Update		100	John	King	SKING	515.123.4567	6/17/2003 12:00 AM	Finance Manager	24,000.00			Executive
Update	Invalid	101	Neena	Kochhar	NKOCHHAR	515.123.4568		Administration Vice President	17,000.00		100	Human Resources
Update	Invalid	102	Lex	De Haan	LDEHAAN	515.123.4569	1/13/2001 12:00 AM	Finance Manager	17,000.00		100	Software

For columns that are associated with a field configured to use a list of values, users can select a value in a search box that displays available values. A user can select 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 search box 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 typed `S` in the search box next to the Search icon to find all the display values that start with `S` (`Sales Manager`, `Sales Representative`, and so on).



> **Note:**
>
> You can alter the search behavior of the list of values using the Business Object Field Editor described in Use Lists of Values in an Excel Workbook.

Users can update the form fields in a Form-over-Table layout if the business object for the form fields supports update. The add-in performs data entry validation before it attempts to upload a user's updates. Data entry failures need to be addressed by the user before the add-in uploads changes. For a form field configured to use a list of values, the behaviour is the same as described for the Table layout. That is, a search box appears when the user selects the form field. In a published workbook or when worksheet protection is enabled, read-only form fields as defined in the service description cannot be edited. You can make a form read-only by clearing the **Form fields are editable for update** checkbox in the Form tab of the Layout Designer.

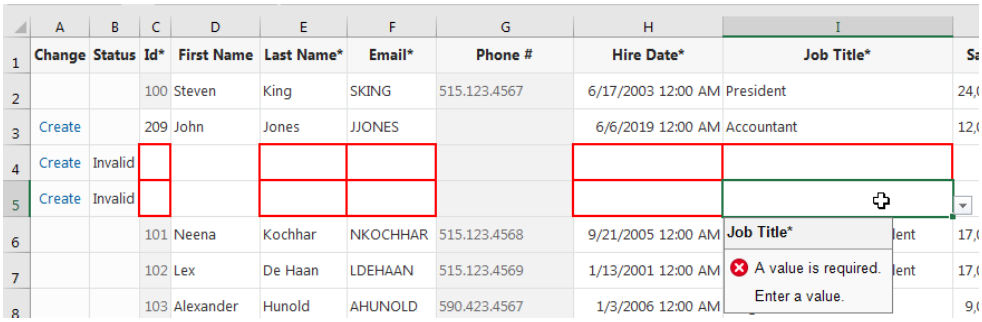
Creating New Rows to Upload to the REST Service

Users can create new rows in the table by using the **Insert Rows** option in the add-in's **Row Changes** drop-down list or by using Excel's context menu to insert a full Excel row inside the current table boundaries.

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

1. Choose the option that you want to use to create a new row:
   - To use the add-in's **Insert Rows** options, select a cell inside the current table boundaries, click the **Row Changes** drop-down list, and select **Insert Rows**.
   - To use Excel's context menu, 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.

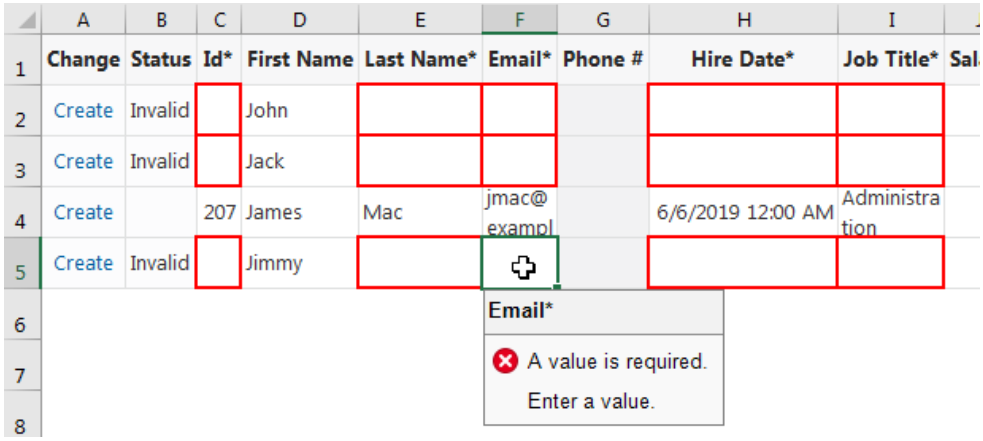
   The following example shows where three new Excel rows have been inserted inside the current table boundaries. A message (`Create`) appears on the left of each new row to indicate this row is new and has not yet been uploaded to the web application that exposes the REST service. Edit the cells in these 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.



2. You can also insert new rows by selecting 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.
3. Click the **Upload Changes** button to upload the newly-created rows to the REST service.
4. For non-published workbooks, 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 empty 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 row, the row below the current row becomes a row where data values can be entered, and so on. This method is not available in published workbooks due to worksheet protection. For published workbooks, users

can enter data in the row that appears below the table header row after they click **Clear** in the Oracle Visual Builder tab.



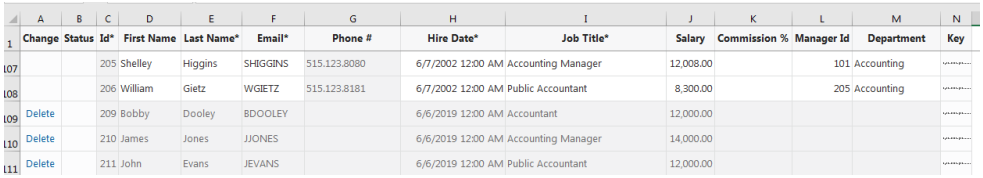
Deleting Data from the REST Service

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:

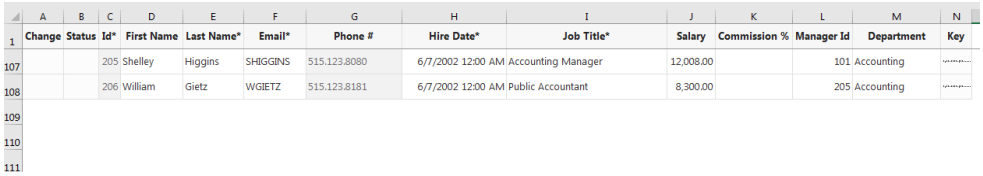
1. Select the table row that you want to delete from the REST service. If you want to select a range of table rows to mark for deletion, hold down your keyboard's Shift key and select the first and last row in the range of table rows that you want to delete.
2. Click the **Row Changes** drop-down list 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 window that appears with a message that the table has pending deletions, click **Yes** to continue.

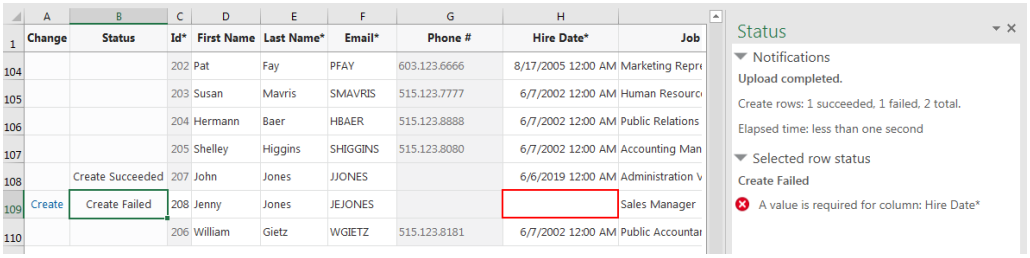
   The add-in uploads the changed rows from the table, which includes the request to delete the rows that you marked for deletion. The table in the Excel workbook refreshes so that the deleted rows no longer appear, as illustrated in the following image.



> **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. You can inspect the reason for the failure to delete in the status in Excel's task pane.



How the add-in uploads modified rows depends on the type of REST service that your workbook uses. For workbooks that use the Oracle RAMP REST Service, the add-in uploads the modified rows in batches. For other types of service, the add-in uploads modified rows one row at a time.

> **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.

View and Edit Data in a Form-over-Table Layout

Viewing and editing data in a Form-over-Table layout is similar in many ways to viewing and editing data in a Table layout. For example, users use the **Download Data** button to download the data that the Form-over-Table layout's business objects reference. There are differences, such as the fact that users need to input a search term in a Search dialog to determine the parent item in the Form-over-Table layout that the add-in downloads.

Users can also update the form fields in a Form-over-Table if the business object for the form fields supports update. The add-in performs data entry validation before it attempts to upload a user's updates. Data entry failures need to be addressed by the user before the add-in uploads changes. For a form field configured to use a list of values, the behaviour is the same as described for the Table layout. That is, a search box appears when the user selects the form field. In a published workbook or when worksheet protection is enabled, read-only form fields as defined in the service description cannot be edited. You can make a form read-only by clearing the **Form fields are editable for update** checkbox in the Form tab of the Layout Designer.

Data Consistency

When a workbook uses an Oracle RAMP REST Service that supports data consistency verification using an entity tag (Etag) mechanism, the add-in detects and reacts to the following scenario:

1. User A downloads information from a business object 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.
   ```

For information about the entity tag (ETag) mechanism, see Data Consistency Tasks in *Accessing Business Objects Using REST APIs*.

If your workbook uses other types of REST service, the last writer wins. So, for the scenario just outlined, User A's changes in Step 3 will overwrite the changes of User B in Step 2.

12

REST Service Support

You can use one or more REST services in the Excel workbooks that you configure to manage data from REST services.

Service Descriptions

To create a layout in the workbook, the REST service(s) that you use must provide a service description that supports the OpenAPI format. The service description can be a URL or a local file. For Oracle RAMP REST services, the URL typically includes a `describe`, as in `https://host/RESTApi/describe`. For an Oracle REST Data Service (ORDS), the URL may be similar to `https://host/ords/great_app/open-api-catalog/employees/`. A service description that you register from a local file on your computer will come from a file where the first element is `openapi` and the file extension is `.JSON`.

The service description that you provide can contain the definition for one or more business objects. A service description that contains multiple business objects is sometimes referred to as a catalog. You cannot provide a data URL (a URL that returns data) as the starting point to creating a layout in the workbook.

You can provide the service description document when you create a Table layout or a Form-over-Table layout by clicking the **Designer** button in the Oracle Visual Builder tab. This approach is described in the sections that describe the creation of Table and Form-over-Table layouts. Alternatively, you can provide the service description document by clicking the **Manage Services** button in the Oracle Visual Builder tab. This latter approach is described in the section that describes how to edit service descriptions.

Definitions

The content that follows in this chapter and elsewhere in this guide makes use of the following terms:

- A *business object* refers to an entity such as employees. A business object includes a collection path, an item path, and other properties.
- A *collection path* is a service path (endpoint) that can be used to fetch multiple rows of data from the business object and/or to perform operations on the collection.
- An *item path* is a service path (endpoint) that can be used to fetch, or operate on, a single row from the business object.

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 supports the following OpenAPI data types (derived from the JSON Schema Specification):

- boolean
- integer
- object
- number
- string

In addition, the add-in recognizes the optional modifier property "format", when it is applied to values of type string. The two formats recognized are "date-time" and "date". There is no support for array-valued fields.

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 use 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.

For more information, see the specifications for OpenAPI and JSON Schema:

- OpenApi: https://github.com/OAI/OpenAPI-Specification/blob/master/versions/3.0.0.md#dataTypes
- JSON Schema: https://tools.ietf.org/html/draft-wright-json-schema-00#section-4.2

The add-in supports Oracle RAMP REST Service and requires REST API framework version 6.

Excel specifications and limits can be found in the Excel specifications and limits page of Microsoft's documentation.

Object-typed Fields and Sub-fields

The add-in supports fields of type `Object`. These fields may expose sub-fields, also known as nested fields.

If, for example, you have an Employee business object with the following fields:

- First Name (type: String)
- Last Name (String)
- Address: (Object)
  - Street (String)
  - City (String)
  - State (String)
  - Zip (String)
  - GPS Coordinates (Object)
    - Latitude (Number)
    - Longitude (Number)
- Hire Date (Date)

In this example, the type of the Address field is `Object` and it contains sub-fields. Object fields should not be confused with arrays. In this example, an Employee has only one Address. The add-in does not support fields that are typed as arrays.

The add-in handles Object fields and their sub-fields in the following manner:

1. First, in the Business Object editor **Fields** tab, only the top-level fields are listed. In this example, the top-level fields are: First Name, Last Name, Address, and Hire Date. To edit the properties for sub-fields of Address, edit Address and find the Sub-fields list on the Field Editor window. The direct sub-fields for Address are Street, City, State, Zip, and GPS Coordinates. Since GPS Coordinates is of type `Object`, its field editor will show its sub-fields (Latitude, Longitude).
2. Next, when creating a Table layout from a business object's fields, the add-in promotes the sub-fields and creates columns for each (leaf) sub-field. This maintains a regular, rectangular structure for the table in the worksheet. So, the above example generates a table with these columns:
   - First Name
   - Last Name
   - Address / Street
   - Address / City
   - Address / State
   - Address / Zip
   - Address / GPS Coords / Latitude
   - Address / GPS Coords / Longitude
   - Hire Date

REST Operations

Table features are enabled as follows:

- Existing row updates are enabled if the item path has either a PUT or PATCH operation
  - For PATCH operations, the add-in includes all data values from editable cells for the changed row(s) on upload, regardless of whether the data value was edited since download.
- Create new rows is enabled if the collection path has a POST operation
- Delete existing rows is enabled if the item path has a DELETE operation

Oracle REST Data Services

The add-in supports Oracle REST Data Services (ORDS). As with other service types, you must provide an OpenAPI description of the service. ORDS with AutoREST can provide an OpenAPI service description.

For example, use `http(s)://myhost.example.com:8888/ords/hr_demo/open-api-catalog/employees/` where:

- `myhost.example.com:8888` is the host and domain portion
- `hr_demo` is the schema/application
- `employees` is the database table

For other ORDS endpoints, you need to find or create an OpenAPI service description. For information about AutoREST, see Automatic Enabling of Schema Objects for REST Access (AutoREST) in *Oracle REST Data Services Installation, Configuration, and Development Guide*.

Only basic authentication is supported when working with ORDS services.

After importing an ORDS service description, you can use the Business Object Field Editor to provide additional information about each field to improve the overall user experience. For example:

- Edit the field titles
- Designate certain fields as required
- Define lists of values. See Use Lists of Values in an Excel Workbook.

Known Issues with Oracle REST Data Services

- 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.

REST Service Support Limitations

Many different request and response schema types are possible and we cannot list all that are compatible with the add-in. If a particular structure is not listed explicitly as supported, it may not work.

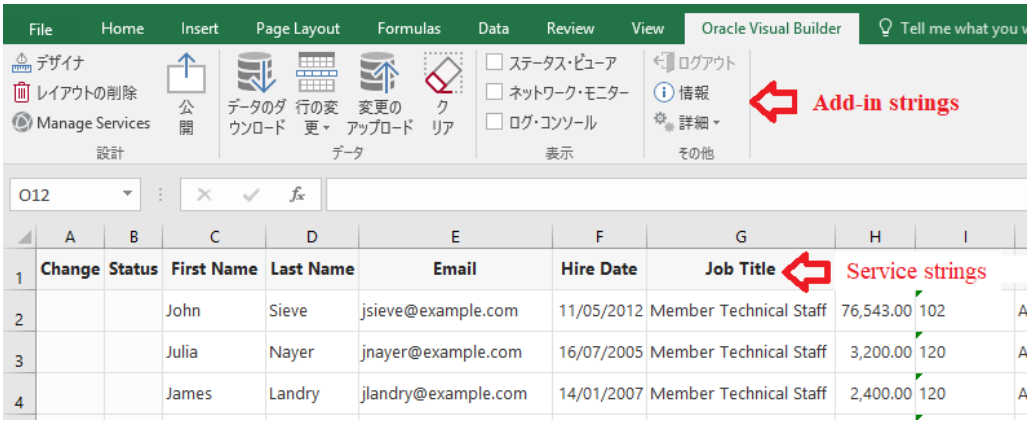
- Asymmetrical field lists
  Since download, editing, and upload all occurs in the same Excel rectangular grid, the add-in counts on having a single set of field IDs (JSON member names) for both download and upload. If the REST service uses different field IDs for the same information when completing different operations, the REST service cannot be used effectively with the add-in.
- Multi-part primary keys are supported only for Oracle RAMP REST services
- Fields with forward slash ('/') in the member name:
  - OpenApi documents contain schema properties that represented in JSON as something like `"memberName" : { . . . properties describing the field ... }`
  - When creating the business object field from the JSON member, the add-in uses the member name as the field ID.
  - Field IDs that include the / character are incompatible with the add-in, so such members will not be represented as fields in the business object.
- When providing a URL for an OpenAPI service description, `?metadataMode=minimal` is not supported.

13
Internationalization

The Oracle Visual Builder Add-in for Excel 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 windows displayed by the add-in. These strings, known as the **Add-in strings**, are owned by the add-in and are localized.
- Strings visible as column headers and field labels are known as **service strings**. These are owned by the REST service.



The add-in sends the `accept-language` header to the service on every request. The language setting specified for Excel is used for the add-in strings, service strings, and for requests to the REST service.

> **Note:**
>
> Because service strings are owned by the REST service, contact the REST service owner for any missing translations or languages.

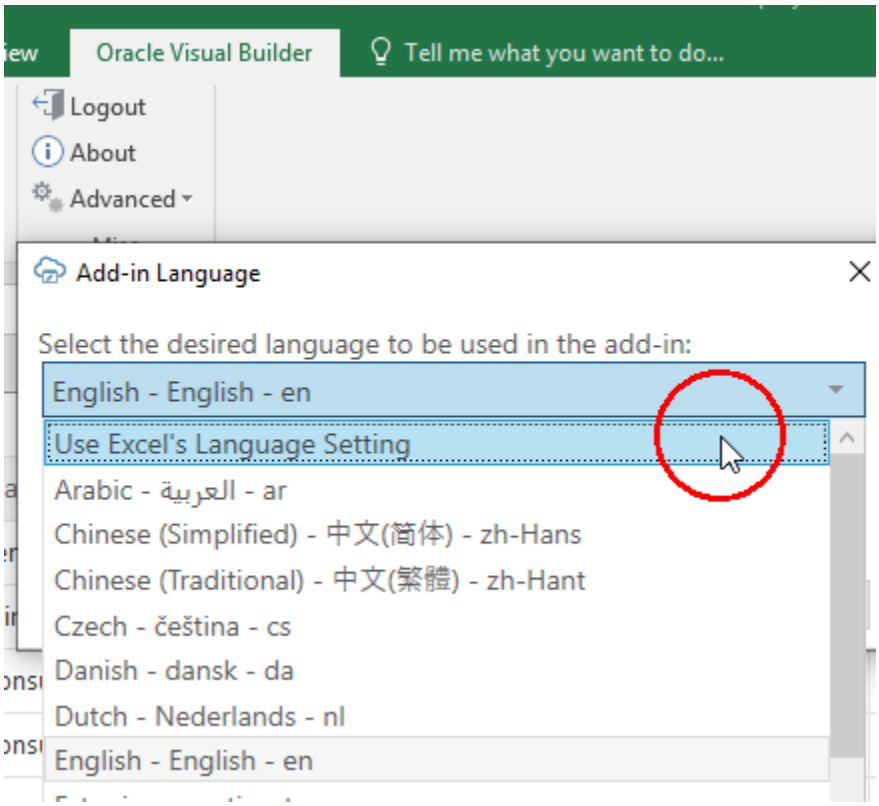
Change the Add-in Language

You can change the add-in language that the add-in uses. Do this if you want to evaluate your integrated 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 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.

   The language that you choose for the add-in language is stored in a local file in the Windows user profile. 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.



Refresh All Field Titles

Excel workbooks that are integrated with Oracle RAMP REST services can refresh field titles when a user has configured Excel to use one language (for example, French) and received another Excel workbook that was configured using another language.

When the user opens the Excel workbook configured in the other language, the add-in uses French to render the add-in strings because this is the language that is configured for the instance of Excel that opens the Excel workbook. The service strings (field titles) do not use French. They use the language that was used when the workbook was configured in Excel.

1. In Excel, click the **Oracle Visual Builder** tab.
2. Choose **Refresh Field Titles** from the Advanced drop-down.
3. In the Field Titles dialog that appears, click **Yes**.

14

Security Best Practices

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

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 computers where you install the Oracle Visual Builder Add-in for Excel.
- Turn off older obsolete security protocols such as SSL.
- Also, consider using Excel's Inspect Workbook feature to review and remove your personal information from the workbook before you distribute it. Do this for un-published workbooks. You access the Inspect Workbook feature from Excel's File menu. Clear the checkbox beside the Hidden Worksheets entry in the Document Inspector where you choose the content to inspect and potentially remove. You must not remove hidden worksheets from the Excel workbook that you distribute. The Oracle Visual Builder Add-in uses hidden worksheets to integrate the Excel workbook with the REST service.

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.

Use of the JSON Web Token (JWT) relay servlet is only available for Fusion applications, as the path to the token relay service that the add-in uses is specific to Fusion applications.

> **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.

15
Troubleshoot Excel Workbooks

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 payloads 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, right-click an entry for a REST service call in the upper table, and choose the **Save request-response details as ...** option to save information about the REST service call to a file.

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.

> **Tip:**
>
> Try the Information log level initially.

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.

16

Known Issues and Limitations

This chapter describes the issues (defects) and limitations that have already been identified for the Oracle Visual Builder Add-in for Excel

Known Limitations

- 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 or update the workbook's service description to match the changes in the service.
- Review REST Service Support to understand limitations around data type support.
- Review Use Lists of Values in an Excel Workbook to understand limitations around using lists of values.

Known Issues

- Excel workbooks that you create or modify with version 2.1 of the add-in are not compatible with prior add-in versions.
- Due to an issue with the Oracle RAMP REST service, all required fields have the **Required** property deselected. To work around this issue, use the add-in's Business Objects Field Editor to select the **Required** property where appropriate.
- When using Oracle REST Data Services earlier than version 19.2, some date-time fields are not recognized as a date-time data type due to faulty service metadata. Use the add-in's Business Object Field Editor to correct the data type.
- For an Oracle RAMP REST service that exposes entity- or view-object attributes that set the **Updateable** property to **While New**, the add-in is currently unable to detect the While New designation and fields that correspond to such attributes will be marked as `Editable on update = true`. As a workaround, use the add-in's Business Object Editor to mark such fields as `Editable on update = false`.

- Certain OpenAPI 3 document properties, such as Description, can contain Markdown text-based formatting markup. The add-in displays the Descriptions verbatim and does not interpret Markdown syntax.

17
Migration

You can migrate an Excel workbook that you integrated with a REST service using a 1.x version of the Oracle Visual Builder Add-in to use the current version 2.1.

Once you migrate an Excel workbook to the current release, you can no longer use it with the 1.x version of the add-in.

Before you install the current release of the add-in:

- Upload any pending changes from the Excel workbooks that use version 1.x of the add-in
- Click **Clear** in the Oracle Visual Builder tab for all layouts in the Excel workbook before you migrate
- Before you migrate an Excel workbook, make sure you are prepared to log in to server configured in the Excel workbook
- If you need to change the service host (the server name that the REST service uses), be sure to do so with version 1.x of the add-in. You cannot change the service host during migration using version 2.0 or later of the add-in

After you install version 2.1 of the add-in:

- During migration of a 1.x workbook, if you fail to log in to the REST service, migration fails. Close the workbook without saving changes and try again later.
- For any layouts containing data, download fresh data before making any changes and attempting to upload

18
Third Party License

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

NewtonSoft.Json, Version 12.0.2

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.

Microsoft.OpenApi, Version 1.1.4

The MIT License (MIT)

Copyright (c) Microsoft Corporation. All rights reserved.

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.

SharpYaml, Version 1.6.5

SharpYaml is a fork of YamlDotNet https://github.com/aaubry/YamlDotNet published with the following license:
