About Application Interface Wizard

Please note that Application Interface Wizard was sold both as a platform for EBS Loaders developed by More4apps and as a Development Platform, enabling customers to build their own loaders. Application Interface Wizard is no longer sold as a standalone development environment and is now in support mode only for customers with existing licenses.
Please speak with your Account Manager if you have any questions.

Welcome to 'Application Interface Wizard' by More4apps.

Application Interface Wizard allows system administrators to design and build their own templates that upload and download data. These can be assigned to EBS Responsibilities for secure deployment to the user community. It can insert and delete data from tables in addition to calling Oracle or your own API's or procedures; passing data from Microsoft Excel to Oracle EBS.

Each Loader has dynamic data entry forms and allows System Administrators to assign LOV's to any field. It also supports Oracle EBS Key and Descriptive Flexfields.

An advanced download facility allows you to create your own queries and run them directly from the spreadsheet. This enables extracts of information for update or as the basis for creating new records.

A special mode for loading Attachments into Oracle EBS is also included.

Connection Method

The Wizard connects to Oracle via the intranet using a similar web connection technology to that used in the Oracle E-Business Suite forms and WebADI. A package needs to be installed in the database and also a Java Servlet for Release 12.

The instructions for product installation, setting up the list of instances, using licence keys and adding named Users are in the Installation and Configuration Guide.

Security

This is a powerful tool so we have provided several security features to control its use.

Trial Version

The trial version of this program is fully functional and has no time-out. It does however only upload five records at a time before it stops and you need to log in again.

Support

If you encounter any problems or have any suggestions, do not hesitate to contact us at https://community.more4apps.com/s/ebs-contact-support/

Disclaimer

This product allows the User to execute transactions on any Oracle object in the database they are connected to. If you are using it on non-public tables or interfaces in an application like Oracle E-Business Suite, the data modifications you perform are NOT supported by Oracle. We are NOT responsible for any data modifications performed using this product.

Overview - Steps to Use Application Interface Wizard

Once Application Interface Wizard has been installed on your Oracle instance, the following steps outline the basic procedure for creating, maintaining and using AIW Templates for loading information into Oracle EBS.

Steps to create and maintain AIW Templates

1. Open the workbook and Login – Connect to the Oracle database.
   Refer to section: Logging In and Out
   Refer to section: Security

2. Set up the Template.
   This section contains a substantial amount of information about how to create a new template, customize the template including LOVs and defaults, set up descriptive or key flexfields, modify templates, and much more.
   Refer to section: Configure the Template

3. Secure the Template to EBS Responsibilities (recommended).
   Refer to section: Secure Template to EBS Responsibilities

Steps to loading data using the AIW Template

1. Open the Worksheet.
   Refer to section: Open the Template Worksheet

2. Create and/or Update the information.
   Refer to section: Enter Information into the Worksheet

3. Load the records into the Interface/API.
   Refer to section: Upload Records

4. Correct errors after uploading the records.
   This step is only required if there are errors.
   Refer to section: Error Handling and Validation

More4apps Provided Loaders

For information on installation of More4apps provide loaders/templates please click here.

There are also several reference sections in this document for further information:

- Debugging Uploads
- Ribbons
- Custom Query

- Getting Started
- Profile Options
- Troubleshooting
- Licence Keys, Named Users & Installation
- Release Notes

Logging In and Out

This section is common to all of our tools.

Click on this link to navigate to Common User Instructions: Logging In and Out

Security

Oracle EBS Connection Security

Database level security is not applicable as the web server logs into the database via the APPS schema. This is enormously powerful, so we have added additional security.

Users can only upload and download data for columns defined in the templates being used. These template definitions are created by your System Administrator. Users must have EBS access to the System Administration responsibility to have access to the template and download configuration options.

Named User Security

When a licence is purchased we will provide a Named User licence key. When the key is installed ONLY registered users will be allowed to use the tool.

In trial mode Named User Security is not enabled. If you request a trial key you can request this security to be enabled so you can test it.

Refer to the Installation Guide to find out more information about how to add Named Users.

Secure Template to EBS Responsibility

Templates are able to be secured by administrators to EBS responsibilities. When users log in, they will select a responsibility and only the list of associated templates will be available.

Refer to section: Secure Template to EBS Responsibilities

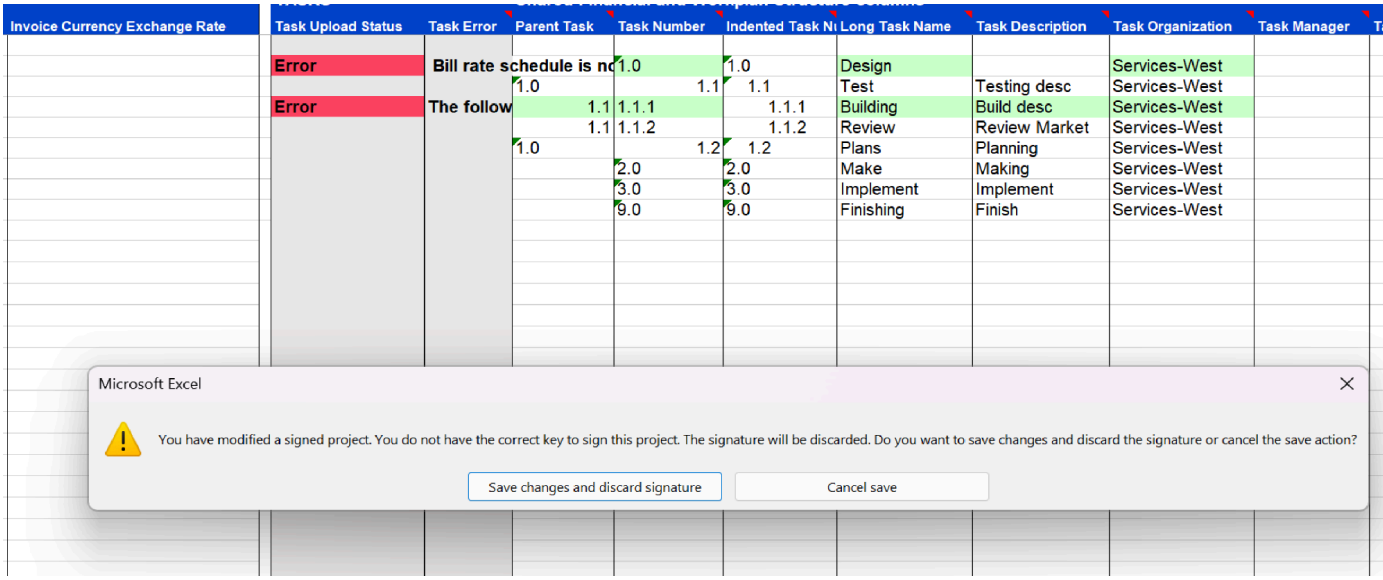
Product Package

Application Interface Wizard will not require product specific packages, therefore these will not be provided in the zip file when downloaded from the More4apps website.

Modifying a Signed Project

Digital Signatures
We now offer more security for our Wizards by having digital signatures automatically applied to all products. Each product is now more secure for you and provides better visibility for tracking issues with broken signatures.

Copying Worksheets
When copying a workbook from an older version of your Wizard, users may receive the following error due to the older version not having a signature:



Error: You have modified a signed project. You do not have the correct key to sign this project. The signature will be discarded. Do you want to save changes and discard the signature or cancel the save action.

To resolve:

1. If you do not want to track the digital signature, then simply select 'save changes and discard signature'.

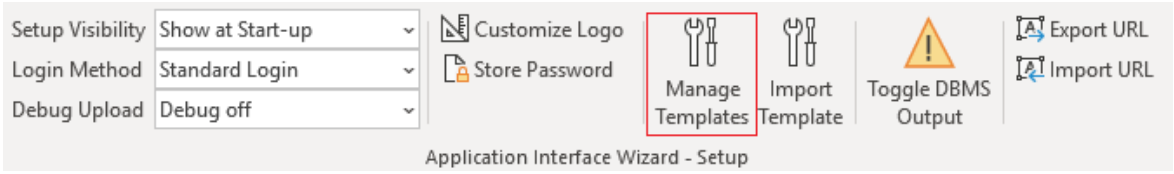
2. If you **DO** want to track the digital signature, then the user will be required to copy the original worksheet(s) into a non-xlsm workbook (e.g. xlsx). Once done this can then be copied into the new Wizard.

Configure the Template

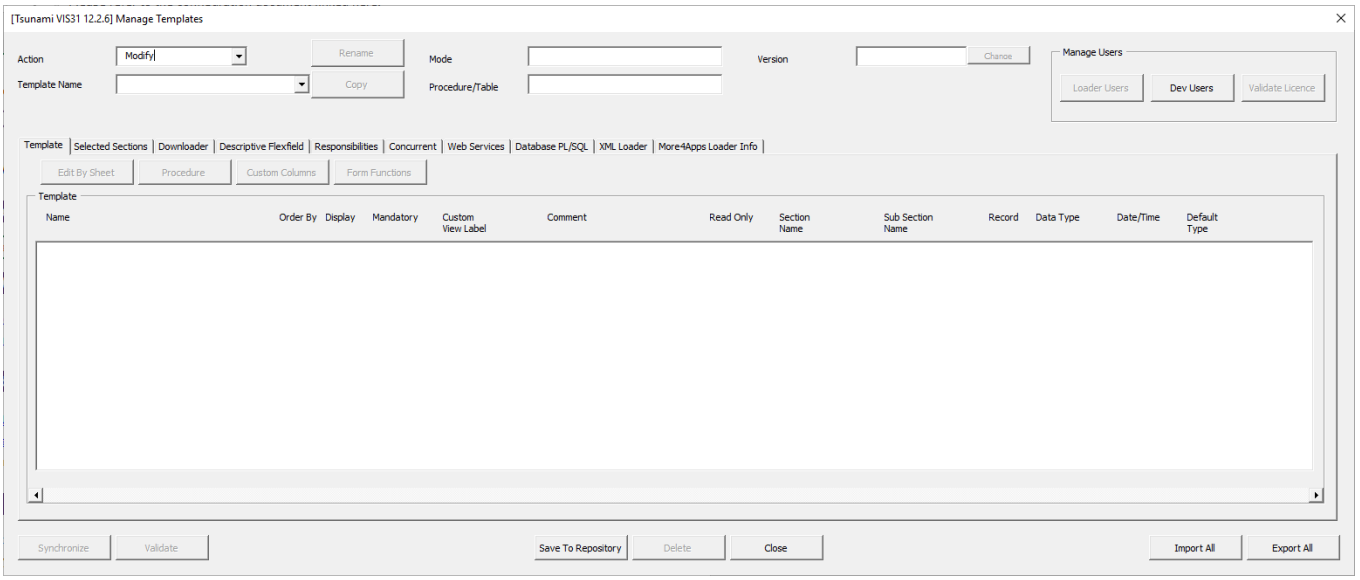
Templates provide System Administrators with the ability to create their own data loaders and save them to the database. For security purposes, they can be assigned to responsibilities, or be available across responsibilities related to the application.

The user defined templates are saved in the database and available to transfer to other instances if required.

The 'Manage Templates' button on the ribbon provides options to create new Templates, modify attributes of existing Templates, or delete obsolete Templates.



Click the 'Manage Templates' button on the ribbon and the 'Manage Templates' form will open:



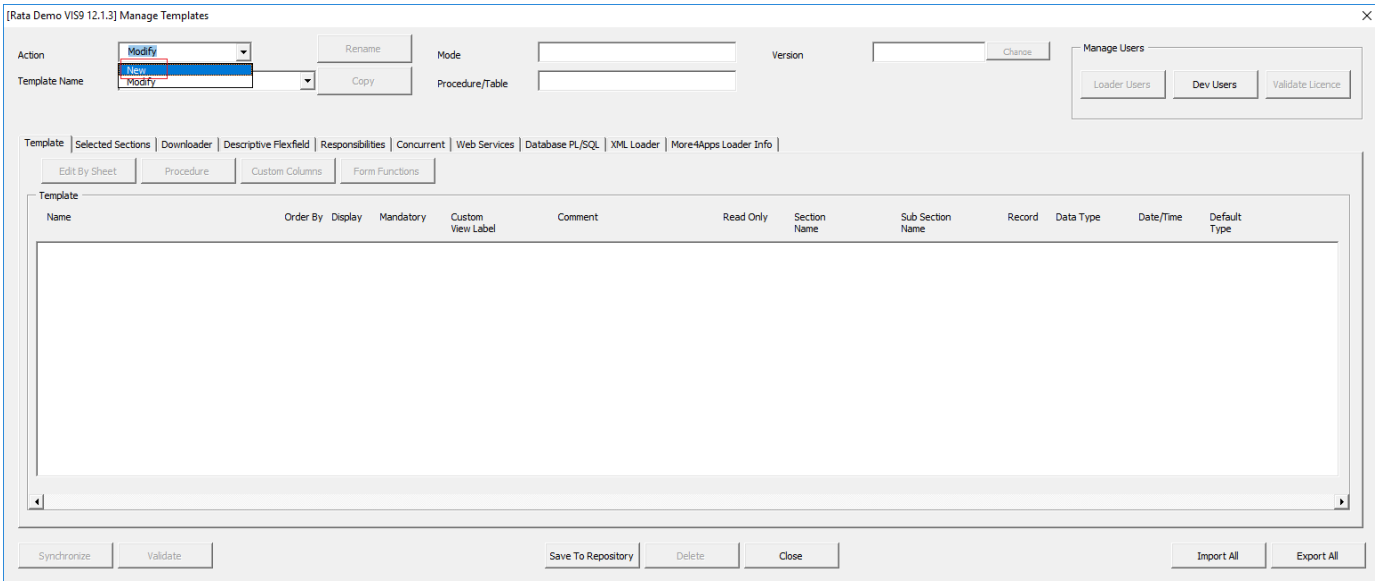
Refer to section:

Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips

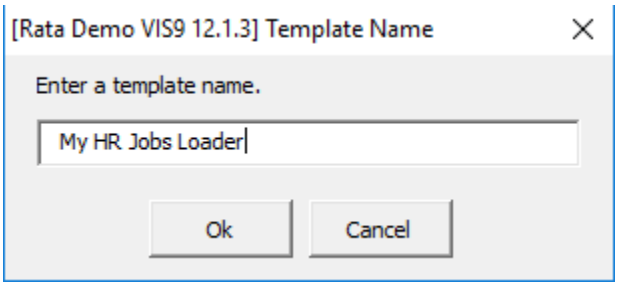
[Synchronize Template](#)
[Validate Template](#)
[Export All](#)
[Update Procedure Name](#)
[Copy Template](#)
[Delete Template](#)
[Modify Template](#)
[Attachment Templates](#)
[Importing Templates](#)
[Downloader Template](#)

Create ‘New’ Template

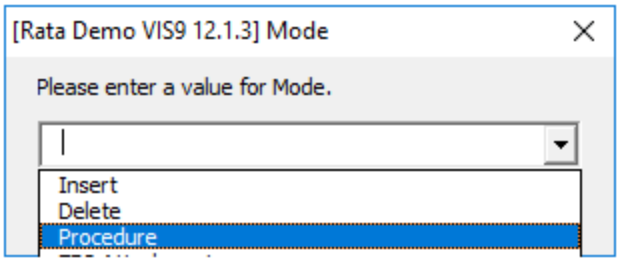
In the ‘Manage Templates’ form, select ‘New’ from the ‘Action’ list of values.



Enter a unique Template Name and click ‘Ok’.



Select a Mode associated with the Template and click ‘Ok’.

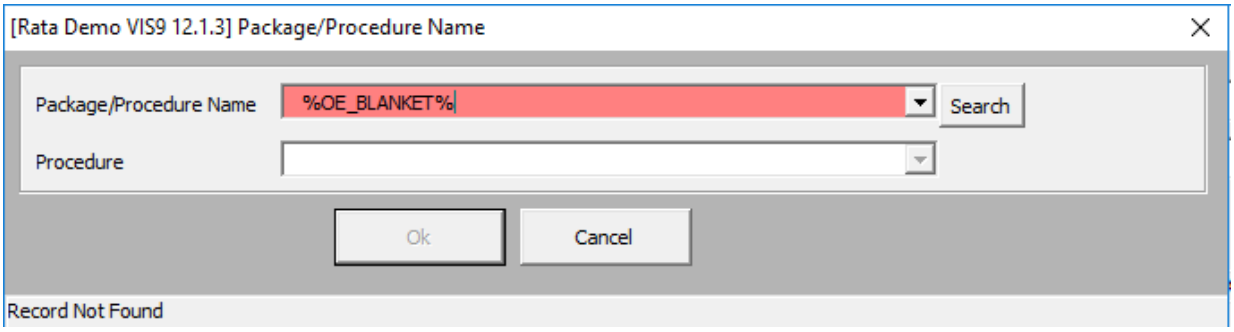


Search and select the Table name for the Insert or Delete modes.

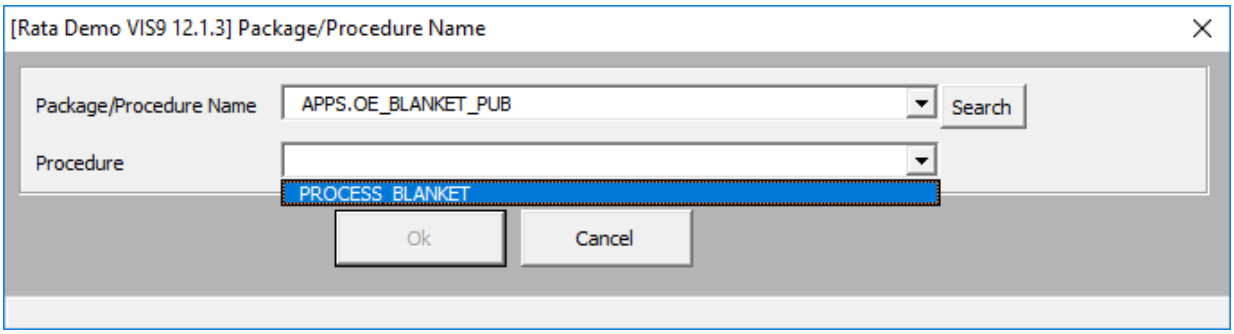
To watch a video on a basic Insert configuration, click here.

Select a Package/Procedure name for 'Procedure' mode.

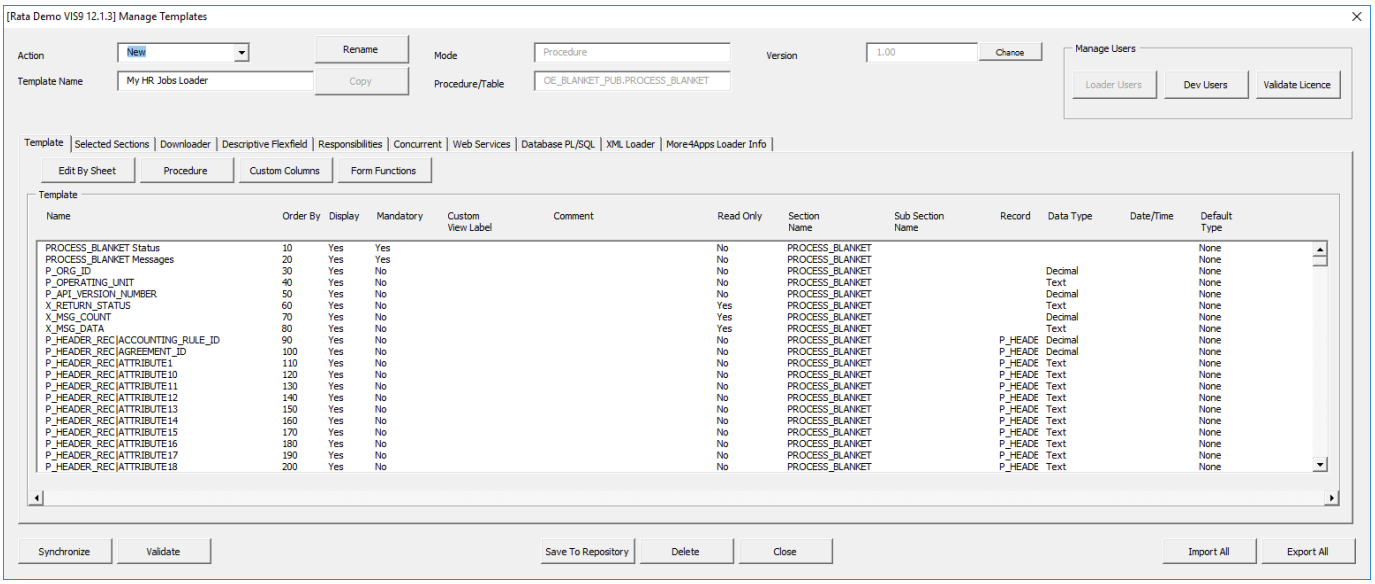
Tip: Use '%' wildcard characters to assist with your search.



In the above example, once the package has been found, select the 'Procedure' from the list of values.



Click 'Ok' to retrieve details of the selected table or procedure from the database into the Wizard template sheet.
Note - Depending on size and complexity this may take a few moments.



The form above will be displayed showing information extracted from the database which can now be used to create a user-friendly template loader.

To save the template to the database now, without changing the definition from the defaults click 'Save to Repository'.

Note - Application Interface Wizard will also automatically add its own status and messages columns to the template definition for each section it locates. These do not have 'Return' values or 'Data Types' as they are not parameters or database columns.

For example, if a section 'PROCESS_JOB' has been returned the Wizard will create the following additional mandatory columns it will then use to report statuses and messages:

PROCESS_JOB Status
PROCESS_JOB Messages

Refer to section:
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Copy Template
Delete Template

Modify Template
Attachment Templates
Importing Templates
Downloader Template

Template Versioning

The 'Version Number' field, where enabled, allows you to assign your own versioning to templates. When template changes are applied to the database the version number will be written into the XML.
Warning - The 'Version Number' field applies to the current template only and multiple templates with the same name and different version numbers cannot be used in the Wizard. This prevents users accidentally using out-of-date templates.

If you want to keep previous versions of templates for historical or review purposes, export the template definition and save it to a secure location on your network. If re-imported later the current template will be overwritten.
Refer to section: XML Loader - Export File

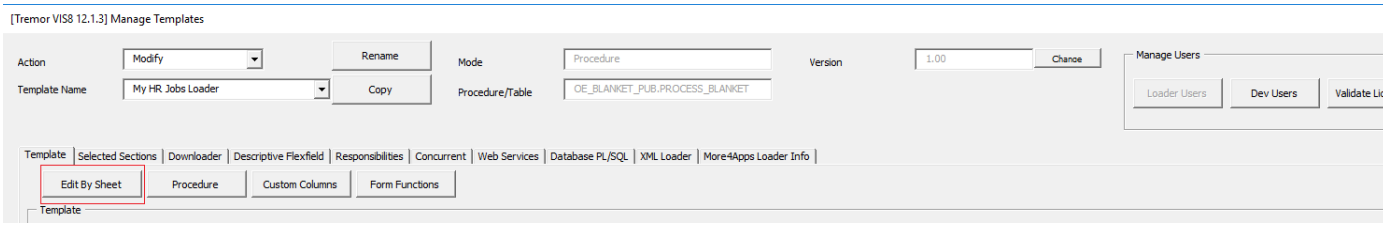
Note - Templates created by More4apps will have their own version number which cannot be updated.

Refer to section:
Create 'New' Template
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Copy Template
Delete Template
Modify Template
Attachment Templates
Importing Templates
Downloader Template

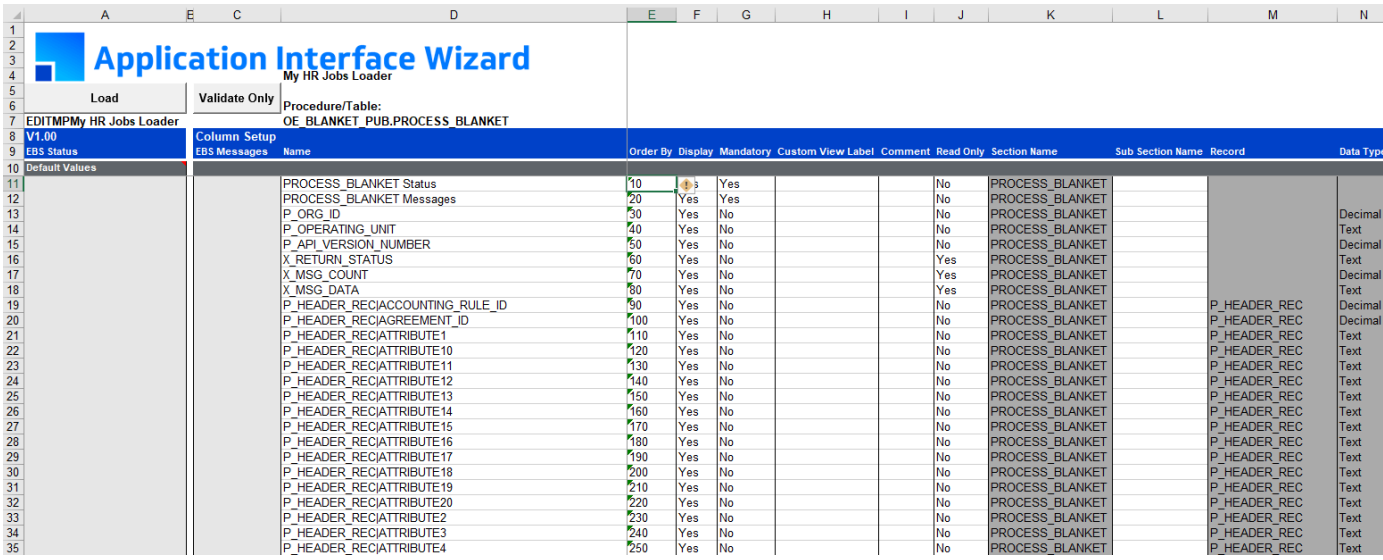
Customize the Look and Feel of the Template

To customize the look and feel of your Template, click on 'Edit By Sheet' to download the template attributes into a spreadsheet for simple and quick editing.
Note: If your Template is based on a More4apps supplied, some of the form tabs will be unavailable.

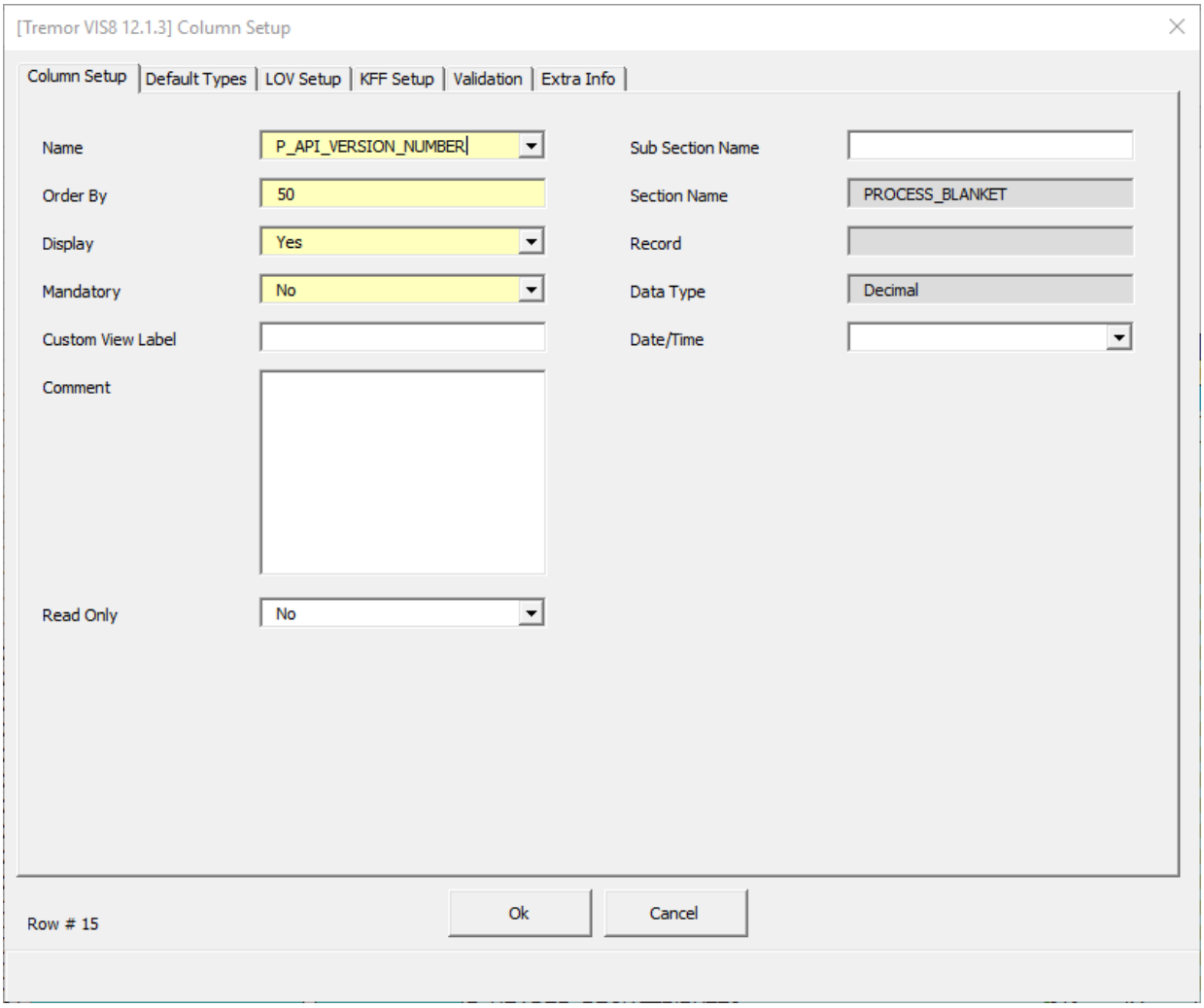
Tip: Ensure you click the 'Load' button in cell 'A1' when you have completed your updates to save them to the database. Saving the Wizard workbook does NOT save the template updates to the database.



An excel worksheet will open in the Wizard similar to the truncated image below:



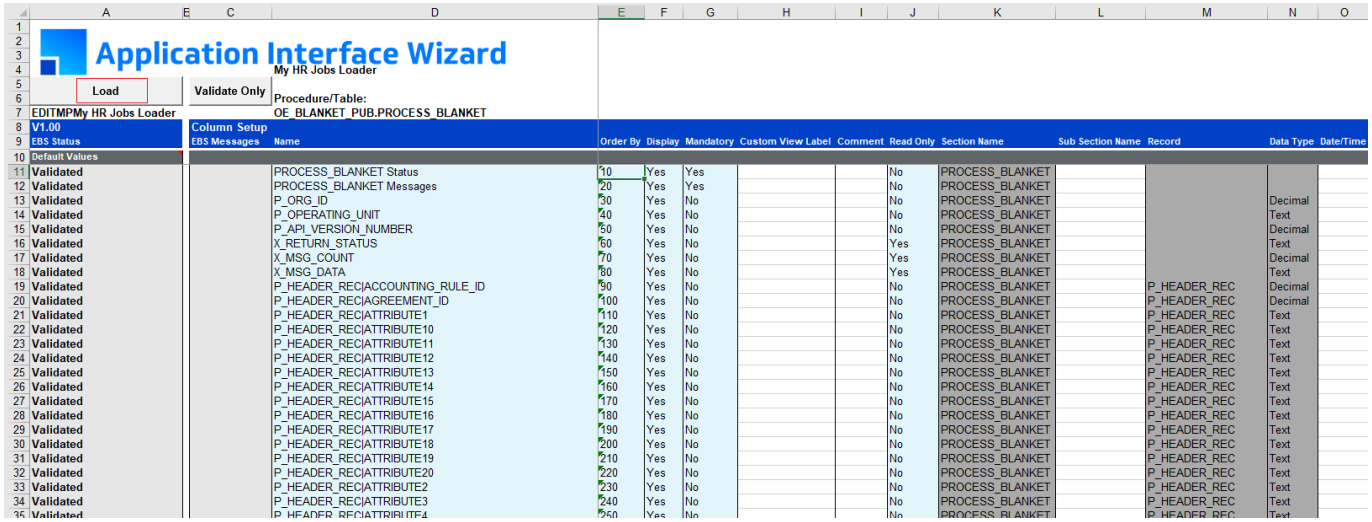
The Edit by Sheet functions in the same way a Wizard does. Each section is accessible via a form by double-clicking the record you wish to create or modify.



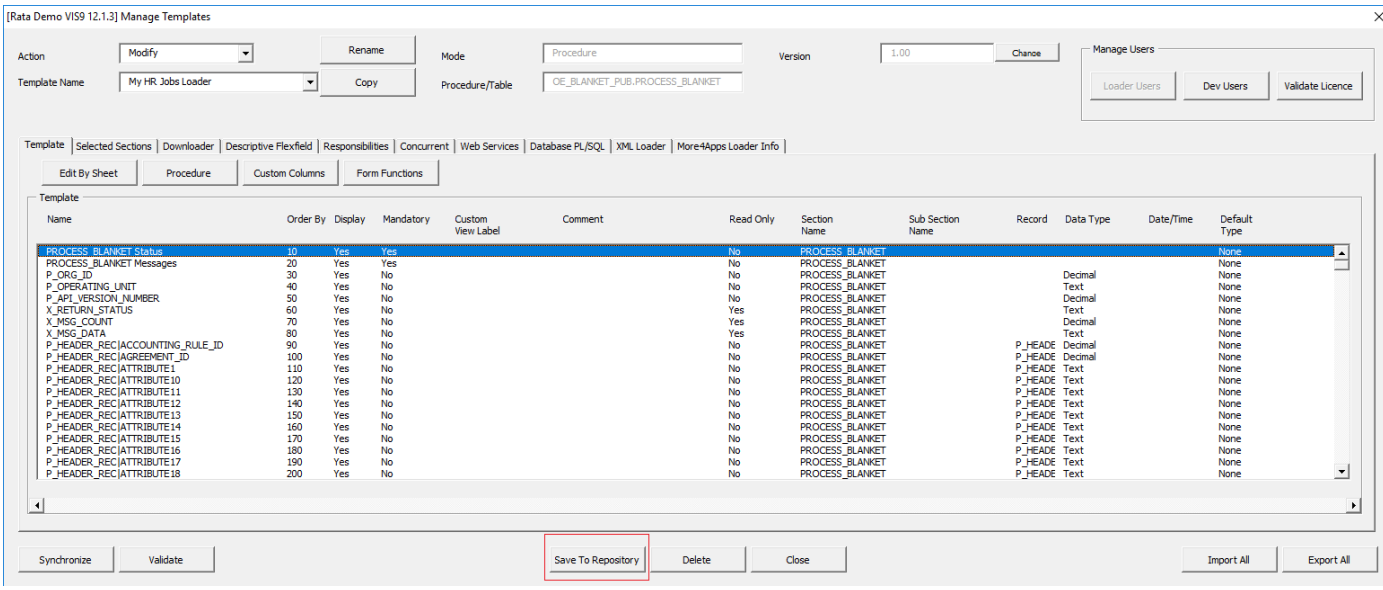
There are six tabs for setting up the columns: Column Setup, Default Types, LOV Setup, KFF Setup, Validation and Extra Info. Each one contains specific fields relating to each column.

Refer to section:
Column Definitions
Configure Defaults
Configure LOV's
Validate LOV's
Configure KFF's

When all changes have been made click 'Validate Only' on the Excel sheet to ensure all columns are correct then click 'Load' to update the Template changes and close the Excel worksheet:



Click ‘Save To Repository’ on the ‘Manage Templates’ form to save changes to the database.

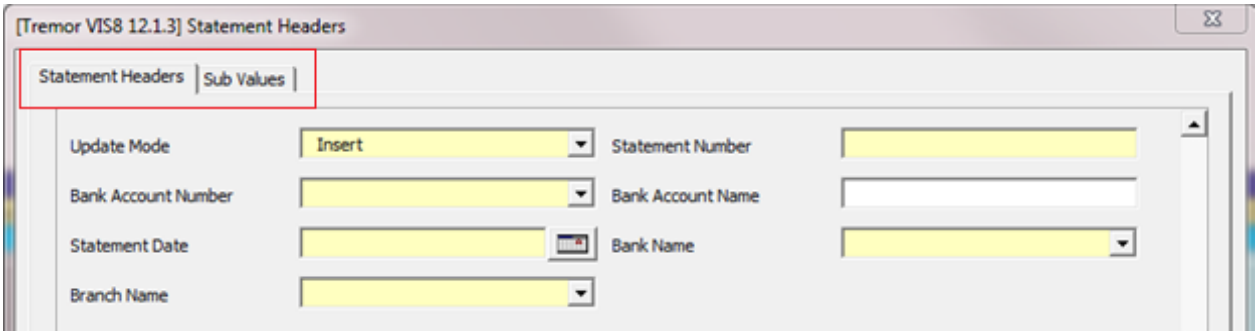


Click ‘Close’.

Refer to section:
Sub-Sectioning
Column Definitions
Configure Defaults
Configure LOV's
Validate LOV's
Configure KFF's
PL/SQL Validation

Sub-Sectioning

In the 'Edit by Sheet' worksheet, there is the functionality to position your loader fields into sub sections. This means we can have user forms for example at the header level of the loader, then within the user form for that section have a multipage for subsection fields.



These sub sections are still treated as the same single section, they are however shown in separate tabs in the user forms for further ease of use in the user forms.

In order to set up this feature we require the 'Edit By Sheet' to be opened. From here we will see that there is a column named 'Sub Section Name'.

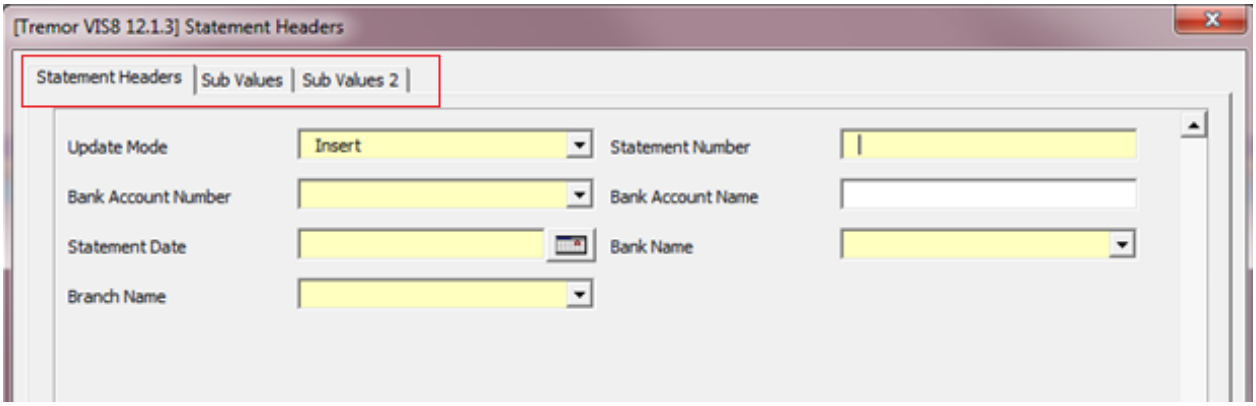
In this column we can name a new sub section for our template. For example 'Sub Values' and 'Sub values2'. The way the 'Edit By Sheet' reads these values is it will detect when a sub section name has been specified, then any rows below the value will be included within the sub section.

Column Setup EBS Messages	Name	Order By	Display	Mandatory	Custom View Label	Comment	Read Only	Section Name	Sub Section Name	Record
	UPLOAD Status	10	Yes	Yes	Upload Status		No	Banks and Branches		
	UPLOAD Messages	20	Yes	Yes	Upload Error		No	Banks and Branches		
	P_COUNTRY_CODE	30	Yes	Yes	Country Code		No	Banks and Branches		
	P_BANK	40	Yes	No	Existing Bank		No	Banks and Branches		
	CUSTOM_COLUMN|1	41	Yes	No	Existing Bank Name		Yes	Banks and Branches		CUST_COL
	CUSTOM_COLUMN|2	42	Yes	No	Existing Bank Number		Yes	Banks and Branches		CUST_COL
	P_BANK_ID	44	Yes	No	Bank ID	Download field only - do not update	No	Banks and Branches	Sub Values 1	
	P_BANK_NAME_NEW	50	Yes	No	New Bank Name		No	Banks and Branches		
	P_BANK_NUMBER_NEW	51	Yes	No	New Bank Number		No	Banks and Branches		
	P_ALTERNATE_BANK_NAME	60	Yes	No	Alternate Bank Name		No	Banks and Branches		
	P_SHORT_BANK_NAME	70	Yes	No	Short Bank Name		No	Banks and Branches		
	P_DESCRIPTION	80	Yes	No	Bank Description		No	Banks and Branches		
	P_BANK_END_DATE	90	Yes	No	Bank Inactive Date		No	Banks and Branches		
	P_BRANCH	111	Yes	No	Existing Branch		No	Banks and Branches		
	CUSTOM_COLUMN|3	112	Yes	No	Existing Branch Name		Yes	Banks and Branches		CUST_COL
	CUSTOM_COLUMN|4	113	Yes	No	Existing Branch Number		Yes	Banks and Branches		CUST_COL
	P_BRANCH_ID	114	Yes	No	Branch ID	Download field only - do not update	No	Banks and Branches	Sub Values 2	
	P_BRANCH_NAME_NEW	115	Yes	No	New Branch Name		No	Banks and Branches		
	P_BRANCH_NUMBER_NEW	116	Yes	No	New Branch Number		No	Banks and Branches		
	P_BRANCH_TYPE	120	Yes	No	Branch Type		No	Banks and Branches		
	P_BRANCH_BIC	130	Yes	No	Branch BIC		No	Banks and Branches		
	P_ALTERNATE_BRANCH_NAME	140	Yes	No	Alternate Branch Name		No	Banks and Branches		
	P_BRANCH_EFT_USER_NUM	141	Yes	No	EFT Number		No	Banks and Branches		
	P_BRANCH_DESCRIPTION	150	Yes	No	Branch Description		No	Banks and Branches		
	P_BRANCH_END_DATE	160	Yes	No	Branch Inactive Date		No	Banks and Branches		
	X_RETURN_STATUS	165	No	No			Yes	Banks and Branches		
	X_ERR_MSG	166	No	No			Yes	Banks and Branches		
	P_BRANCH_ADDRESSES_TBL Status	190	Yes	Yes	Branch Address Status		No	Branch Addresses		

Once the 'Edit By Sheet' has been loaded and a template is created we can see the new sections have been made and also the user form now has two new multipage sections.

Sub Values 1 Bank ID	New Bank Name	New Bank Number	Alternate Bank Name	Short Bank Name	Bank Description	Bank Inactive Date	Existing Branch	Existing Branch Name	Existing Branch Number

Sub Values 2 Branch ID	New Branch Name	New Branch Number	Branch Type	Branch BIC



Refer to section:
[Column Definitions]
[Configure Defaults]
[Configure LOV's]
[Validate LOV's]
[Configure KFF's]
[PL/SQL Validation]

Column Definitions

Every column/parameter retrieved from the database has the following configuration options available:

Worksheet Column Label	Purpose
Name	Table column name or procedure parameter name retrieved from the database. This value cannot be changed and is used to map your configuration back to the database object.
Order By	The order that the columns appear in the worksheet and on the data entry forms (lowest number to highest).
Display	Choose to display or hide the column from the user. Columns that are not displayed can still be used to pass values to the database object and are useful when you need to control default values. If columns are not required entering 'No' will create smaller more manageable templates for your users.
Mandatory	Indicate if the column should be mandatory. If the selected table/procedure requires a value for this field, the value will be automatically set to Yes (and coloured grey). This will appear as a yellow field in the data entry form and be validated by the Wizard as a not null column.
Custom View Label	Change your Wizard column headings to more meaningful values for your users. Spaces in column headings are acceptable.

Comment	Enter your own comments which will be displayed as Excel cell comments. These are useful when you need to describe more about the column to users.
Read Only	Choose to make a column read-only where data is displayed but not uploaded.
Section Name	This data is read only and based on the definition retrieved from the database. Sections can be renamed in the 'Selected Sections' tab in the 'Manage Templates form. Note that parent sections should only have one Header row of data for each of their multiple related child rows. Refer to Selected Sections for selecting and deselecting sections.
Record	Read only column, indicates type of column/parameter.
Data Type	Defaults to datatype of the database column/parameter. This can be changed if appropriate using the list of values on the cell.
Default Type	Enter defaults using the list of values where appropriate. Defaults are hidden from the user in the worksheet and only populate during validate or when the form is opened. ***Tip:*** *Double-click in any of the Default cells if you would like to open a form to assist with data-entry.* Select from the following options: **None –** No default **Value –** To set a constant **SQL Statement –** To dynamically select a value **Parent –** Set this field to the value of another field in the list **Oracle Function –** Call an Oracle function on upload to set the default.
Default Value	***Important:*** *If the Template definition has a 'Default Value' assigned to a column, and the user has entered a value in Row10, the Row10 value will override the Template default.* Enter the default value based on the 'Default Type' of the column: **Value Example:** FALSE *This is a constant value* **SQL Statement Example:** select sysdate from dual **Parent Example:** P_DATE_FROM *This is the 'Name' value as extracted from the database object rather than the 'Custom View label'.* **Oracle Function Example:** sys_context('FND','PER_BUSINESS_GROUP_ID')

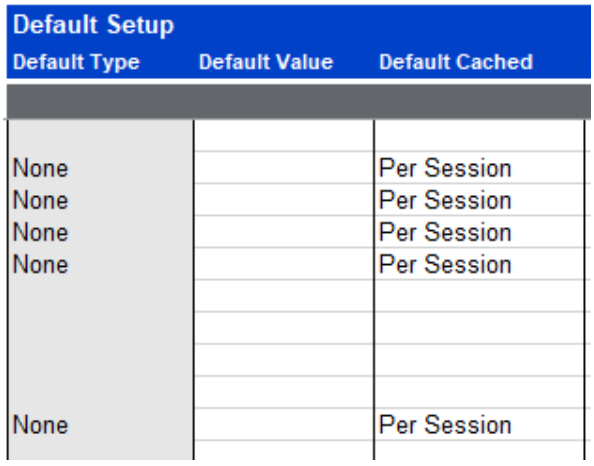
	The function syntax is always hidden from the user and the value is pasted into the sheet on upload. ***Tip:*** *Hide the column by setting Display=No if you do not want your user to be able to modify the default value that you are setting.*
Default Cached	For default values select the appropriate caching option from the list: **None** - The function/sql statement will be re-run for every row uploaded. Necessary for Oracle sequences as an example. **Per Session** - The function/sql statement will be run once per session and all other rows will be uploaded with the same value until the user logs out or the session is disconnected. This will improve performance. **Per Upload** - The function/sql statement will be run once per upload of selected records. This is useful when values like group_id's need to be different per upload, but the same for all records in the upload.
LOV Type	Enter List of Value (LOV) information where appropriate. ***Tip:*** *Double-click in any of the LOV cells if you would like to open a form to assist data-entry and sql validation.* Select from the following options: **Shortlist –** Creates a dropdown list in the form that the user can select a value from. **Longlist** – Search criteria can be entered (usually using the wildcard '%') and matching values are returned for the user to select. **Key Flexfield –** If this field is a key flexfield in Oracle EBS and you want a list of values, use this option. Refer to section: Configure KFF's
Distinct	Select 'Yes' to ensure only distinct values are displayed in the LOV. This may affect performance on long lists.
Extra LOV Fields	Enter the database column names of other fields to be displayed in the LOV, but not used during validation. Separated with a comma.
From:	Table name(s) where LOV columns are sourced from.
Where:	Table joins and other conditions where LOV columns are sourced from.
Grouping Rule	Grouping/Order By information where LOV columns are sourced from.
LOV Upload Field	The name of the field from the LOV definition to be sent to the database object. Leave blank if the same as the Display field.
LOV Display Field	The name of the field that displays in the Wizard sheet from the LOV definition.
LOV Display Width	The Wizard attempts to default this but at times it becomes too complex. If a LOV column is too wide, or narrow in the Wizard form, modify the template and change/remove this value.

LOV Cache Type	The LOV Cache Type can be Single-Row or Multi-Row, this will set whether the wizard queries each value (Single-Row) or 200 rows at a time (Multi-Row) during Upload. In order to use Multi-Row the LOV query needs to be optimized to run effectively.
Application Name	Application name for the Key Flexfield definition Refer to section: Configure KFF's
Key Flexfield Code	Key Flexfield Code.
Key Flexfield Number	This needs to resolve the defining column of the key flexfield structure (explained below). Refer to section: Configure KFF's
Return	Default as read from the database indicating the type of parameter.

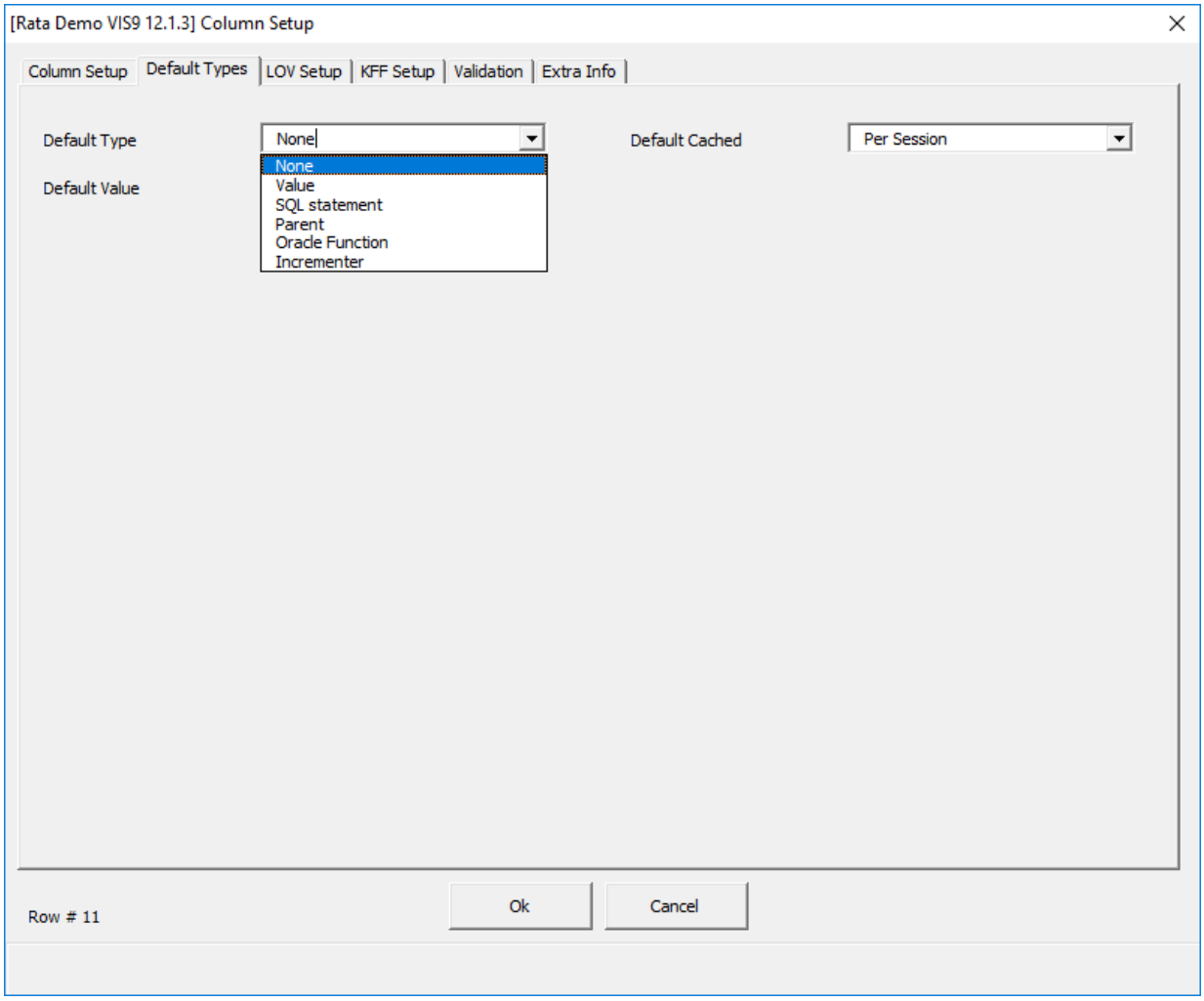
Refer to section:
Sub-Sectioning
Configure Defaults
Configure LOV's
Validate LOV's
Configure KFF's
PL/SQL Validation

Configure Defaults

Double-click on any of the 'Default' columns in the template worksheet:



This opens the 'Column Setup' form on the ‘Default Types’ tab:



For default types of 'Value', 'SQL statement' and 'Oracle Function' simply enter the relevant information into the Default Value field. Some examples are listed below:

Value Example:
FALSE
This is a constant value

SQL Statement Example:
select sysdate from dual
Select wip_job_schedule_interface_s.nextval from dual

Oracle Function Examples:
sys_context('FND','PER_BUSINESS_GROUP_ID')
The function syntax is always hidden from the user and the value is pasted into the sheet on upload.

sysdate
fnd_global.user_id

Click ‘Validate’ to check the default is set correctly.

You can use the **Parent** default type if you want to use a value from a sql definition of another field on the sheet.

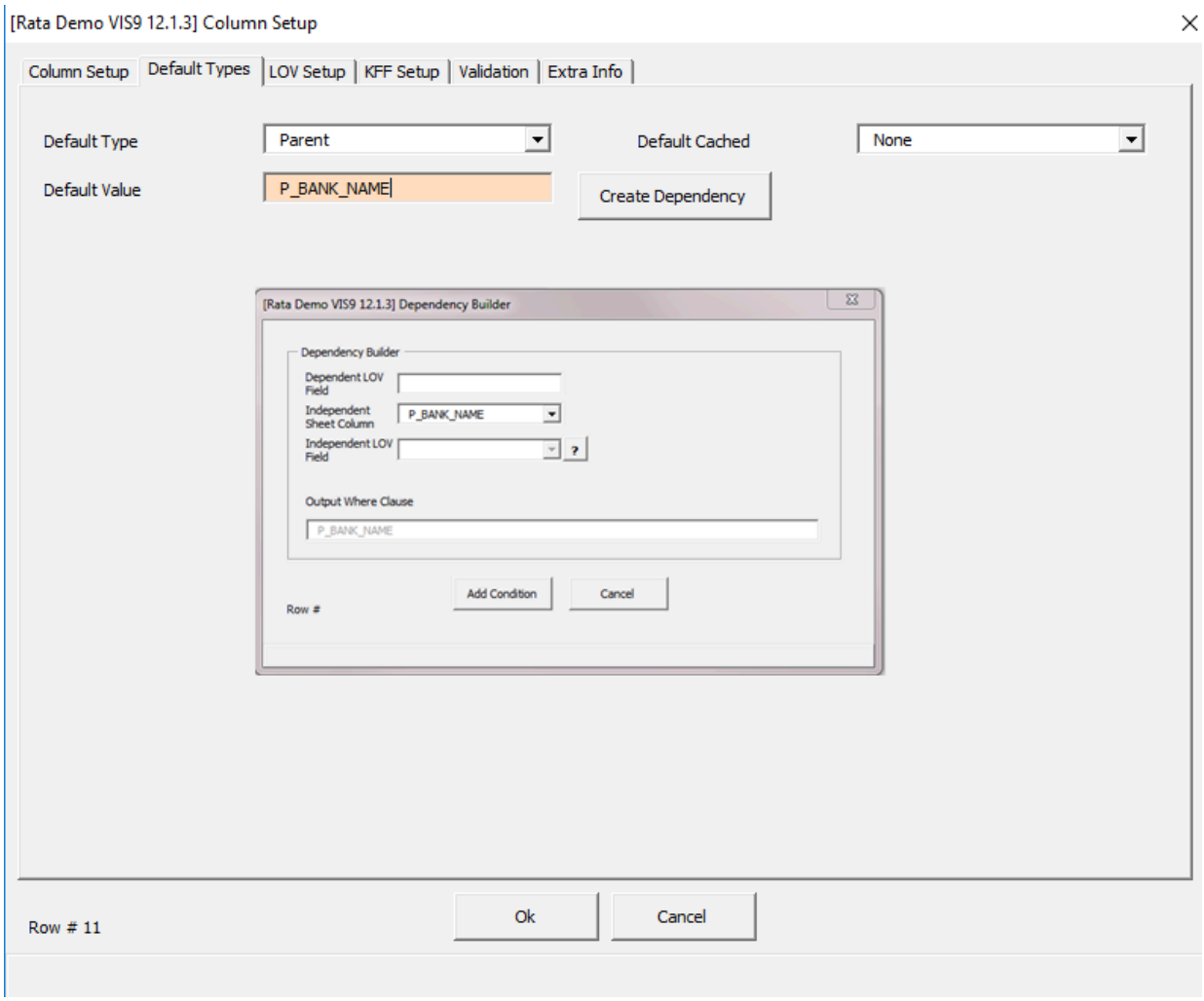
Select 'Parent' in the 'Default Type 'field.

For the Default Value select the Wizard column name that the LOV exists against.
For the Independent LOV Field, select the field from the referenced LOV that you want to use for the current field (it can be a display, upload or extra field in that definition).

In the following example the 'Bank Account Number' field has a LOV defined against it that displays the 'Bank Branch Name' value as an Extra LOV field.

When the user selects the 'Bank Account Number' from the LOV, the 'Bank Branch Name' will be defaulted provided this field is currently blank.

Clicking ‘Create Dependency’ will open up the Dependency Builder form.



Note - If the user subsequently changes the LOV selection the default will not be updated. This will be resolved in a later version of Application Interface Wizard.

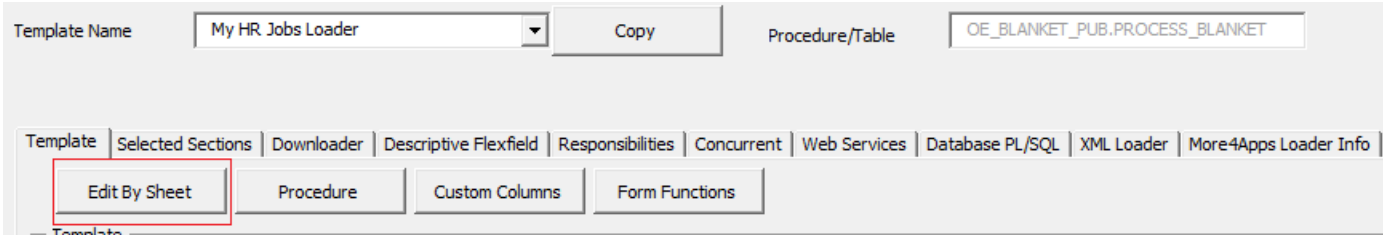
Parent Examples:
WIP.WIP_JOB_SCHEDULE_INTERFACE|ALTERNATE_ROUTING_DESIGNATOR:COMPLETION_SUBINVENTORY
WIP.WIP_JOB_SCHEDULE_INTERFACE|ALTERNATE_ROUTING_DESIGNATOR:LOCATOR

Configure LOV's

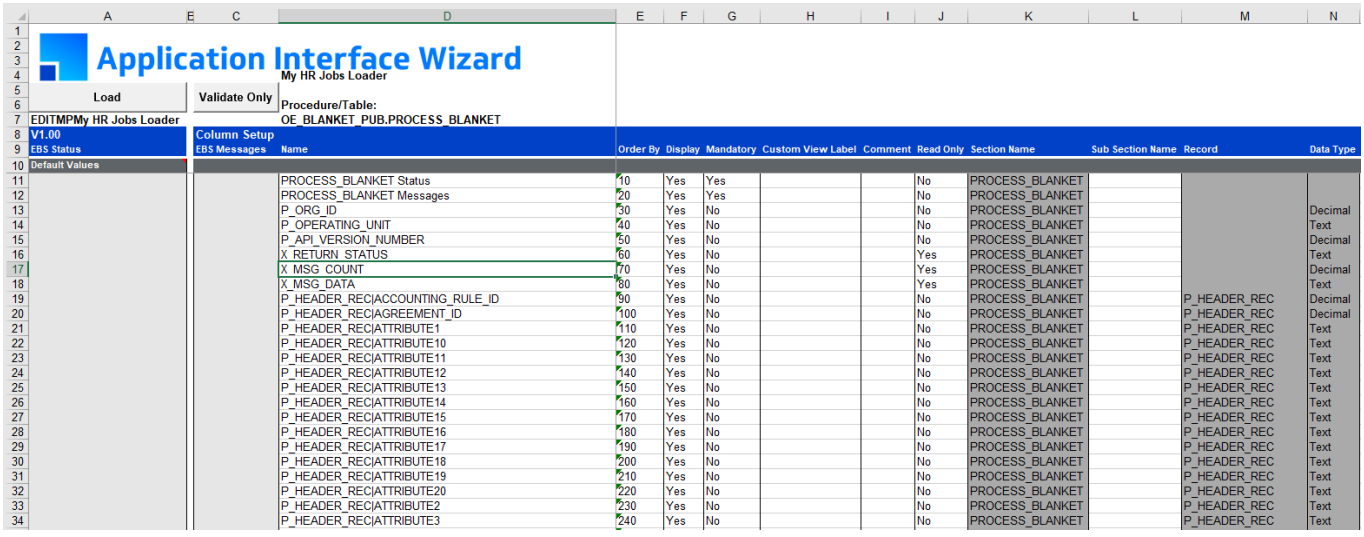
Application Interface Wizard has dynamic data entry forms. Users can double-click on any cell below Row10 and see all columns listed in the form. Administrators can add LOV's to any column to control the validation of data in the Wizard before uploading information to Oracle.

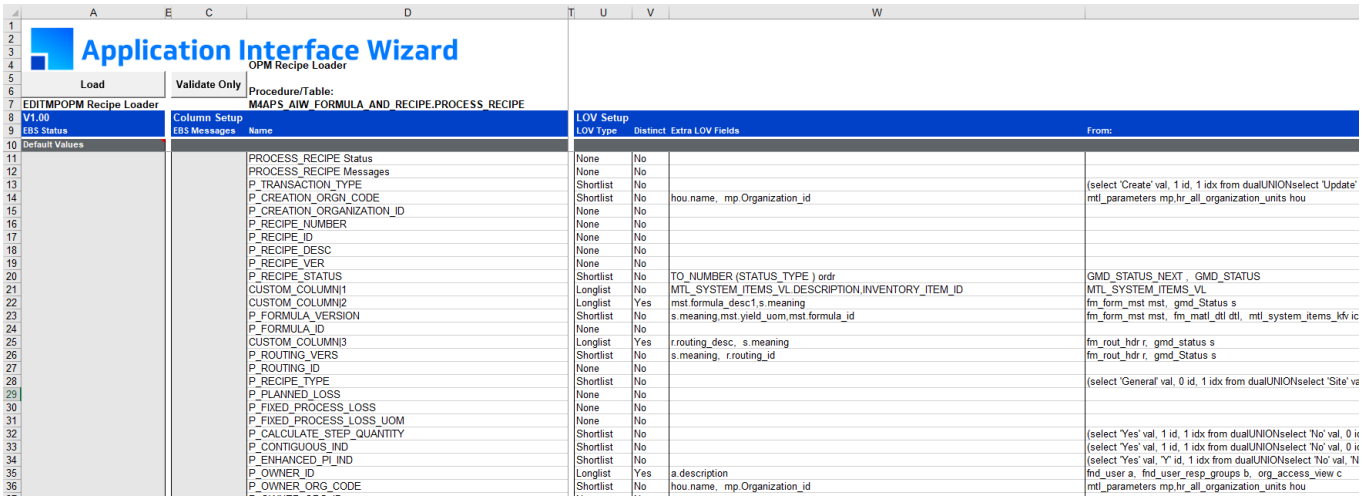
The LOV Builder is used to configure a LOV for a column. This is useful for turning a column/parameter such as Org ID, into a user-friendly column such as Org Code, where the Wizard sheet displays the value but during upload the code is sent to the database.

To create LOV's, click on 'Edit By Sheet' to download the template attributes into a spreadsheet for simpler and quicker editing.

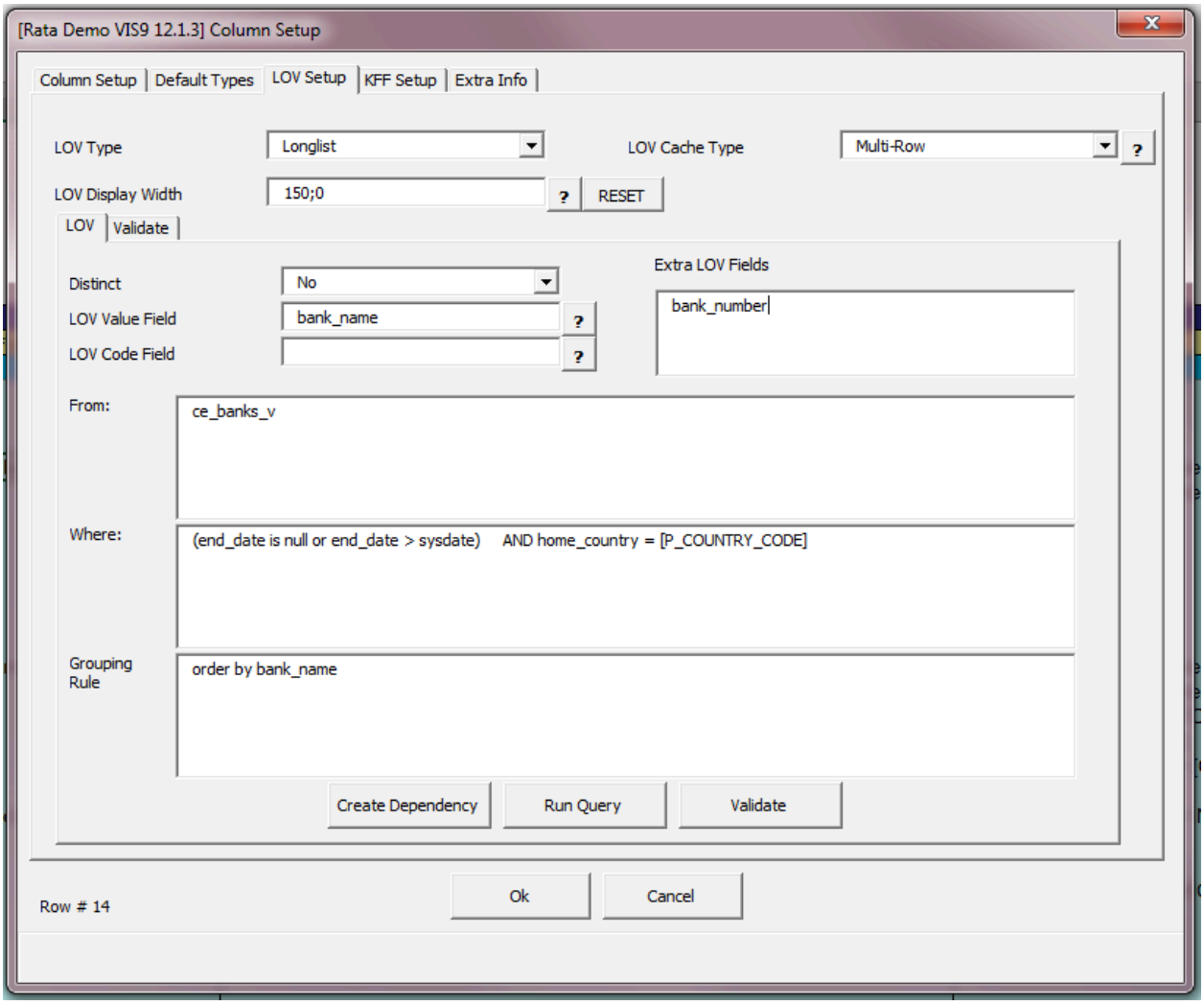


An excel worksheet will open in the Wizard like the truncated image below:



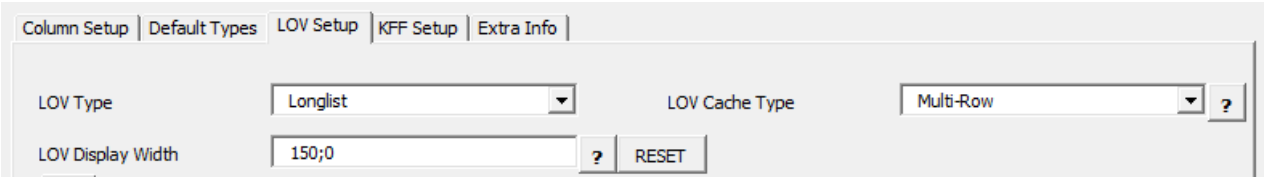


Scroll across to any LOV Setup section of the columns in this worksheet and double-click on one of these cells in the row that contains the column/parameter you want to add a LOV to.



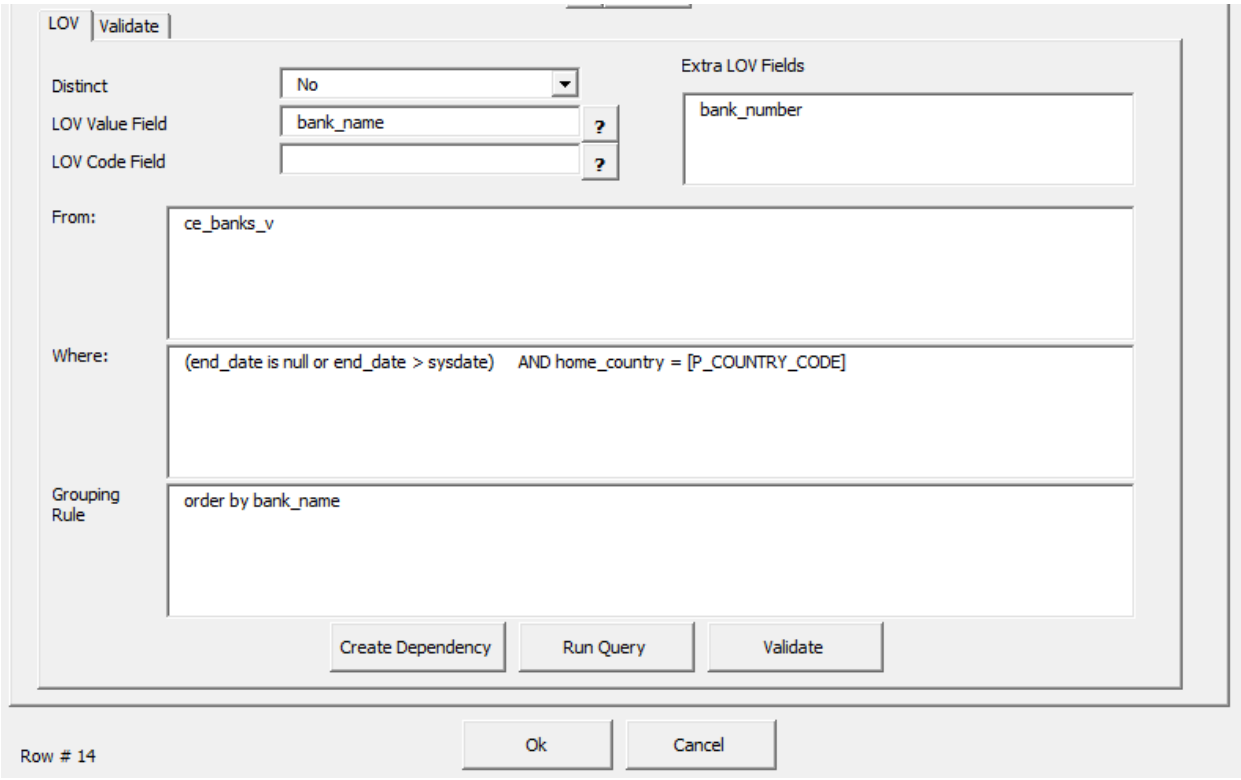
This form is intended to be used by a System Administrator. It requires a SQL statement to be split up into separate fields on the form. There is also a dependency builder under 'Create Dependency' which will output a dependency clause to be used in the LOV Setup tab to ensure the correct syntax is used.

Setup Page
The setup details the LOV Setup for the SQL Statement being run.



LOV Type	The LOV Type can be a Short, Long or Key Flex Field. This will determine how the LOV is cached and if there is a search control for the form. A Shortlist will have a static, small list of values. A longlist will have a search control, validation will be slower. Free Type allows us to create a LOV where you can either select a value from the list or you can free type into the LOV field. ***Note – On upload the parameter that will be uploaded for the list of values will be the value shown on the sheet as it will read as a new LOV value.*** This functionality is recommended to be used for simple fields where for example we have a list of values for a description, however we want to assign a different description to a record that does not yet exist in the list of values.
LOV Display Width	The LOV Column width is automatically determined when parsing SQL if it is blank. You can then modify the widths if the control drop-down list columns are too small or unreadable.
LOV Cache Type	The LOV Cache Type can be Single-Row or Multi-Row, this will set whether the Wizard queries each value (Single-Row) or 200 rows at a time (Multi-Row) during Upload. In order to use Multi-Row the LOV query needs to be optimized to run effectively.

LOV Page
The LOV page is used to set up the SQL statement. The select clause is made up of the LOV Value Field, LOV Code Field and the Extra LOV Fields.



Select Section

Distinct	Tick to ensure distinct values are displayed in the LOV where this can't be controlled by the SQL statement.
LOV Value Field	This is the value displayed and validated in the Wizard sheet.
LOV Code Field	This value will be uploaded into the parameter/column.
Extra LOV Fields	These are display only data that appear in the LOV when opened using the user data entry forms.

From Section

The FROM field is where all the tables that can be added into the 'from' clause. Multiple tables are separated by commas and can also be carriage returned for easier readability. Inline SQL statements can also be added in this section.

Where Section

The WHERE field is where any joins and filtering criteria can be added. Dependencies will also be automatically placed into the field when the 'Add Condition' button is clicked on the Setup page in the dependency builder.

Grouping Rule/Order By Section

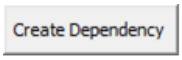
The GROUPING RULE/ORDER BY field is where you enter your additional grouping and ordering SQL.

Important - If you prefer to leave redundant code in your LOV SQL it's better practice to comment the code out using the '/* … */ syntax rather than using the double hyphens '--'at the beginning of the code.

While the LOV may still successfully be validated with the double hyphens you may experience unexpected errors when opening data entry forms.

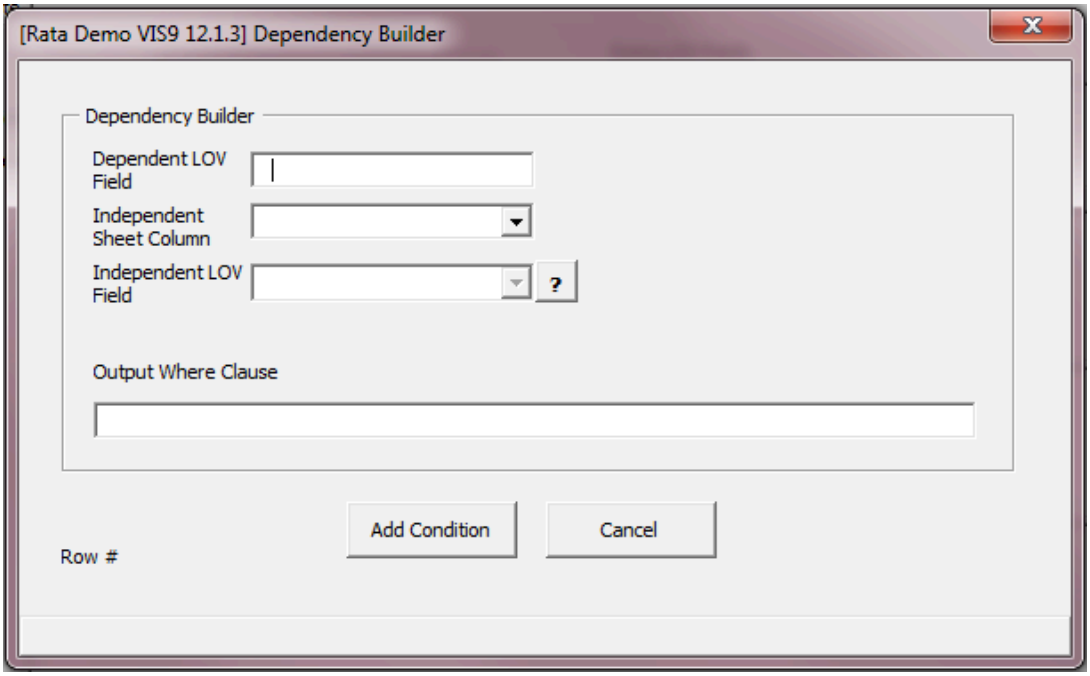
Dependency Builder
The dependency builder is used to generate dependency clauses using values from other LOV's in a format that the wizard can translate. This provides a guided form to ensure the syntax where SQL conditions that use Wizard columns are correct.

Note - Any column available in a LOV can be used to create dependency. This means the 'LOV Display Field', 'LOV Upload Field' and any of the 'Extra LOV Fields' values can be used.
In order for dependency to work as expected, ensure the dependent column has an order by value less than the current LOV column to ensure it is evaluated first.

Click the 

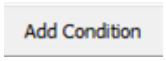
button in the LOV Page to open the Dependency Builder form.

Important - The 'Create Dependency' feature will ensure the correct syntax is used. If you are experiencing any issues with your LOV's, particularly if you have manually entered them into the LOV definition, it's often better to remove them and use this feature to recreate them.



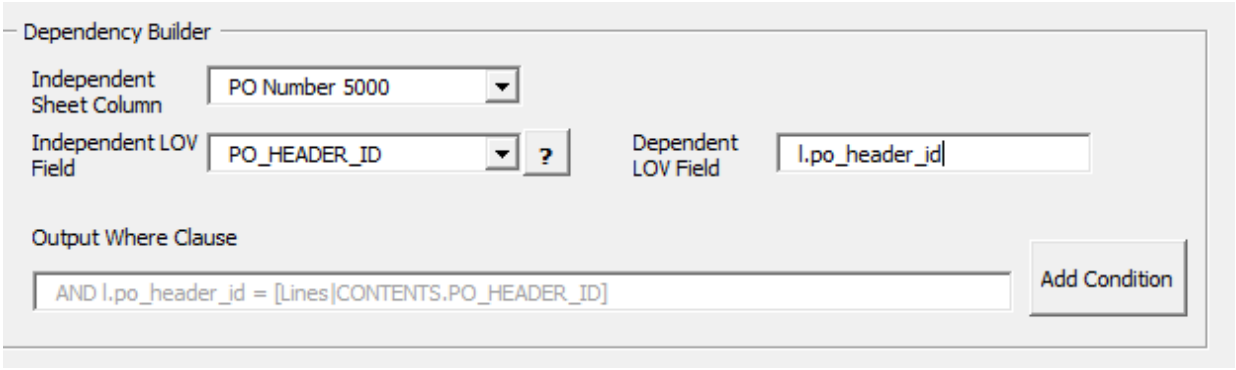
Dependent LOV Field	The dependent LOV Field is a SQL field from the current LOV definition.
Independent Sheet Column	The independent sheet column is a column on the TEMPLATE. This is the parent template column that the dependency value will be derived from. Use the LOV to select a Wizard column.
Independent LOV Field	The independent LOV field is a field from the parent template columns LOV. If the parent column has a LOV, this field will be made available to select from. If this value is blank, it will use the sheet value to derive the dependency.
Output Where Clause	Read Only. This is the clause that will be added to the WHERE clause in the LOV tab. Clicking the "Add Condition" button will add the clause to the WHERE field. The clause can then be moved/edited if necessary.

As values are selected the 'Output Where Clause' dynamically updates.

Click the 

button to write the condition into the WHERE section.

Independent LOV Field example:



Note - Once you are familiar with the syntax the Wizard expects for dependencies you can simply type these directly into the WHERE section as needed.

Tip - The syntax below refers to a value that is already displayed in the worksheet at the time this LOV is invoked:
[APPS.M4APS_TIMECARD_LINES|PROJECT_NAME]

Tip - The syntax below refers to a value that exists in a LOV column that isn't displayed in the worksheet:
[APPS.M4APS_TIMECARD_LINES|PROJECT_NAME:PROJECT_ID]

In the above example PROJECT_ID is the 'LOV Code Field' value from the PROJECT_NAME field LOV.

Various LOV Syntax Examples

Select:

LOV Display Field	Meaning
LOV Upload Field	Code

From:

(SELECT meaning, 'CREATE' code, 1 orderMe FROM fnd_lookup_values_vl WHERE lookup_type = 'EAM_WORK_ORDER_PM_ACTION' AND lookup_code = '1' UNION SELECT meaning, 'UPDATE' code, 2 FROM fnd_lookup_values_vl WHERE lookup_type = 'AT_SQL_ACTION_TYPE' AND lookup_code = 'U' UNION SELECT meaning, 'DELETE' code, 3 FROM fnd_lookup_values_vl WHERE lookup_type = 'AT_SQL_ACTION_TYPE' AND lookup_code = 'D' UNION SELECT 'Update Grades' meaning, 'UPDATE GRADES' code, 4 FROM DUAL)

Where:

Select:

LOV Display Field	displayed_name
LOV Upload Field	job_group_id

From:

per_job_groups

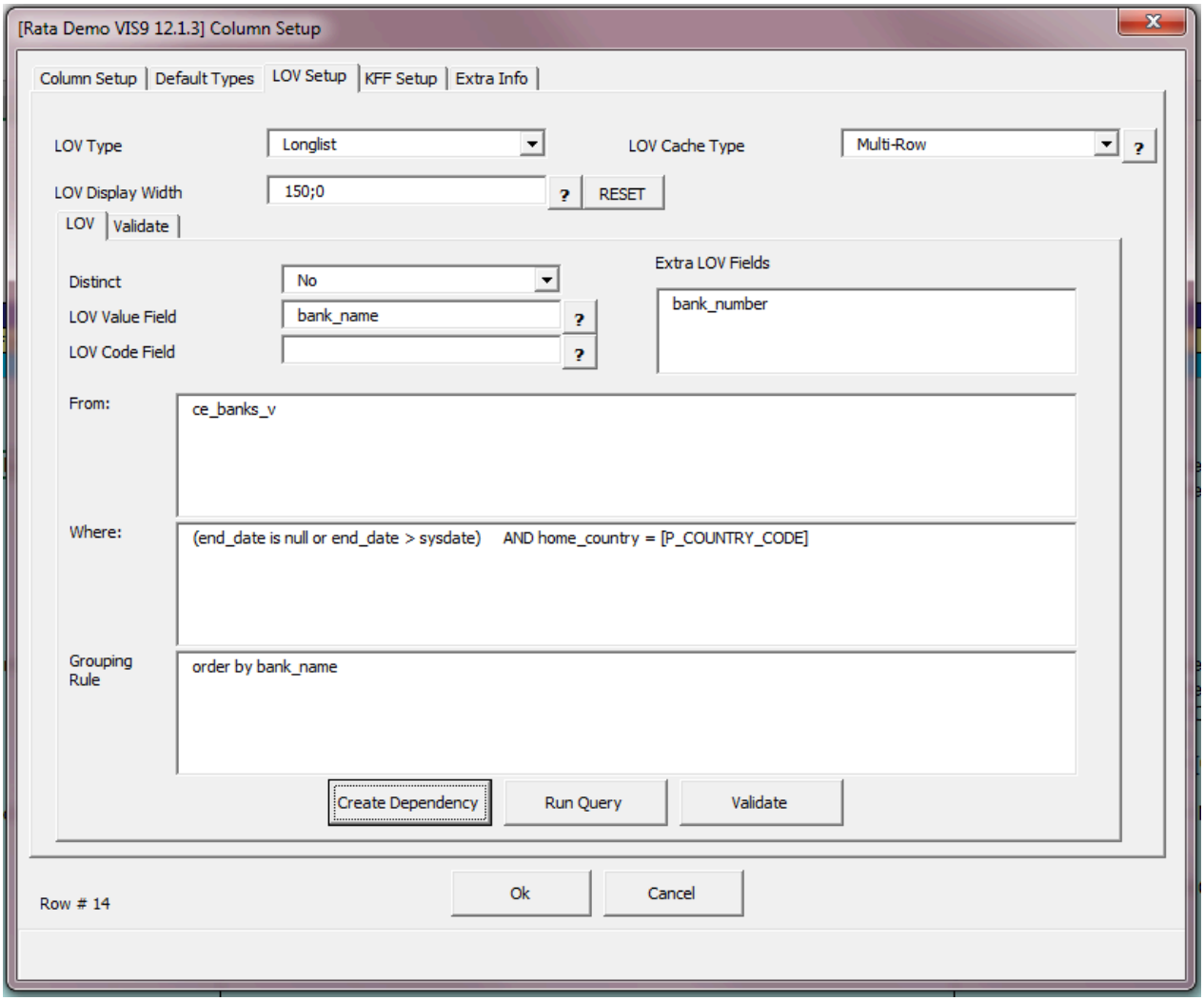
Where:

business_group_id = sys_context('FND','PER_BUSINESS_GROUP_ID')

Validate LOV's

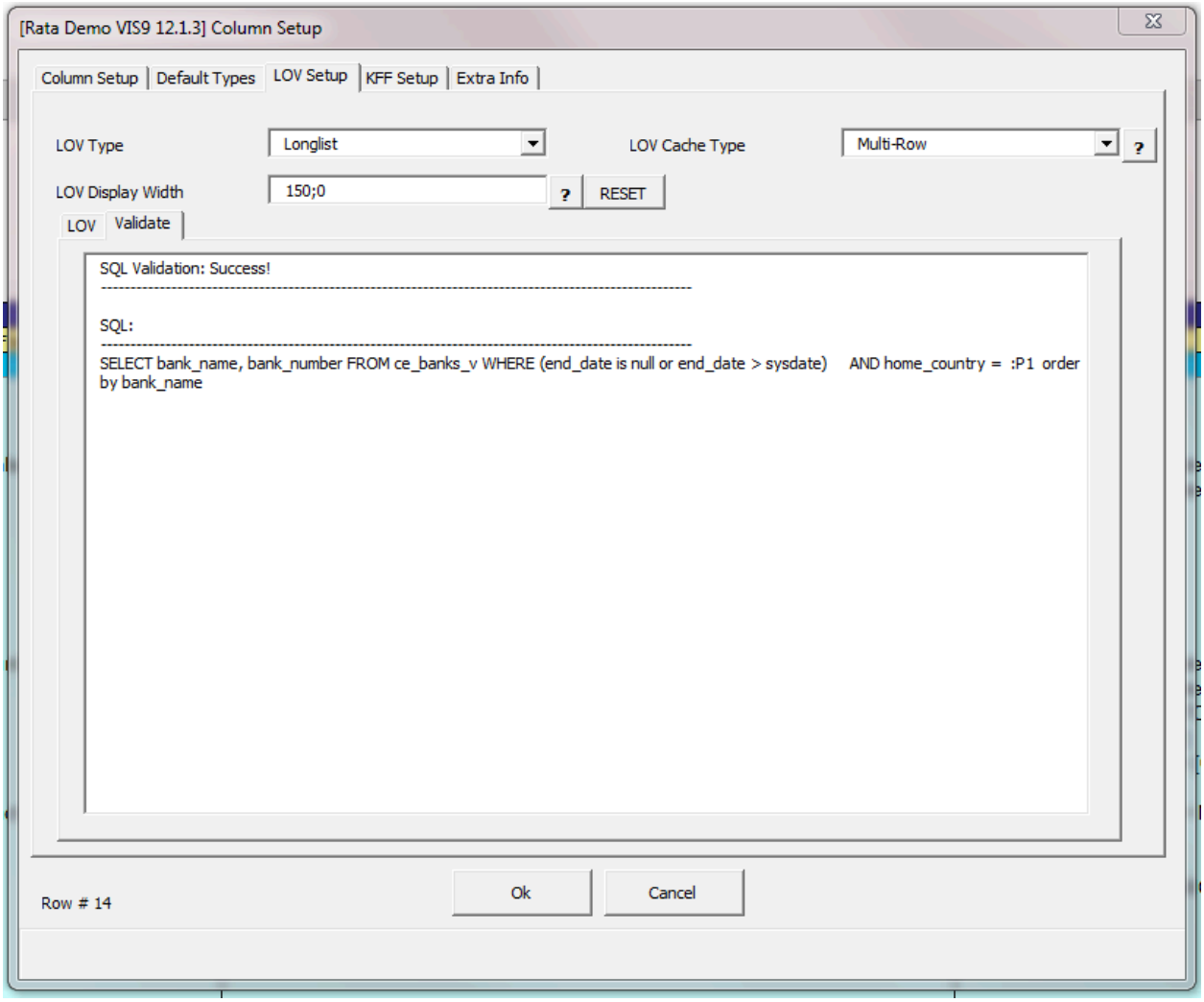
As each LOV is configured it should be validated to ensure the construction is correct.

Click the 'Validate' button to initiate this check:

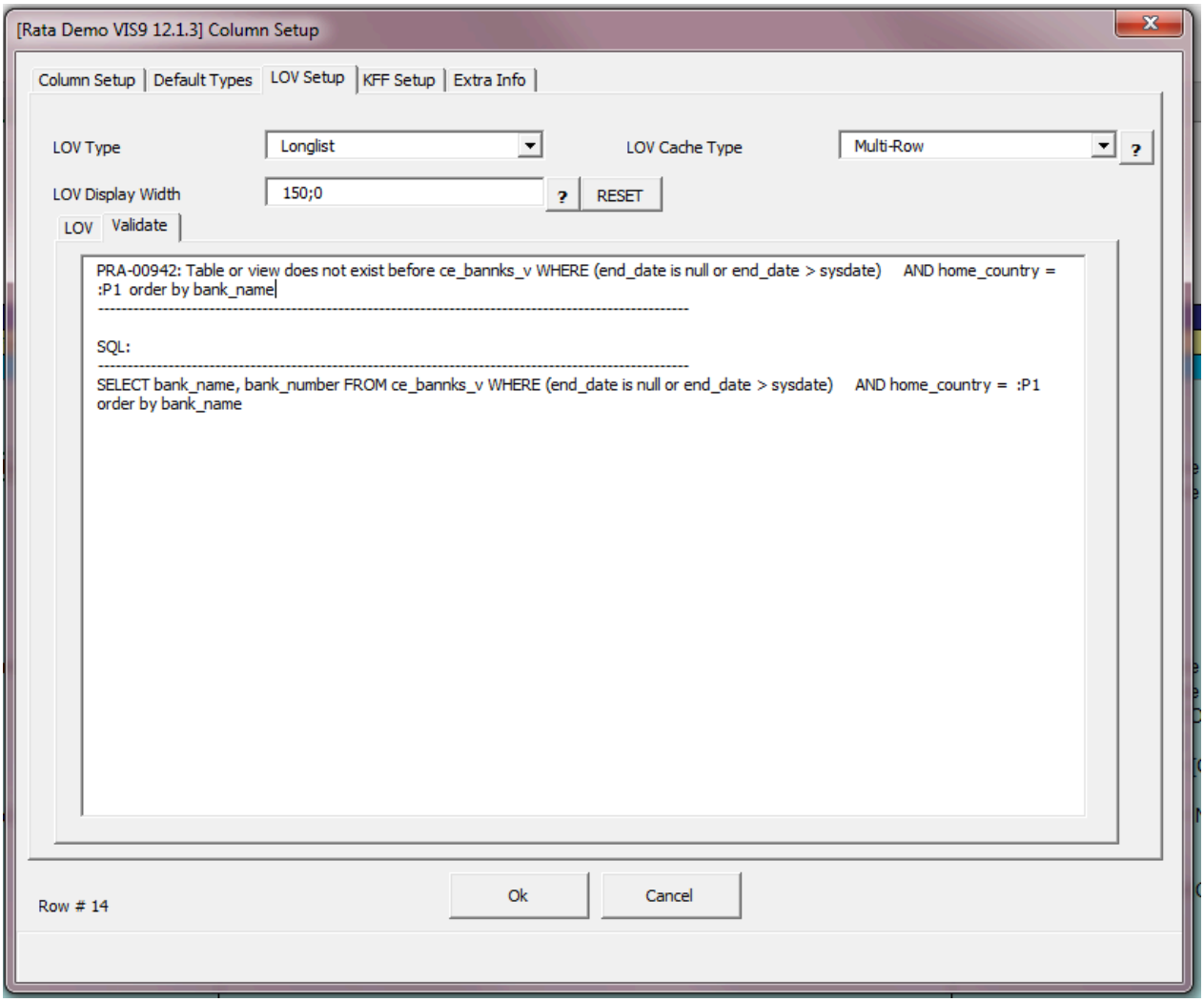


The 'Validate Messages' page will display any SQL errors first. The SQL statement will also be outputted into the results page, allowing for simple copy and paste into your SQL development tool if required.

Example of Successful Validation

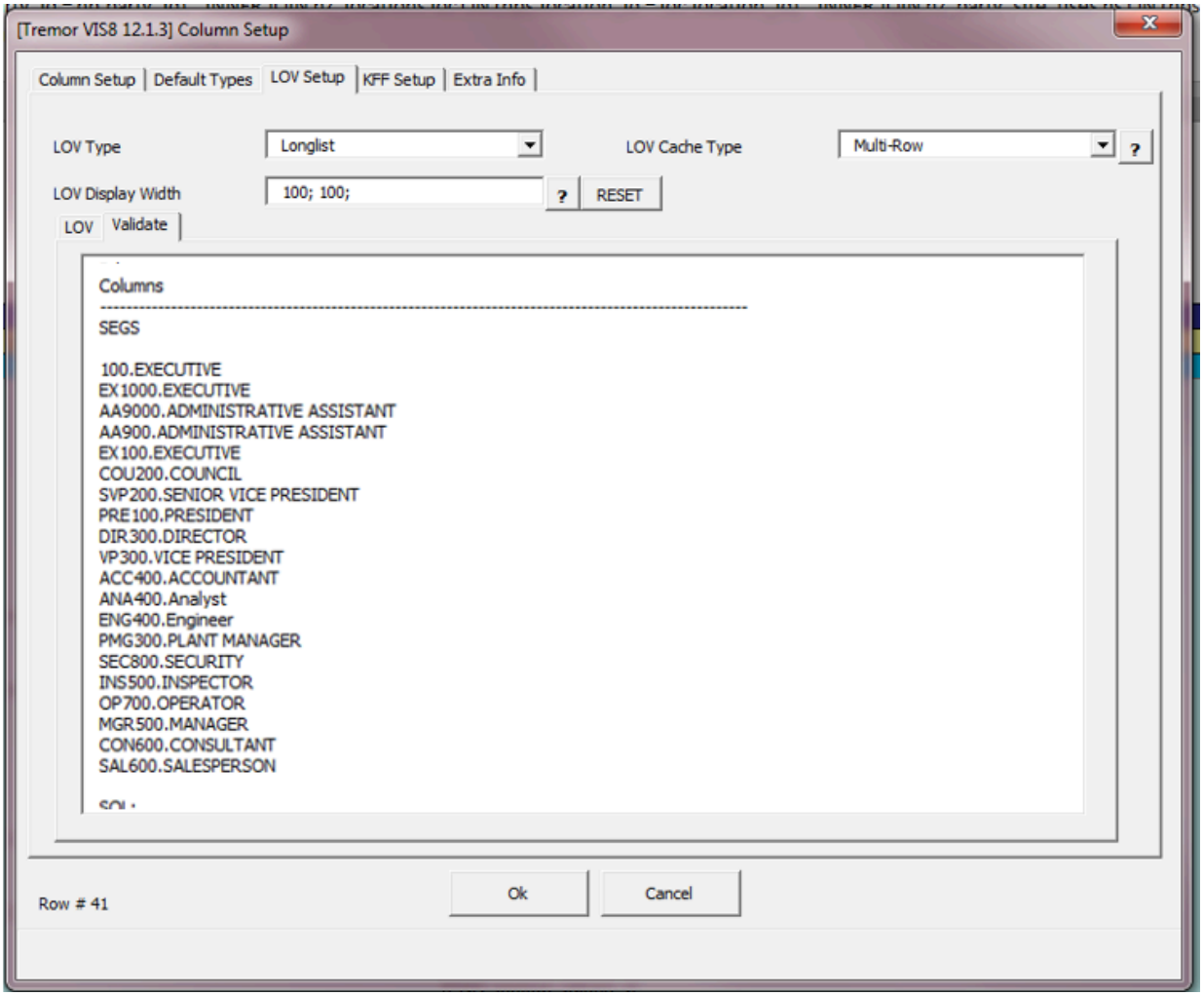


Example of Unsuccessful Validation



Click the Run Query button under the LOV Tab to view the SQL results.

Note - Records will be truncated in this form if too many rows are returned.



Refer to section:
Sub-Sectioning
Column Definitions
Configure Defaults
Configure LOV's
Configure KFF's
PL/SQL Validation

Configure KFF's

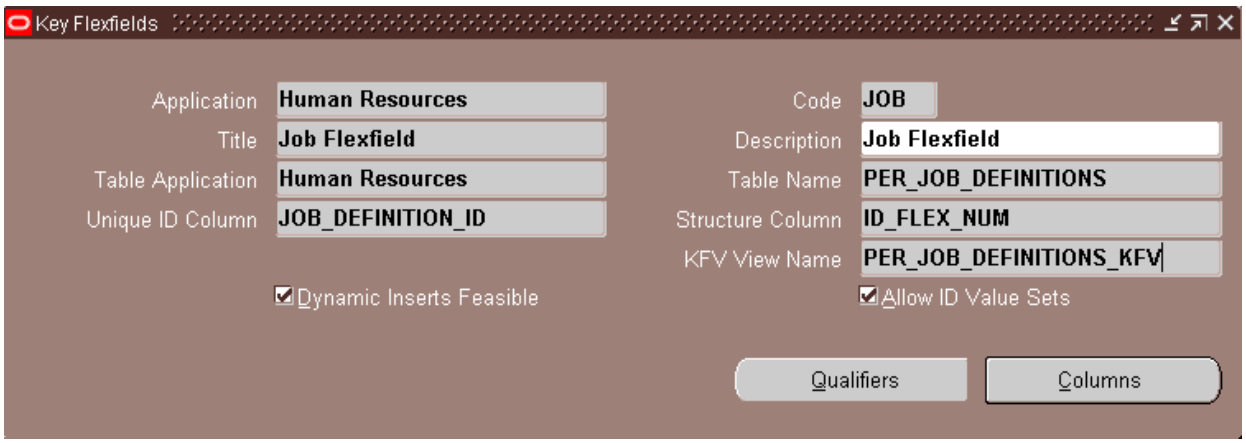
Application Interface Wizard has dynamic data entry forms. Users can double-click on any cell below Row 10 and see all columns listed in the form. Administrators can add LOV's that reference their Key Flexfield (KFF) configuration.

Key flexfields must be set up using the LOV builder. The key flex fields require a SQL statement to return existing concatenated segments, as well as providing flex information such as an Application.

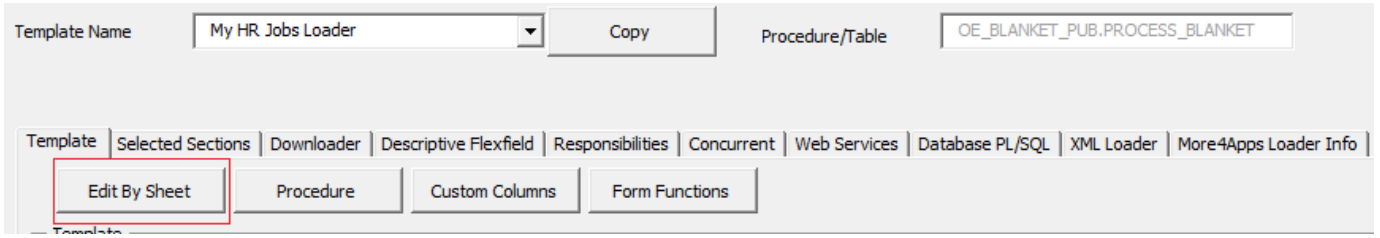
A LOV must be set up against the column to return existing combinations. The LOV Value Field should be the concatenated segments field and the LOV Code field should be the upload ID.

The example below shows the Job Flexfields under Human Resources. If the LOV id_flex_num (Structure ID) is dependent on another column then the dependency builder in the Setup tab must be used. In this case, the id_flex_num (structure ID) can be returned from a query that gets run during initialisation.

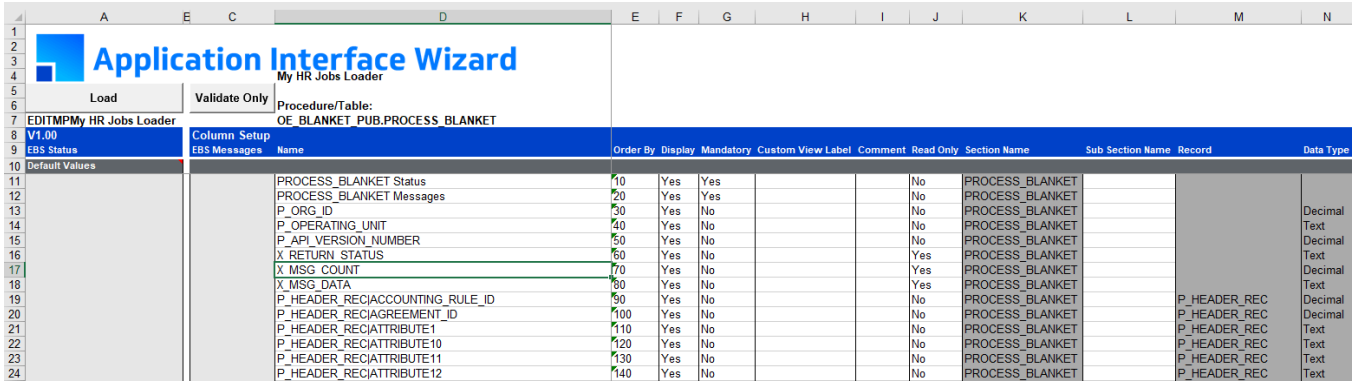
The initial information requirements can be sourced from EBS in the 'Application Developer' Flexfield | Key | Register screen.



To create LOV's, click on 'Edit By Sheet' to download the template attributes into a spreadsheet for simpler and quicker editing.

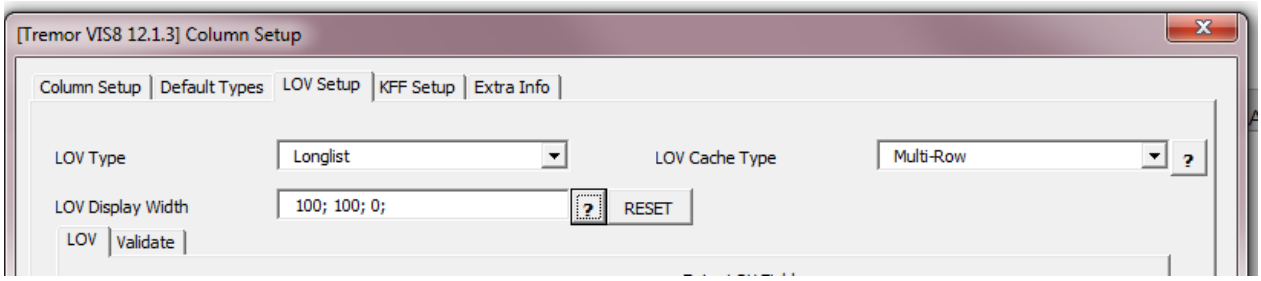


An excel worksheet will open in the Wizard similar to the truncated image below:

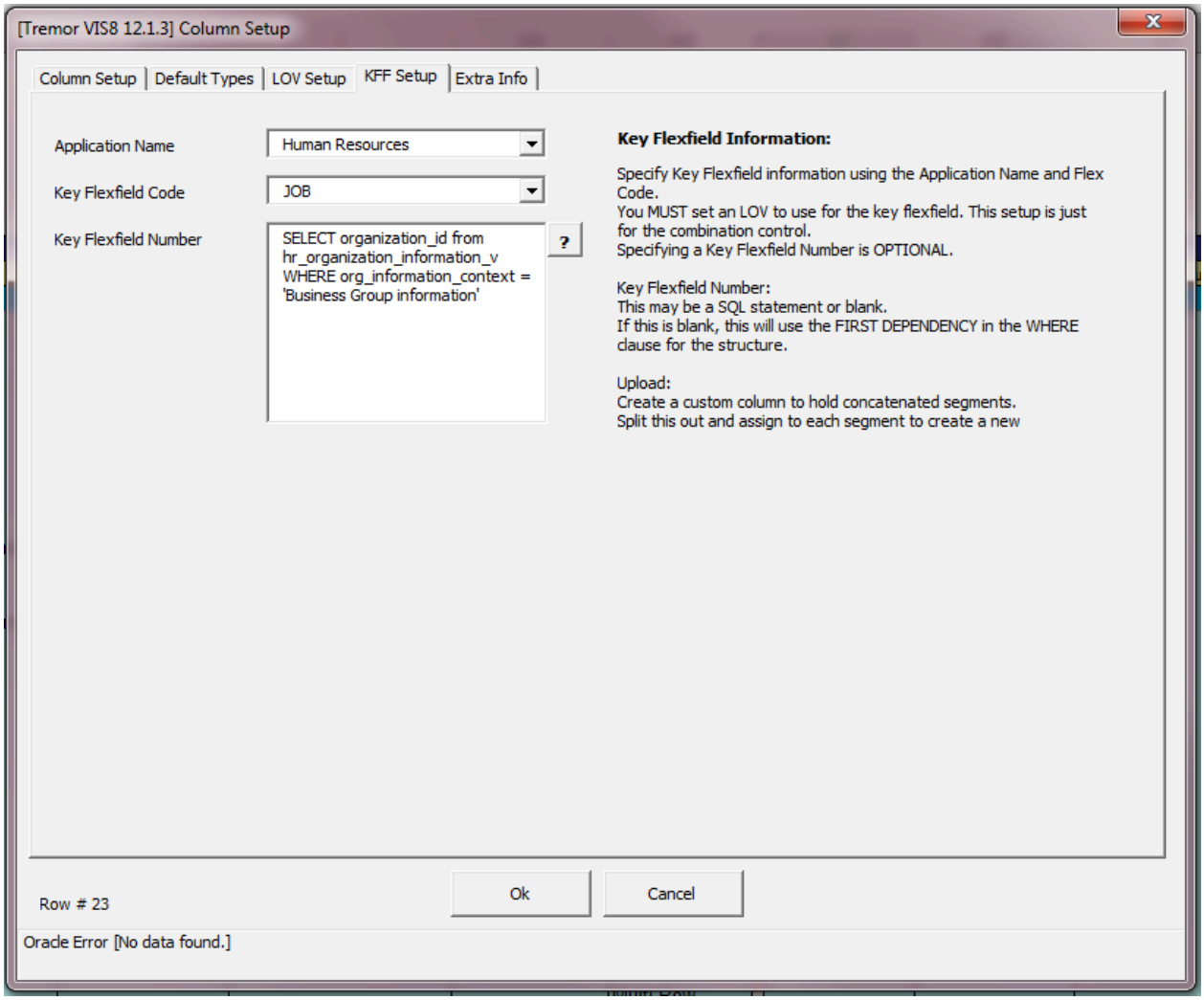


Scroll across to any of the LOV columns in this worksheet and double-click on one of these cells in the row that contains the Key Flexfield column/parameter. Display the 'LOV Setup' tab.

The LOV setup tab requires some information to get the Key Flexfield working correctly. Initially, a LOV Type of Longlist must be entered. This will allow for Key Flexfield information to be entered.



Once entered, navigate to the 'KFF Setup' tab.



The Application Name and Key Flexfield code must be entered.

LOV Setup

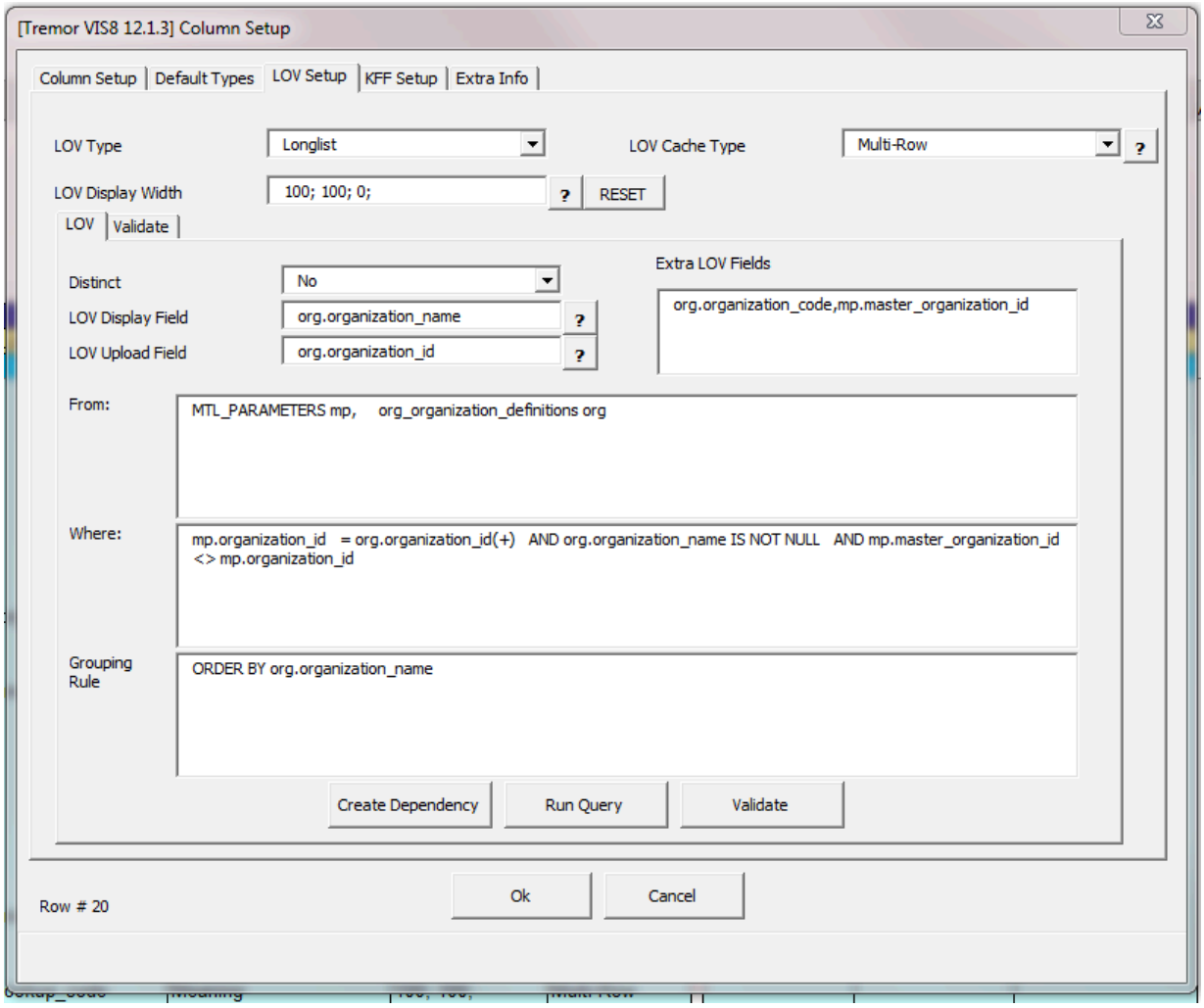
LOV Type	Key Flexfield
LOV Display Width	The LOV Column width is automatically determined when parsing SQL if it is blank. You can then modify the widths if the control drop-down list columns are too small or unreadable.

Key Flexfield

Field	Purpose	Example
Application Name	Enter the EBS 'Application' name of the KFF	Human Resources
Key Flexfield Code	Enter the 'Code' value	JOB
Key Flexfield Number	Either a SQL statement or the parent column	See examples below
KFF Dynamic Insertion	Specified Dynamic Insertion will be used for the KFF Segment Mapping	See examples below
KFF Segment Prefix	Used to specify the prefix of the KFF segments being used in the KFF fields of the template	See examples below

Tip: *Take a look at the relevant structure in FND_ID_FLEXS table. Find the SET_DEFINING_COLUMN_NAME value. The sql to find the Key Flexfield Number must resolve the value for whatever this field shows. E.g. for the Accounting Flexfield this would be the CHART_OF_ACCOUNTS_ID.*

Example of LOV page for Key Flexfield column:



KFF Mapping

KFF Mapping was introduced as a means of mapping the segments of a Key Flexfield in a single column value to associated database columns on upload.

To set up KFF Mapping, the fields 'KFF Dynamic Insertion' and 'KFF Segment Prefix' need to be populated. For example, as seen below in the 'Edit By Sheet' there are multiple columns that are for each segment in a Key Flexfield:

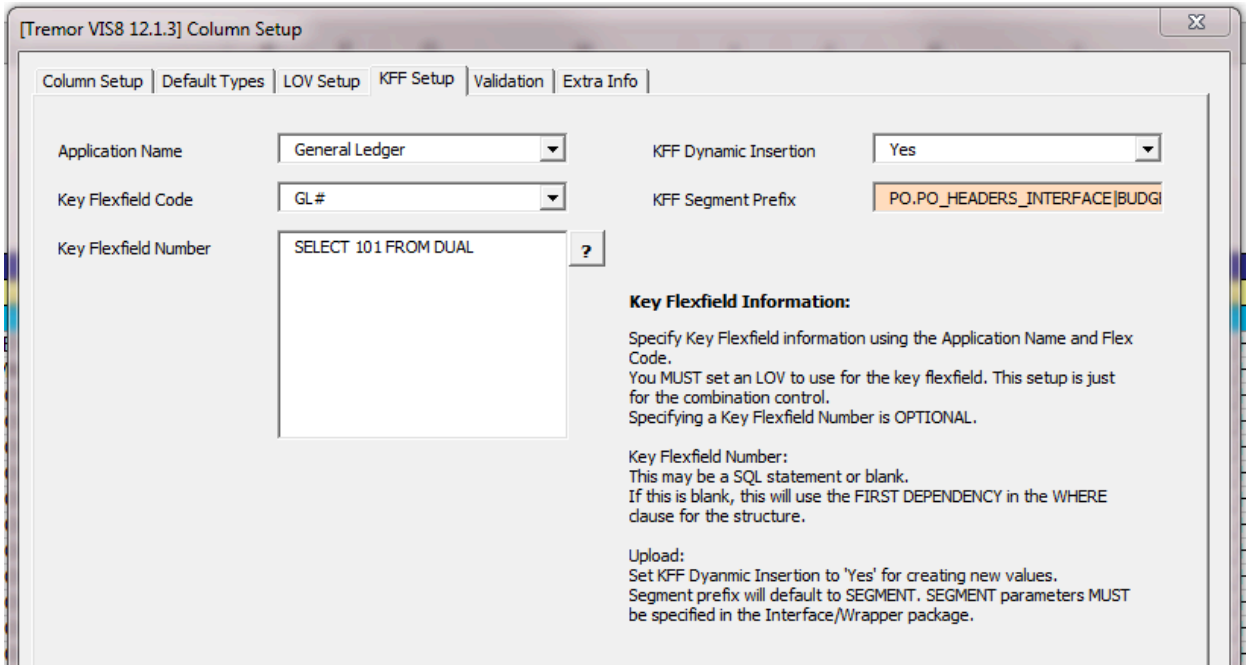
Column Setup

EBS Messages	Name	Order By	Display	Mandatory	Custom View Label	Comment	Read Only	Section Name
	PO.POHEADERS_INTERFACE|AMOUNT_TO_ENCUMBER	10	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|CHANGE_SUMMARY	20	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT1	30	Yes	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT2	40	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT3	41	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT4	42	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT5	44	No	Yes			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT6	50	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT7	51	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT8	60	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT9	70	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT10	80	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT11	90	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT12	111	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT13	112	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT14	113	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT15	114	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT16	115	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT17	116	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT18	120	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT19	130	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT20	140	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT21	141	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT22	150	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT23	160	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT24	165	No	No			No	Flex Values
	PO.POHEADERS_INTERFACE|BUDGET_ACCOUNT_SEGMENT25	166	No	No			No	Flex Values

Instead of needing to have every column on the sheet in order to upload multiple segments, we can use the KFF Mapping functionality to have all fields handled in one column. To do this only the SEGMENT1 field needs to be set to Display = 'Yes', all other segments can have Display set to 'No'.

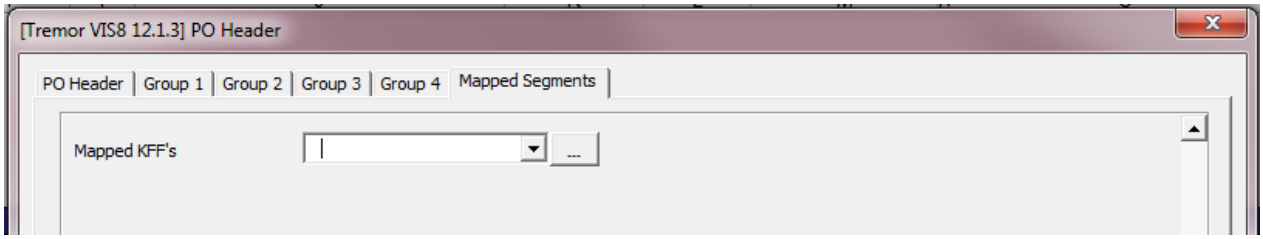
Next the 'Edit By Sheet' form can be opened and navigate to the 'KFF Setup' tab. From here set the 'KFF Dynamic Insertion' to 'Yes'.

Next populate the 'KFF Segment Prefix' with the initial prefix of the segment column names excluding the segment number. For example from the above we would want this prefix to be "PO.PO_HEADERS_INTERFACE\BUDGET_ACCOUNT_SEGMENT".

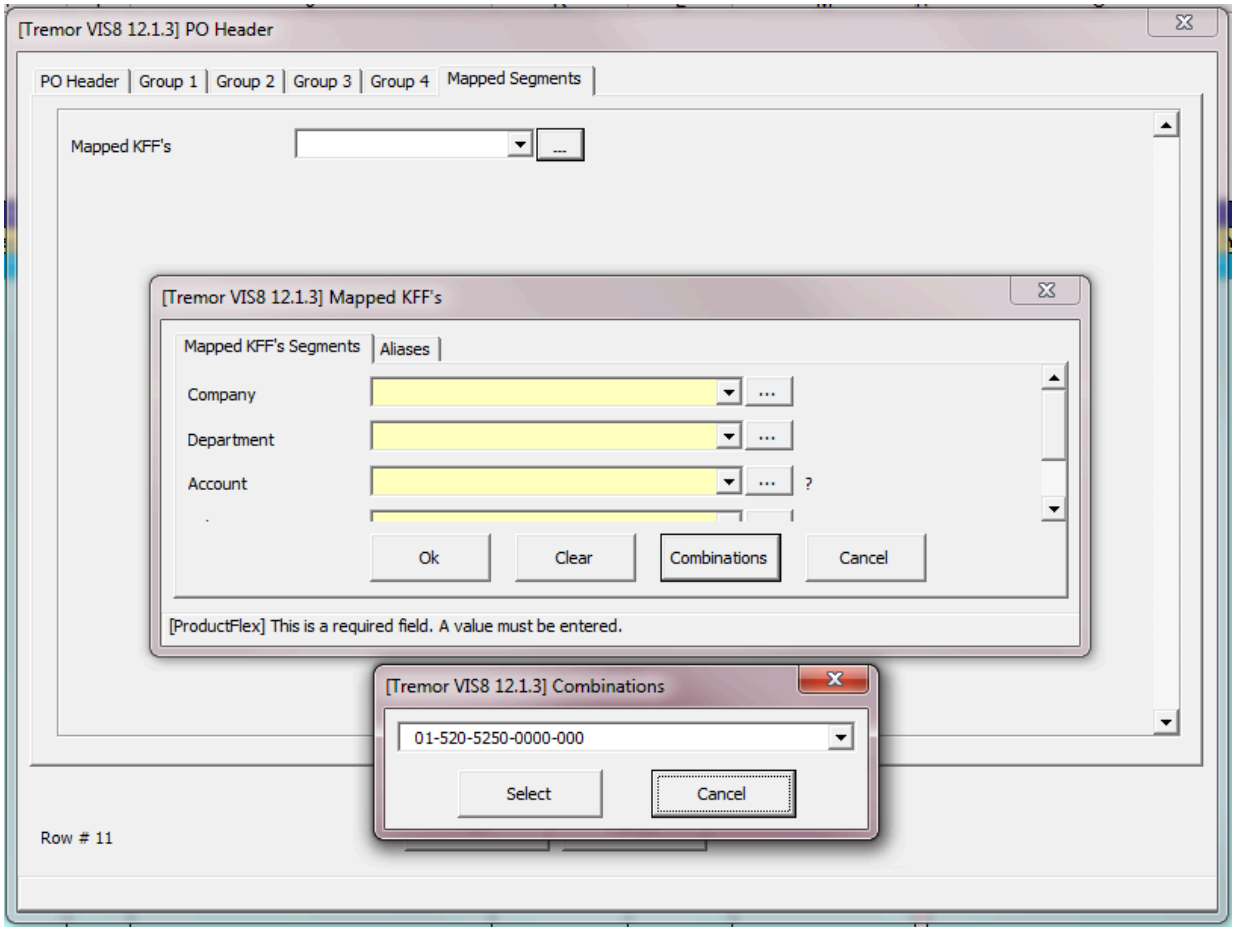


Once this has been done the 'Edit By Sheet' can be reloaded back against the template and the template can be applied.

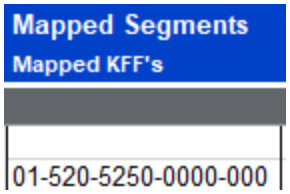
Opening the user form with a template that contains this functionality would appear as shown below:



Clicking the ellipses button will bring up the KFF form with the segments associated to the Key Flexfield.



Upon upload of these values each segment will dynamically be written against the associated segment column in the database e.g.



SEGMENT1: 01
SEGMENT2: 520
SEGMENT3: 5250
SEGMENT4: 0000
SEGMENT5: 000

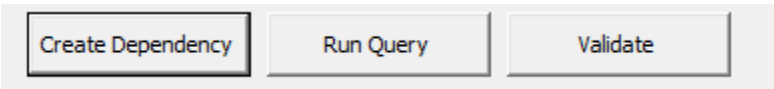
Refer to section:
Sub-Sectioning
Column Definitions
Configure Defaults
Configure LOV's
Validate LOV's
PL/SQL Validation

PL/SQL Validation

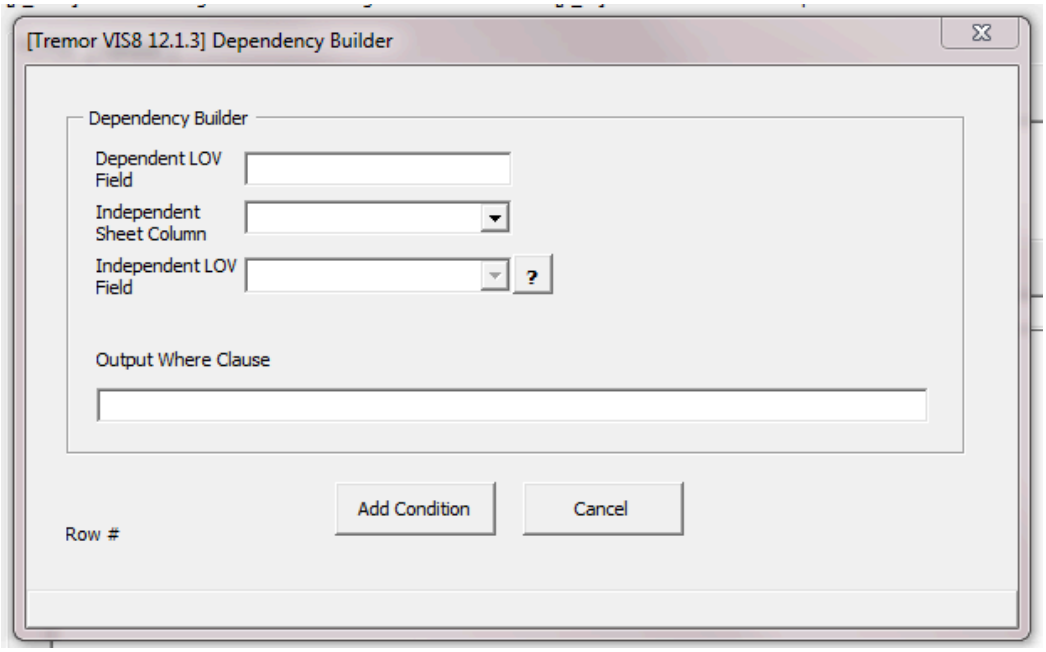
Under the Validation tab we are able to also build our own PL/SQL validation for our loaded.

The form is made up of three sections:

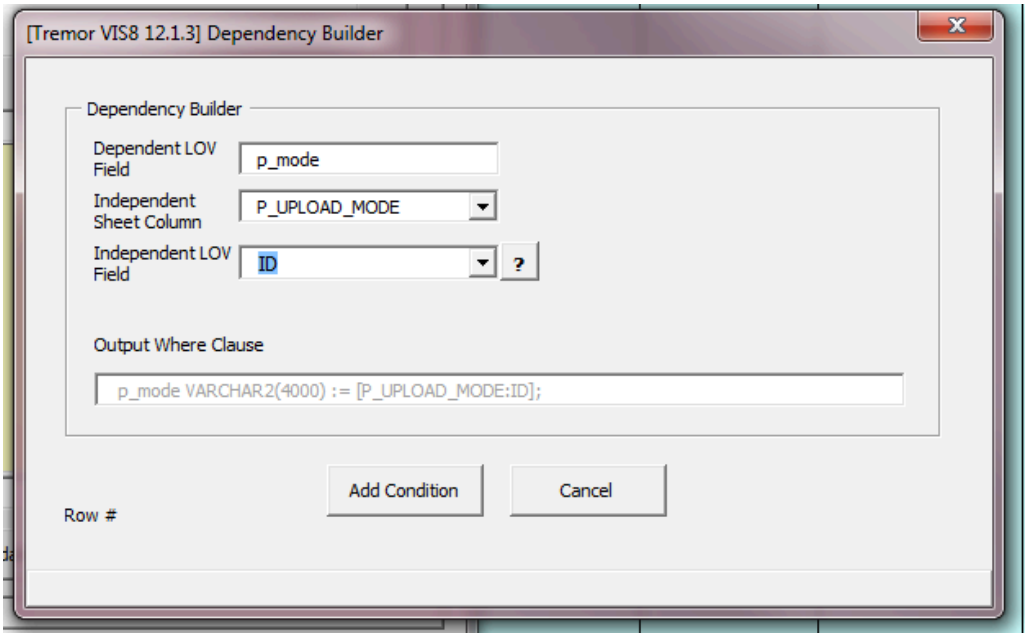
1. PL/SQL Variables
   This box is where your PL/SQL variables will populate:
   - p_ID will be the LOV Upload field value (If a LOV is assigned to the column. Or else will be NULL.
   - p_MEANING is the sheet value. This will always be populated regardless of a LOV.
   - p_ERROR can be set to an error message to fail validation and write to the sheet.
   - b_REQUIRED can be set to TRUE to make a column mandatory for upload (or false for the opposite.
2. PL/SQL Declare
   Your variables and dependencies must be created here, use the dependency builder to build variables to which you can then use those in the PL/SQL Body section.



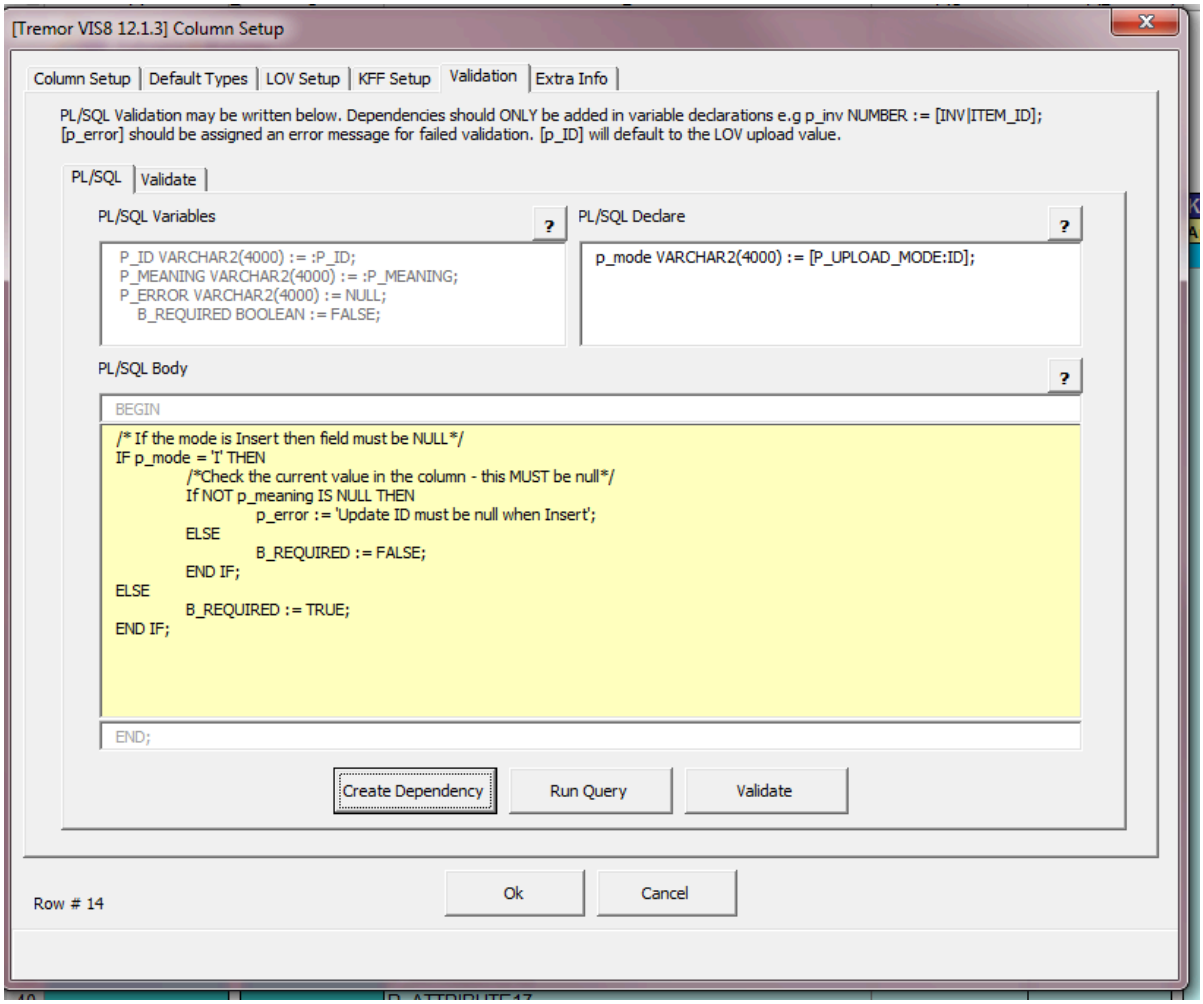
The following form will appear:



We are then able to add in our own dependencies where we can specify the dependent LOV fields for example:



3. PL/SQL Body
   In this section the PL/SQL validation can be entered, for example we want to set a column to be required when we are doing an update or delete. The following image shows this:



Note the following regarding the PL/SQL variables:

-p_ID can be assigned a new ID for upload, or if a LOV is tied to a column p_ID will be the ID value from the LOV.

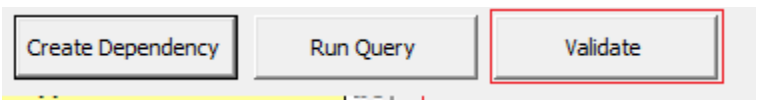
-p_ERROR can be assigned to write messages back. This does NOT have to be set if b_REQUIRED is used, otherwise this MUST be set with a message.

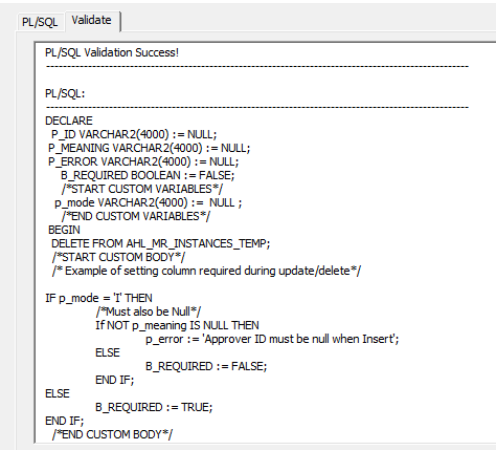
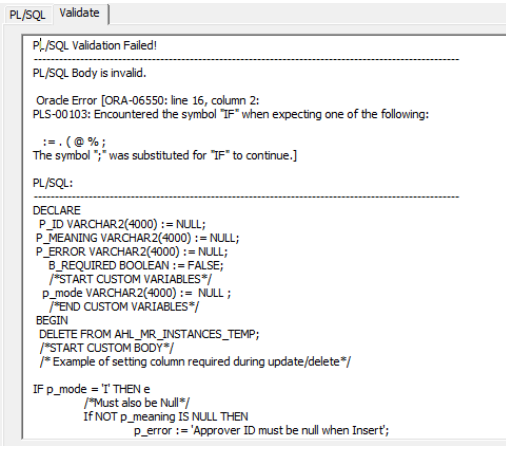
-b_REQUIRED is used to set a columns mandatory property for upload. If this is true then a value MUST be provided for upload.

-b_REQUIRED can be set to TRUE or FALSE here independently of p_ERROR.

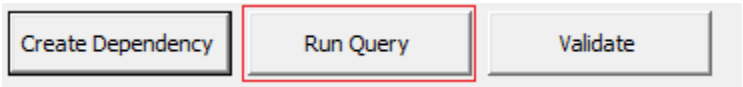
-b_REQUIRED will default to the columns mandatory setting.

The PL/SQL can then be validated using the Validate button. This will then either display a success message or an error in which you will need to fix the validation.

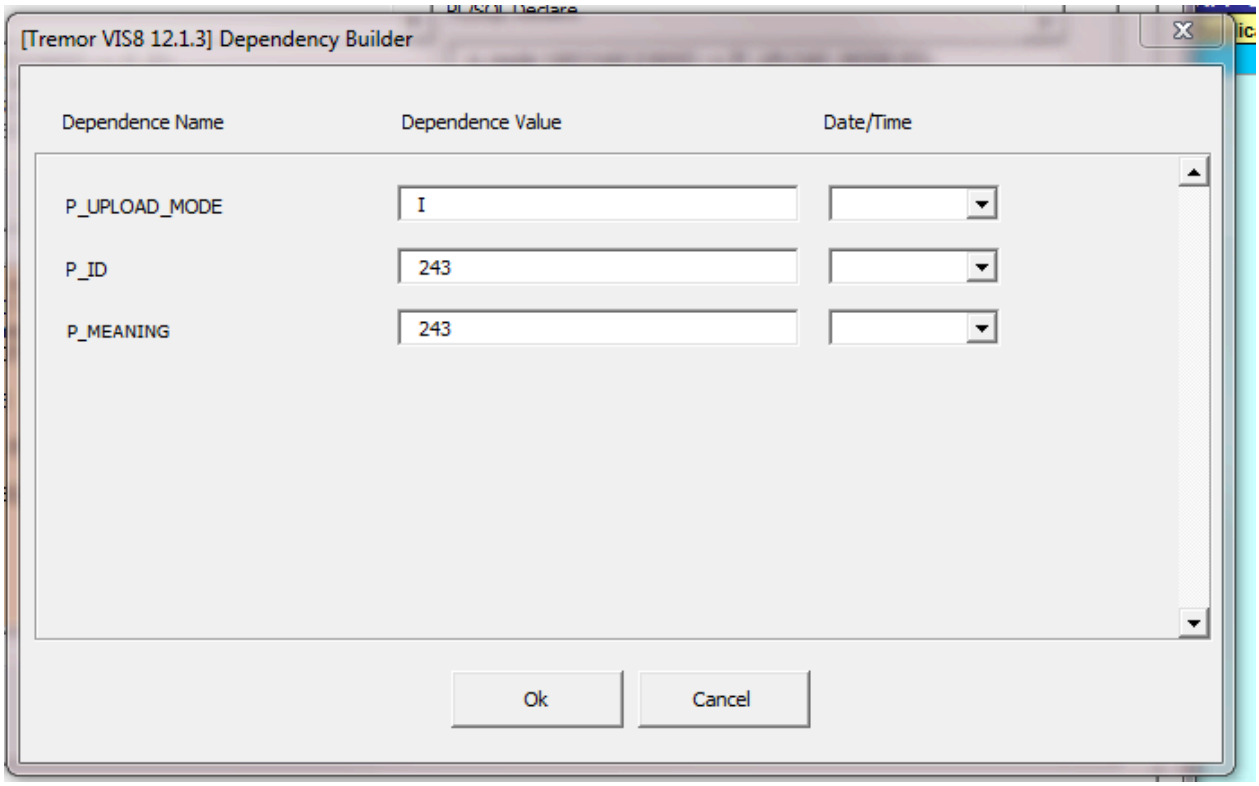


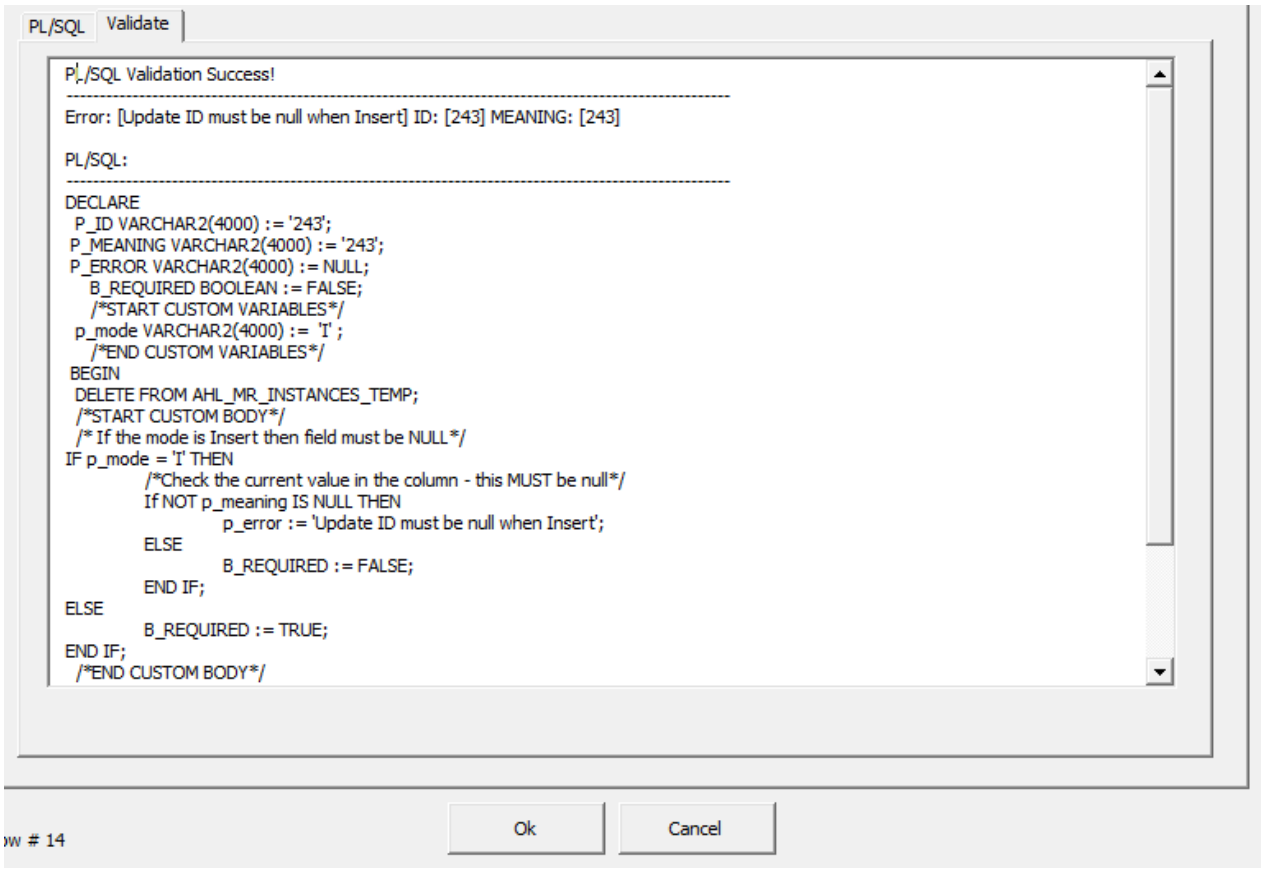
Success	Fail
	

We can then also use the 'Run Query' to test the validation:



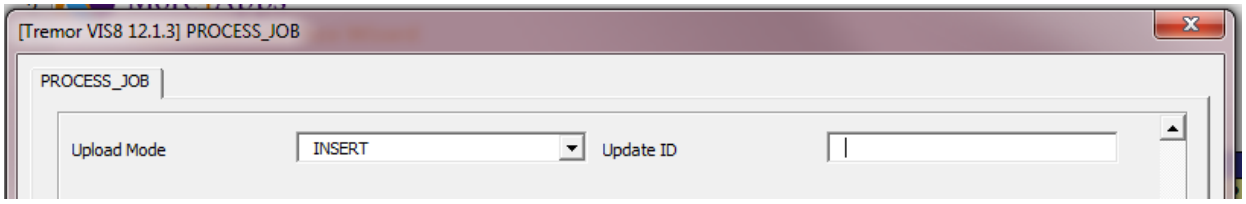
This functionality gives us the ability to provide values to be put through the validation, as seen from the illustration below:



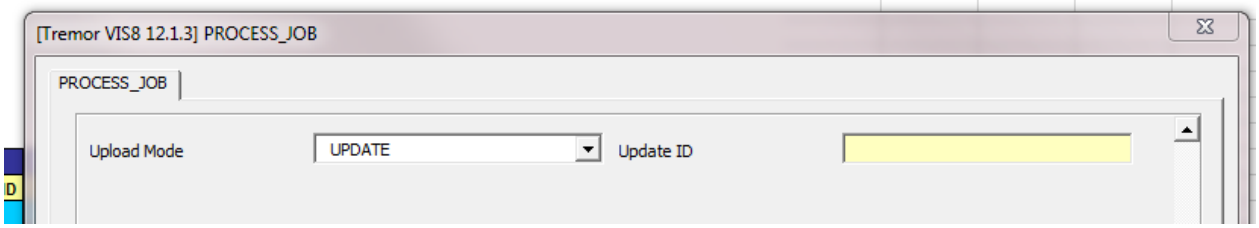


Once this is complete and we have saved our PL/SQL we can see in the form that the fields change based on what Upload Mode is selected based on our PL/SQL written:

INSERT Mode:



UPDATE Mode:



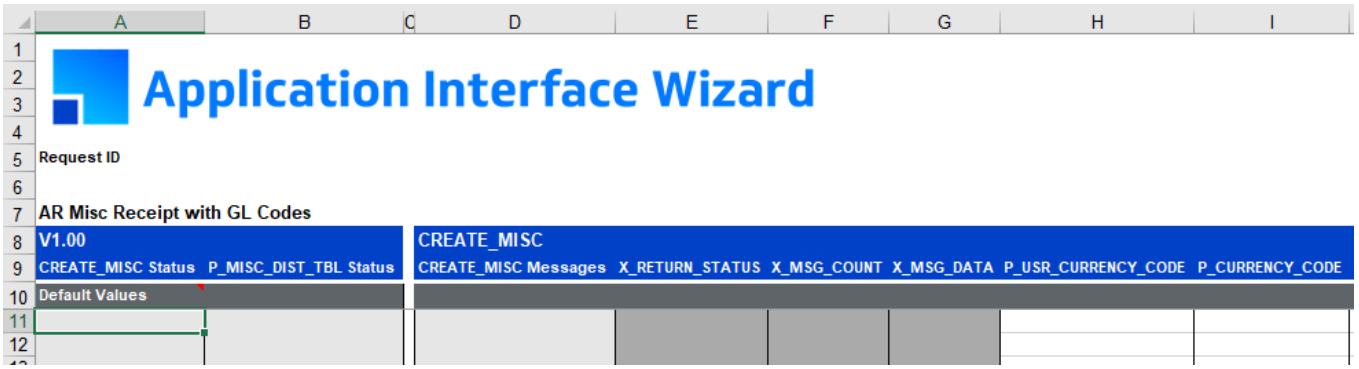
Refer to section:
Sub-Sectioning
Column Definitions

[Configure Defaults]
[Configure LOV's]
[Validate LOV's]
[Configure KFF's]

Selected Sections

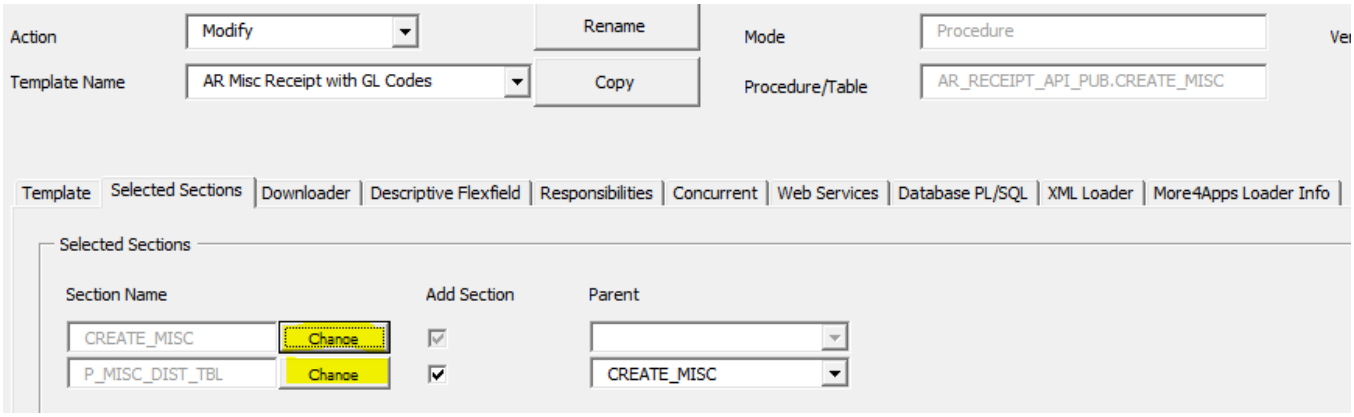
An Administrator who has access to create Templates can modify section names in the layout.

Pre Update Wizard Layout:

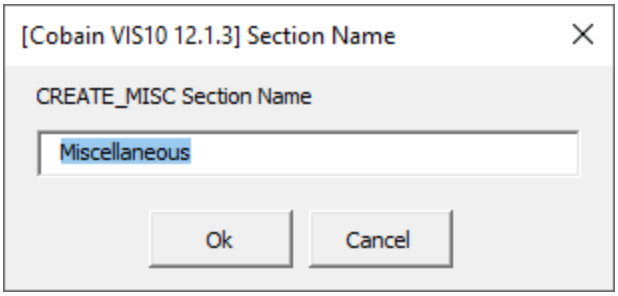


Select 'Modify' in the Action field, then the 'Template Name' in the 'Manage Templates' form.

Select the 'Selected Sections' tab.



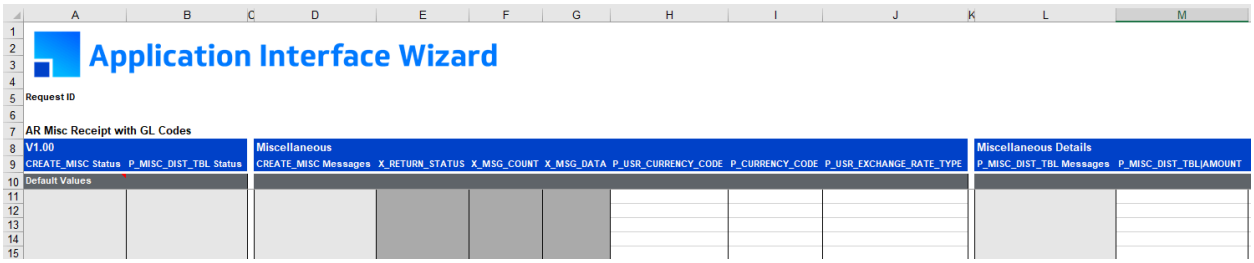
Click each 'Change' button to update the section label as appropriate and click 'Ok'.



Click 'Save To Repository' to save the changes to the database.

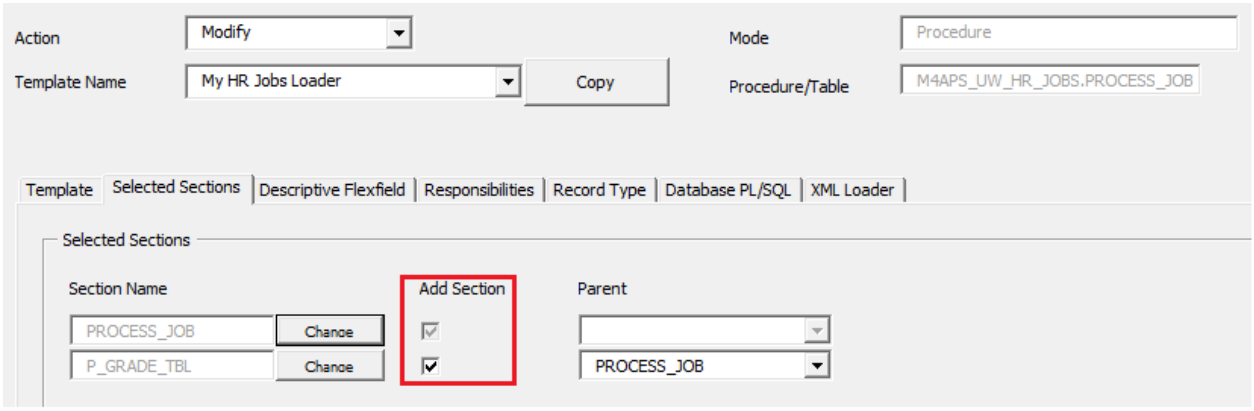
When a new sheet is generated from the template it will now display the updated section names, as per the image below:

Post Update Wizard Layout:



The 'Selected Sections' tab also allows for sections to be excluded from the Template. By default all sections are selected in the Template. An Administrator can deselect any section(s) that are not required by the Wizard. If a section is deselected, no columns in the section will be generated in the worksheet.

Deselect the section flag(s) that are not required if appropriate using the 'Add Section' checkboxes.



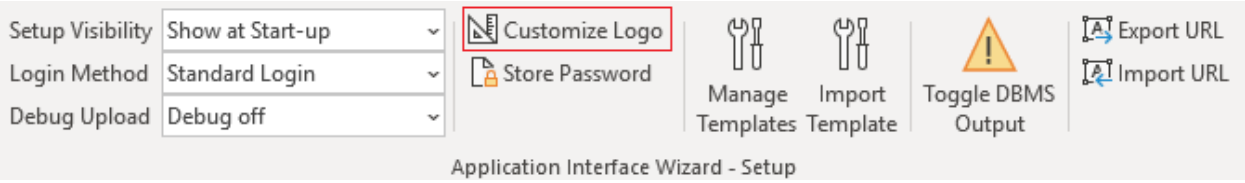
Click 'Save To Repository' to save the changes.

The parent column should be set against any child section to determine which section is its parent. You may have multiple child sections in the template for a parent section.

Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Copy Template
Delete Template
Modify Template
Attachment Templates
Importing Templates
Downloader Template

Customize Logo

The 'Template123' sheet stores the current logo that is used by default in the Wizard templates but can be changed at the user's discretion so that new sheets created have your Company Logo.



This section is common to all of our tools.
Click on this link to navigate to Customize the More4apps Logo

Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All

Update Procedure Name
Copy Template
Delete Template
Modify Template
Attachment Templates
Importing Templates
Downloader Template

Descriptive Flexfields (DFF's)

Descriptive Flexfield values can be entered two different ways using this Wizard depending on what best suits your users.

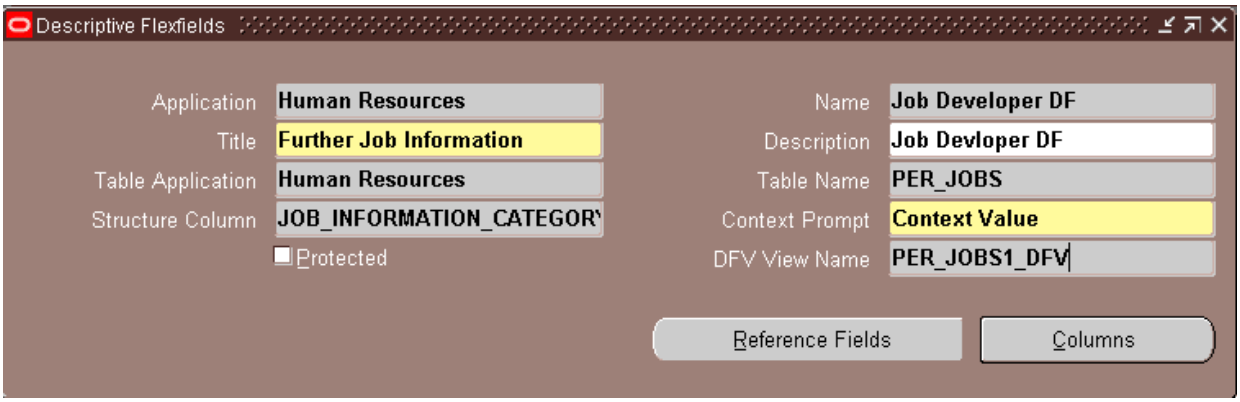
DFF's are typically attribute columns in tables or p_attribute parameters in procedures. As the Wizard retrieves the database object definition into the 'Manage Templates' form, all the attribute columns already exist as individual columns.

Therefore, using the previously described 'Edit by Sheet' functionality these columns can be individually controlled. In this regard, each column would have a separate field in the Wizard data entry form. Assigning LOV's to each column when using this methodology can be more effort as administrators would need to add the individual SQL statements to each column.

DFF's can also be setup where the definition of the attribute segments are also retrieved from the database. This means the Wizard will automatically detect the LOV and mandatory nature of the attribute segment, meaning no additional definition is required.

In the following example we will configure the Human Resources 'Further Job Information' DFF segments in the Wizard.

The initial information requirements can be sourced from EBS in the 'Application Developer' Flexfield | Descriptive | Register screen.



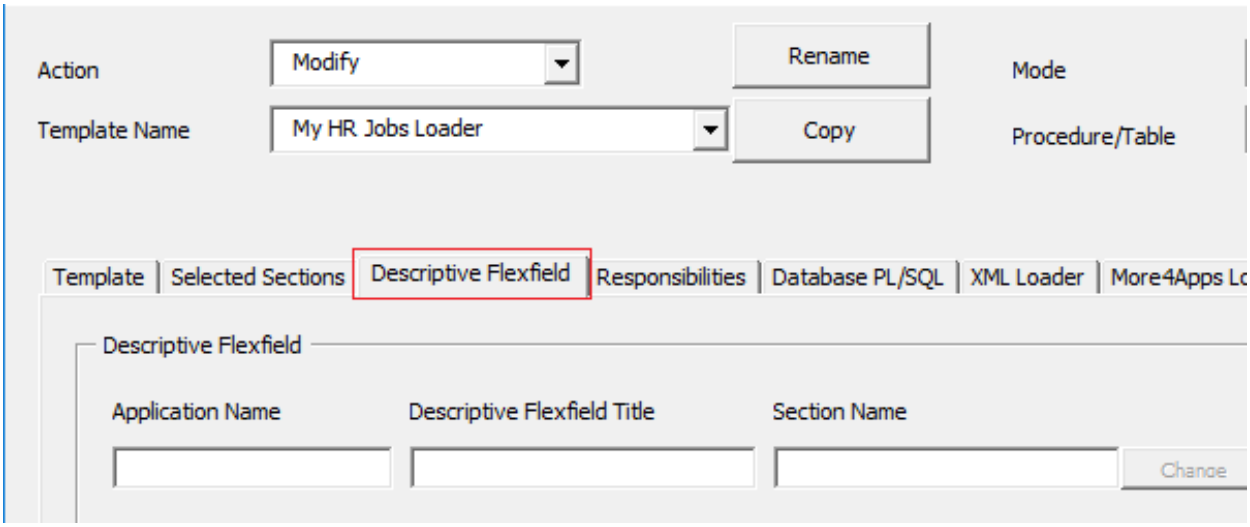
Refer to section:

Configure DFF's
Form References
Removing DFF's
Using DFF Columns

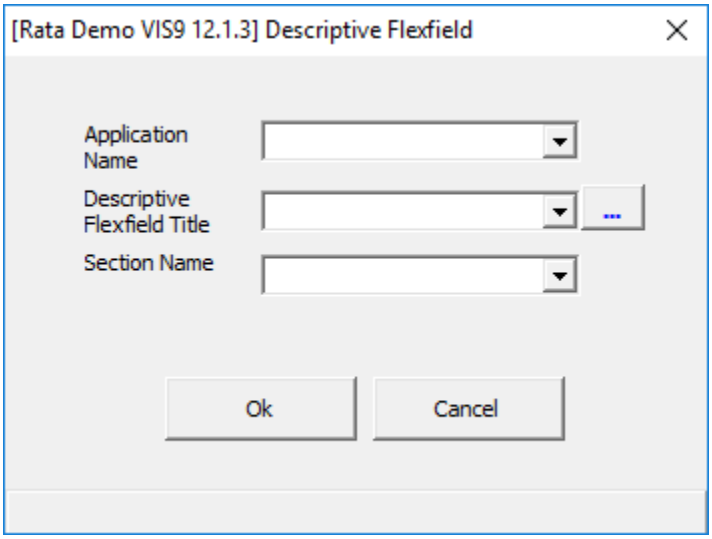
Configure DFF's

Select ‘Modify’ in the Action field, then the ‘Template Name’ in the ‘Manage Templates’ form.

Select the ‘Descriptive Flexfields' tab.



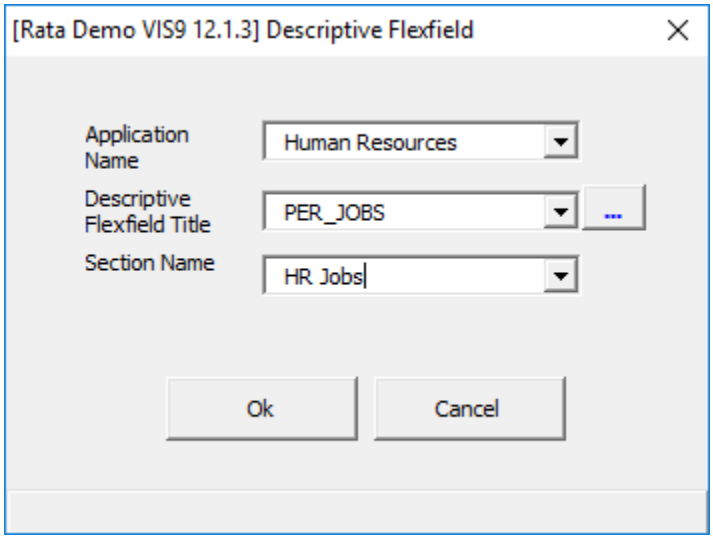
Click the 'New' button to edit the field data.



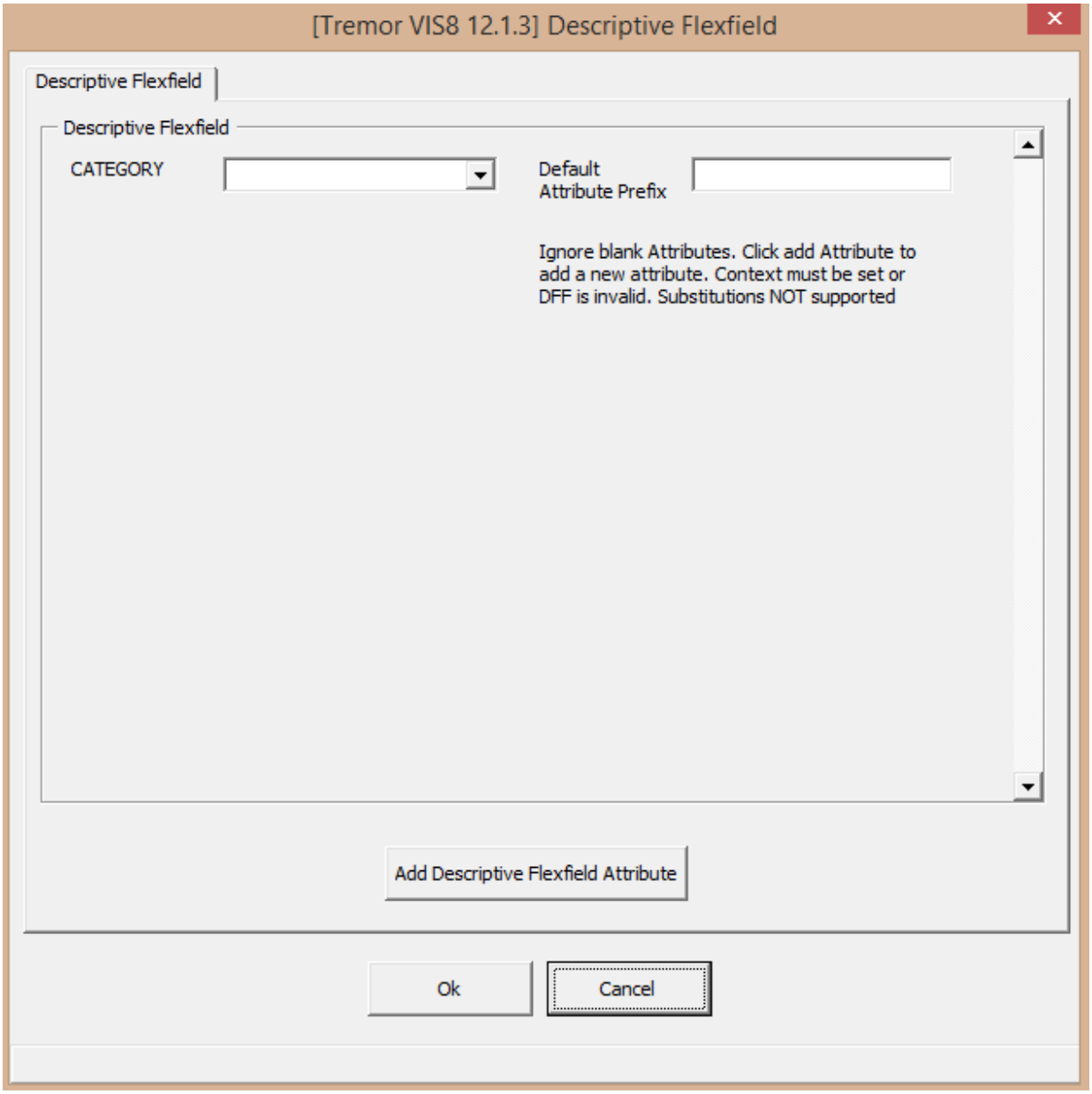
Search and select appropriate values for each field based on the table below:

Field	Purpose	Example
Application Name	Enter the EBS 'Application' name of the DFF	Human Resources

Descriptive Flexfield Title	Enter the 'Table Name' value	PER_JOBS
Section Name	Select the section from your Template definition the DFF columns are to be added to	HR Jobs

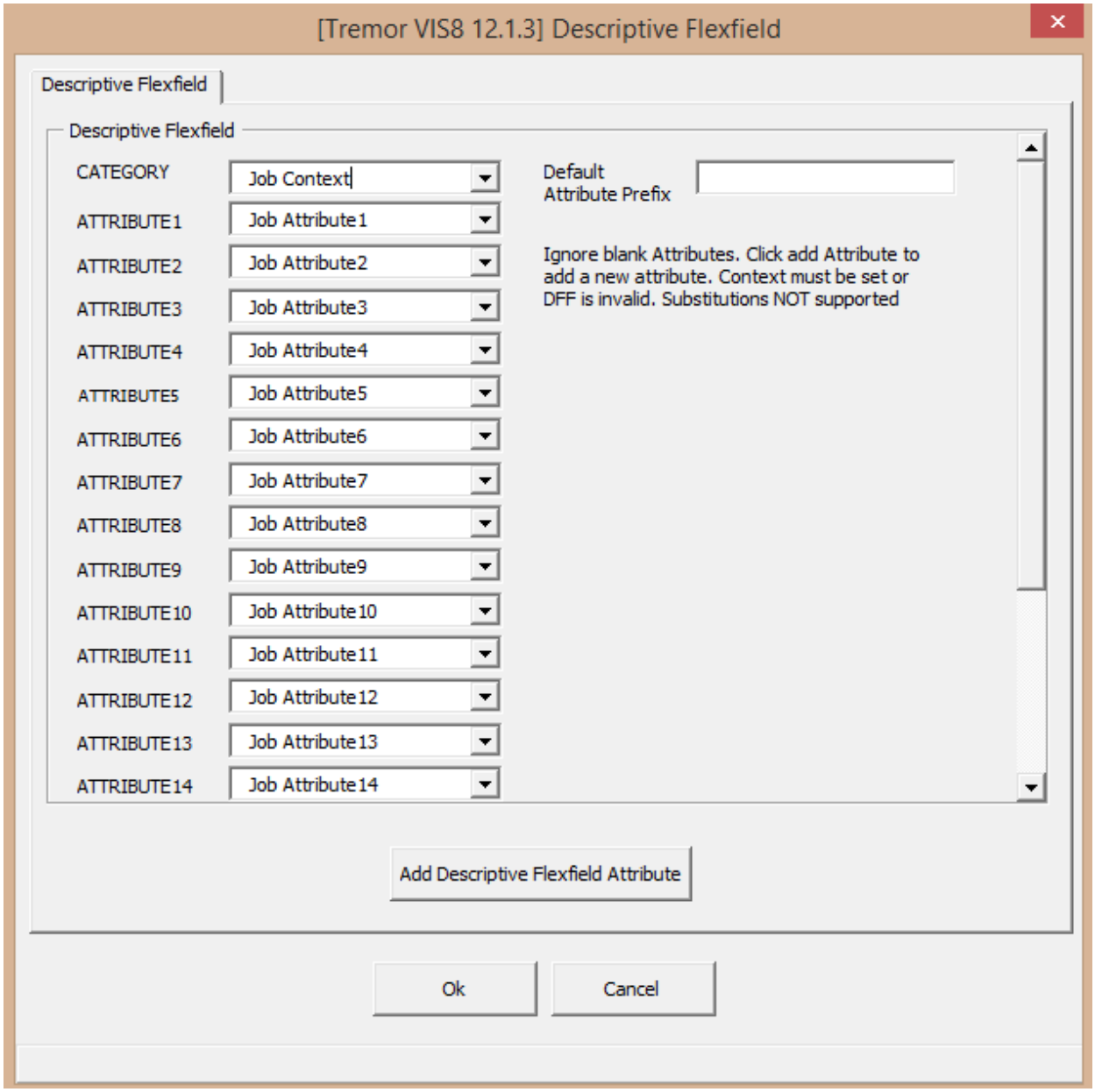


Click 'Ok' to display the Column mapping form.



In the 'CATEGORY' field select the attribute category for the DFF from your Template definition.

To select the available attribute fields, click the 'Add Descriptive Flexfield Attribute' button to create the next available attribute field in the form. Once all the attribute columns have been added, select the appropriate Wizard Columns as defined in your Template definition.



Tip - You can enter a prefix in the 'Default Attribute Prefix' field to default column attributes.

Click 'Ok' when all attributes have been added.

Refer to section:
Form References
Removing DFF's
Using DFF Columns

Form References

If BLOCK.FIELD Form References are used in the DFF value sets, Oracle will give a warning similar to the error message below:

The descriptive flexfield "PO Headers" of application "Purchasing" uses features which are only supported in Oracle Forms. This flexfield may not function in other environments such as OA Framework or PL/SQL APIs.

The number of warning messages: 2.

The segment "Blanket Context Test" of context "BLANKET" uses value set "Standard Document Type" which has BLOCK.FIELD reference "PO_HEADERS.TYPE_LOOKUP_CODE" in its Where clause or Additional Columns.

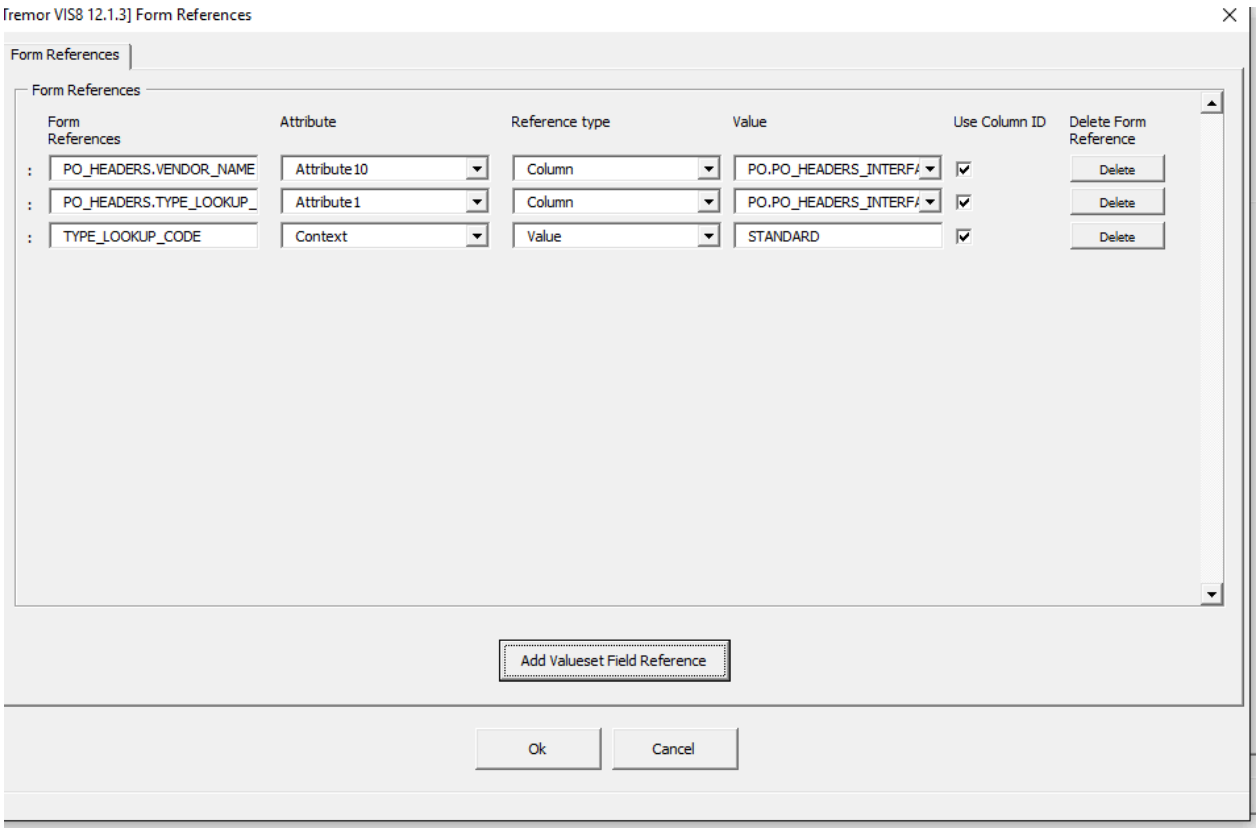
BLOCK.FIELD references are only supported in Oracle Forms.

The segment "Standard" of context "STANDARD" uses value set "Vendor Sites" which has BLOCK.FIELD reference "PO_HEADERS.VENDOR_NAME" in its Where clause or Additional Columns.

BLOCK.FIELD references are only supported in Oracle Forms.

Any form references must be specified and added to the Form References form in the DFF setup. Otherwise the Descriptive Flexfield will cause exceptions within Application Interface Wizard and will not function correctly.

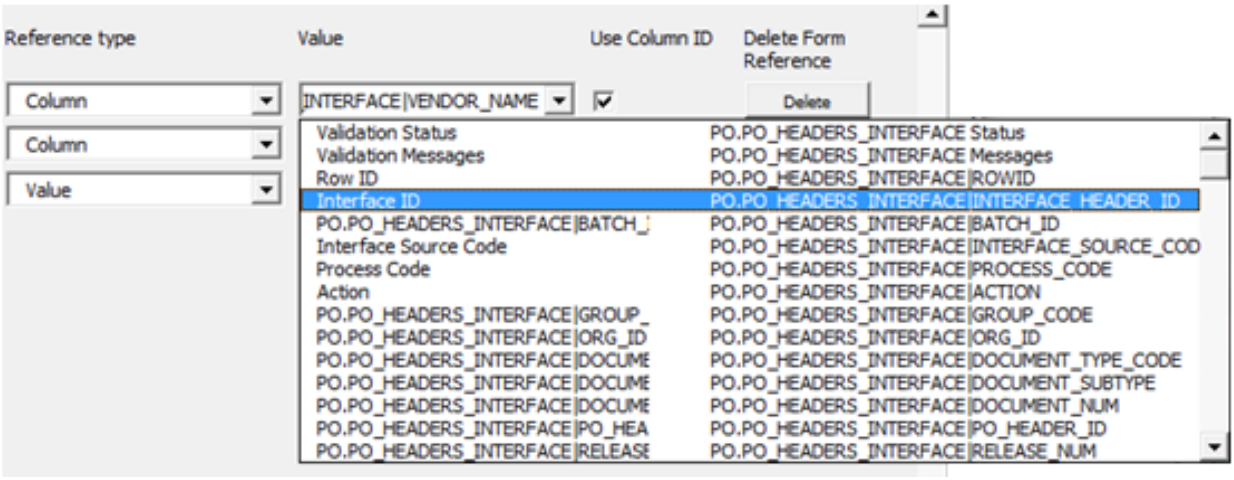
An example has been provided below on how to set up Form References:



- Each Form Reference MUST have the ‘Form References’ field populated WITHOUT the colon ':'
- The associated Wizard Attribute column must then be selected.
- The Reference Type can either look at a column value in the section, or a static value, more information on reference types is below.
- The value populated depends on the type.
- Column ID will use the upload ID if selected, or else will use the sheet value.
- You may delete form references using the ‘Delete’ button.

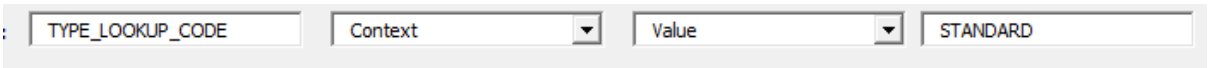
Reference Types - Column

The value selected must be a sheet column. This will derive the value from a column within the current template. The column ID checkbox determines whether the sheet value or the Upload ID is used for the substitution (LOV Only).



Reference Types - Value

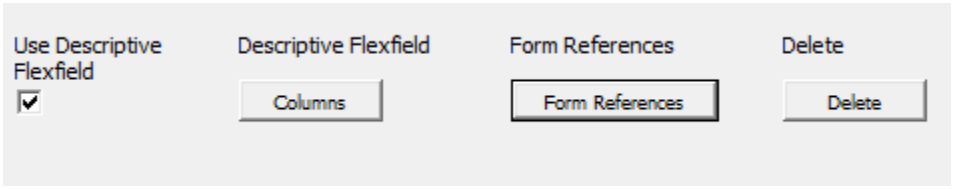
A Reference Type of Value replaces the Form Reference with a static value.



Refer to section:
Configure DFF's
Removing DFF’s
Using DFF Columns

Removing DFF’s

DFF sections can be either removed (hidden) or deleted from the worksheet.



DFF's can be unselected for use on the template by unchecking the 'Use Descriptive Flexfield' checkbox.

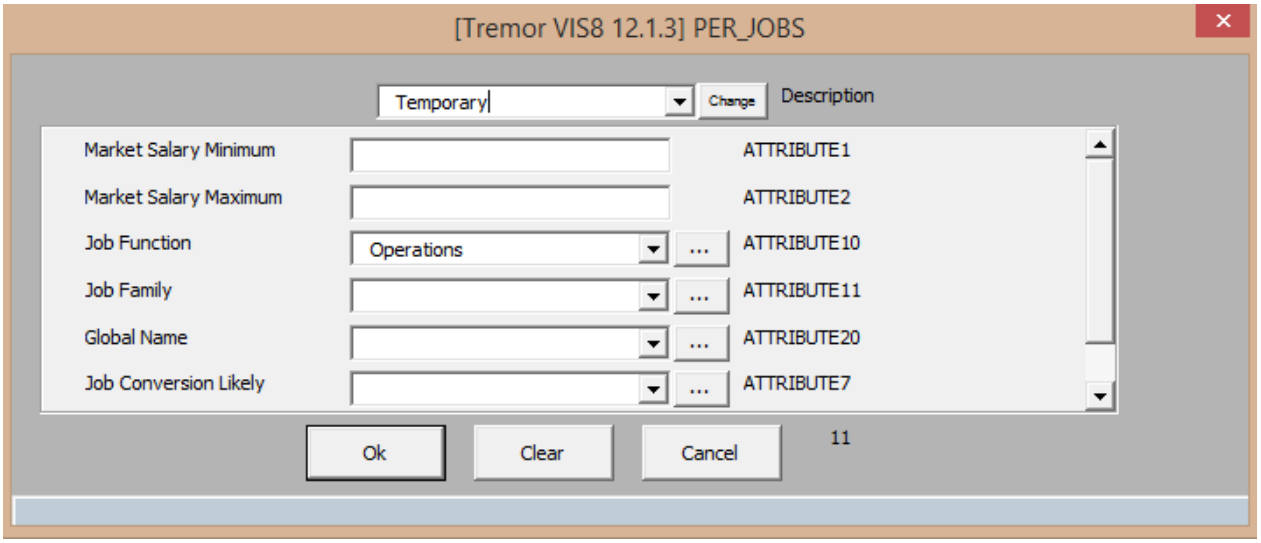
DFF's can be deleted by clicking the 'Delete' button. All associated columns will be reset back to standard columns and form references will be deleted.

Refer to section:
Configure DFF's
Form References
Using DFF Columns

Using DFF Columns

When creating a new sheet, the DFF columns will appear as individual columns in the layout, but their data entry form will have dynamic LOV's as already defined in EBS and work in a similar manner to our feature rich Wizard product suite.

When double-clicking on any of the attribute fields, the same data entry form will appear, similar to the example below where all DFF fields are in the same form:



Note - A disabled field indicates the DFF column is missing from the Worksheet.

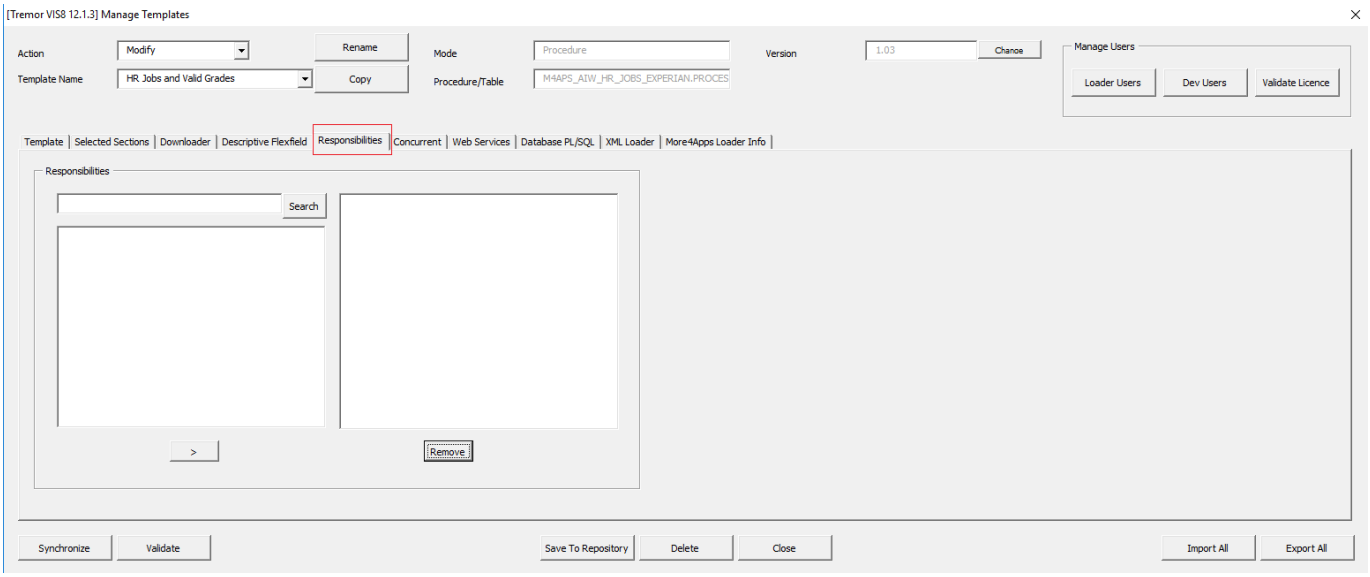
Refer to section:
Configure DFF's
Form References
Removing DFF's

Secure Template to EBS Responsibilities

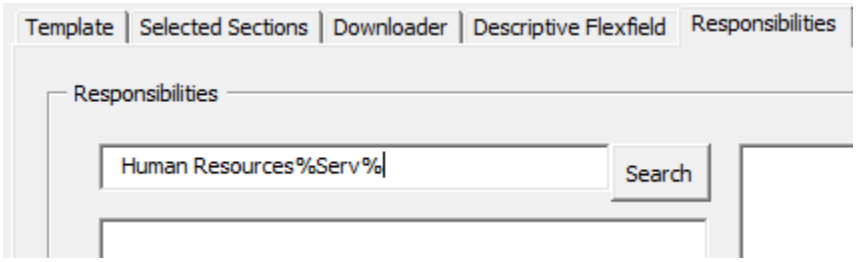
An Administrator who has access to create Templates can grant user access to specific templates by assigning the appropriate responsibility. Users logging in with this responsibility will have access to any templates assigned to the same responsibility and all templates that haven't been assigned to any responsibilities.

Select ‘Modify’ in the Action field, then the ‘Template Name’ in the ‘Manage Templates’ form.

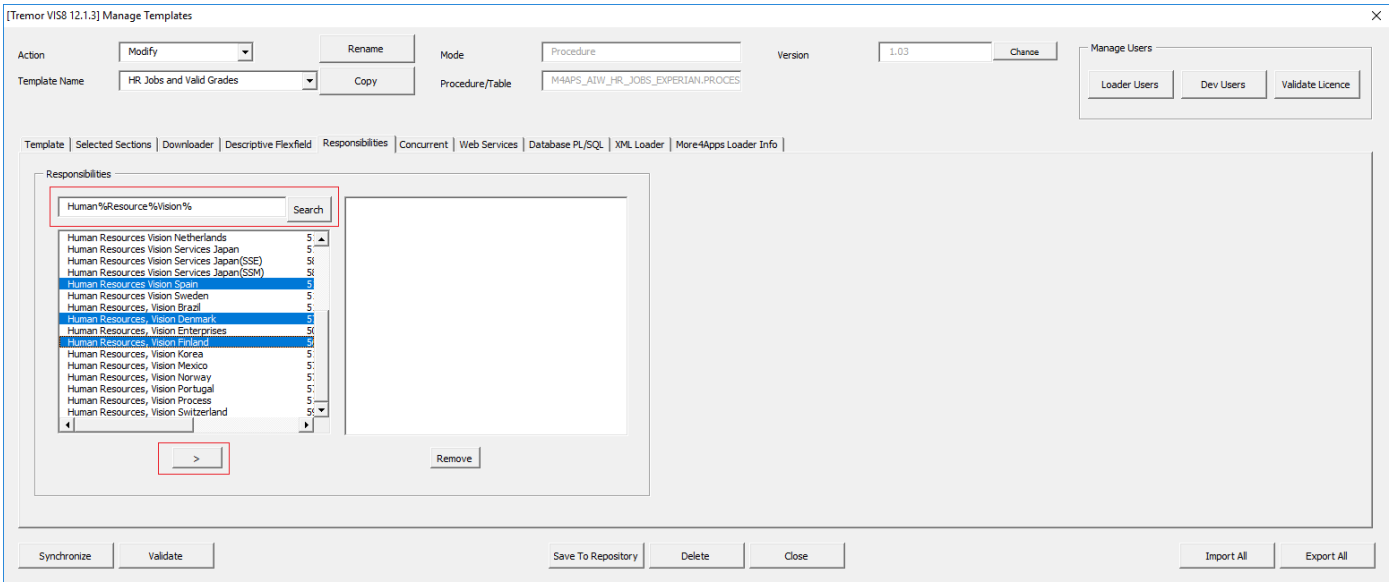
Select the ‘Responsibilities’ tab.



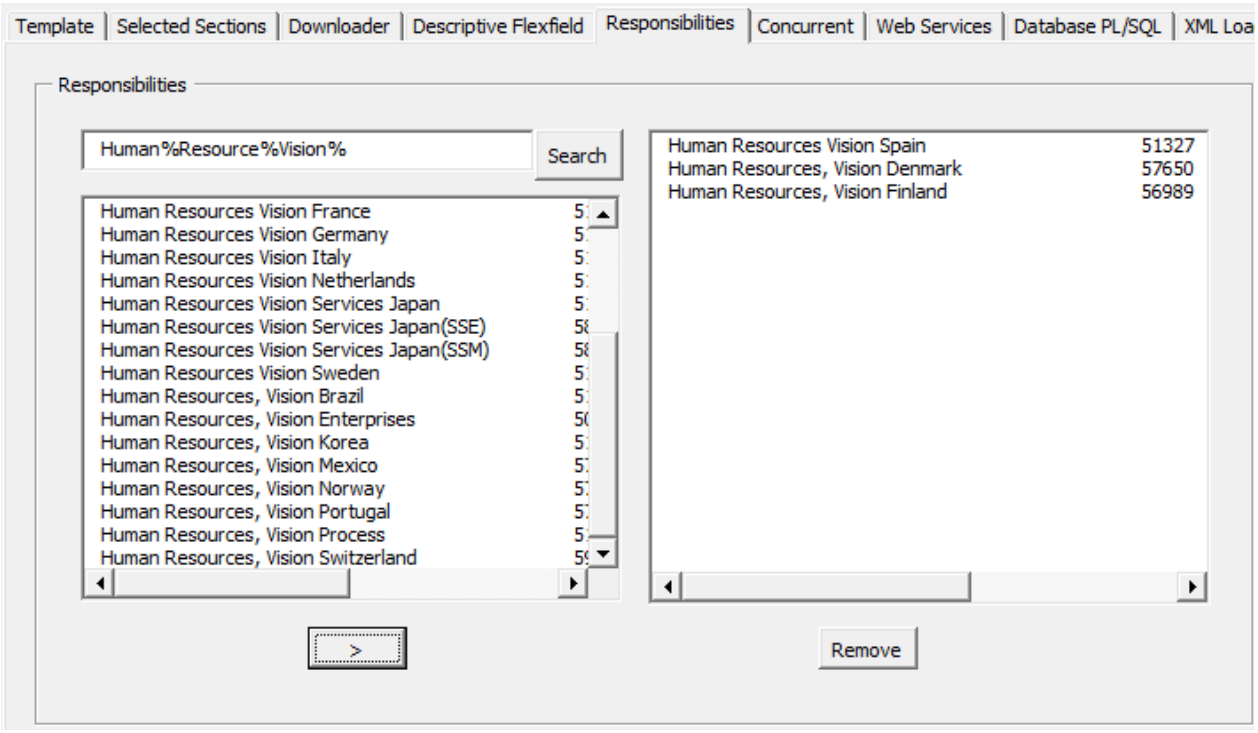
Search for the responsibility you wish to assign access to the Template.



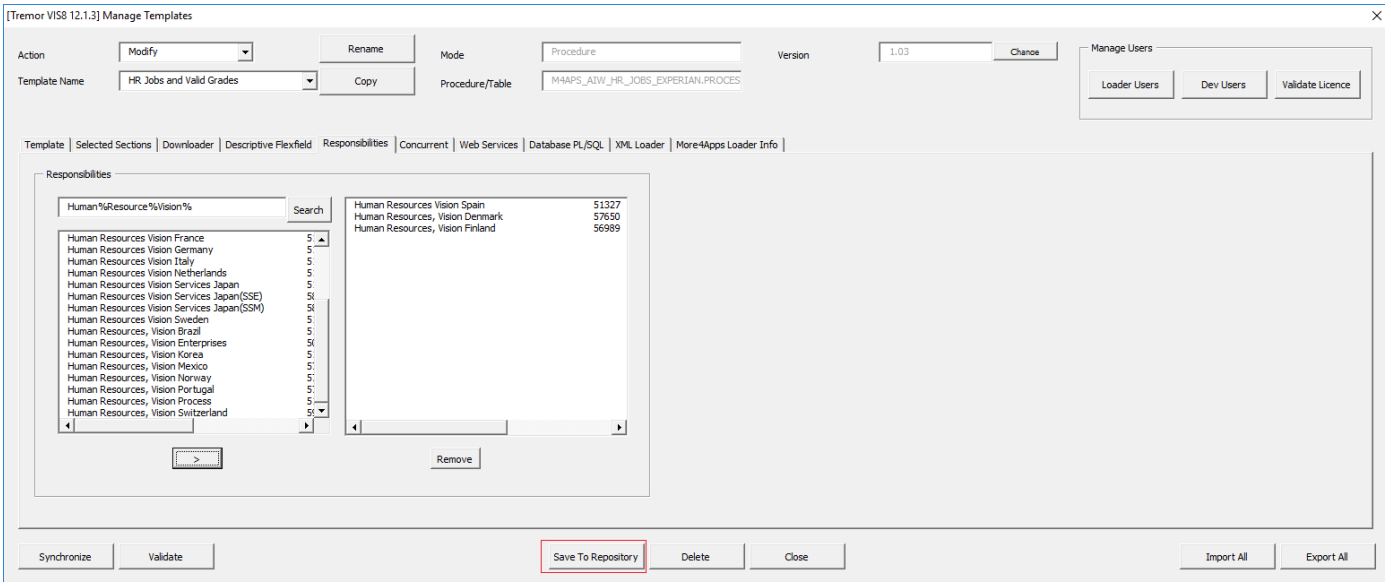
Highlight the Responsibilities you intend assigning the template to, using Ctrl-Click or Shift-Click, and click the arrow button to move the responsibilities to the right-hand pane.



Perform additional searches and copies to the right-hand pane until all required responsibilities have been selected.



Click 'Save To Repository' to assign the Template to the selected responsibilities.



Note - If Templates are not assigned to specific responsibilities ALL EBS users can use them.

Refer to section:

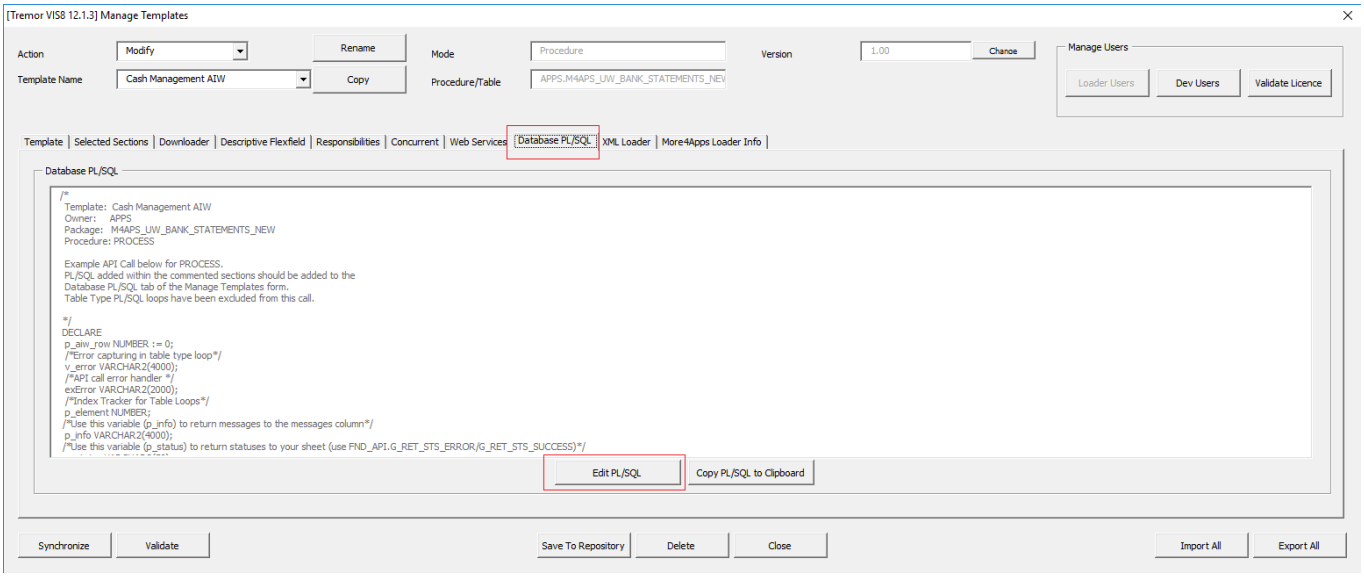
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Copy Template
Delete Template
Modify Template
Attachment Templates
Importing Templates
Downloader Template

Database PL/SQL

PL/SQL can be written into the upload using the 'Database PL/SQL' Tab in Manage templates. This provides additional functionality without needing to use additional packages or your own wrapper packages. For example, you may want to call your own validation and exit the upload before the API is called. This can be achieved very simply using this feature.

Select ‘Modify’ in the Action field, then the ‘Template Name’ in the ‘Manage Templates’ form.

Select the ‘Database PL/SQL’ tab.



A read-only form displaying summary information of the procedure call is now visible.

Tip - Use the 'Copy PL/SQL to Clipboard' button at any time to copy the contents of this form to use in your SQL Editor.

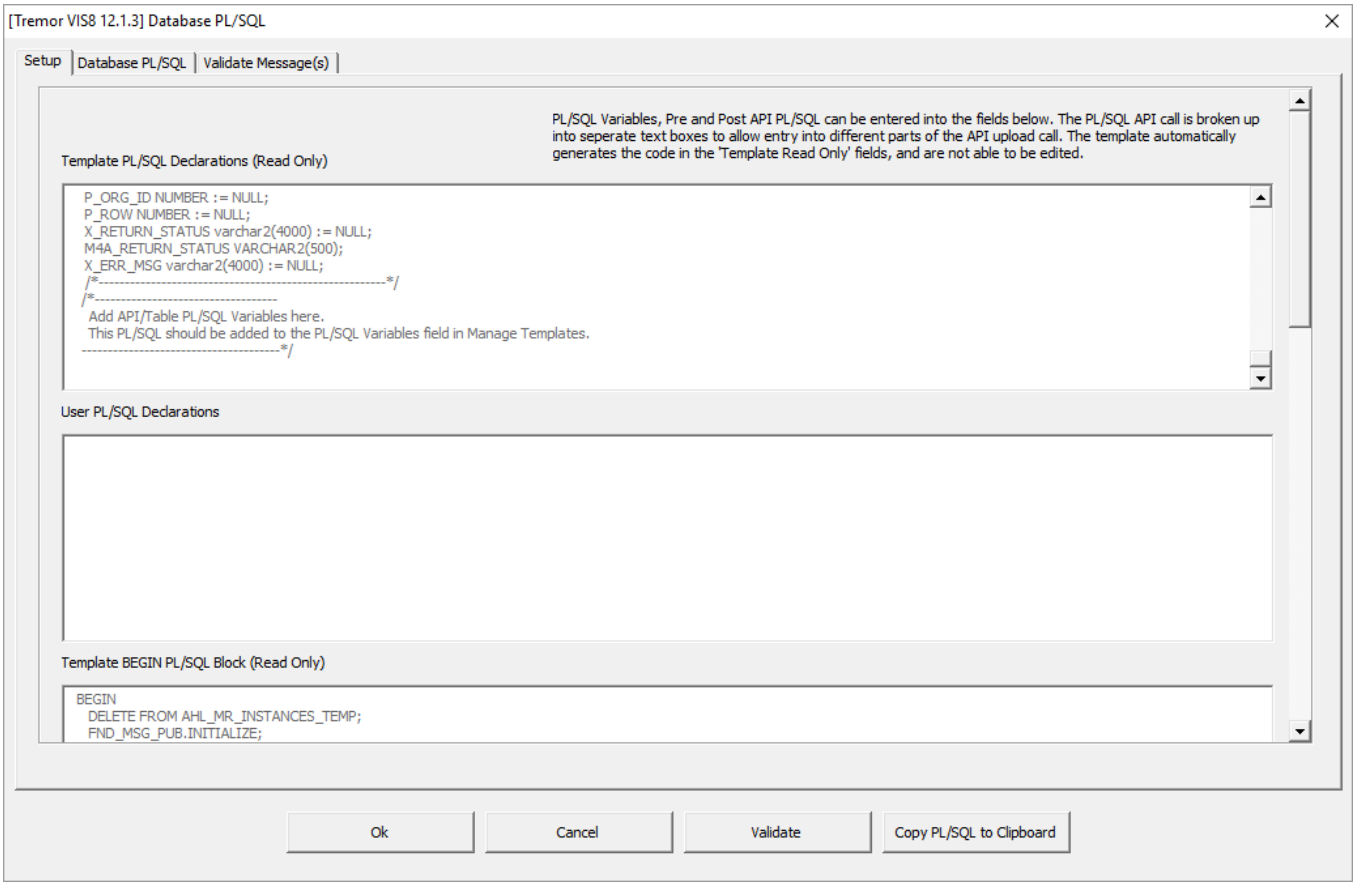
Refer to section:
Define Additional PL/SQL
Copy PL/SQL to Clipboard
Erroring Child Sections

Define Additional PL/SQL

For more information on adding business validation, click here to watch a video.

Click the 'Edit PL/SQL' button to open the Database PL/SQL Editor.

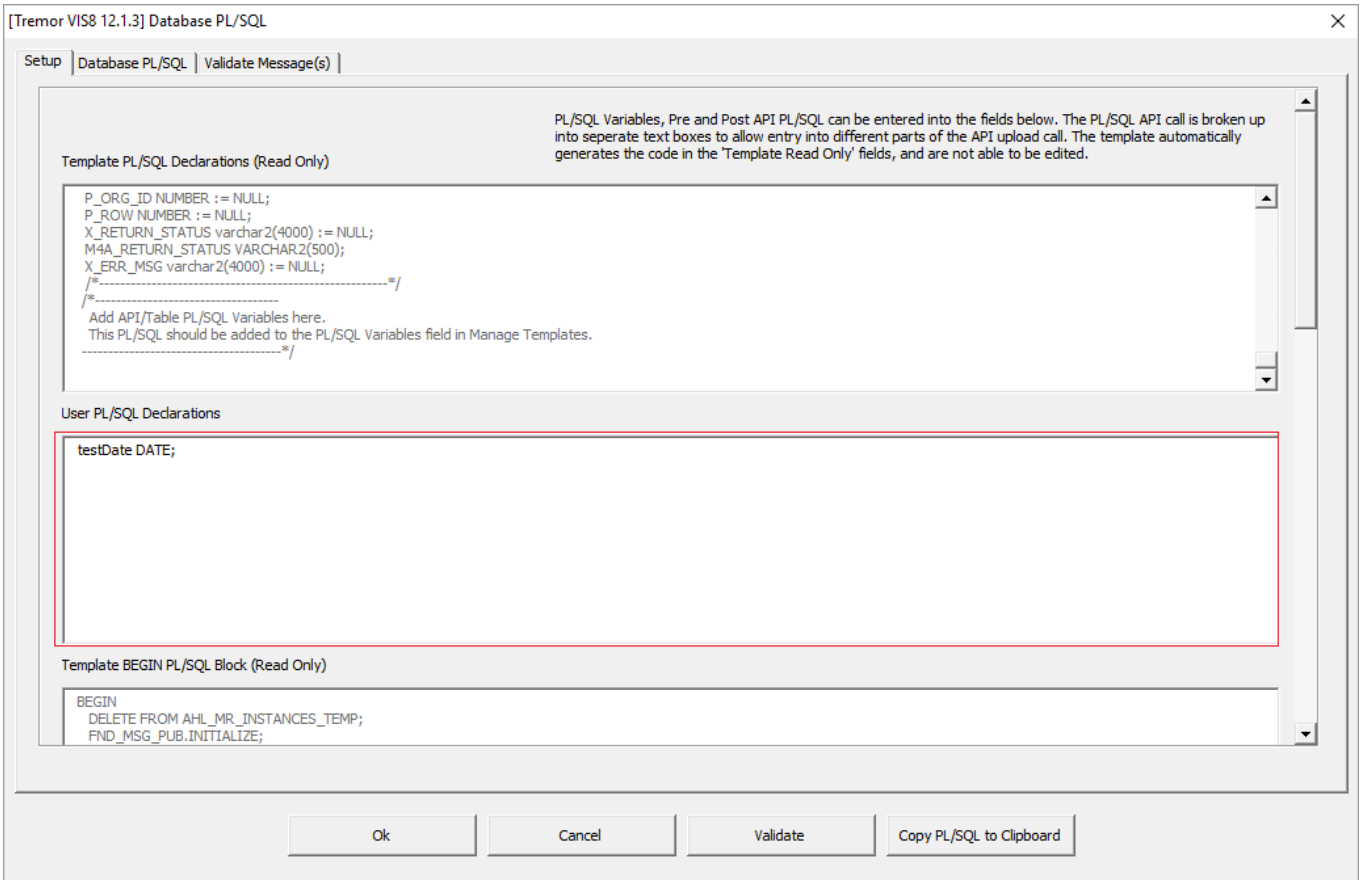
The following form will now be available:



This form contains read-only sections and user editable sections where you can add your own PL/SQL Declarations and pre and post API PL/SQL.

Define User PL/SQL Declarations

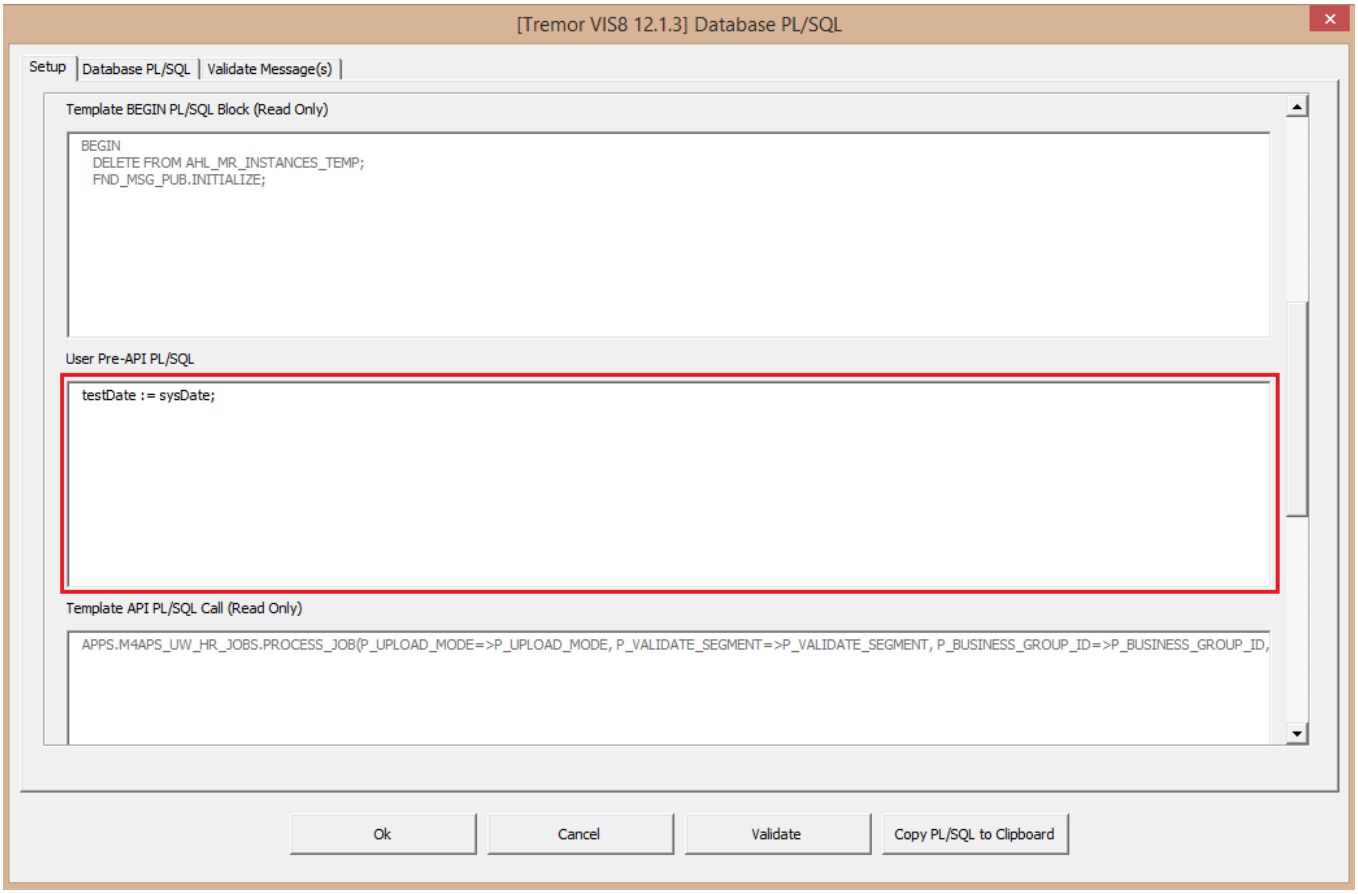
First define any variables you intend to use in your PL/SQL. These will be dynamically defined by the Wizard during the upload as additional variables for use:



Now type or paste your PL/SQL Variables into the pane using the correct PL/SQL syntax. Multiple variables should be separated by carriage returns.

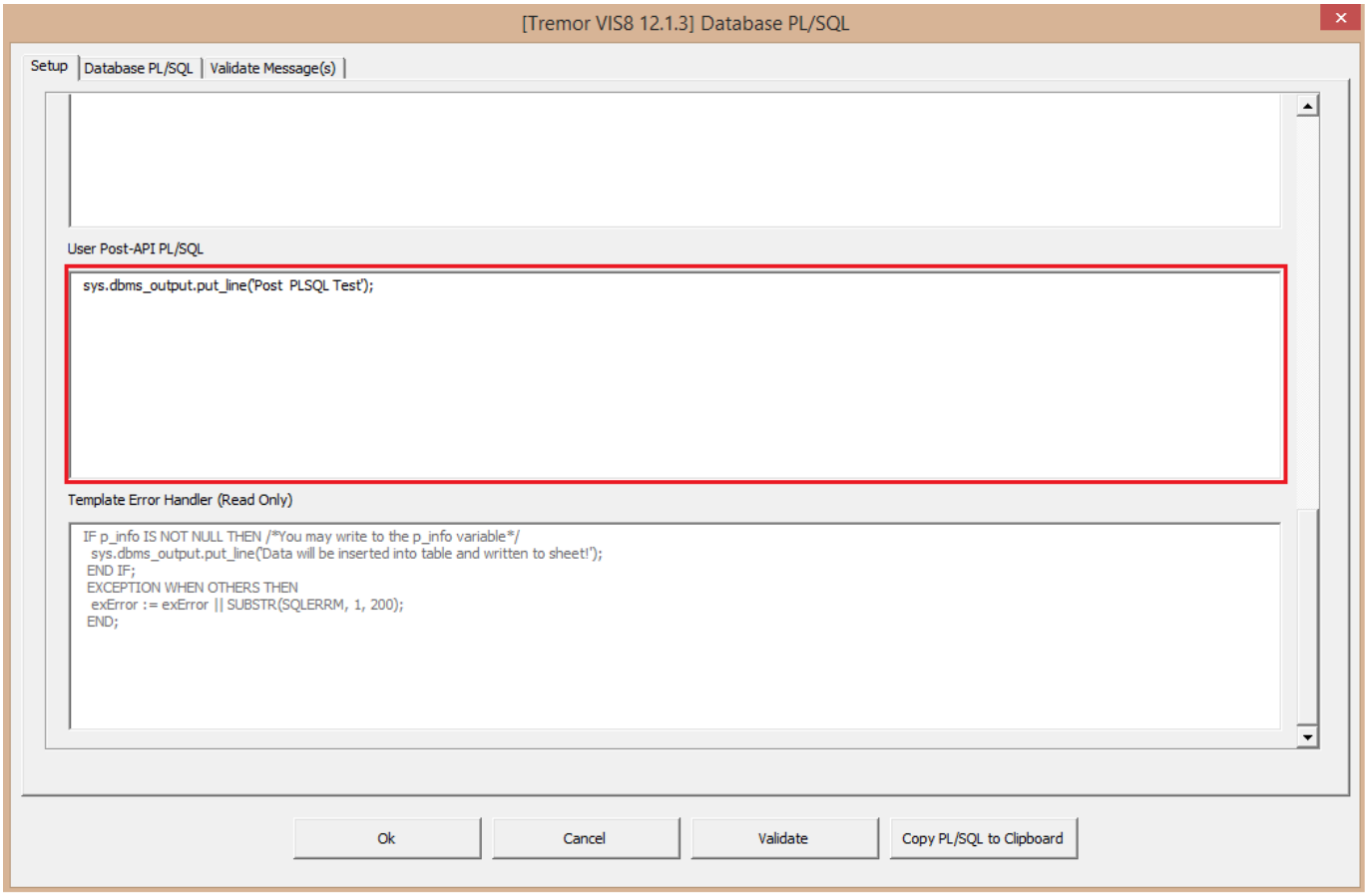
Define User Pre-API PL/SQL

Use the vertical scroll bar to move to the next editable section 'User Pre-API PL/SQL' and add any pre upload PL/SQL code you require the Wizard to undertake before calling the procedure:



Define User Post-API PL/SQL

Use the vertical scroll bar to move to the next editable section 'User Post-API PL/SQL' and add any post upload PL/SQL code you require the Wizard to undertake after calling the procedure:



When all variables have been added into the left-hand pane click 'Validate' to check the syntax and 'OK' when ready to close this form.

Note - These changes will not be saved to the template definition on the database until the 'Save To Repository' button in the main template form has been clicked.
P_info is an additional pre-configured parameter already available for use by the Wizard. In the pre and post Upload PL/SQL blocks you can return information into this pre-defined variable which will then be appended to the messages in the Header section automatically.

Tip - To add additional pre-validation into the template and have the record rejected before calling the upload API code, add code similar to the example below:

User PL/SQL Declarations
testdate DATE;

Pre-API PL/SQL
testdate := sysdate;

if p_trx_date < testdate then

x_err_msg := 'Transaction date cannot be prior to today';
x_return_status := 'E';

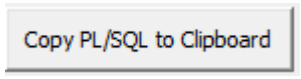
else

Post-API PL/SQL

end If;

Refer to section:
Copy PL/SQL to Clipboard
Erroring Child Sections

Copy PL/SQL to Clipboard

By clicking this 

button, the API Upload shell will be copied to clipboard. You can then paste this into SQL Developer/Toad and test the pre/post PL/SQL in your development application.

Refer to section:
Define Additional PL/SQL
Erroring Child Sections

Erroring Child Sections

On occasion when creating your own loader there may be scenarios where a 1 header to many child type relationship exists. This usually causes there to be two more or more sections in the loader, header and lines. When an upload is performed either the child section can be errored or the header can be errored or vice versa.

Below we demonstrate how to do this for both table based and procedure based loaders.

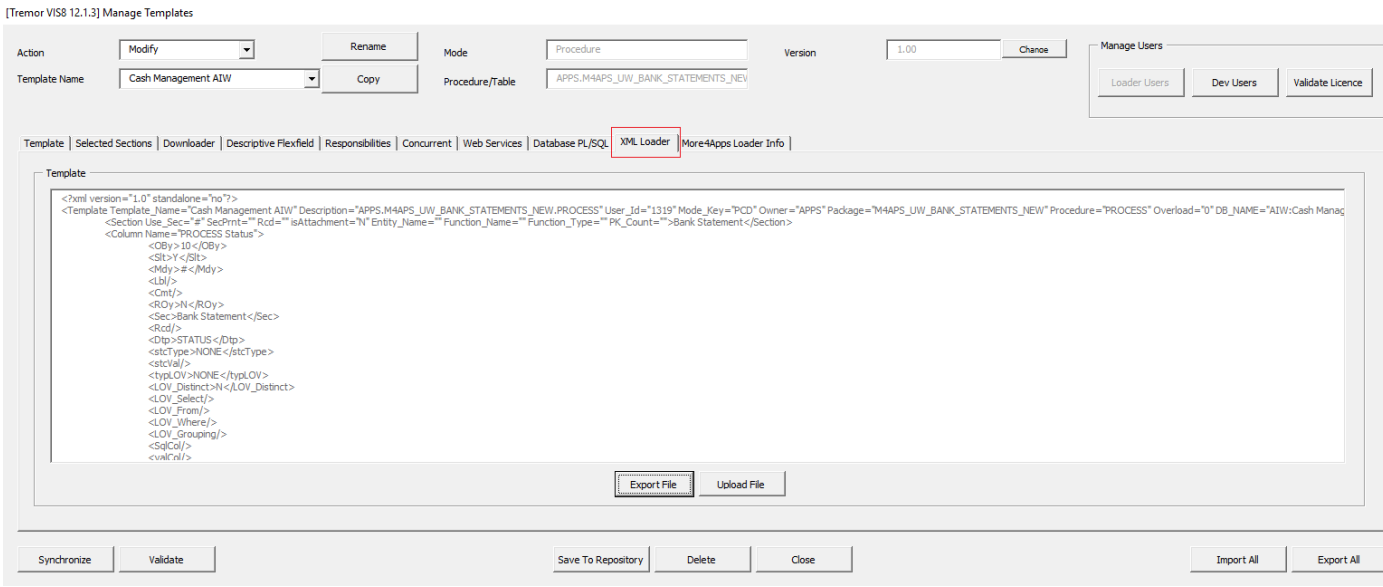
Refer to section:
Define Additional PL/SQL
Copy PL/SQL to Clipboard

XML Loader

This section displays the XML for the Template definition and provides various options to manage the definition across your EBS instances.

Select 'Modify' in the Action field, then the 'Template Name' in the 'Manage Templates' form.

Select the 'XML Loader' tab.



Refer to section:
Export File
Upload File

Export File

To create a XML file for the complete Template definition click the 'Export File' button.

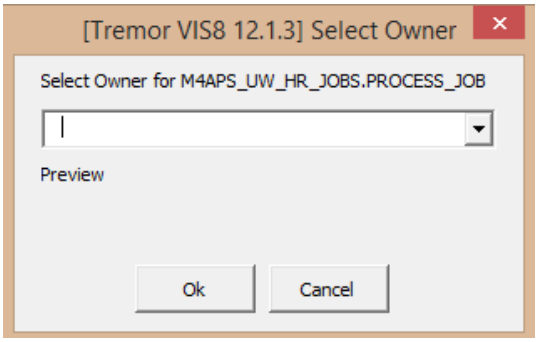
Select a directory to save the XML file to and rename the file if necessary.
Acknowledge the notification messages as appropriate.

Refer to section:
XML Loader
Upload File

Upload File

Upload Template definitions into your instance easily from an existing XML file by clicking the 'Upload File' button.

Browse to the appropriate directory and select the XML file.
Click 'OK' to upload the file.



Select the Owner of the procedure from the LOV and click 'OK'
Click 'Save To Repository' to save the template definition to the database.

Refer to section:
XML Loader
Export File

Wrapper Package Status Syntax

Advanced users of Application Interface Wizard may want to create their own wrapper packages in the Oracle Database. This may call various API's, have complex validation and a greater level of control. In this case when setting up the new template you would enter a mode of 'Procedure' and enter the custom package and procedure names.

There are some things to be aware of when creating your wrapper package and they are highlighted below.

If you are using your own wrapper packages the following parameters and return values are required in your procedure to return status and messages to Application Interface Wizard:

x_return_status	out	varchar2
x_err_msg	out	varchar2

Use this syntax in your procedures to report a successful upload:

```
x_return_status := FND_API.G_RET_STS_SUCCESS;
```

Use this syntax in your procedures to report an upload failure with an associated message:

```
x_return_status := FND_API.G_RET_STS_ERROR;
x_err_msg := 'HR Job Valid Grade assignment Failed - ' || sqlerrm;
```

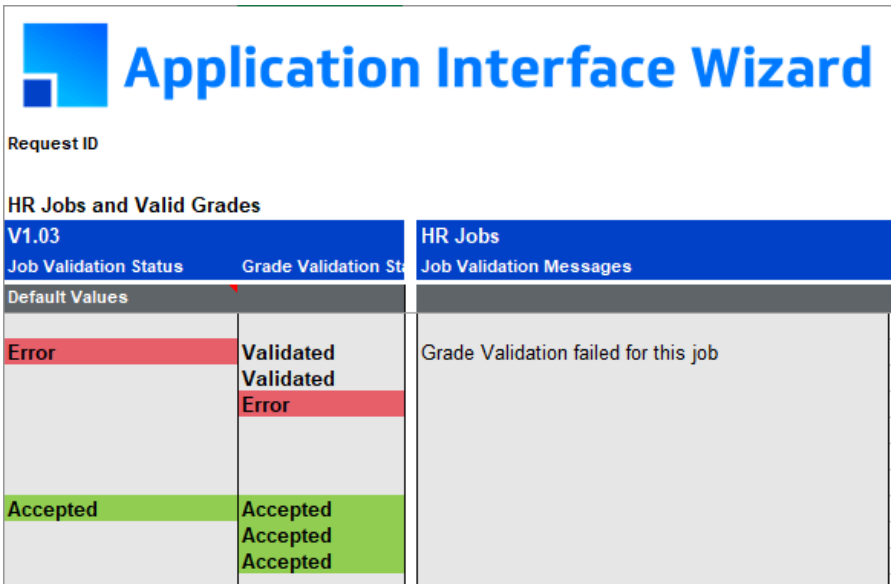
If you have parent/child relationships in your package then each table type must also have these columns to enable statuses and message to be feed back into the Wizard. In addition to this in the main procedure definition the table type parameter must be an in/out return type.

If X_RETURN_STATUS is displayed in the Template the above value of the API constant would be written to this column, for example 'S' or 'E'.

If this column is not displayed on the sheet the above status results are then resolved into Application Interface Wizard syntax and written to the mandatory section status column. For example 'E' would display 'Error' and 'S' would display 'Accepted'

If X_ERR_MSG is displayed in the Template it will contain the same value as the sections 'Messages' column.

Example Wizard Status and Messages Results:



Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Copy Template
Delete Template
Modify Template
Attachment Templates
Importing Templates
Downloader Template

Package Tips

This section displays the various options available for the Package Specification and Package Body.

Refer to section:
Package Specification
Package Body

Package Specification

AUTH_ID:
The *authid current_user* is used when you want a piece of code (PL/SQL) to execute with the privileges of the current user, and NOT the user ID that created the procedure. This is used mainly for BOLINF purposes - when this is installed under BOLINF but run as APPS.

COMMENTS:
The package spec should be formatted as below with revisions and comments

HEADER_VERSION:
The header version should match the specification date, and the version number and be updated for any code changes to be released. The Header Version will appear in the about section in the AIW spreadsheet.

PROCEDURE VERSION:
The procedure version is required for the about section in the AIW spreadsheet.

PROCEDURE UPLOAD:
This is the main procedure that will be called by the loader.

Any columns to be included in the parent section can be called directly as parameters
Any columns to be included in the child sections are included as a table type.
x_return_status - will return Success or Failure from the package body for parent section
x_err_msg - will return the error message for the parent section

Record type:
This is a record type containing all the child columns. The columns that are required are:
x_err_msg - returns an error message for the child section
x_return_status - returns Success or Failure for the child section
p_row_num - required field - keeps track of the child row for AIW

```
create or replace
PACKAGE M4APS_AIW_XYZ authid current_user AS
/*************************************************************************
   NAME:       M4APS_AIW_XYZ
   PURPOSE:    Put the purpose here

   REVISIONS:
   Ver    Date         Author           Description
```

```
   -----  -----------  ---------------  ------------------------------------
   1.0.0  31 Aug 2017  S Smith          Created
*************************************************************************/

   header_version  VARCHAR2 (20) := '1.0.0 31-Aug-17';

   -- Return the header and body version in XML format
   procedure version (p_header_version   out   varchar2,
                      p_body_version     out   varchar2);

   type asset_lines_rec is record
   (
    project_asset_line_id   number,
    asset_id                number,
    amount                  number,
    percentage              number,
    x_err_msg               varchar2(4000) ,
    x_return_status         varchar2(1),
    p_row                   number
   );

   TYPE asset_lines_tbl_type IS TABLE OF asset_lines_rec index by binary_integer;

    PROCEDURE UPLOAD (
         p_project_id               in     number,
         p_asset_lines              in out asset_lines_tbl_type,
         x_return_status            out    varchar2,
         x_err_msg                  out    varchar2
   ) ;

end M4APS_AIW_XYZ;
```

Refer to section:
Package Tips
Package Body

Package Body

COMMENTS:
The package body should be formatted as below with revisions and comments:

BODY_VERSION:
The body version should match the body date and version number and be updated for any code changes to be released. The Body Version will appear in the about section in the AIW spreadsheet.

PROCEDURE VERSION:
The procedure version is required for the about section in the AIW spreadsheet.

PROCEDURE UPLOAD:
This is the main procedure that will be called by the loader.
Sample header/parent error and success code has been provided.
Sample line/child error and success code has been provided.

CHILD LOOP:
Sample code for looping and processing child rows has been provided.

EXCEPTION HANDLER:
Package body should always include an exception handler. A sample WHEN OTHERS exception handler has been provided.

```
create or replace
PACKAGE BODY M4APS_AIW_XYZ AS
/******************************************************************************
   NAME:       M4APS_AIW_XYZ
   PURPOSE:    Put the purpose here

   REVISIONS:
   Ver    Date         Author           Description
   -----  -----------  ---------------  ------------------------------------
   1.0.0  31 Aug 2017  S Smith          Created
******************************************************************************/

body_version     varchar2 (20) := '1.0.2 07-Feb-18';

PROCEDURE VERSION(
    p_header_version OUT VARCHAR2,
    p_body_version OUT VARCHAR2 )
IS
BEGIN
  p_header_version := header_version;
  p_body_version   := body_version;
END VERSION;

PROCEDURE UPLOAD (
         p_project_id               in     number,
         p_asset_lines              in out asset_lines_tbl_type,
         x_return_status            out    varchar2,
         x_err_msg                  out    varchar2
   )  AS

BEGIN
  --SUCCESS
  x_return_status := fnd_api.g_ret_sts_success ;
  --ERROR
  x_return_status := fnd_api.g_ret_sts_error ;
  x_err_msg       := 'This project must be set up to use Client Extension in the Asset
Cost Allocation Method';

  for i in 1..p_asset_lines.count loop
    --SUCCESS
    p_asset_lines(i).x_return_status := fnd_api.g_ret_sts_success;
    --ERROR
    p_asset_lines(i).x_return_status := fnd_api.g_ret_sts_error;
    p_asset_lines(i).x_err_msg       := p_asset_lines(i).percentage || ' percent of ' ||
l_current_asset_cost || ' should be ' ||
  end loop;

exception
  when others then
    x_err_msg := 'Unknown error in M4APS_AIW_PA_UNASSIGNED_ASSET.upload - '||
dbms_utility.format_error_backtrace;
    dbms_output.put_line(x_err_msg);
    dbms_output.put_line(dbms_utility.format_error_stack);
    dbms_output.put_line(dbms_utility.format_error_backtrace);
    x_return_status := FND_API.G_RET_STS_ERROR;
END UPLOAD;

END M4APS_AIW_XYZ;
```

Refer to section:
Package Tips

Package Specification

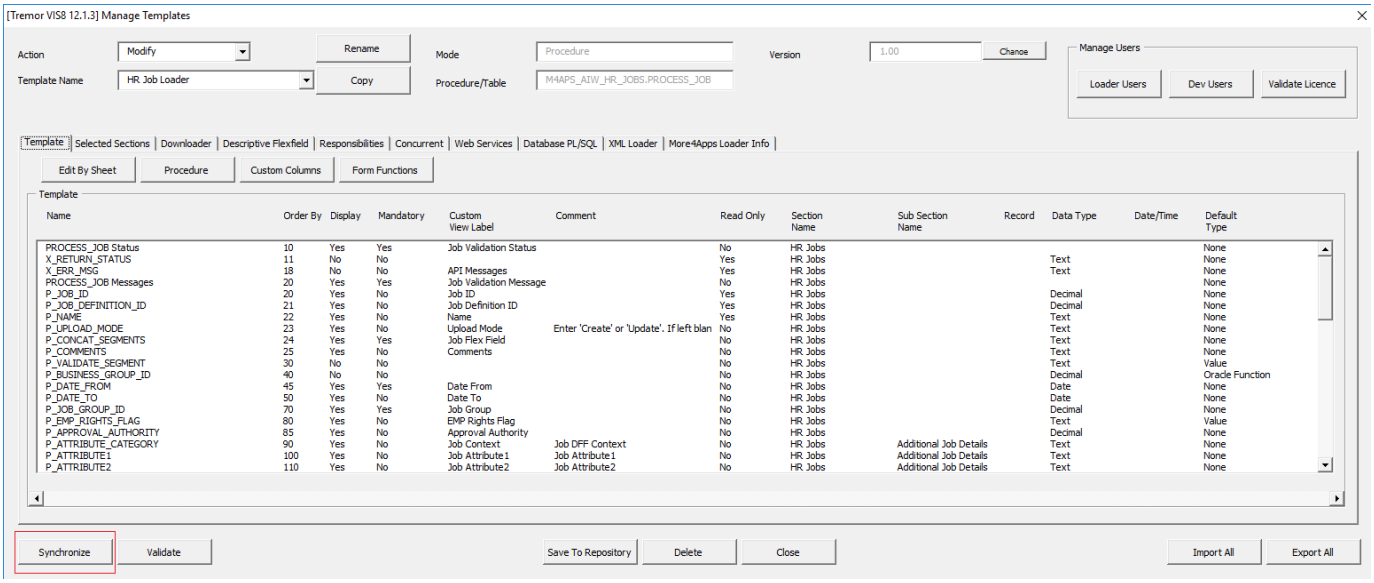
Synchronize Template

This functionality allows any changes to the database procedure (since the Template was created) to be seamlessly incorporated into the Template definition.

An example of this could be new parameters/columns added to the database object that do not exist in the Template yet.

Select 'Modify' in the Action field, then the 'Template Name' in the 'Manage Templates' form.

Click the 'Synchronize' button. The additional attributes will be migrated to this particular template.



Note - Any existing Template configuration will remain. The new columns may require additional configuration to be consistent with the current Template design.

Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Validate Template
Export All

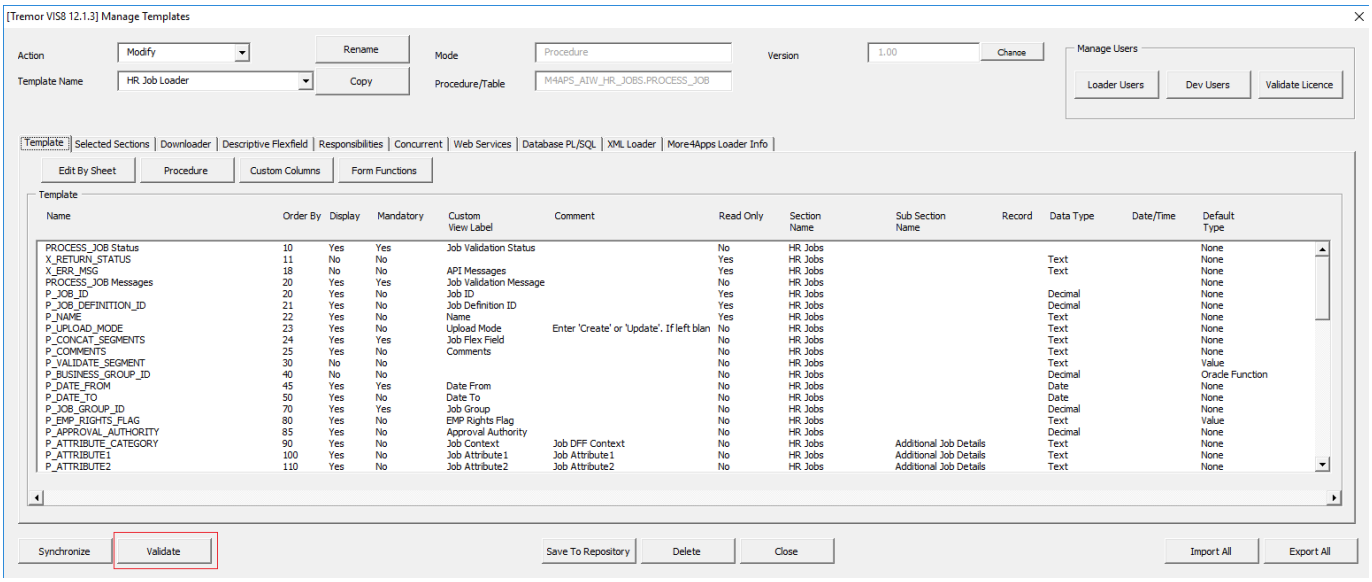
[Update Procedure Name](#)
[Copy Template](#)
[Delete Template](#)
[Modify Template](#)
[Attachment Templates](#)
[Importing Templates](#)
[Downloader Template](#)

Validate Template

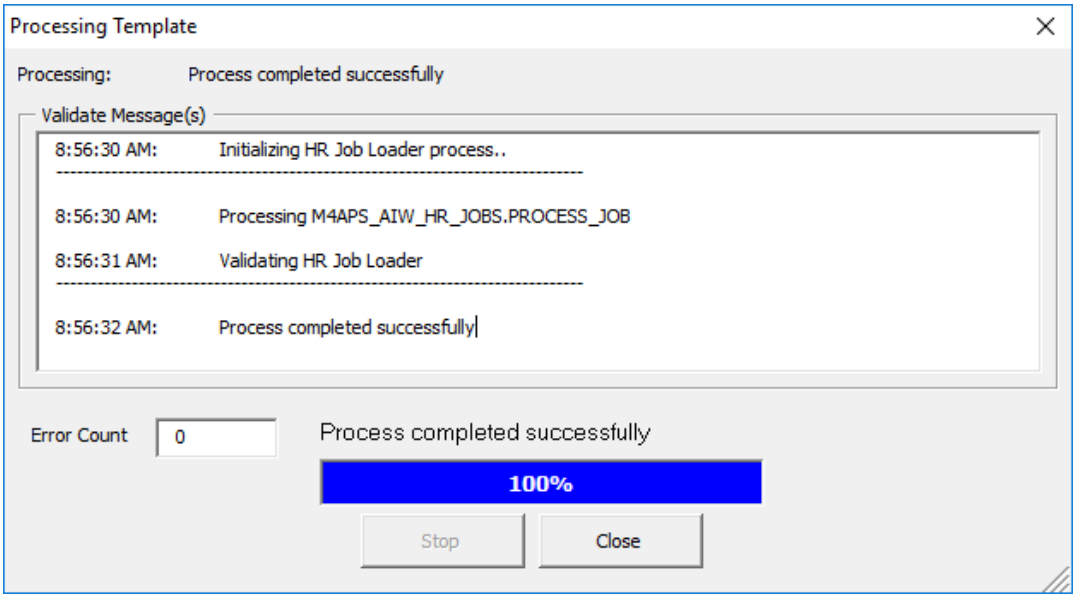
After Synchronization it's a good idea to validate the template to ensure all the dependencies are intact.

Select ‘Modify’ in the Action field, then the ‘Template Name’ in the ‘Manage Templates’ form.

Click the 'Validate' button to begin the process.



Review the Processing Template for exceptions and resolve them until Validation completes successfully.

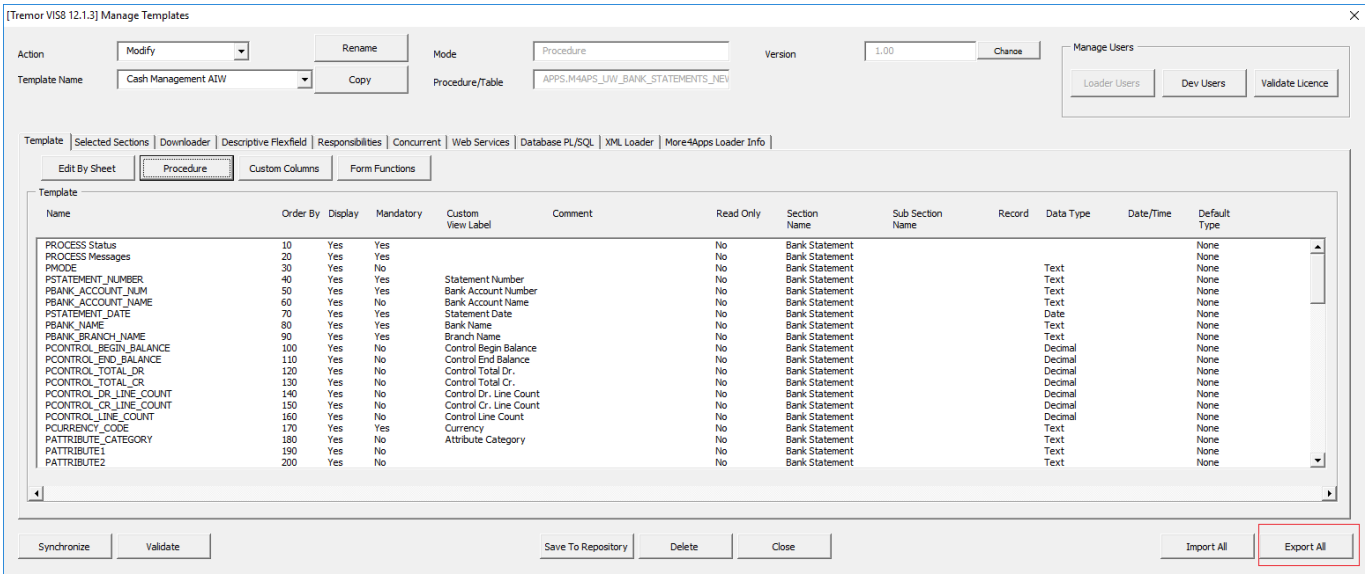


Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Export All
Update Procedure Name
Copy Template
Delete Template
Modify Template
Attachment Templates
Importing Templates
Downloader Template

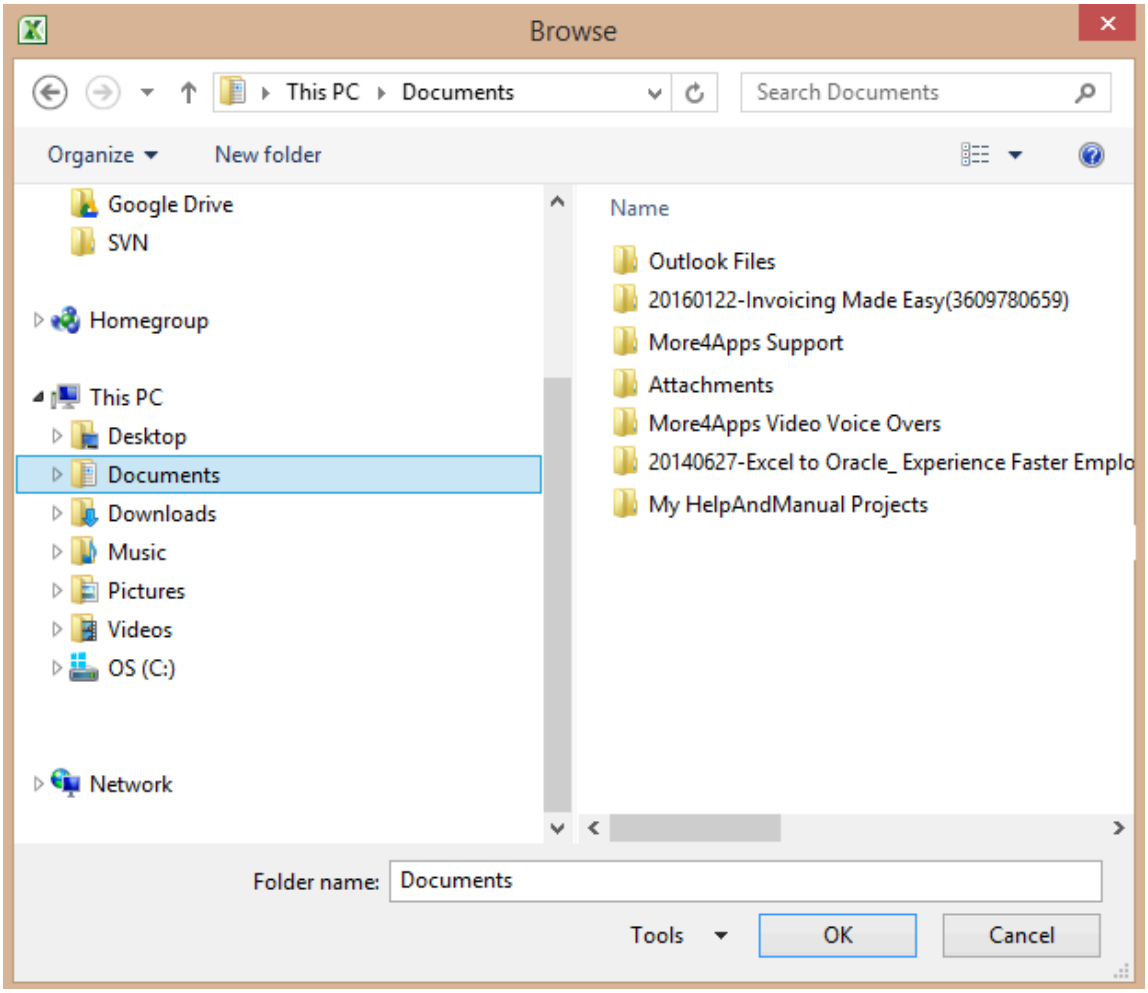
Export All

This functionality allows you to export all templates to individual XML files in a nominated directory.

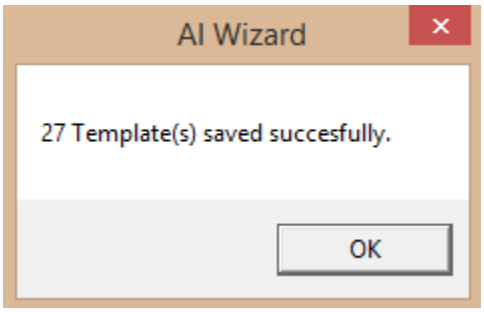
To create XML files for all Templates in the current instance click the 'Export All' button:



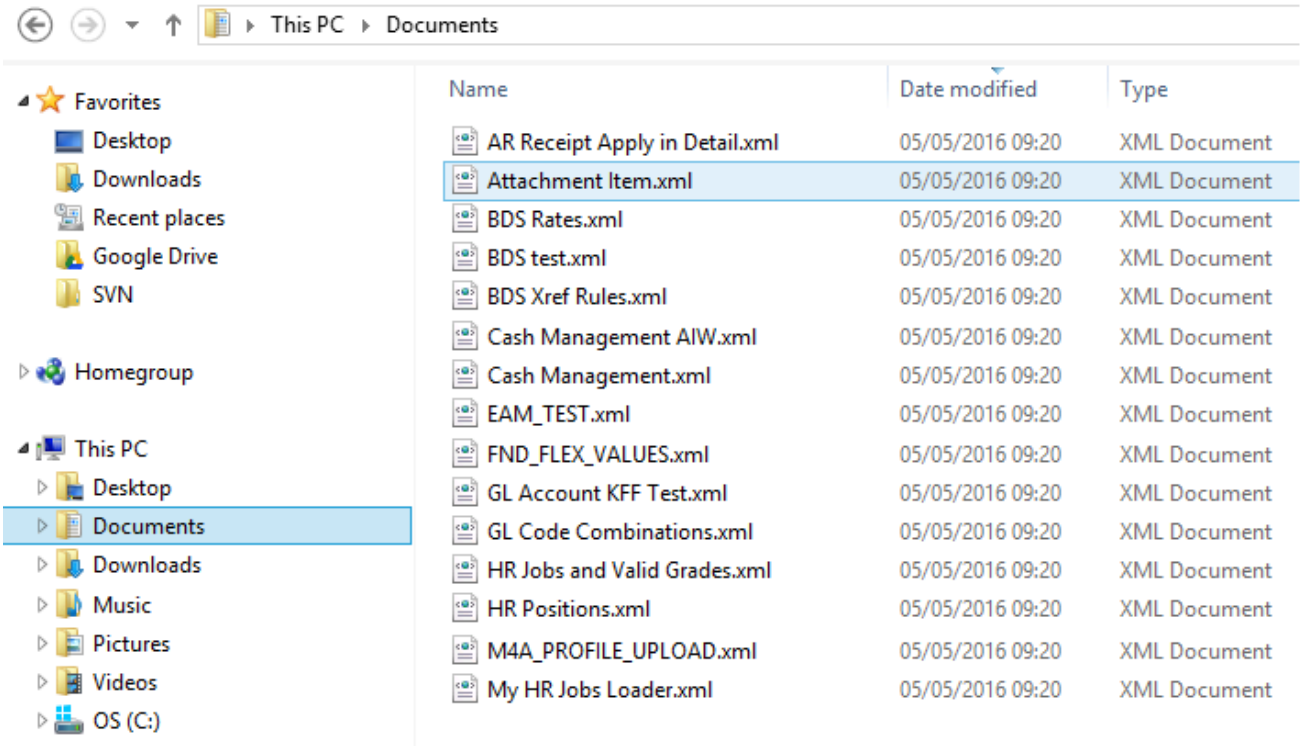
This will open a dialog box for you to select the appropriate folder to save the files to:



Select the appropriate directory to save the files to and click 'OK'.



The XML files will be saved to the selected directory:



Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template

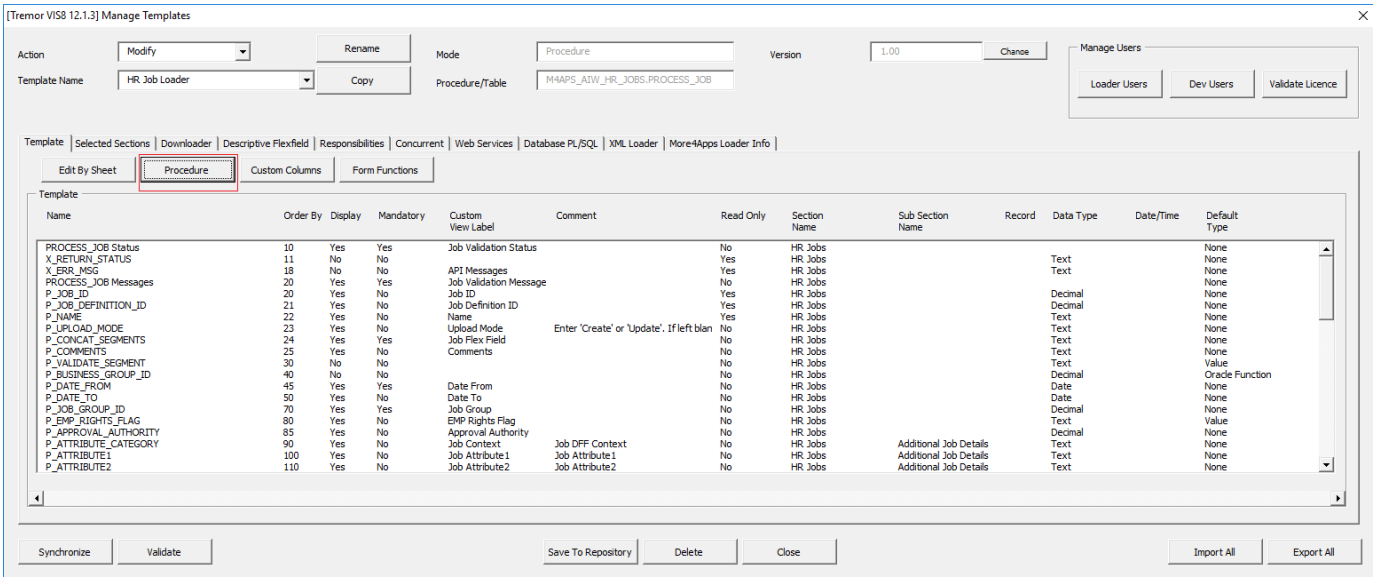
[Update Procedure Name](#)
[Copy Template](#)
[Delete Template](#)
[Modify Template](#)
[Attachment Templates](#)
[Importing Templates](#)
[Downloader Template](#)

Update Procedure Name

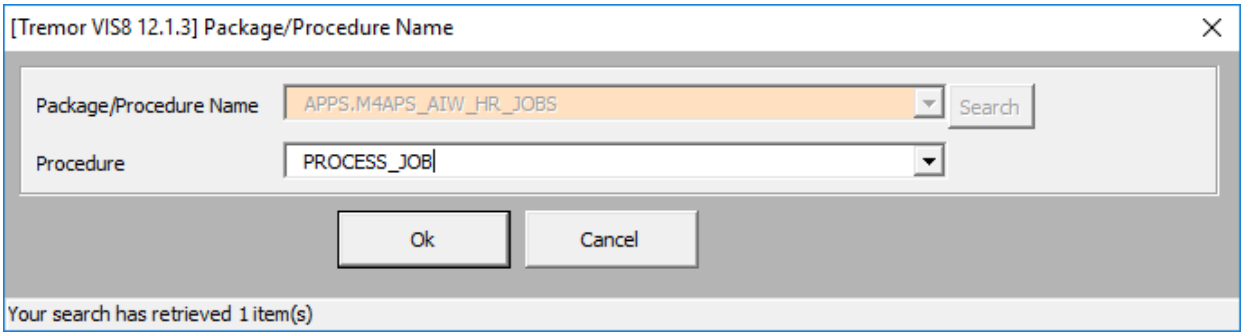
During your design phase you may need to update the procedure name associated with the Template. A situation may arise during your development phase, for example, where it becomes simpler to create a new database procedure as a copy of the current procedure and then make a few changes.

If this occurs, you can simply change the name of the procedure your Template is associated with and then SYNCHRONIZE the existing Template to the new procedure.

To change the Template's associated procedure, click the 'Procedure' button:



Select the new procedure from the dropdown list. Note the Package Name cannot be changed.



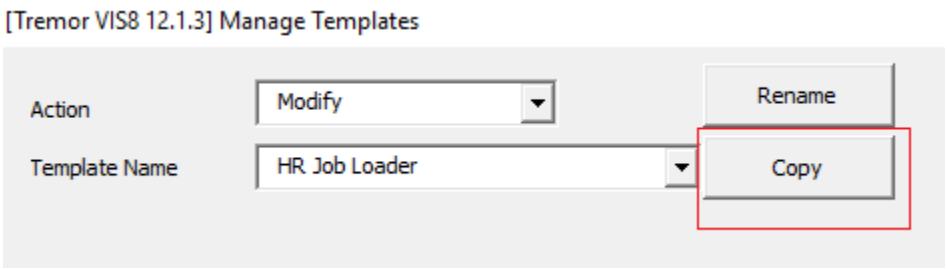
Then to ensure your Template matches the new procedure click the 'Synchronize' button on the 'Manage Templates' form.

Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Copy Template
Delete Template
Modify Template
Attachment Templates
Importing Templates
Downloader Template

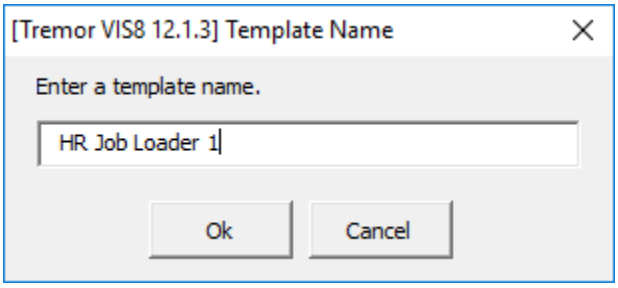
Copy Template

The 'Copy' functionality allows an existing Template to be copied and given a new name. The new Template is then saved to the database.

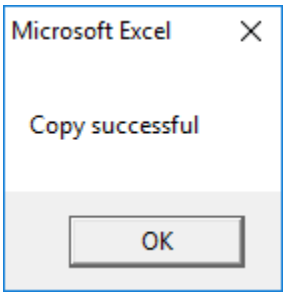
For 'Action' select 'Modify' and the 'Template Name':



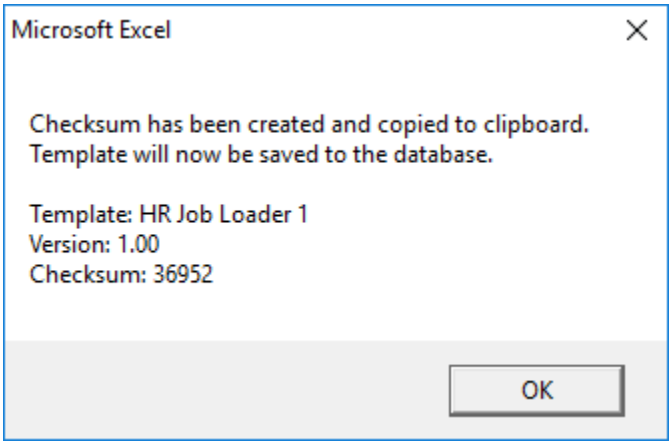
Click ‘Copy’ and enter a unique template name:

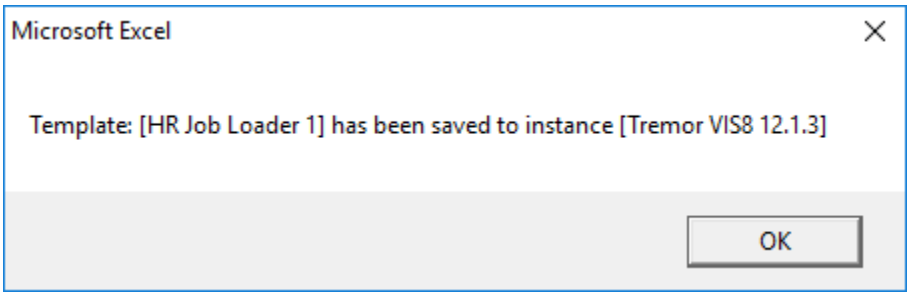


Click ‘OK’ to copy the selected Template attributes to the new Template Name and the ‘Copy successful’ message will display:



Click ‘OK’ and then click ‘Save To Repository’ to save the changes to the database. The following messages will display to confirm.



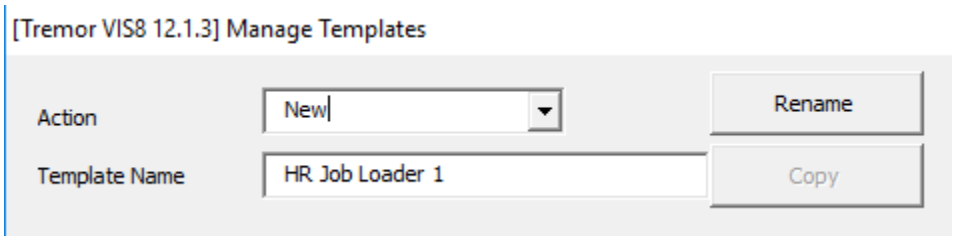


Refer to section:
Create ‘New’ Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Delete Template
Modify Template
Attachment Templates
Importing Templates
Downloader Template

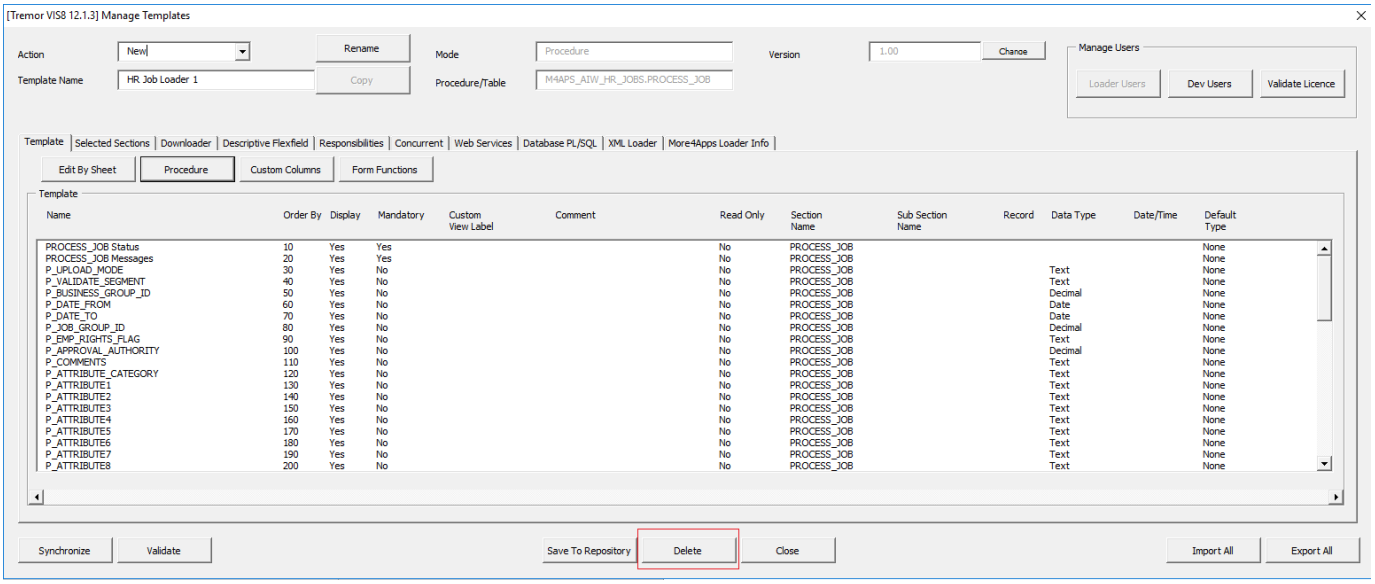
Delete Template

‘Manage Templates’ has the option to delete any obsolete Templates from the database.

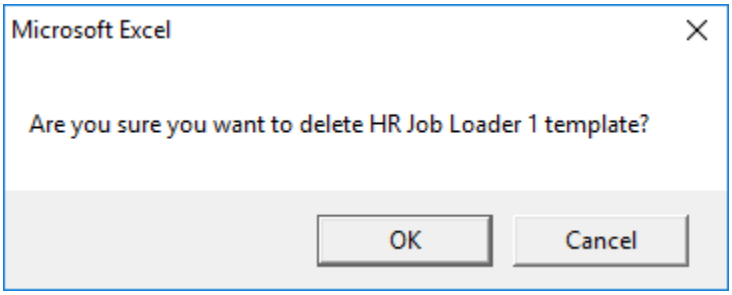
Select the ‘Template Name’ to delete.



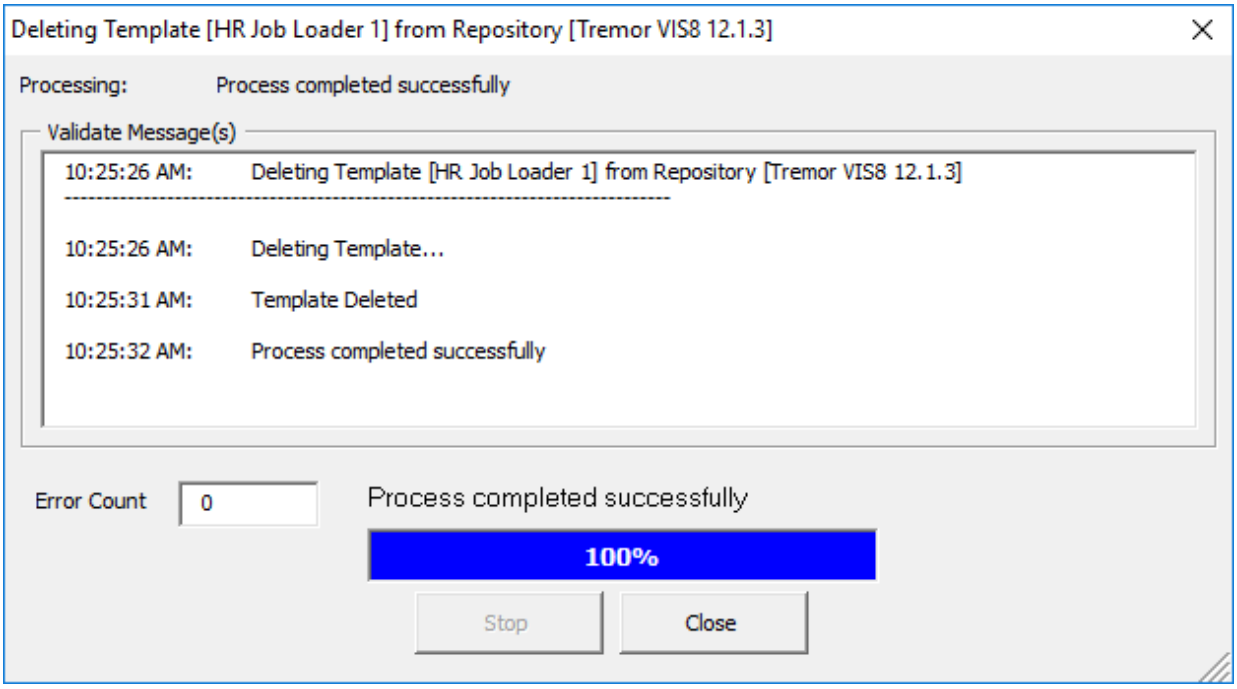
Click the ‘Delete’ button on the bottom of the form to delete the selected template.



Respond to the popup message by clicking ‘OK’ if you are sure you want to delete the selected Template.



The Template has been deleted.



The Template is no longer in the list of values and has been removed from the Oracle database.

Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Copy Template
Modify Template
Attachment Templates
Importing Templates
Downloader Template

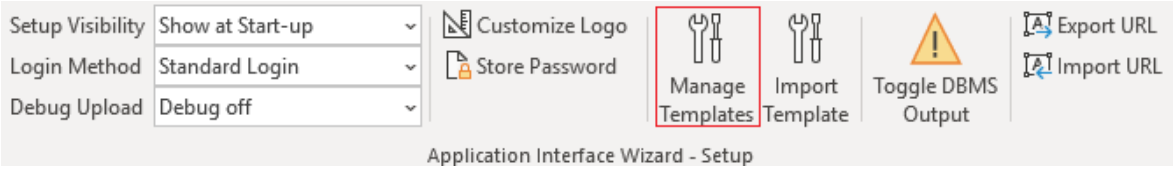
Modify Template

Template functionality provides users with System Administrator access to create their own data loaders and save them to the database. For security purposes, they can be

assigned to responsibilities for even greater control or be available across responsibilities related to the Wizard application.

The user defined templates are saved in the database and available to transfer to other instances if required.

The 'Manage Templates' icon on the ribbon provides options to create new templates, modify attributes of existing templates, or delete obsolete templates.



Refer to section:

Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Copy Template
Delete Template
Attachment Templates
Importing Templates
Downloader Template

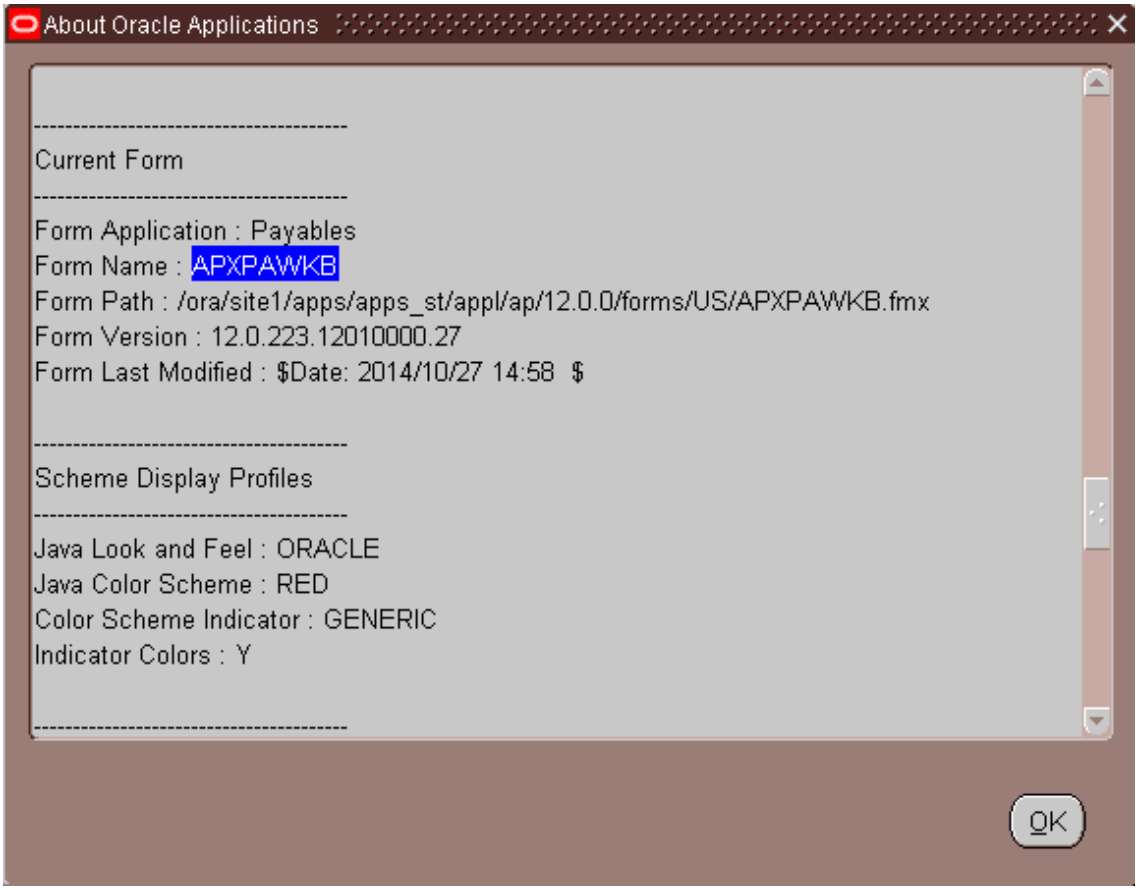
Attachment Templates

Templates can also be defined to upload attachments into Oracle EBS. Before creating these the Attachment function information needs to be identified, as it's required during the Template configuration.

Deriving Name for the required Attachment

First determine the Oracle EBS Form Name.
Open the Oracle Form where you want the Attachment to be uploaded and select 'Help | About Oracle Applications' from the menu.

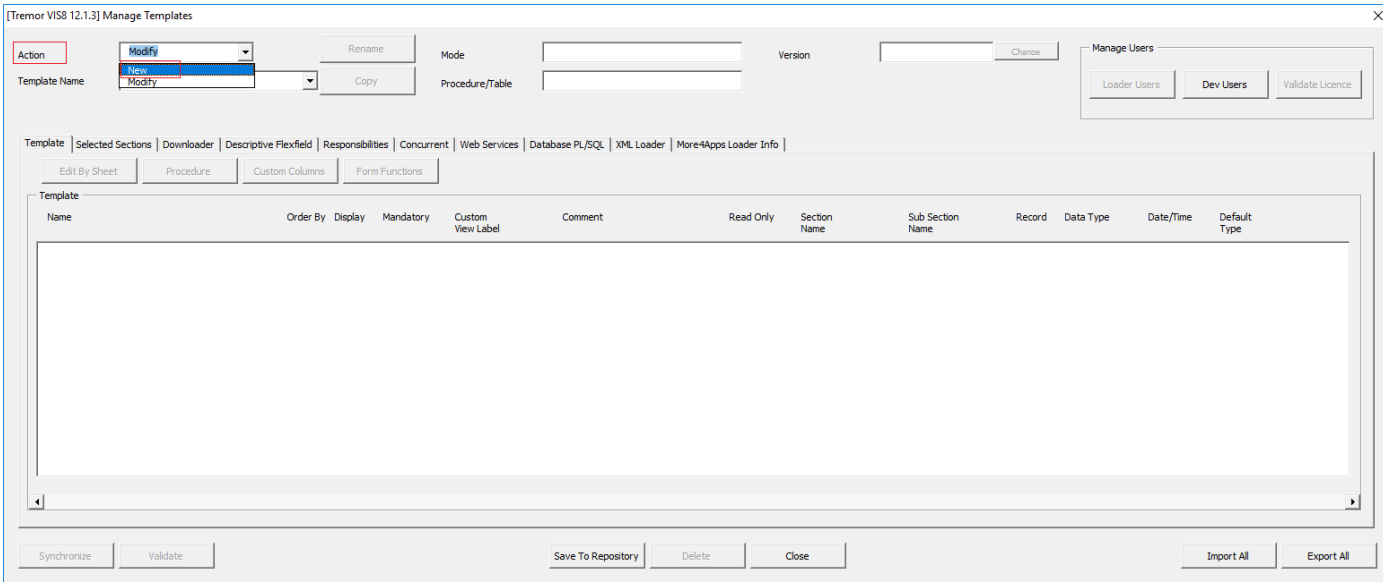
Scroll down the form until the 'Current Form' section is in view and copy the 'Form Name' value.



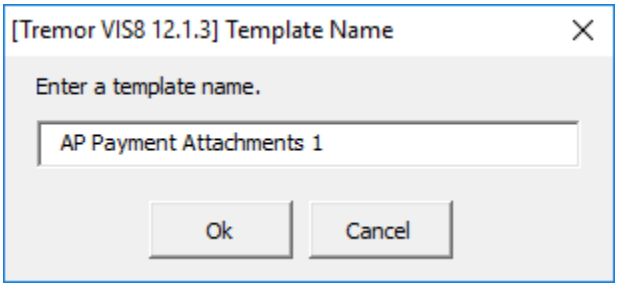
In this example we will be creating a template to load Attachments to Oracle Payables Payments.

Form Name: APXPAWKB

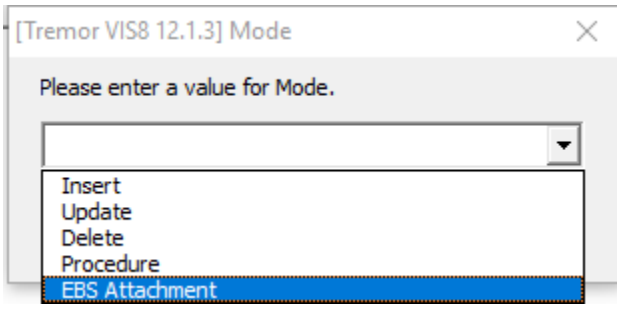
Select ‘New’ from the ‘Action’ list of values.



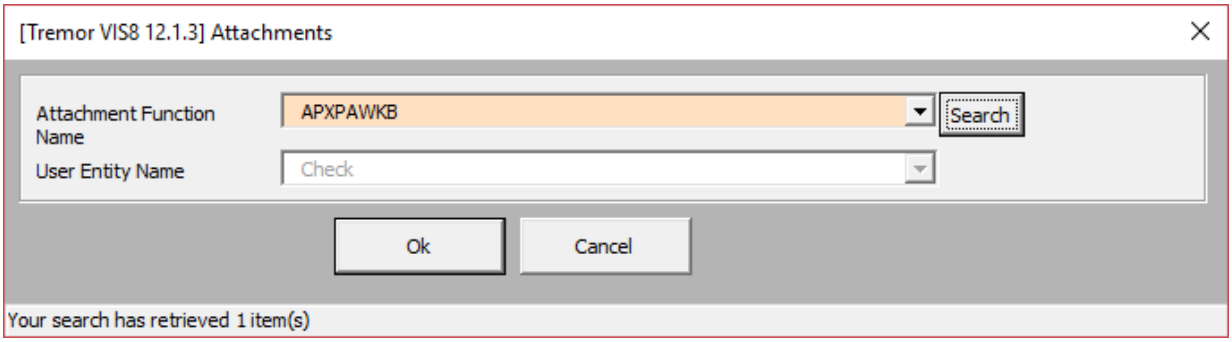
Enter a unique Template Name and click ‘Ok’.



Select the EBS Attachment Mode and click ‘Ok’.

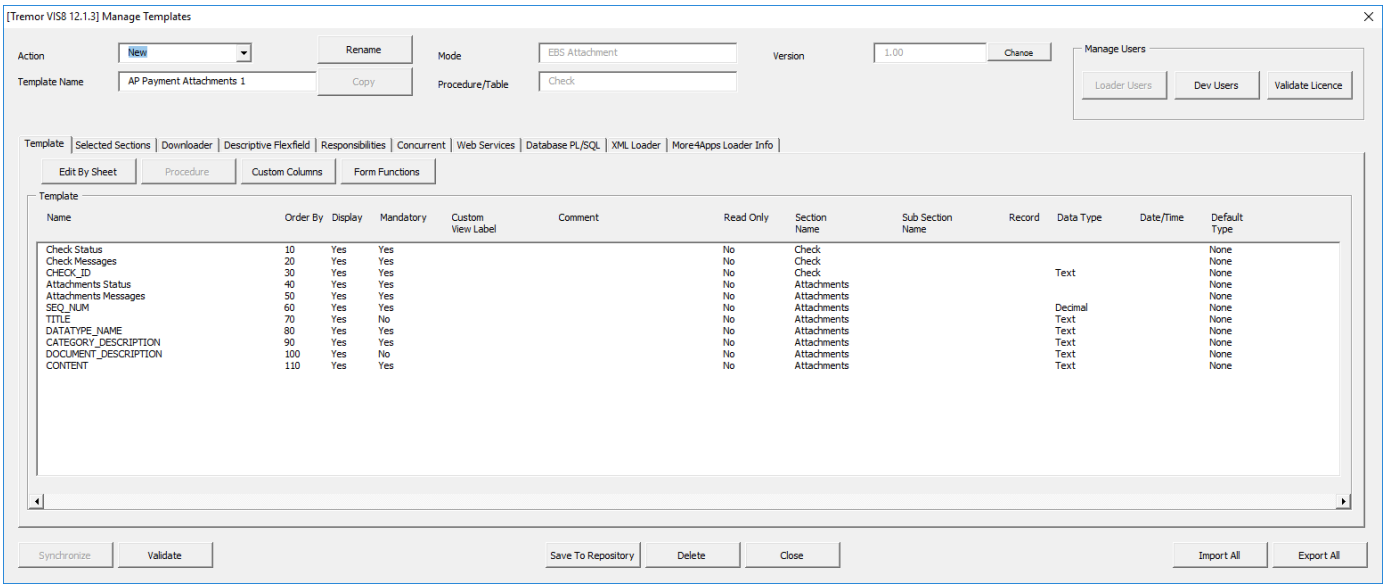


Search for and select the Attachment Function Name.



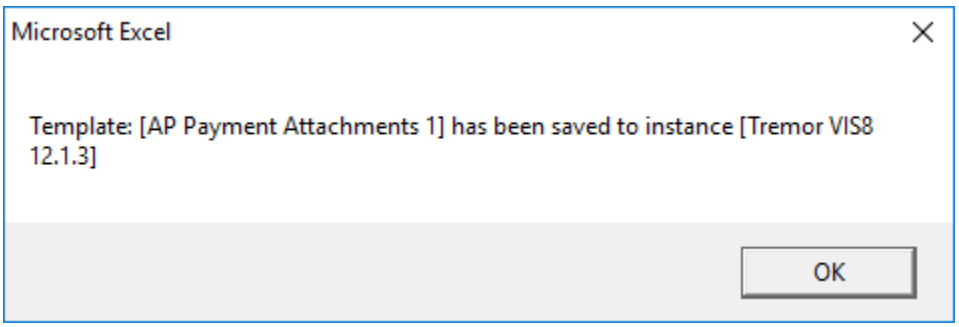
Select from the drop-down list if multiple values exist.

Click 'Ok' to create the Template.



As in other Templates, System Administrators can also configure the layouts and forms as necessary.

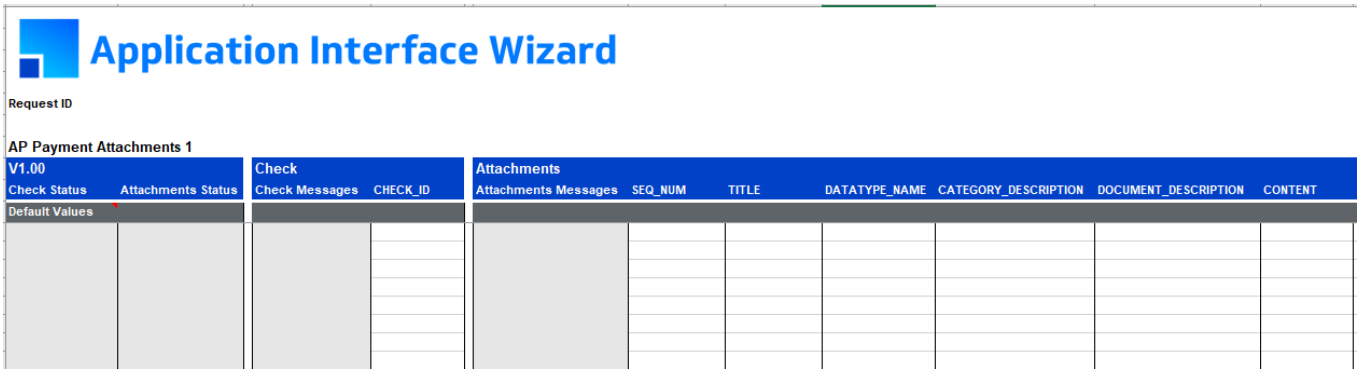
Click 'Save To Repository' to save the Template to the Database when all changes have been made.



Click ‘OK’.

Click 'Close' to exit the 'Manage Templates' form.

In the Main ribbon select your Attachments Template and create a new worksheet.



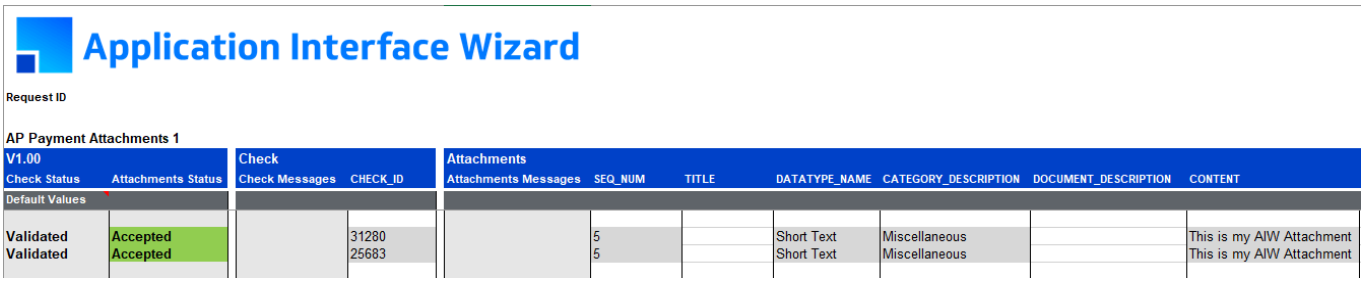
Note - The two sections where the source document exists separately to the 'Attachments' section. Both sections have data entry forms, accessed by double-clicking on any cell below row 10.

In this example the CHECK_ID value has been entered fron the Oracle AP Payments and Attachments uploaded against these using the 'Upload' buttons.

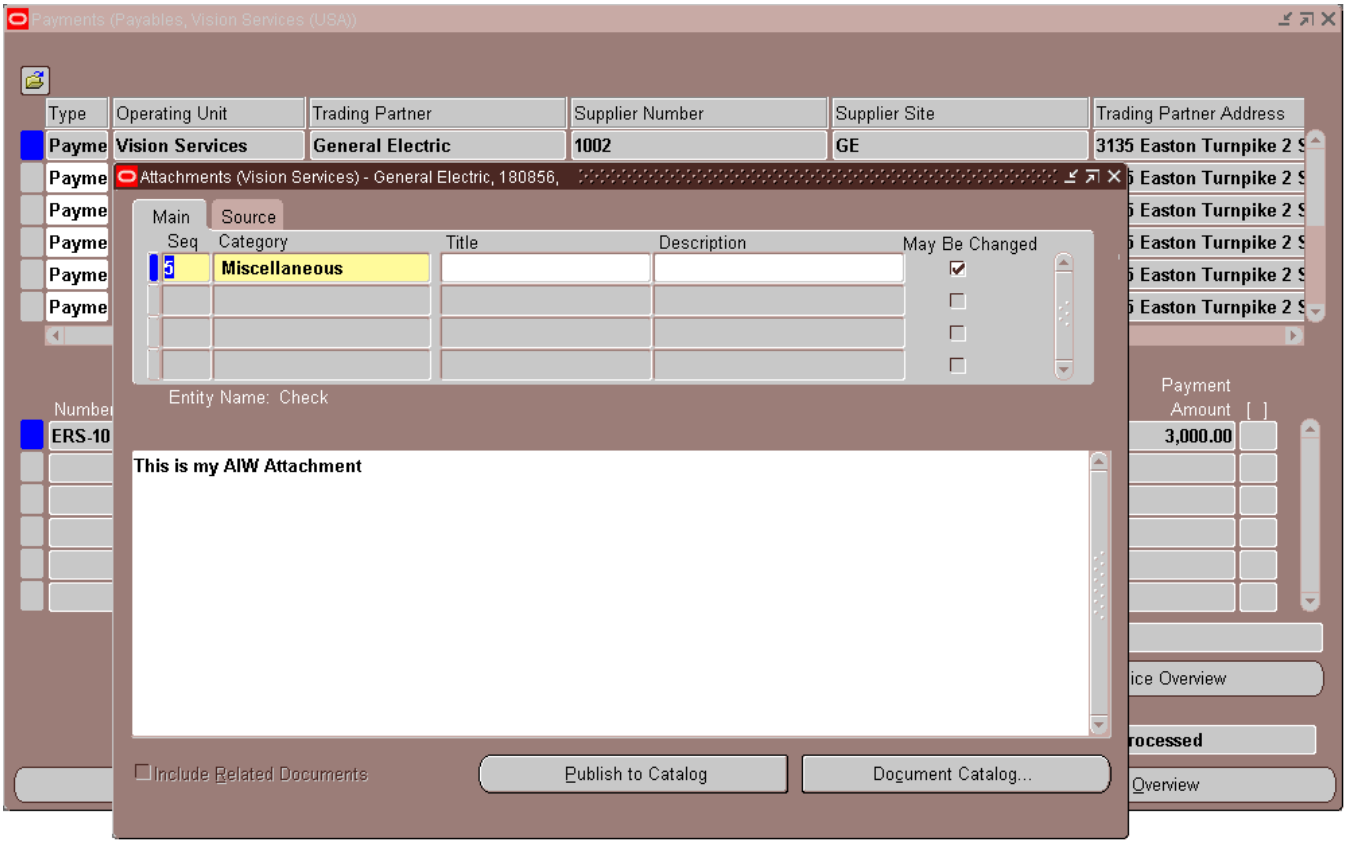
Select the rows for upload and click on the appropriate ‘Upload’ button on the ribbon. The Wizard then processes the rows and displays the results back to the sheet in the 'Attachment Status' and 'Attachment Messages' columns.

Double-click on the Attachment row to open the form and make changes to the individual records. Supported documents like webpage and file provided in the ‘Content’ box can be viewed by clicking the ‘View Document’ button. After changes have been completed the user can upload using the ‘Upload Now’ button on the form.

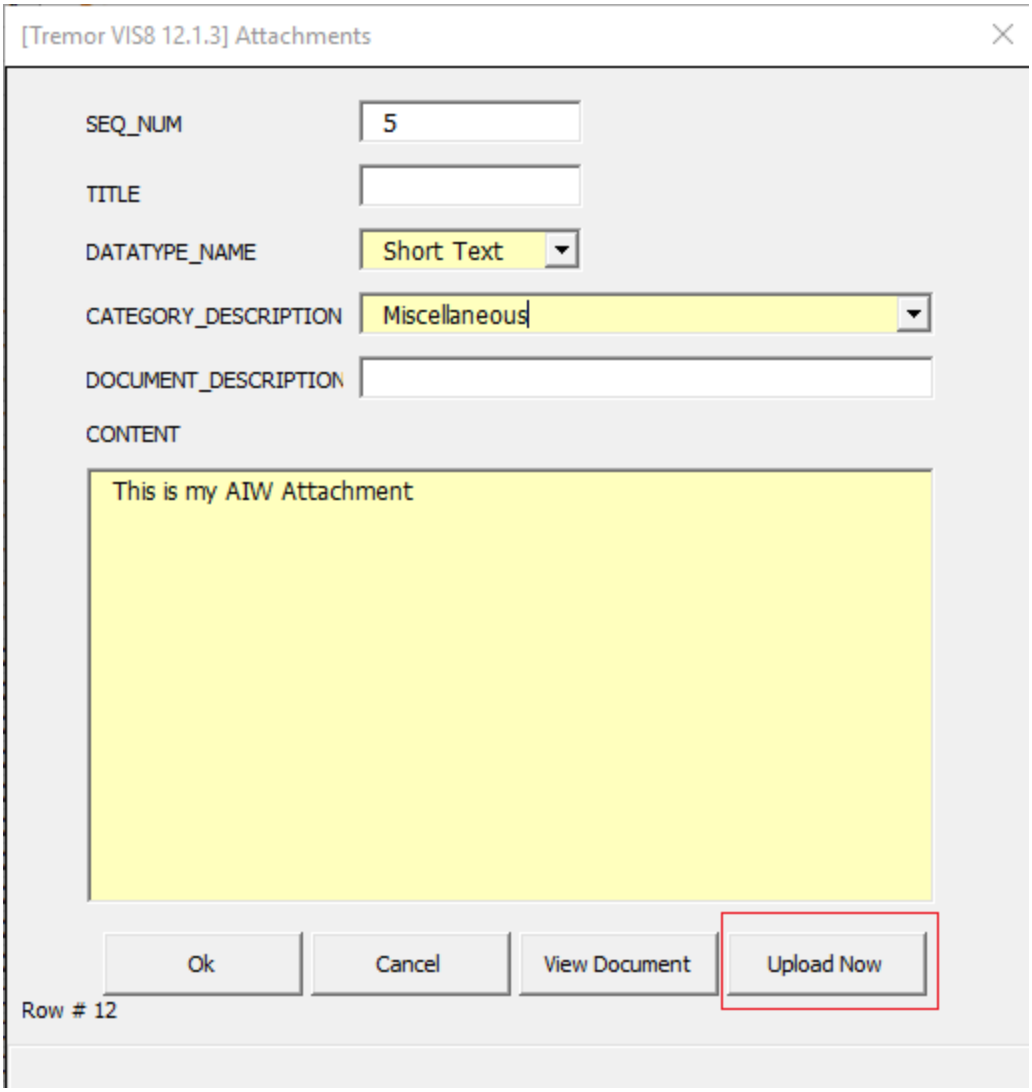
Example of successful Attachment uploads:



The resulting Attachment now exists against the AP Payment record.



Note - Attachments can also be uploaded directly clicking the 'Upload Now' button on the ‘Attachments’ data entry form:



Refer to section:

[Create 'New' Template]
[Template Versioning]
[Customize the Look and Feel of the Template]
[Selected Sections]
[Customize Logo]
[Descriptive Flexfields (DFF's)]
[Secure Template to EBS Responsibilities]
[Database PL/SQL]
[XML Loader]
[Wrapper Package Status Syntax]
[Package Tips]
[Synchronize Template]
[Validate Template]
[Export All]
[Update Procedure Name]
[Copy Template]
[Delete Template]

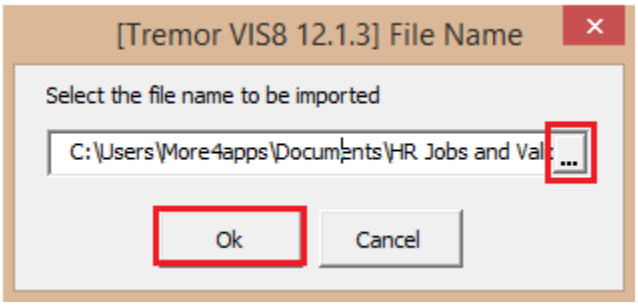
[Modify Template](#)
[Importing Templates](#)
[Downloader Template](#)

Importing Templates

When More4apps sends you either new or updated templates, save the XML files to a directory on your PC or network, ready for later import.

To import the template definition into your database, click the 'Import Template' button in the Application Interface Wizard Setup ribbon:

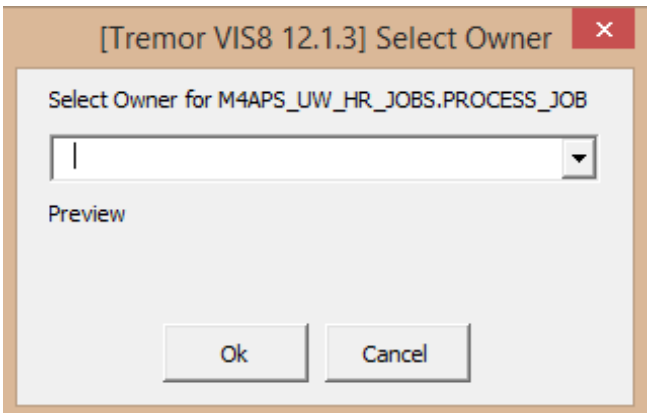
Browse to the appropriate directory and select the XML file.

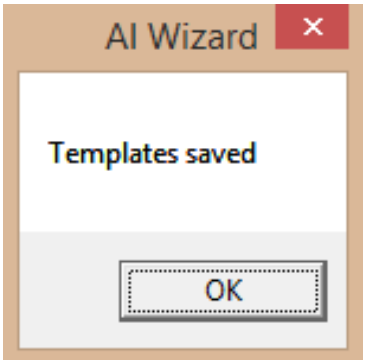


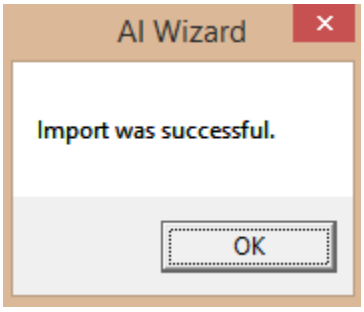
Click 'Ok' to upload the file.

Select the Owner of the procedure from the LOV and click 'Ok'.

Click 'Ok' to validate and save the template definition to the database.







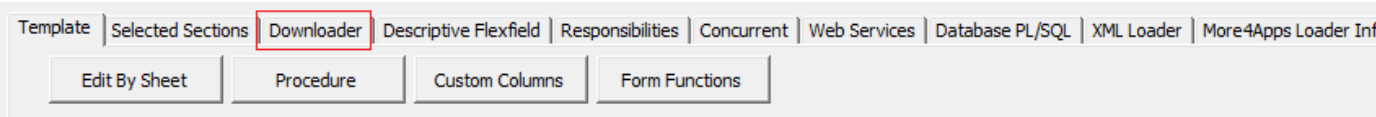
Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Copy Template
Delete Template
Modify Template
Attachment Templates
Downloader Template

Downloader Template

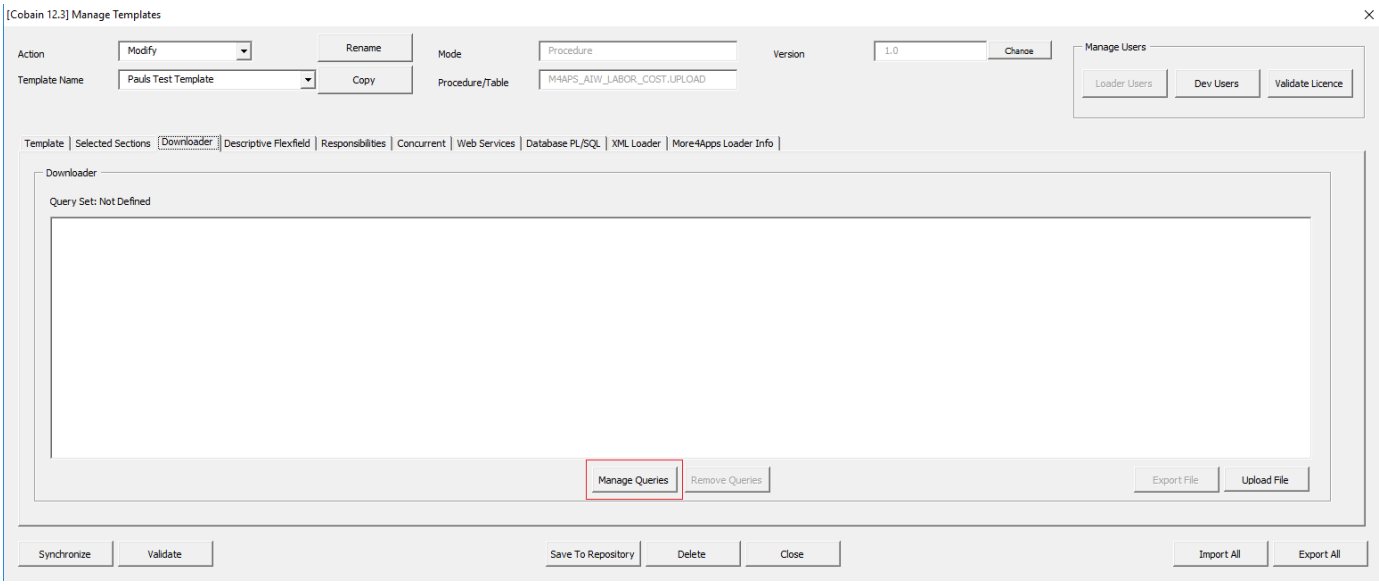
This section informs on how to set up a download query against an Application interface loader. For information on initiating a download, please refer to section(s): Custom Query - Download Form or Custom Query - Download from Sheet

AIW Developer

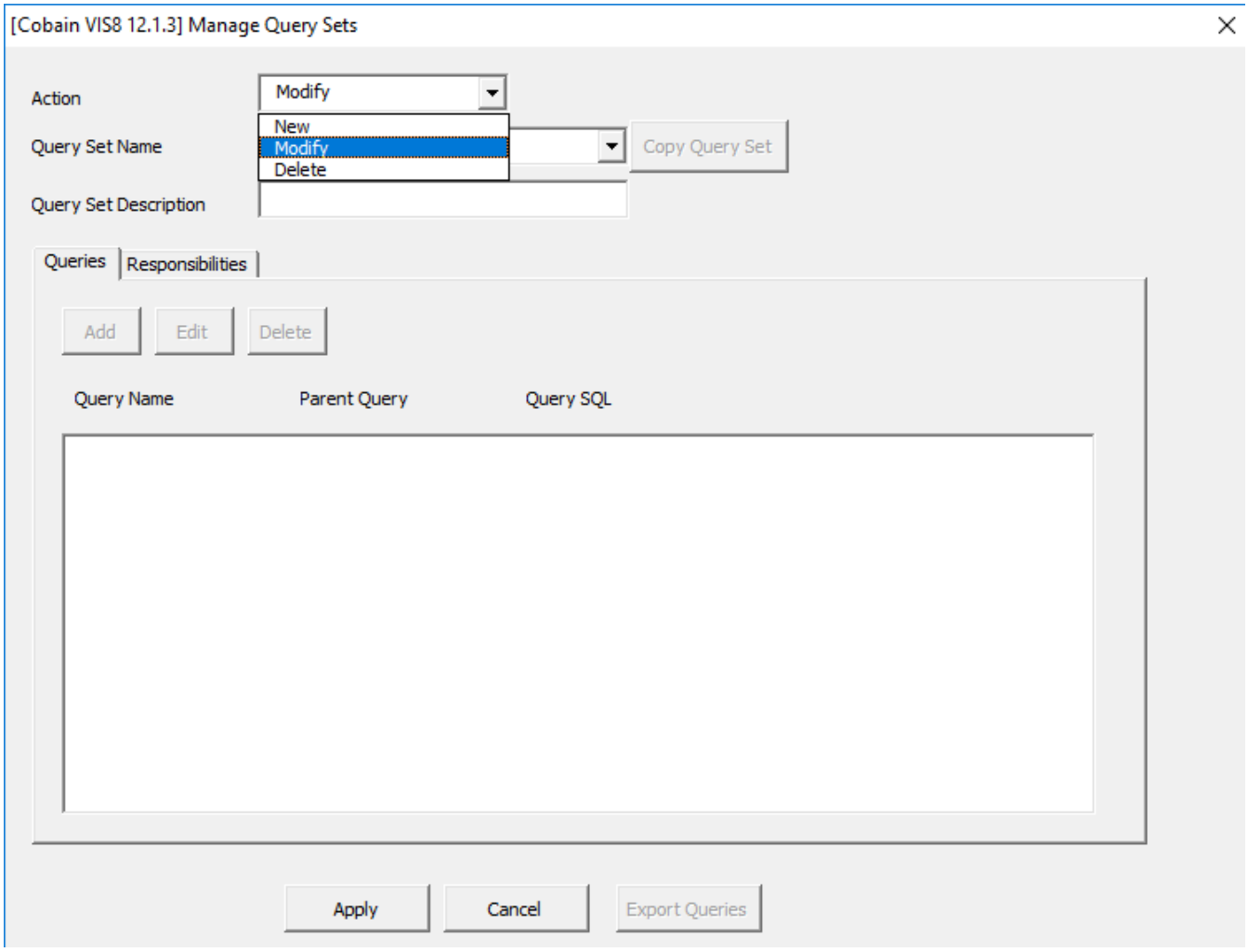
From the 'Manage Templates' form, the Application Interface Wizard has the ability to save a download query against a created loader. Once selecting the AIW template and navigating to 'Manage Templates' via the ribbon, there is a tab called 'Downloader' e.g.



Click on the 'Manage Queries' button.



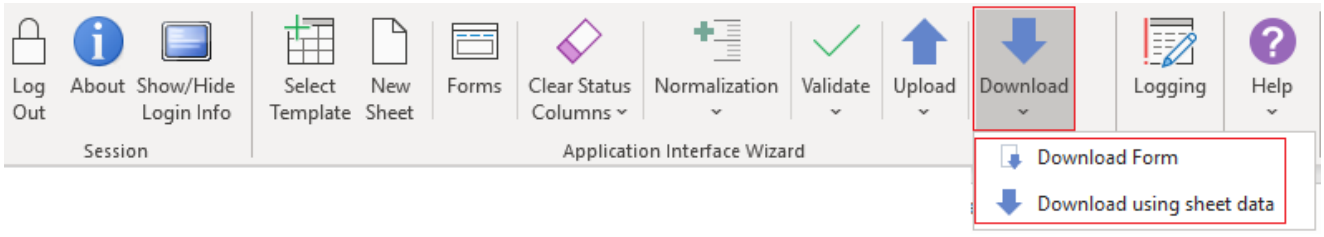
This will open the 'Manage Query Sets' form which gives the user the ability to Create, Modify or Delete a download query for the created loader.



Downloads are able to be built and the same functionality is available as the original 'Custom Query' Tab available in Application Interface Wizard. Once a query has been built click 'Apply' to allocate the query to the loader.
For more information on building downloads please refer to Custom Query in this document.

Once the download query has been built, validated and applied, the query is automatically assigned to the 'Download Form' and 'Download using sheet data' buttons on the Application Interface Wizard ribbon.

**Note - When a query is assigned to the loader, the same query will not be visible from the Query Set list of values found in the 'Custom Query' ribbon.*

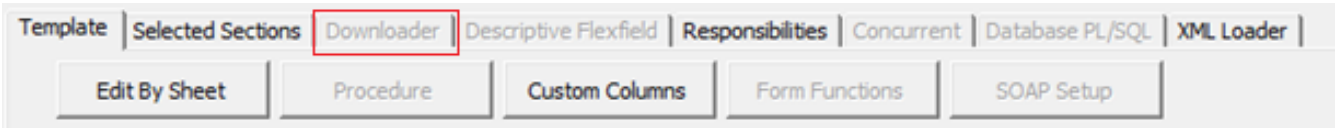


Once this is done, downloads can be executed in the same manner that we would with other More4apps Wizards.

Only one query can be assigned to a loader at a time, however further downloads can be built using the 'Custom Query' tab on the ribbon.

Supported Loader

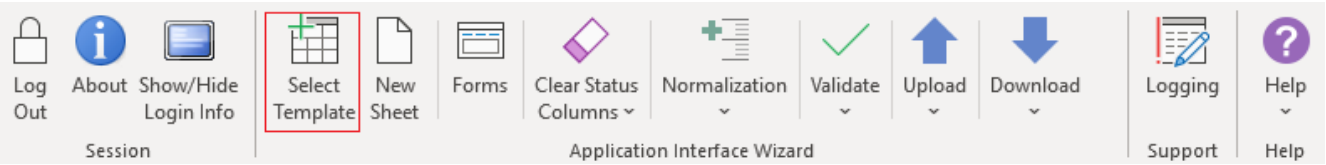
If a supported More4apps loader template is being used, the 'Downloader' tab will be greyed out. This is to ensure no changes can be made to the supported download of the loader.



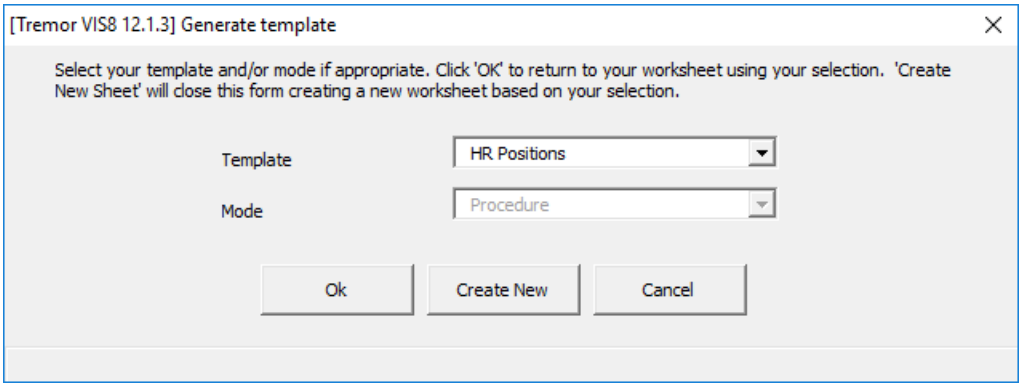
Refer to section:
Create 'New' Template
Template Versioning
Customize the Look and Feel of the Template
Selected Sections
Customize Logo
Descriptive Flexfields (DFF's)
Secure Template to EBS Responsibilities
Database PL/SQL
XML Loader
Wrapper Package Status Syntax
Package Tips
Synchronize Template
Validate Template
Export All
Update Procedure Name
Copy Template
Delete Template
Modify Template
Attachment Templates
Importing Templates

Open the Template Worksheet

Select the template by clicking on the 'Select Template' button on the Application Interface Wizard ribbon.
Note - You will be prompted to log in and select a Responsibility if you are not logged in.

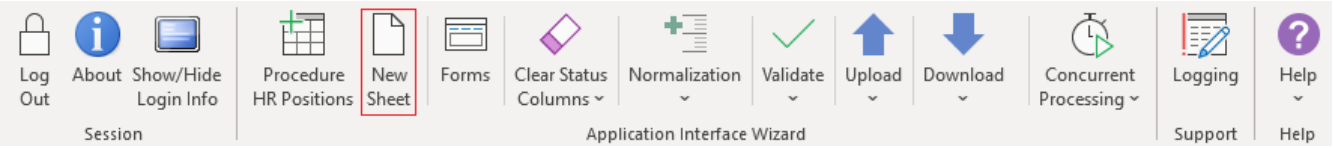


The 'Generate template' form will appear.
Use the down arrow on the Template field to select an available Template and click 'Create New' to create a new worksheet for the selected Template.



If you click 'Ok' you will not create a new worksheet but can carry on working with those that already exist for this Template.
Note - Only templates assigned to the Responsibility currently being used in the Wizard, or templates not assigned to any responsibilities will be available for selection.

Once a Template has been selected and if you click 'New Sheet' on the Application Interface Wizard, a new worksheet for the selected Template will be created:



Downloading Records

One of the most powerful features of Application Interface Wizard is the ability to extract/download information from Oracle and use it as the basis for new information.

There are three methods provided to download existing information into your worksheet from Oracle:

- Method 1 - Use the Download using Sheet Data option and enter multiple criteria into your worksheet to download based on the data entered.
- Method 2 – Use a Download Form which provides multiple criteria for you to target the required customer lists.
- Method 3 – Use the Custom Query Tab. This feature allows users to run pre-defined queries created by their Administrators where they are assigned to the same Oracle Responsibility being used in the Wizard/Loader.

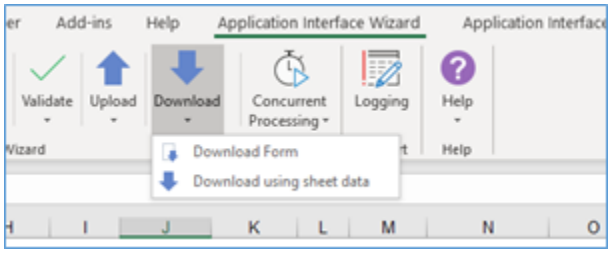
Each method will return data depending on the criteria entered. This provides the ability to download existing information into the worksheet. Information can then be modified to create or update existing records.

A download is a fast and effective way to illustrate what data goes in what column. Consequently, it is a good idea to use an existing record as a template for new data.

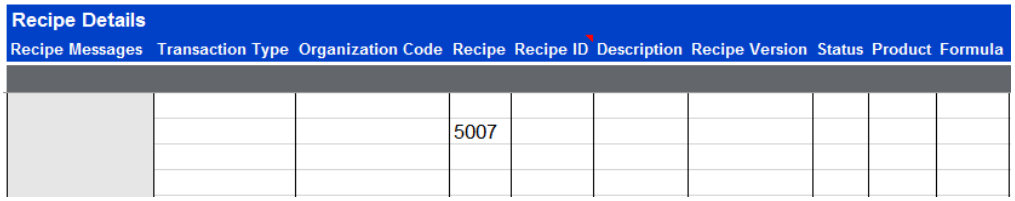
Refer to section:
Download using Sheet Data
Download Form
Custom Query Tab

Download using Sheet Data

The Wizard/Loader provides for the download of records based on a selection of downloadable columns in the worksheet.

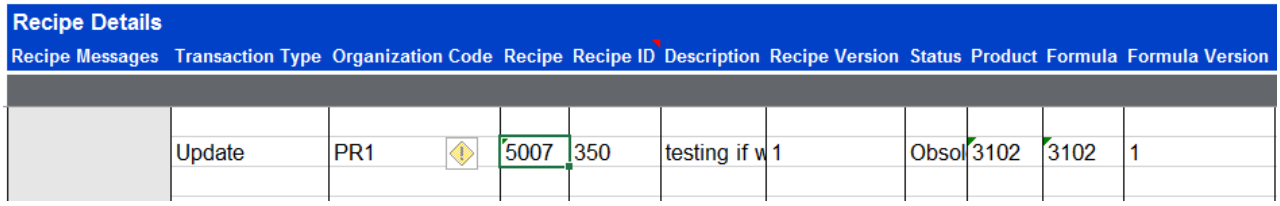


To download records, using the sheet data, first enter one or more the download filters on the loader sheet as below.



Select the rows to download and click the 'Download' icon and select 'Download using sheet data'.

The results of the download are returned directly to the highlighted range.



Note – When using the 'Download via sheet' option in the Wizard, the 'Default Values' row (Row 10) is not used.

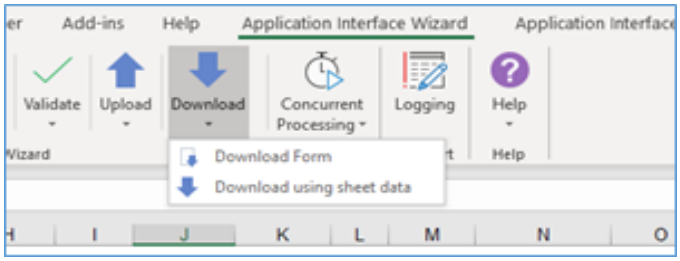
Once you have downloaded some invoices you can now make changes to them to create new invoices by keying directly into the fields, or double-click on the transaction to update via the Wizard form. Once data entry has been completed, upload the transactions into Oracle Payables.

Refer to section:
Download Form
Custom Query Tab

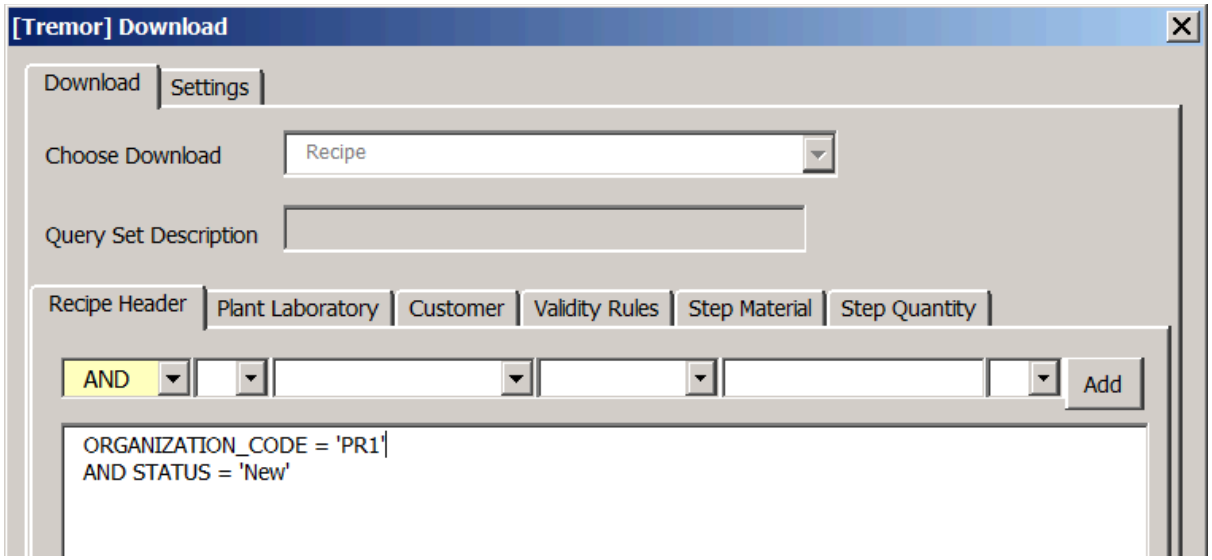
Download Form

Position the cursor on the row where you would like the download to start. Ensure you leave enough free rows below the cursor for all of the records that you are downloading, otherwise data may be overwritten.

Click the 'Download' icon on the ribbon and select 'Download Form':



A form like the following will appear:



Filters can be added here either by selecting values using the drop down lists or by typing directly into the text box.

Once all filters are entered, click on the "Download" button and it will download the records from the database into the AIW sheet. The following is an example of downloaded data.

Recipe Details

Recipe Messages	Transaction Type	Organization Code	Recipe	Recipe ID	Description	Recipe Version	Status	Product	Formula	Formula Version	Routing
	Update	PR1	3822	208	MC Test R	1	New	3822	3822	1	
	Update	PR1	5010	368	testtt	1	New	3202	5249	1	8110

Plant/Laboratory

Plan Laboratory Messages	PL Transaction Type	PL Organization	PL ID	PL Planned Loss	PL Fixed Loss	PL Fixed Loss UOM	PL Contiguous
		PR1	14	2	0	Ea	No
		PR1	174	15	20	AHR	No

Customer

Customer Messages	Customer Transaction Type	Customer Name	Ship To Locator	Operating Unit
		World of Busi	3754	Vision Italy

Step Quantity

Step Quantity Messages	Step Quantity Transaction Type	Step	Step ID	Operation ID	Operation	Operation Description	Step Quantity
		10	182	3	1-COMPI	Tabletting Operation	200004
		20	183	14	1-QC	QC Operation	54

For more information about downloading using Custom Query, see the Custom Query documentation.

Enter Information into the Worksheet

Populate information into the worksheet using the following methods:

(a) Use the 'Custom Query' ribbon to download existing records. You can then overwrite values in the cells as applicable. You can use this as the basis for new records or to update existing records.
Refer to section: Custom Query Ribbon
Refer to section: Custom Query

(b) Import a file, type or paste data directly into the worksheet cells provided.

(c) Each template contains dynamic forms for each section in the worksheet. If your System Administrator has also assigned List of Values (LOV's) to any columns these will appear in the forms. Simply double-click on a cell to open the corresponding data entry assistance form.
Refer to section: Data Entry Forms

Refer to section:
Default Values in Row 10
Data Entry Forms

Default Values in Row 10

The following functionality applies to Insert and Update modes.

Row 10 has been reserved to hold Default Values for the worksheet:

- You can enter defaults into the cells in row 10 and then leave the corresponding cell in the record blank. If the cell in the record is blank and there is a default, it will use the default.
- If the cell in the record has a value, this will override any default in Row 10.
- If you enter [BLANK] into the cell in the record, then the default value in Row 10 will be ignored. This syntax also means any API defaults will also be ignored.
- This functionality is also available in the 'Edit By Sheet' when loading columns.

Important - If the Template definition has a 'Default Value' assigned to a column, and the user has entered a value in Row10, the Row10 value will override the Template default.

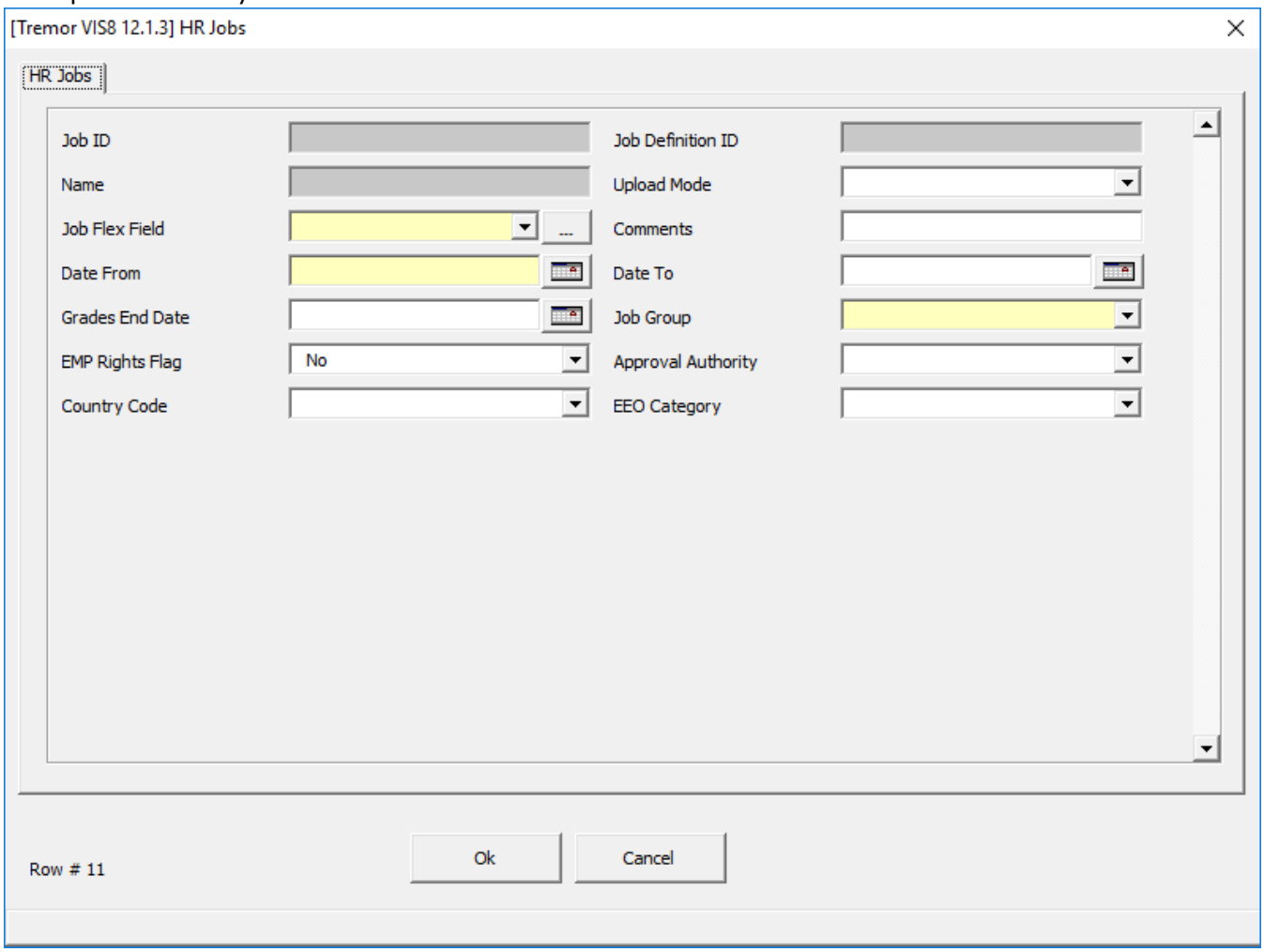
Refer to section:
Enter Information into the Worksheet
Data Entry Forms

Data Entry Forms

Each Wizard section contains a dynamic data entry form. While use of this is not required, they can assist with ensuring valid data is populated in the sheet prior to upload.

To open these forms simply double-click on any cell below Row10 and the form for that section will be displayed.

Example Data Entry Form



In the above form example, the fields display the results of the Template configuration, namely:

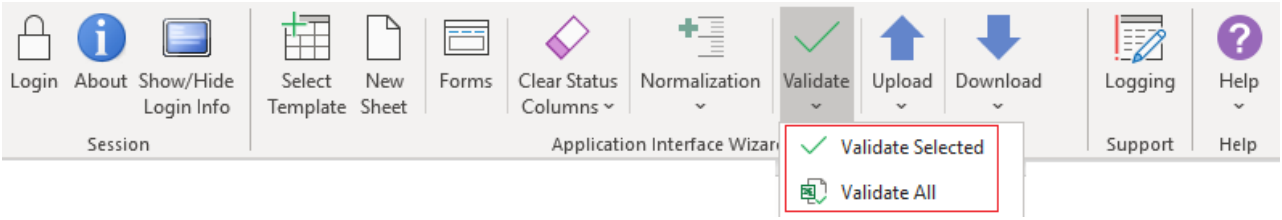
- Mandatory fields are shaded yellow, optional fields are unshaded.
- Shortlist LOV's have a dropdown arrow at the end of the field and these lists are pre-populated when the form is opened.
- Longlist LOV's have a dropdown arrow and an ellipsis search button at the end of the field. These lists are not pre-populated when the form is opened as users must search for values using the ellipsis button.
- Columns are ordered according to the template.

- Custom labels will be displayed (if set).
- Date fields are automatically assigned calendar buttons.
- Only fields set as 'Display' in the Template definition appear on the form.
- Read Only columns are unshaded but can't be selected.

Refer to section:
Enter Information into the Worksheet
Default Values in Row 10

Upload Records

1. Before uploading the information in the worksheet, the user can validate the information using one of the following options on the ribbon. Click the Validate icon and select your option from the LOVs:



The choices are:

Validate Selected – Will validate selected row(s) in the current worksheet.

Validate All – Will validate all rows in the current worksheet.

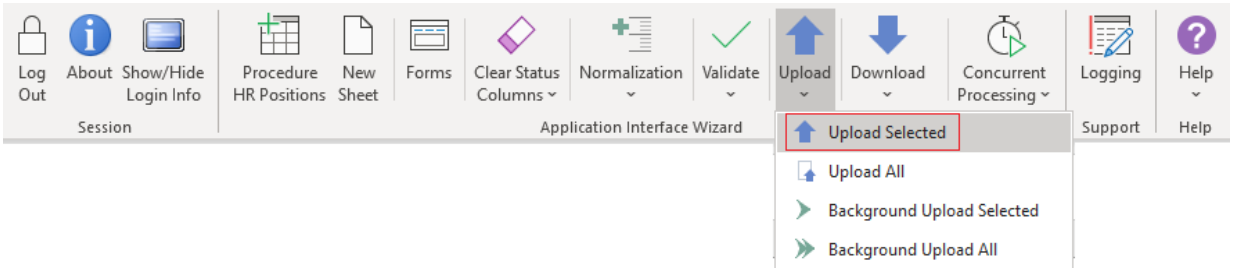
2. There are four options to Upload Records.

(a) **Upload Selected** records - Use your cursor to select the rows ready to be validated and loaded.

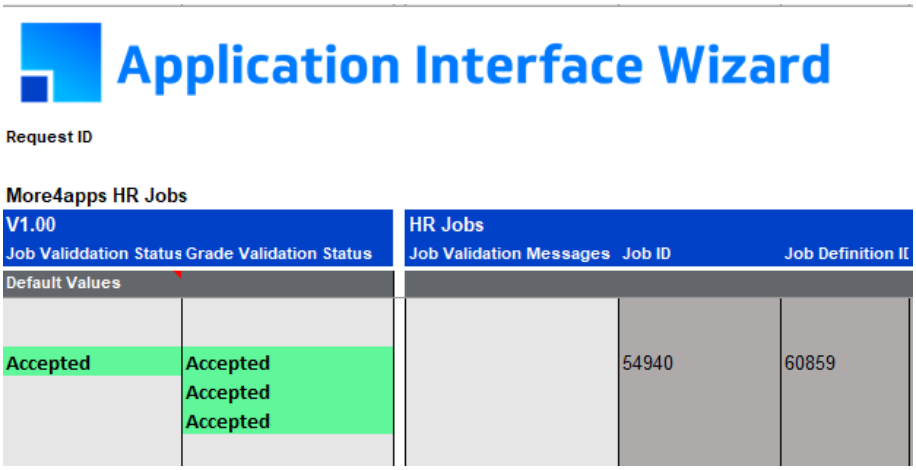
<u>Select a Single Row</u>
Place your cursor anywhere on a row containing the information you need to load. It does not have to be any particular column.

<u>Select a Range of Rows</u>
Use your cursor to highlight the rows you wish to upload. It does not have to be any particular column and you do not need to highlight the entire row. Use the Ctrl key or Shift key to select multiple rows.

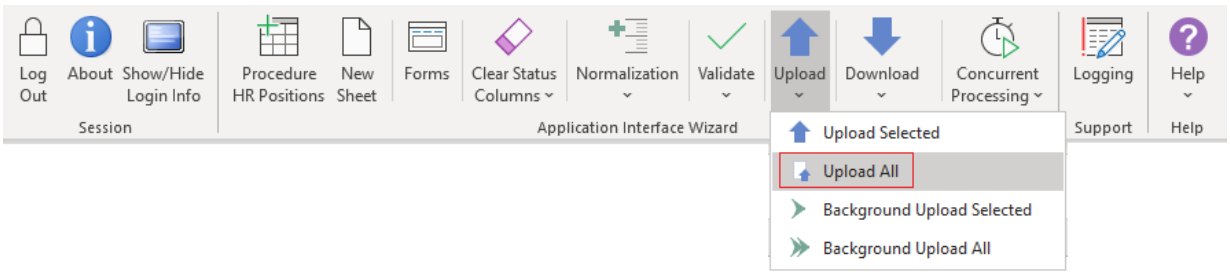
Click the 'Upload' icon and select 'Upload Selected' on the ribbon to send the selected data to EBS.



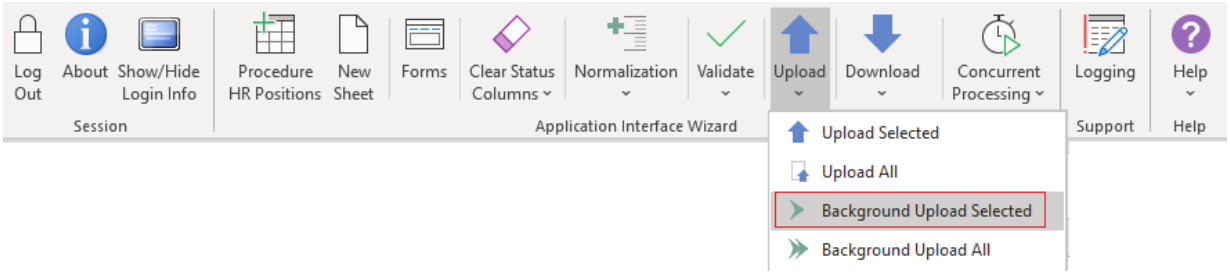
If the record is successfully uploaded, then a commit will be performed and there is no rolling back the transaction. The status **Accepted** will appear in the appropriate status columns, as per the example below:



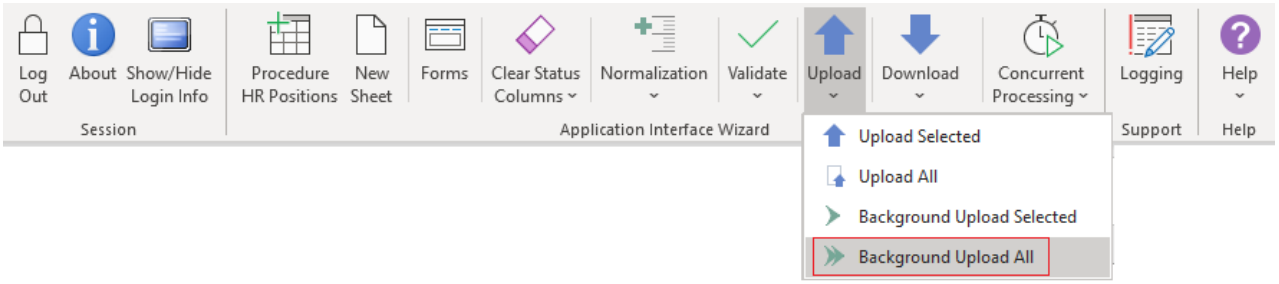
(b) **Upload All** records - Alternatively, to send all unprocessed rows to EBS simply click the 'Upload' icon and select 'Upload All' on the ribbon. This will validate and upload all records.



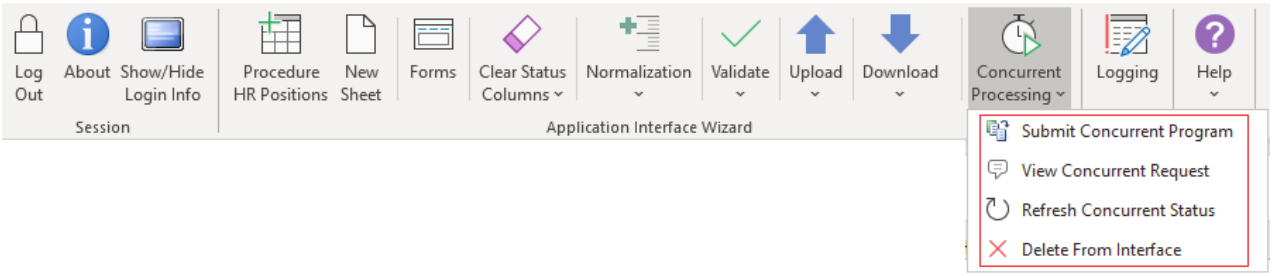
(c) **Background Upload Selected** records - This option is only available in a 'Procedure' mode. Validates and uploads selected records as per 'Upload Selected' option, however will cause the Wizard to create a concurrent request to upload records.



(d) **Background Upload All** records - This option is only available in a 'Procedure' mode. Validates and uploads all records as per 'Upload All' option, however it will cause the Wizard to create a concurrent request to upload records.



3. To view the request for the concurrent background processes, click on the 'Concurrent Processing' icon and select 'View Concurrent Request' from the product ribbon. Once the request has 'Completed' click 'Refresh Interface Status' from the ribbon and the upload messages will return to the worksheet.
   Note - If this option is not available on your version of the Wizard then you will need to download the latest version and packages from our website.



Improving Upload Performance by Removing Cell Defaulting and Colouring

When uploading large volumes of data, the Wizard performance can start to diminish when Excel reaches its memory limits.

The following Wizard worksheet standard operations can affect this:

- Writing default row values (Row 10) down into the data section of the worksheet.
- Writing cell comments for messages and statuses.
- Formatting cell colouring in the data section.

This has always been slow, because each aspect of saving must go cell by cell. Another aspect is that colouring causes the file size of Excel to increase, which eventually has a performance hit on the code and stability of Excel.

Error Handling and Validation

There are two levels of validation:

- Spreadsheet Validation – The Wizard will validate the information based on the data type of the column e.g. date, number, a string of characters and any associated LOV's.

- Server side Validation – The upload process will perform the spreadsheet validation and then attempt to execute the transaction. Any database errors, or user-defined errors that prevent the transaction being uploaded will be written back into the worksheet.

Refer to section:
Status and Message Columns
Correcting Errors and Reloading

Status and Message Columns

Every worksheet will be created with relevant 'Status' and 'Messages' columns for each section.

The section 'Status' is used to record the overall status for each record. The section 'Status' may be:

Status	**Status Explanation**
Blank	Prior to running the upload process.
Accepted	The information in this row has successfully created a record in the Oracle database.
Error	This row has failed to upload process. Refer to the section's messages column for a descriptive reason for the error.

Refer to section: Correcting Errors and Reloading

Correcting Errors and Reloading

If you have an **Error** status, you need to read the corresponding message in the corresponding Message column and investigate the data. The error messages relate to data on the same row.

To correct the information, you can key directly into the cell in the worksheet.
Once the data has been corrected, you can re-upload selected rows into the Oracle database.

Those records that already have a Status of **Accepted** will not be processed again.

Refer to section:
Upload Records
Status and Message Columns

Debugging Uploads

There are a couple of useful tools to assist System Administrators with debugging their loaders.

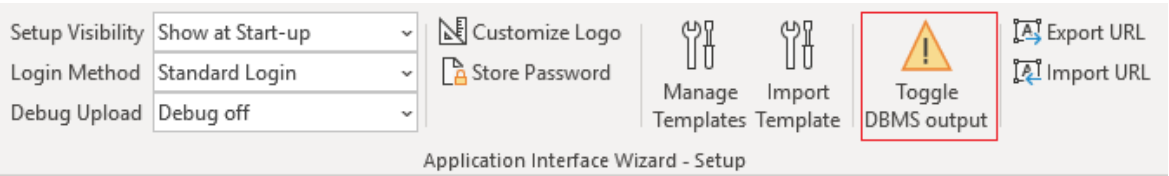
Refer to section:
DBMS_OUTPUT Messages
DEBUG LOG

DBMS_OUTPUT Messages

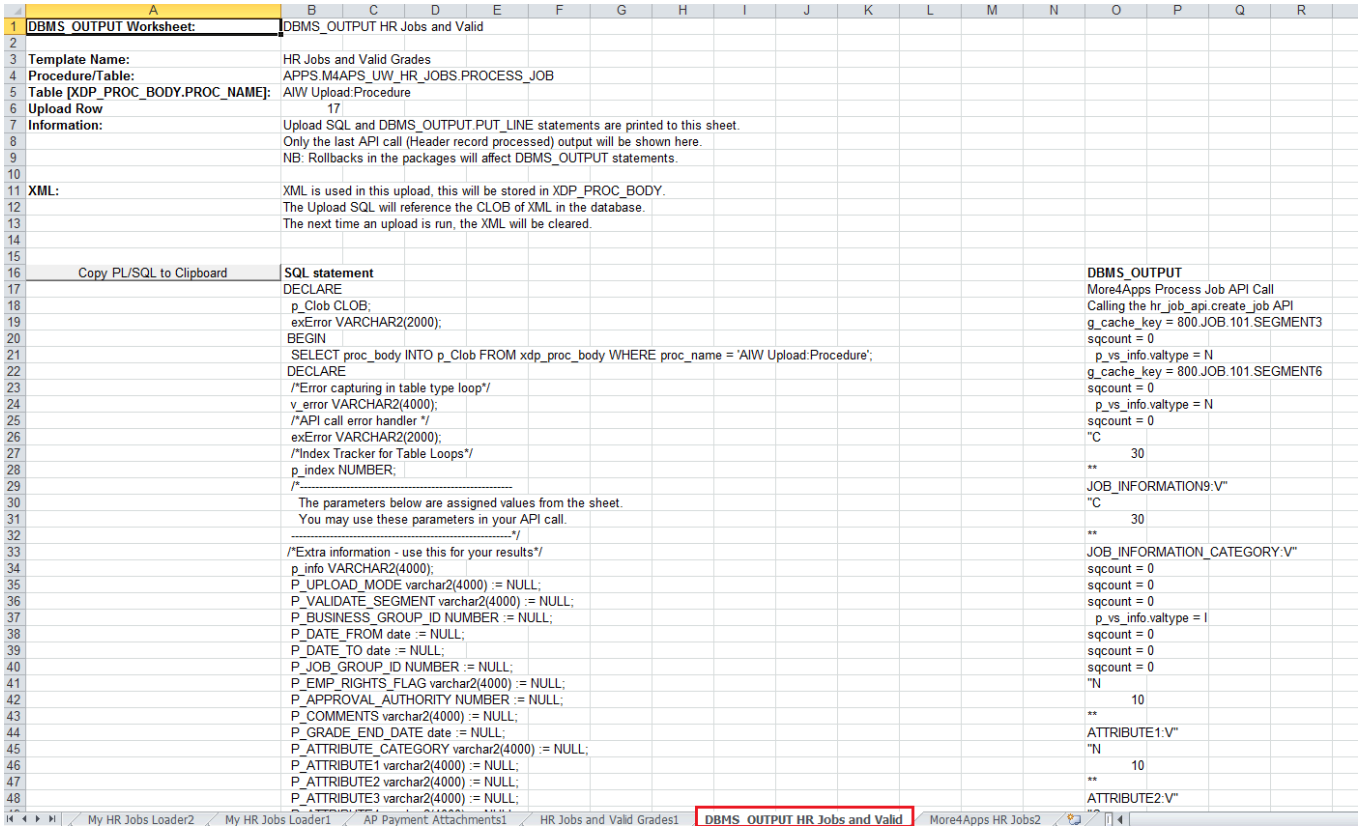
System Administrators can elect to have DBMS_OUTPUT messages written to a separate worksheet while they are designing and testing their Templates that are either Table or Procedure based.

This is particularly useful as there is no requirement to log into a separate SQL development application.

To enable DBMS_OUPUT messages click the 'Toggle DBMS output' icon from the 'Application Interface Wizard - Setup' ribbon.



When you upload your data a new worksheet will be automatically created similar to the image below:



	A	B
1	**DBMS_OUTPUT Worksheet:**	DBMS_OUTPUT HR Jobs and Valid
2		
3	**Template Name:**	HR Jobs and Valid Grades
4	**Procedure/Table:**	APPS.M4APS_UW_HR_JOBS.PROCESS_JOB
5	**Table [XDP_PROC_BODY.PROC_NAME]:**	AIW Upload:Procedure
6	**Upload Row**	17
7	**Information:**	Upload SQL and DBMS_OUTPUT.PUT_LINE statements are printed to this sheet.
8		Only the last API call (Header record processed) output will be shown here.
9		NB: Rollbacks in the packages will affect DBMS_OUTPUT statements.
10		
11	**XML:**	XML is used in this upload, this will be stored in XDP_PROC_BODY.
12		The Upload SQL will reference the CLOB of XML in the database.
13		The next time an upload is run, the XML will be cleared.
14		
15		
16	Copy PL/SQL to Clipboard	**SQL statement**
17		DECLARE
18		p_Clob CLOB;
19		exError VARCHAR2(2000);
20		BEGIN
21		SELECT proc_body INTO p_Clob FROM xdp_proc_body WHERE proc_name = 'AIW Upload:Procedure';
22		DECLARE
23		/*Error capturing in table type loop*/
24		v_error VARCHAR2(4000);
25		/*API call error handler */
26		exError VARCHAR2(2000);
27		/*Index Tracker for Table Loops*/
28		p_index NUMBER;
29		/*----------
30		The parameters below are assigned values from the sheet.
31		You may use these parameters in your API call.
32		----------*/
33		/*Extra information - use this for your results*/
34		p_info VARCHAR2(4000);
35		P_UPLOAD_MODE varchar2(4000) := NULL;
36		P_VALIDATE_SEGMENT varchar2(4000) := NULL;
37		P_BUSINESS_GROUP_ID NUMBER := NULL;
38		P_DATE_FROM date := NULL;
39		P_DATE_TO date := NULL;
40		P_JOB_GROUP_ID NUMBER := NULL;
41		P_EMP_RIGHTS_FLAG varchar2(4000) := NULL;
42		P_APPROVAL_AUTHORITY NUMBER := NULL;
43		P_COMMENTS varchar2(4000) := NULL;
44		P_GRADE_END_DATE date := NULL;
45		P_ATTRIBUTE_CATEGORY varchar2(4000) := NULL;
46		P_ATTRIBUTE1 varchar2(4000) := NULL;
47		P_ATTRIBUTE2 varchar2(4000) := NULL;
48		P_ATTRIBUTE3 varchar2(4000) := NULL;

O	P	Q	R
DBMS_OUTPUT			
More4Apps Process Job API Call			
Calling the hr_job_api.create_job API			
g_cache_key = 800.JOB.101.SEGMENT3			
sqcount = 0			
p_vs_info.valtype = N			
g_cache_key = 800.JOB.101.SEGMENT6			
sqcount = 0			
p_vs_info.valtype = N			
sqcount = 0			
"C			
30			
**			
JOB_INFORMATION9:V"			
"C			
30			
**			
JOB_INFORMATION_CATEGORY:V"			
sqcount = 0			
sqcount = 0			
sqcount = 0			
p_vs_info.valtype = I			
sqcount = 0			
sqcount = 0			
sqcount = 0			
"N			
10			
**			
ATTRIBUTE1:V"			
"N			
10			
**			
ATTRIBUTE2:V"			

The worksheet name will be DBMS_OUTPUT appended to the template name.

DBMS_OUTPUT messages are written into column 'O' of the worksheet.

The SQL Statement is also displayed in column 'B' for your information.

The 'Copy PL/SQL to Clipboard' button in the worksheet allows for a simple copy of the entire Wizard call to the database to be pasted into your SQL Development tool should this be required for debugging purposes.

When you have completed debugging your Template click the 'DBMS output option' icon again from the 'Setup' ribbon to turn off the debugging.

Note - This sheet can be deleted when development is complete. Each time DBMS_OUTPUT is enabled it will be regenerated.

DBMS_OUTPUT information is only displayed for the last record in multiple row uploads.

DBMS_OUTPUT information is not available for 'Attachment' Templates.

Refer to section: DEBUG LOG

DEBUG LOG

All More4apps Wizards will print SQL information to any worksheet named 'Debug'.
This can also be useful for tracking issues and sending data to More4apps.
Insert a new worksheet (Shift + F11) into the Wizard Workbook and rename it 'Debug'.

While the 'Debug' sheet exists in the workbook, SQL information will continue to be written into the sheet.

Note - The Wizard will append information to existing data in the 'Debug' sheet. If you delete data from the 'Debug' sheet, new information may be written further down in the sheet than expected.

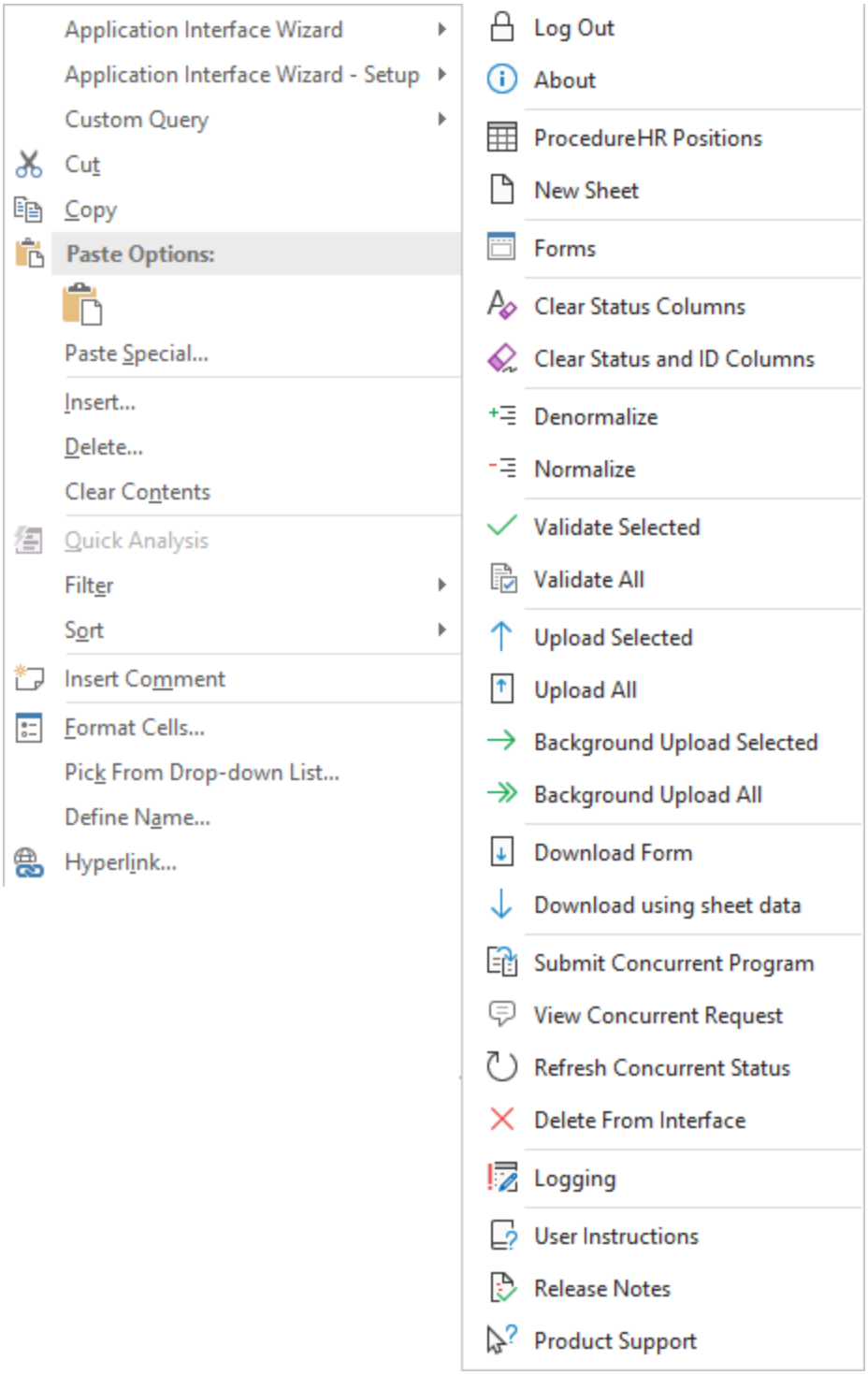
Refer to section: DBMS_OUTPUT Messages

Ribbons

There are three tabs in the ribbon menu which belong to Application Interface Wizard:

- **Application Interface Wizard - Setup Ribbon** is used by administrators to set up the Wizard and to create or maintain Templates.
- **Application Interface Wizard Ribbon** is used to carry out functions when working in your workbook e.g. Validate and Upload.
- **Custom Query Ribbon** is used to maintain and execute the stored queries to extract information. For more information refer to section: Custom Query

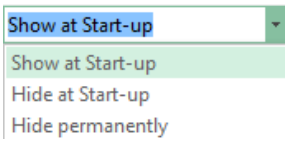
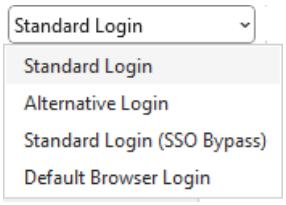
The menus are also available using the right mouse click functionality within Excel. If you press the right mouse button you will see the Application Interface Wizard ribbon options.

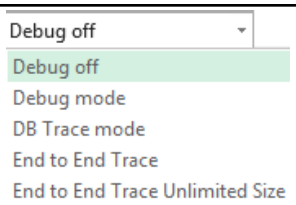


Refer to section:
[Application Interface Wizard - Setup Ribbon]
[Application Interface Wizard Ribbon]
[Custom Query Ribbon]

Application Interface Wizard - Setup Ribbon

The following table provides a quick reference to the Application Interface Wizard - Setup Ribbon and its functions.

Icon	Description/Functionality
Setup Visibility 	Once your set-up is done you may not want your Users to change the set up, you can hide the setup ribbon. There are three options: **Show at Start-up** – The setup menu always appears. **Hide at Start-up** – The setup menu is hidden but can be found using: View > Macros > ShowWizardToolbars and click 'Run' **Hide Permanently** – The setup menu is hidden and cannot be displayed. Revert back to a copy of the original workbook to display the menu.
Login Method 	Choose one of the following methods for logging in: **Standard Login** – Will use single sign-on where available. **Alternative Login** – Will use application level sign-on. **Standard Login (SSO Bypass)** – This will use Single Sign-On Bypass and can be used if you have SSO installed but want to bypass the SSO mechanism. The user will be authenticated to the internal FND_USERS table rather than the external SSO. **Default Browser Login** - This will use the default browser on your PC to display the EBS Login page. This is the best method to use if the 'Standard Login' method does not work for you.
Debug Upload	For the 5x 'Debug Upload' options mentioned below, the user will be under instruction from More4apps Support (used for Support purposes

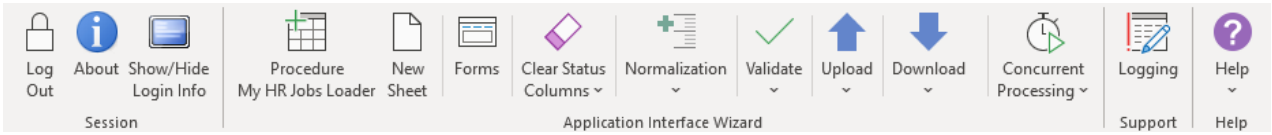
	only). The default value is 'Debug off' unless instructed by More4apps to change: **Debug off** – Disables any of the above options that are enabled. **Debug mode** – Turns on the More4apps debug facility to generate debug messages to assist with product troubleshooting. **DB Trace mode –** Generates a database trace for database Version 9i. **End to End Trace** – Database trace generated for performance analysis and product troubleshooting. **End to End Trace Unlimited Size** – This option allows the maximum size of trace files created whilst debugging to become unlimited. This resolves the scenario when trace files are too large that they were being truncated. A message box will be displayed when this option is chosen allowing the user to cancel and not set the trace off: "Please only use this under More4apps advice. A very large trace file may be created. If you are unsure, please use the End to End Trace' option rather than the 'End to End Trace Unlimited Size' option". **Refer to section:** Common User Topics
Customize Logo 	Show or Hide Template worksheet. If you modify the template worksheet, subsequent worksheets created using the 'New Sheet' icon will be based on this template. Refer to section: Customize the Template Sheet This sheet also stores the current logo that is used by default in the Wizard templates but can be changed at the user's discretion. Refer to section: Customize Logo
Store Password 	You can protect your worksheet after data-entry to prevent further changes. However, you must store the password in order to use the upload/validate functions. Click on this button and store the password that you have used to protect the worksheet. For further information refer to Common User Instructions: Worksheet Protection
Manage Templates 	The Manage Templates icon provides options to create new Templates, modify attributes of existing Templates or delete obsolete Templates.

Import Template	The Import Template icon enables templates stored as XML files on your PC to be uploaded into the Wizard.
Toggle DBMS output	To see the results of DBMS Output messages from your code, set this profile to 'Create DBMS Output'. When the data is uploaded a new worksheet will appear in the Wizard workbook displaying all DBMS Output messages. Select 'Ignore DBMS Output' to suppress this functionality. **Note** - The Sheet can simply be hidden, or deleted when finished with. DBMS Output messages will only be displayed for the last row loaded where there are multiple lines.
Export URL	Export Login Configuration Settings to separate sheet. This option is only applicable to the 'Oracle EBS' connection method. The login configuration allows you to store links to multiple instances of Oracle Applications. Create a new blank sheet (insert worksheet) and click on this icon. The current login configuration will be copied into cell A1. This can then be copied into another workbook (into cell A1) and imported using the option below.
Import URL	Import Login Configuration Setting from separate sheet. This option is only applicable to the 'Oracle EBS' connection method. The login configuration that has been copied into cell A1 of the current sheet will be used to set up the list of instances available in this workbook.

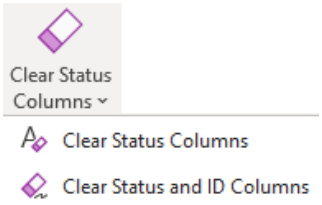
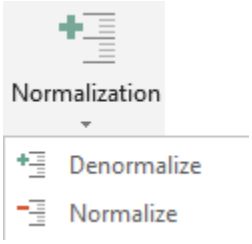
Refer to section:
Application Interface Wizard Ribbon
Custom Query Ribbon

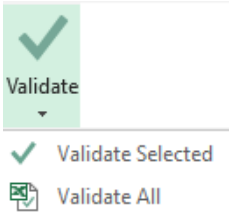
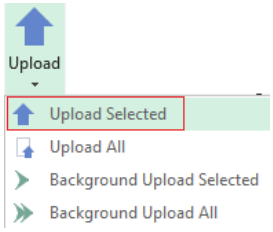
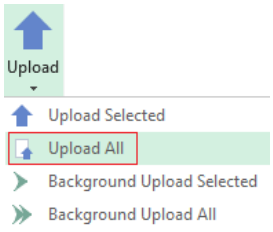
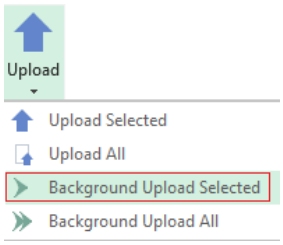
Application Interface Wizard Ribbon

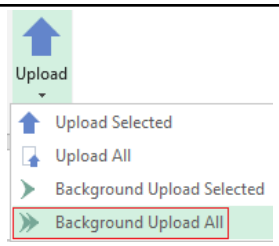
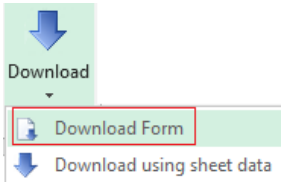
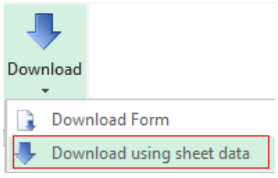
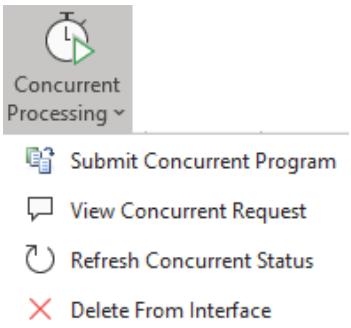
The following table provides a quick reference to the Application Interface Wizard Ribbon and its functions.

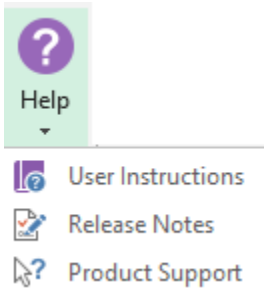


Icon	Description/Functionality
Log In/Log Out	This will either log you into or out of EBS depending on your status (terminate or reinstate your Oracle database connection). The icon will change depending on your current status. If you are currently logged in then the icon will show as 'Log Out'. If you are not logged in the icon will show as 'Log In'.
About	Displays product and package version information along with licencing and usage data. You need to click on this button to load a new Licence Key, Modify/add a user of the Wizard or 'View Usage'. Refer to the Common User Topics documentation for more information.
Show/Hide Login Info	This button toggles the display of the 'Login Info' information on the ribbon.
Select Wizard Mode/Template	This icon will open the 'Generate template' form to select a template created by your System Administrator. This icon will change to display the type of template and template name once selected.
New Sheet	A new worksheet will be created based on the selected template.

	If the Template is invalid or not yet selected, the 'Generate template' form will appear.
Forms 	The 'Forms' icon opens the ….??? in the current sheet. You do not have to use forms, you can copy and paste or key directly into the cells of the worksheet if you prefer. If you do use these forms, you can make use of the 'Lists of Values' and other validation as you are entering data. Double-clicking in the particular section in the worksheet will also open the required form.
Clear Status Columns 	This icon enables you to remove status column information from the worksheet: **Clear Status Columns** – This only clears the Status Columns and not the ID's. **Clear Status and ID Columns** – This clears the Status Columns and ID's. This functionality is helpful if you have downloaded an existing record and would like to clear the ID's to enable you to use this as a template to create a new record. Note – If a Wizard does not have ID columns and this option is selected, a message will appear stating there are no ID columns but all statuses have been cleared.
Normalization 	Used for formatting data by adding duplicate parent rows based on child rows and removing duplicate parent rows respectively. **Denormalize** the data is useful for when you need to sort the data in Excel. Click the Denormalize button and it will duplicate header data for the rows containing child data. Once you are finished sorting the data then you can normalize the data to remove the duplicate headers and prepare it for upload. **Normalize** the data is used to prepare data for upload in the Wizard, whether you've just denormalized it for sorting or imported the data from an external source with duplicate headers. Normalization will only affect sections that have children with multiple rows of data. Both functions will affect all the records on the active sheet.

Validate 	Choose whether to 'Validate Selected' or 'Validate All' when uploading lines into the interface table. This selection is used in conjunction with the 'Upload Selected' and 'Upload All' rows icon (as shown below).
Upload Selected 	The selected rows (as indicated by the cursor) will be validated/loaded into Oracle.
Upload All 	All rows in the worksheet will be validated/loaded into Oracle.
Upload Background Upload Selected 	All selected rows (as indicated by the cursor) in the worksheet are validated/uploaded and processed using Oracle Concurrent processing. This is useful when processing large amounts of data and you wish to continue working in the More4apps Wizard. This process can be viewed using the Requests form in Oracle.
Upload Background Upload All	All rows in the worksheet are validated/uploaded and processed using Oracle Concurrent processing.

	This is useful when processing large amounts of data and you wish to continue working in the More4apps Wizard. This process can be viewed using the Requests form in Oracle.
Download Form 	A form will open that will enable you to download information into the worksheet based on selected criteria. Please note a download query needs to be assigned to the AIW loader template first. **Refer to section:** Downloader Template
Download using sheet data 	After entering data or information into the worksheet, use your cursor to highlight the details and then use this icon to download all details which are associated with these values. Please note a download query needs to be assigned to the AIW loader template first. **Refer to section:** Downloader Template
Concurrent Processing 	This option processes the concurrent program: **Submit Concurrent Request** – This option submits the concurrent program associated to the Mode. **View Concurrent Request** – Enables the user to view the concurrent requests on the Oracle Concurrent Manager. **Refresh Concurrent Status** – This refreshes the line status of the records in the spreadsheet after the concurrent request has run. It will show errors or warning messages as applicable. **Delete From Interface –** Deletes data from the interface.
Logging 	Clicking the button will create a new sheet and name it 'Debug' which will record all actions being taken by the Wizard on upload. If a sheet already exists with the name Debug, this existing sheet will be named Debugx where x is the next available number e.g. If a sheet exists called 'Debug', it will be renamed 'Debug1'. If a sheet exists called 'Debug1', it will be renamed 'Debug2' etc.

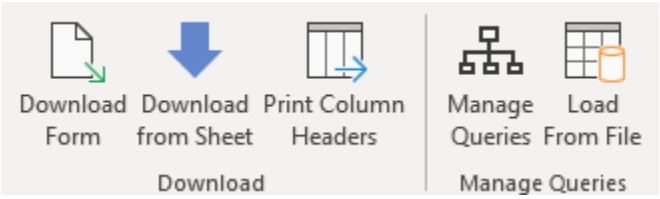
	This is helpful for support to ensure that More4apps has all the information required to resolve your issue.
Help 	Click this icon for help on any of the functionality of this Wizard. The three options are: **User Instructions** – Link to the More4apps Webpage to find detailed explanation on the Wizard. **Release Notes** – Link to the latest releases for the Wizard. **Product Support** – Link to the 'Contact Us' page.

Refer to section:
Application Interface Wizard - Setup Ribbon
Custom Query Ribbon

Custom Query Ribbon

The following table provides a quick reference to the Custom Query Ribbon and its functions.

Refer to section: Custom Query



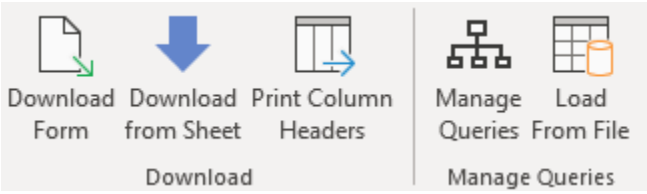
Icon	Description/Functionality
Download Form	A form to enter the Query Set for downloading records to Excel. Here you can also indicate if you want to print parent information, whether you need to print column headers or use existing ones and, if needed, what filters you would like to use to limit the records to download.
Download from Sheet	When your headers are already printed on the spreadsheet you can position your cursor right below them and click on this option to download records. You will just need to pick the Query Set and then click the Download button.
Print Column Headers	Before downloading records directly from the sheet you need to click on the Print Column Headers option. Here you will also select the row where the columns will be printed to.
Manage Queries	This option will give you access to the Manage Query window where you can create, modify and delete SQL queries and responsibility access to query sets.

Load from File 	This option will import xml query set data contained in selected worksheet cells into the instance you are connected to. This feature is useful when writing custom Query Sets in one instance and then needing to copy these across to other instances. To achieve this simply use the 'Export Queries' feature in the Custom Query 'Edit Queries Using Form' to copy the xml definitions into your current worksheet. Then login to a different instance, select the cells containing the downloaded query sets and click the 'Load From File' button to import them into the current instance.

Refer to section:
Application Interface Wizard - Setup Ribbon
Application Interface Wizard Ribbon

Custom Query

Application Interface Wizard has our latest flexible downloading option available in a separate 'Custom Query' ribbon.



Refer to section: Custom Query Ribbon for more information on each button.

This feature allows users to run pre-defined queries created by their Administrators where they are assigned to the same Oracle Responsibility being used in the Wizard.

This powerful download option enables the simple mapping of Oracle E-Business Suite data directly into the appropriate Wizard columns, making subsequent uploading a breeze.

The pre-defined queries can include run-time filters or you can add you own additional filters before downloading the data.

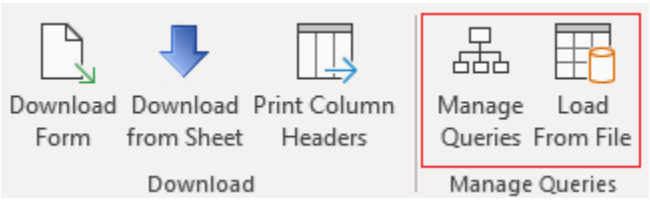
There are three main user buttons available on the Custom Query ribbon:

a) **Download Form** - Where additional filters can be added using a form.

b) **Download from Sheet** - Where sheet criteria is used for additional download filters.

c) **Print Column Headers** - Where the column headers are printed to the worksheet.

Note - The other two buttons in the 'Manage Queries' section on the ribbon are only available to users with System Administrator access:

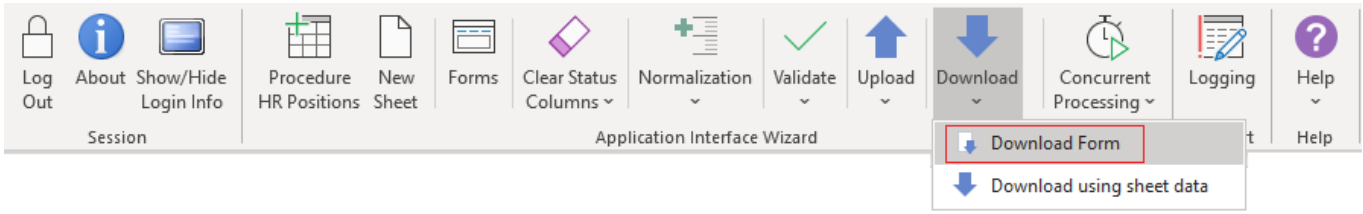


Information on how to use these options to Create, Maintain, and Assign Query Sets, can be found on the More4apps website [Common User Instructions: Custom Query Add-In]
This functionality can be suppressed by setting the profile option '*More4Apps: AIW Suppress Custom Query Functionality*' to 'Yes' at the user, responsibility or site level.
When this profile option is set (at the appropriate level) selection of any custom query icon will produce a message of 'The custom query functionality has been disabled by your System Administrator. Please contact your System Administrator'.

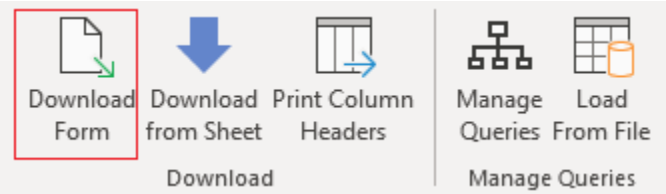
Refer to section:
Custom Query - Download Form
Custom Query - Download from Sheet
Custom Query - Print Column Headers
Examples of Custom Query used in PO Wizard

Custom Query - Download Form

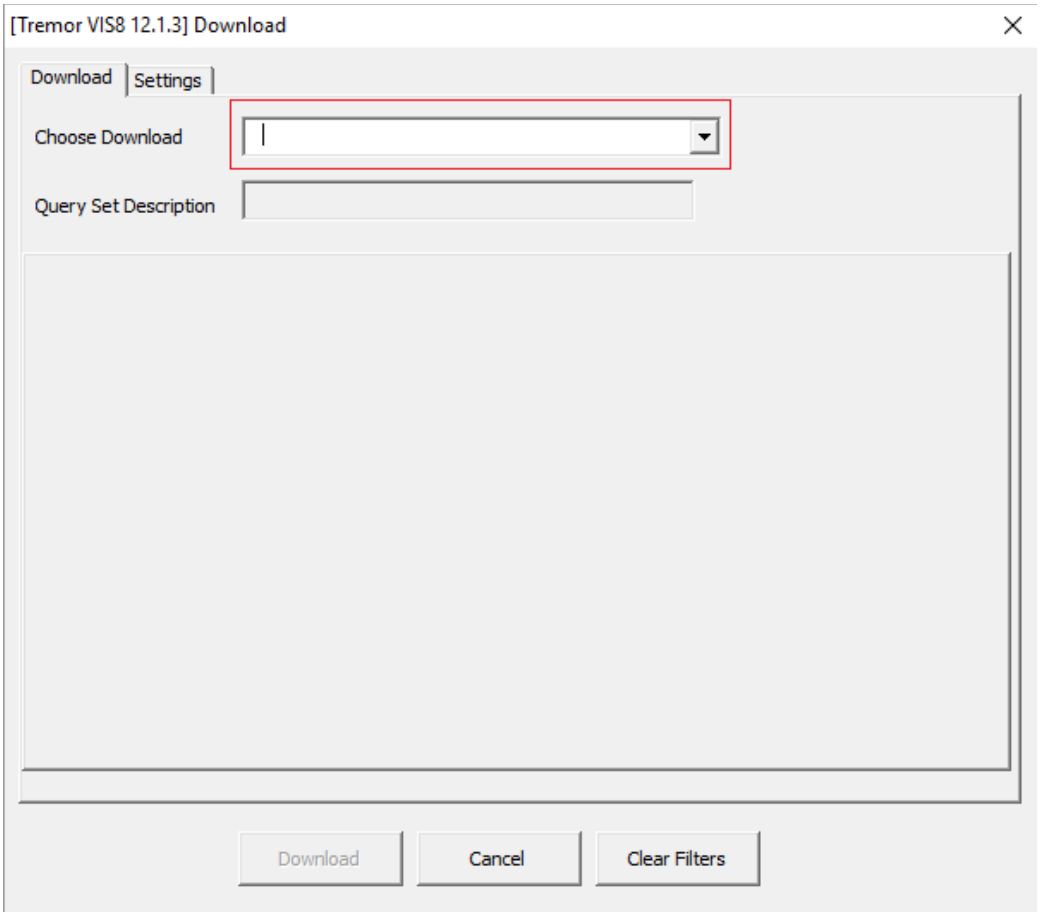
Use the Download form to select a pre-defined query to run, where you want to use additional filters that don't exist in the query definition.
Note - If the download is being initiated using the primary template 'Download Form' (see illustration below) only steps 4 and 5 will need to be followed.



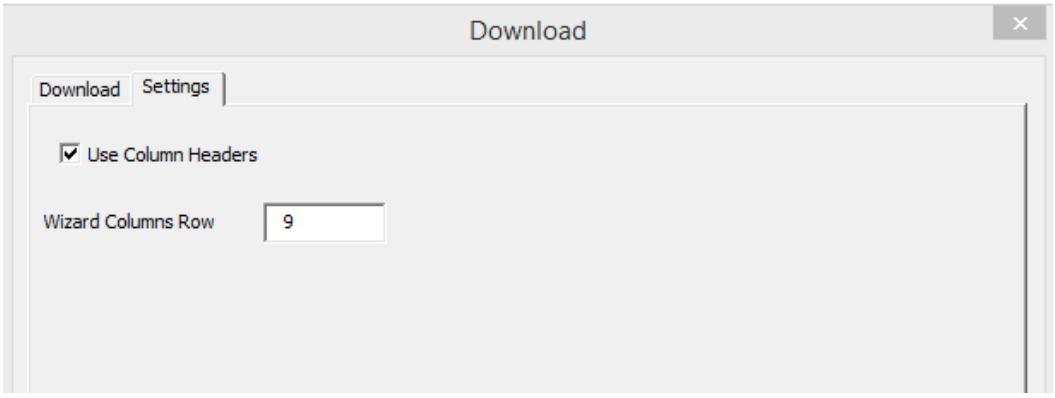
1. Select the 'Download Form' option from the Custom Query ribbon.



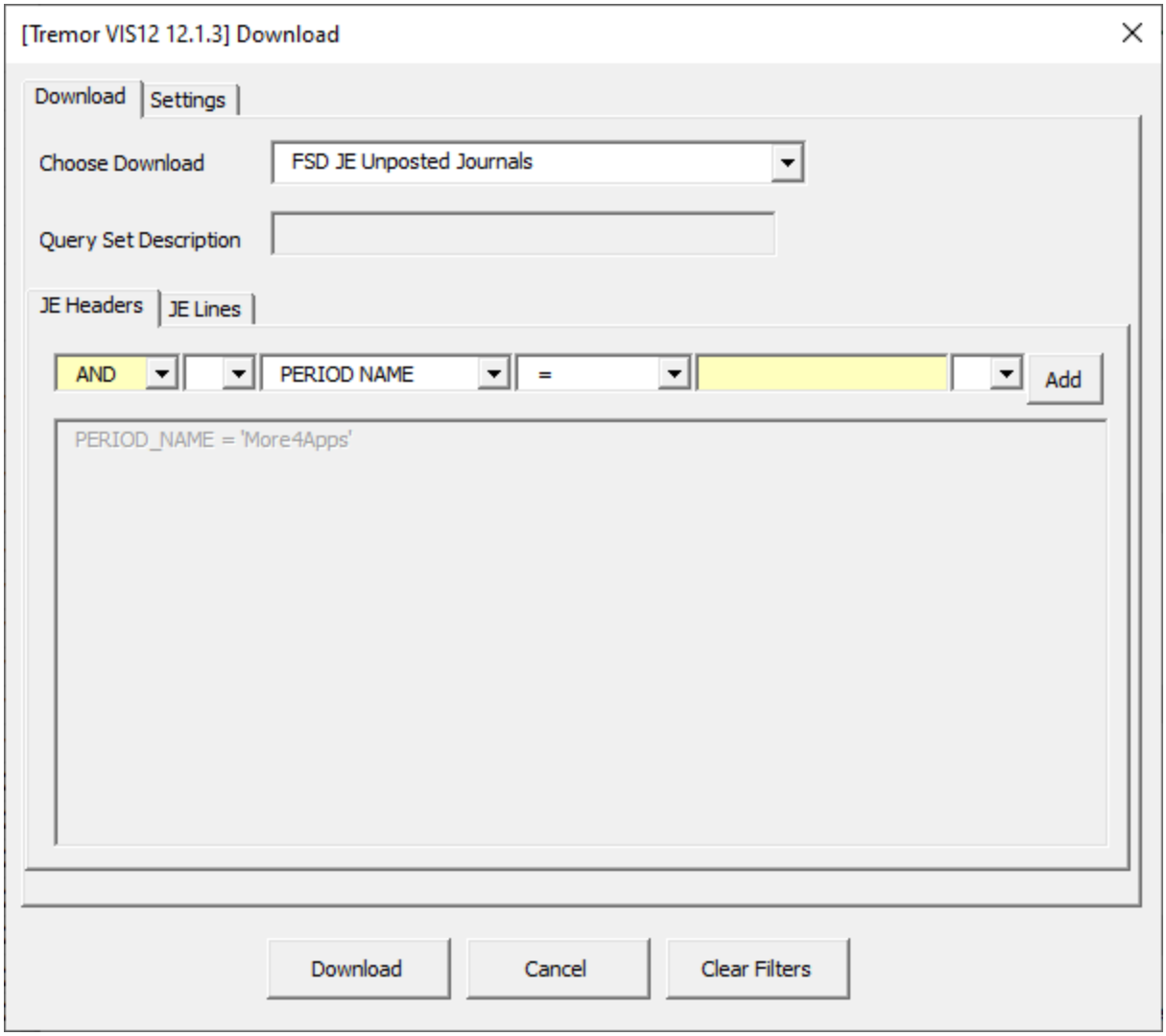
2. Choose the Query Set from the dropdown list provided.

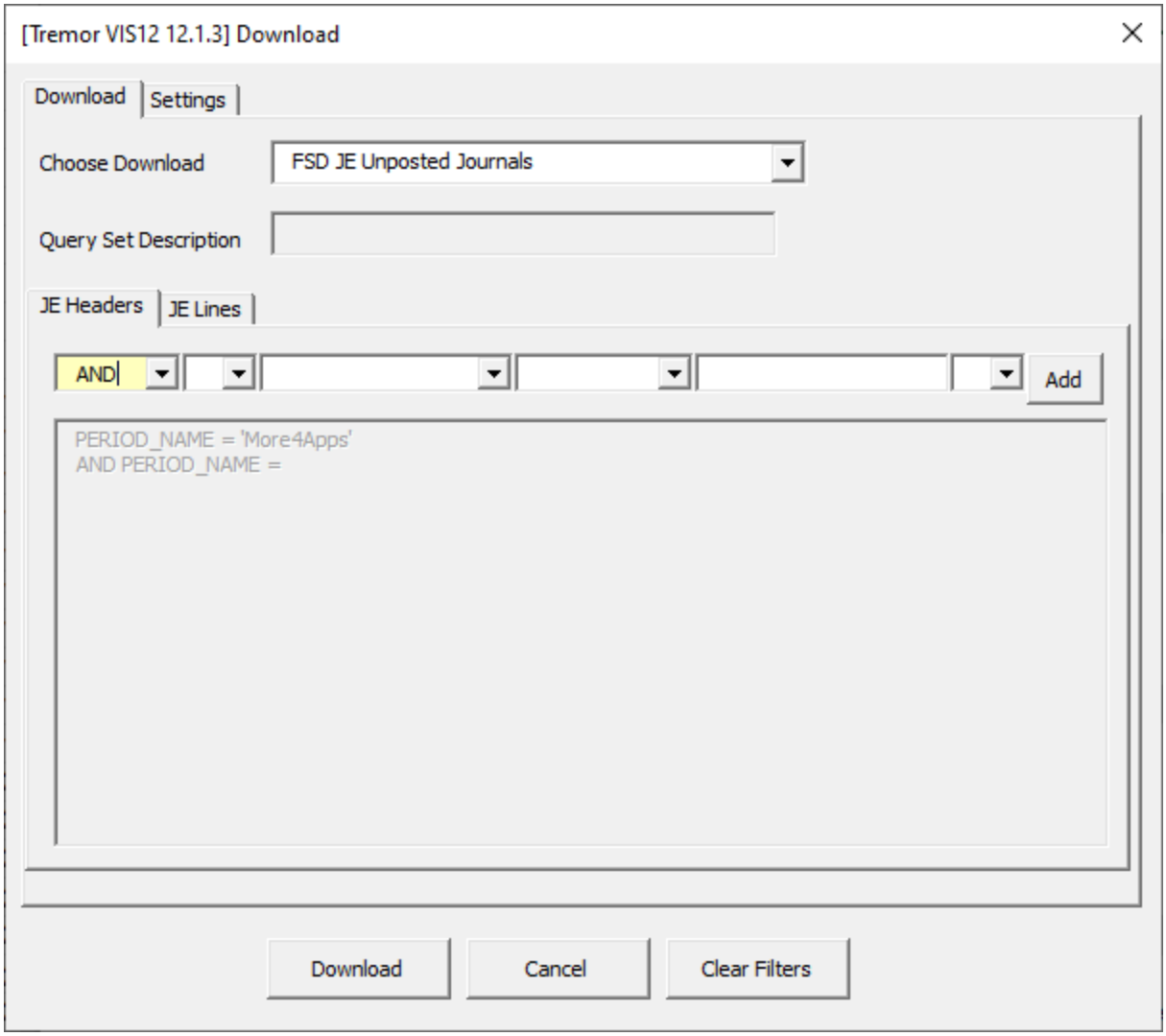


3. On the 'Settings' tab indicate if you will be using existing Column Headers and if so in what row they are held. If using your existing worksheet they will be in row 9.



4. Add a filter first and then complete the appropriate details in the filter fields for each query and click the 'Add' button to create the run-time filter.


[Tremor VIS12 12.1.3] Download
Download
Settings
Choose Download
FSD JE Unposted Journals
Query Set Description
JE Headers
JE Lines
AND
PERIOD NAME
=
Add
PERIOD_NAME = 'More4Apps'
Download
Cancel
Clear Filters



5. Click the 'Download' button to retrieve the records based on your selected Query Set.

Note - To ensure data is downloaded into the appropriate columns in the sheet, ensure your System Administrator uses column aliases in the select statement that map exactly to Wizard columns.

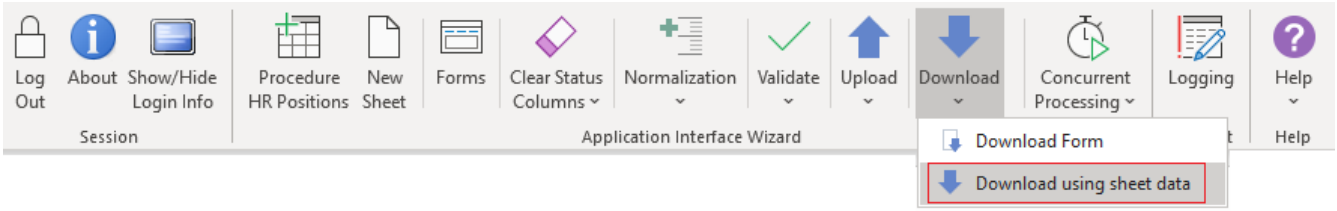
For example, this Wizard column can be aliased using either of these formats:

select name "Batch Name" from gl_je_batches
select name batch_name from gl_je_batches

Refer to section:
Custom Query - Download from Sheet
Custom Query - Print Column Headers
Examples of Custom Query used in PO Wizard

Custom Query - Download from Sheet

Use the 'Download using sheet data' option to select a pre-defined query to run where you want to use additional filters by using data entered in Wizard columns.

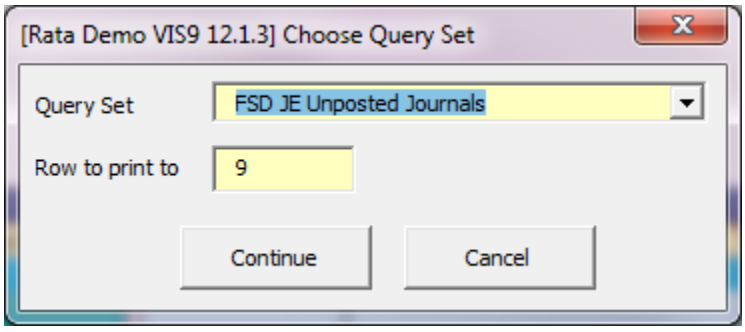


Note - If you are initiating the download using the 'Download using sheet data' option from the primary template download (see illustration below) please follow steps 2, 3 and 4.

1. If the worksheet doesn't already have the column headings, follow the step below, otherwise skip to Step 2.

   Select 'Print Column Headers' option from the Custom Query ribbon.

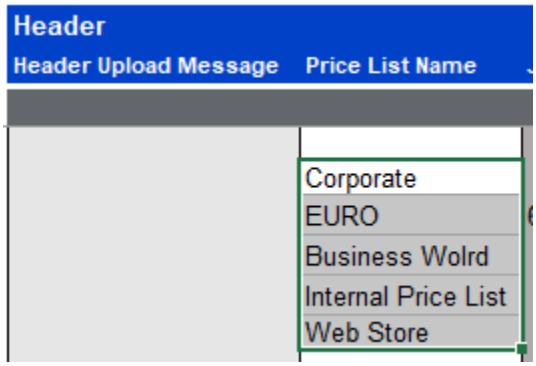
   Select the 'Query Set' and confirm the row number to print the headings on:



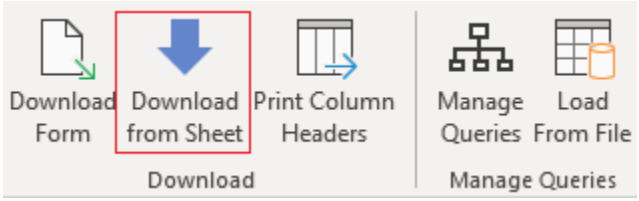
   Click 'Continue' to print the column headings from your Query Set into the nominated row.
   Note – If copying into a More4apps Wizard, the Column Headings MUST be on Row 9.

2. Click on a blank row below the column headings. Be sure to click below the 'Default Values' (row 10) if you are intending to download the data into a More4apps Wizard template.

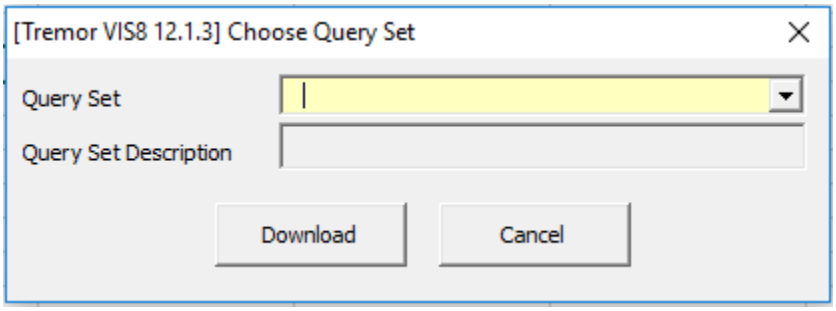
3. If you want to apply a filter to your download, enter the details in the appropriate column of your worksheet and highlight the rows covering these details as shown in the following screenshot:



4. Click the 'Download from Sheet' button on the Custom Query ribbon.



5. Choose the appropriate Query Set and then click on the 'Download' button.



Note – The column names used by the query are dictated by the 'Setting' tab of the 'Download Form' as illustrated below. This should be 9 if you are downloading to a Wizard template.

Refer to section:
Custom Query - Download Form

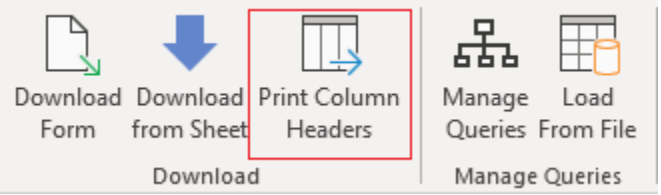
Custom Query - Print Column Headers
Examples of Custom Query used in PO Wizard

Custom Query - Print Column Headers

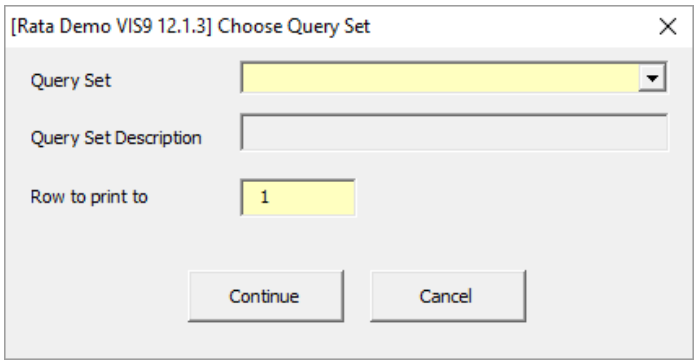
Use the 'Print Header' option to write the column names from the selected query into a worksheet. This can be any worksheet and doesn't need to be in a Wizard format. It is a useful method to first create column headers and then write data into the same sheet using either of the above 'Download' buttons.

1. Create a new Worksheet.

2. Click the 'Print Column Headers' button.



3. Select the appropriate Query Set and Row number to print the headings in.



4. Click 'Continue' to write the column headings.
   Note – Column headers are pasted into the nominated row commencing in column 'A'.

5. You can now use the download options to write information into the sheet.
   Note the following:

   (a) You will need to have your cursor below row 10.
   (b) If you are using a 'Non Wizard' worksheet then you will need to dictate where you have created the headings. For example in the screenshot above the headings will have been written to row 1 so you need to ensure the query uses this row to determine the column names. This is provided on the 'Settings' tab of the Download form.

Refer to section:
Custom Query - Download Form

Custom Query - Download from Sheet
Examples of Custom Query used in PO Wizard

Examples of Custom Query used in PO Wizard

Examples where the Custom Query would be beneficial to users are described below:

Data Conversion
For conversion purposes, the custom query functionality is ideal as you can select all outstanding Purchase Orders to identify those that are yet to be billed or have only been partially billed.

The query criteria can also include those that have been partially receipted or yet to be receipted. Then you can use the Close PO functionality to close off these lines and create new lines in the new E-Business instance for the outstanding amounts.

You may have other criteria that you wish to use to identify Purchase Orders to be converted to your new E-Business instance.

Open Purchase Orders
The custom query functionality is handy for finding Purchase Orders that failed to close due to receipting and invoice tolerances that are negligible in an environment that has zero tolerance.

It also enables you to write a query to find all Purchase Orders that are either open for invoicing or receiving where the quantity is receipted and the quantity billed is not equal to the quantity ordered, or some variation of these conditions.

You can then download the list of the Purchase Orders into the Close PO template and close all of them off in one action.

The example below provides SQL queries and a step-by-step guide for creating a simple query set for downloading Blanket Header's and associated Lines:

Blanket Header Query

```
SELECT DISTINCT h.po_header_id Header_ID
  ,h.segment1 PO_Number
  ,h.vendor_name Supplier
  ,h.vendor_site_code Site
  ,h.blanket_total_amount Amount_Agreed
  ,h.start_date Header_Effective_Date
  ,h.end_date Header_Expiration_Date
  ,h.amount_limit Amount_Limit
  ,h.GLOBAL_AGREEMENT_FLAG Global
FROM po_headers_v h
WHERE type_lookup_code = 'BLANKET'
```

Blanket Line Query

```
SELECT l.po_header_id HeaderID
  ,l.po_line_id LineID
  ,l.item_description Line_Description
```

```
  ,l.unit_meas_lookup_code UOM
  ,l.line_num Line_Number
  ,l.line_type Line_type
  ,mtl.concatenated_segments Item
  ,cat.concatenated_segments Category
  ,l.unit_price Price
  ,l.quantity Line_Quantity
FROM po_lines_v l
,mtl_system_items_kfv mtl
,mtl_categories_kfv cat
where
mtl.inventory_item_id(+) = l.item_id
 AND mtl.organization_id(+)   = 204
 AND cat.category_id(+)       = l.category_id
```

Refer to section:

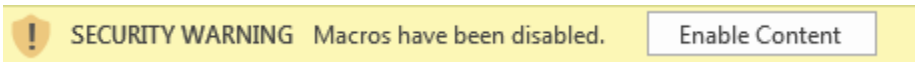
Custom Query - Download Form
Custom Query - Download from Sheet
Custom Query - Print Column Headers

Getting Started

This section provides a new user with tips on how to get started using the Wizard.

1. **Enable Macros** – The worksheet requires the macros to be enabled.

   When the worksheet is first opened on the menu bar the following will appear. Click on 'Enable Content'.



2. **Unable to see Wizard Ribbons -** If this occurs then your security settings are too high. Application Interface Wizard is written in VBA, and you must allow macros to run before you can run the program. Close the worksheet from the Excel menu then navigate to the following based on your version of Microsoft:

   Navigate to the 'Developer' tab and in the 'Code' group click 'Macro Security'. In 'Macro Settings', select the 'Disable all macros with notification'. Re-open the worksheet and you should receive the message shown above.
   Note - If the 'Developer' tab did not display, click File > Options > Customize Ribbon. In the 'Main Tabs' section, select 'Developer' and click 'OK'. You should now be able to go back to the 'Developer' Tab to make the security changes.

   Refer to section: Enable Macros/Cannot Access Wizard Ribbons

3. **Wizard Ribbons –** Please see the note below for details on the ribbons provided by the Wizard.
   Refer to section: Ribbons

4. **Trial Version** - If you are using the trial version it will require you to log out after five records have been downloaded or uploaded. You will need to log in again to continue. There are no other constraints in the trial version.

5. **Row 9** - The column names are stored in row 9. These are fixed and cannot be changed. Do NOT insert or delete rows before row 11 (Row 10 Default Values row). The case of the column names is not important but spelling is. If you change the column name in row 9, you have in effect deleted the column as far as the program is concerned. You can move the columns anywhere you like, hide, or even delete them if they are not required.

6. **Row 10** - This row is used to enter Default Values. Any data here will be defaulted into the same record where no data exists in the worksheet.
   Refer to section: Default Values in Row 10

7. **Column Naming** – Use the Templating functionality to rename your columns. You can move the columns anywhere you like, hide, or even delete them if they are not required.

8. **Populate the Worksheet** - There are a number of options available to populate the worksheet:

- You can type or paste the relevant values directly into the worksheet (no validation is performed until you run the upload program).

- You can download existing data and modify the information to update the record or you can download existing data and use this as the starting point to create new records.
  Note - If you are downloading an existing record to create a new one you need to remove the Ids of the existing records.

Refer to section: Enter Information into the Worksheet

9. **Working with More4apps Specific Loaders** - If More4apps has supplied a specific loader for your use, refer to this section for instructions on how to start using it.
   Refer to section: Installing More4apps Specific Loaders

10. **Troubleshooting** - If you are having any problems using the Wizard, click on the link to search our online resources https://community.more4apps.com/s/

If you encounter any problems or have any suggestions, do not hesitate to contact us at https://community.more4apps.com/s/ebs-contact-support

Installing More4apps Specific Loaders

If More4apps has supplied 'Specific Loaders' for your business, follow the instructions below to install and start using them:

1. Complete the appropriate instructions for running the install scripts for the Wizard and setting up the list of instances as described in the Installation and Configuration Guide.

   The Readme file supplied in the zip file should explain what needs to be installed on the server.

2. Import the More4apps Template.

   (a) Log into the Wizard as a user who has the System Administrator responsibility.
   (b) Select 'Import Template' from the Application Interface Wizard ribbon.
   Note - If the supplied template requires packages, the Wizard will check they have been successfully compiled during this step.

   Refer to section: Importing Templates

3. Select the new Template.
   Important: The template must be selected before adding licence keys and named users, otherwise these will be added to incorrect products.

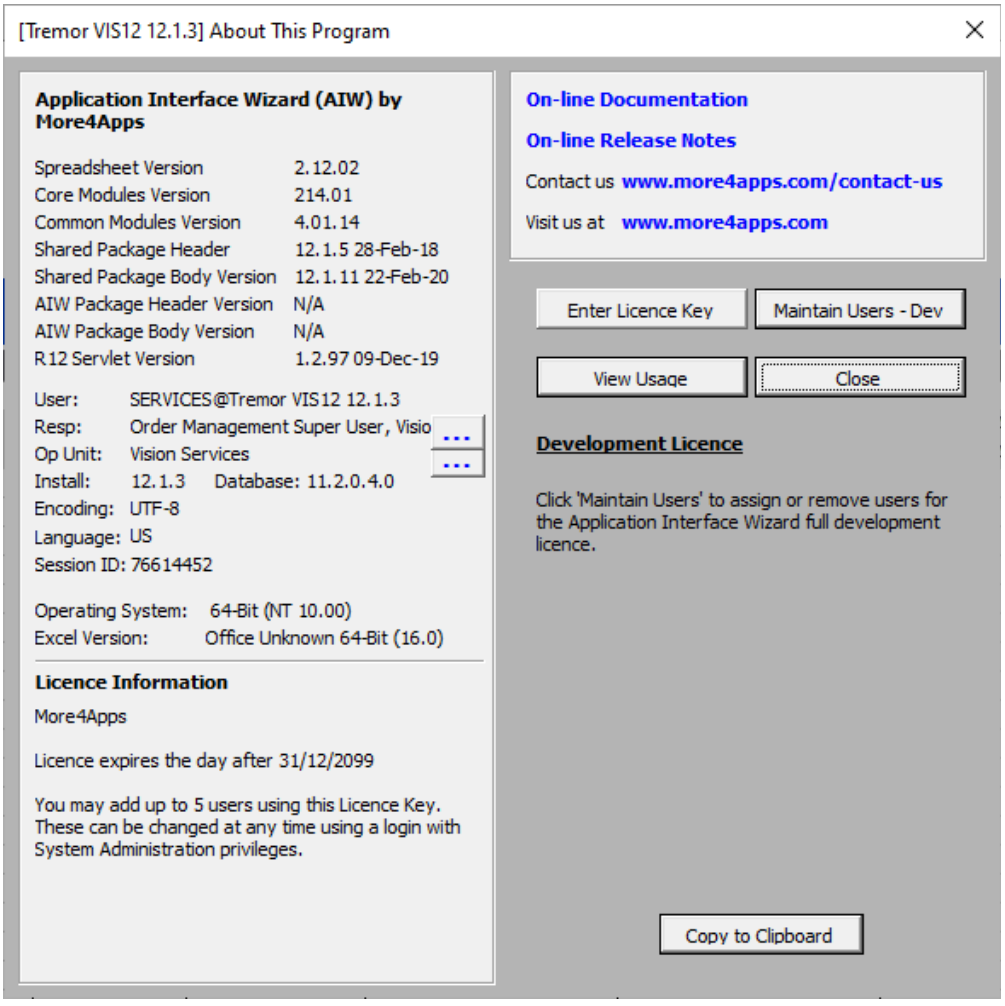
   Refer to section: Open the Template Worksheet

4. Enter the Licence Key and Named users for the Specific Loader:

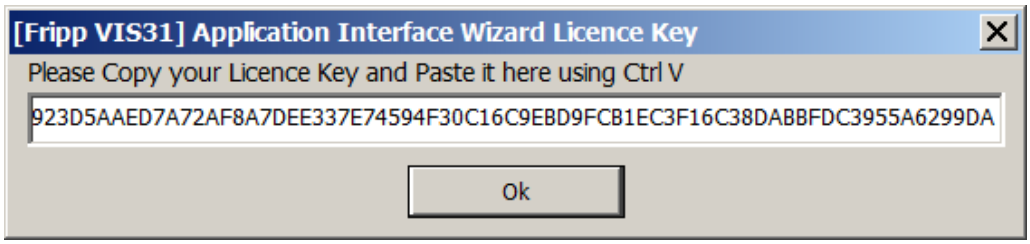
   (a) Click 'Manage Templates' on the Application Interface Wizard ribbon.

(b) From the 'Action' field LOVs, choose 'Modify' and select the 'Template Name'. Click the 'Loader Users' button.

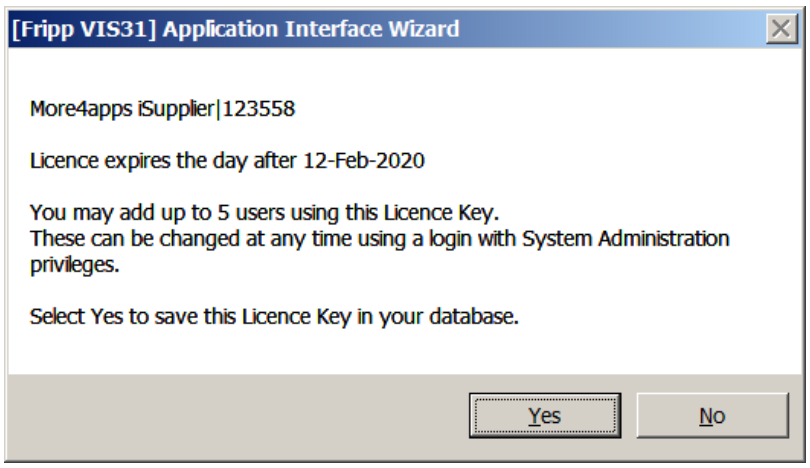
The 'About This Program' form will display:



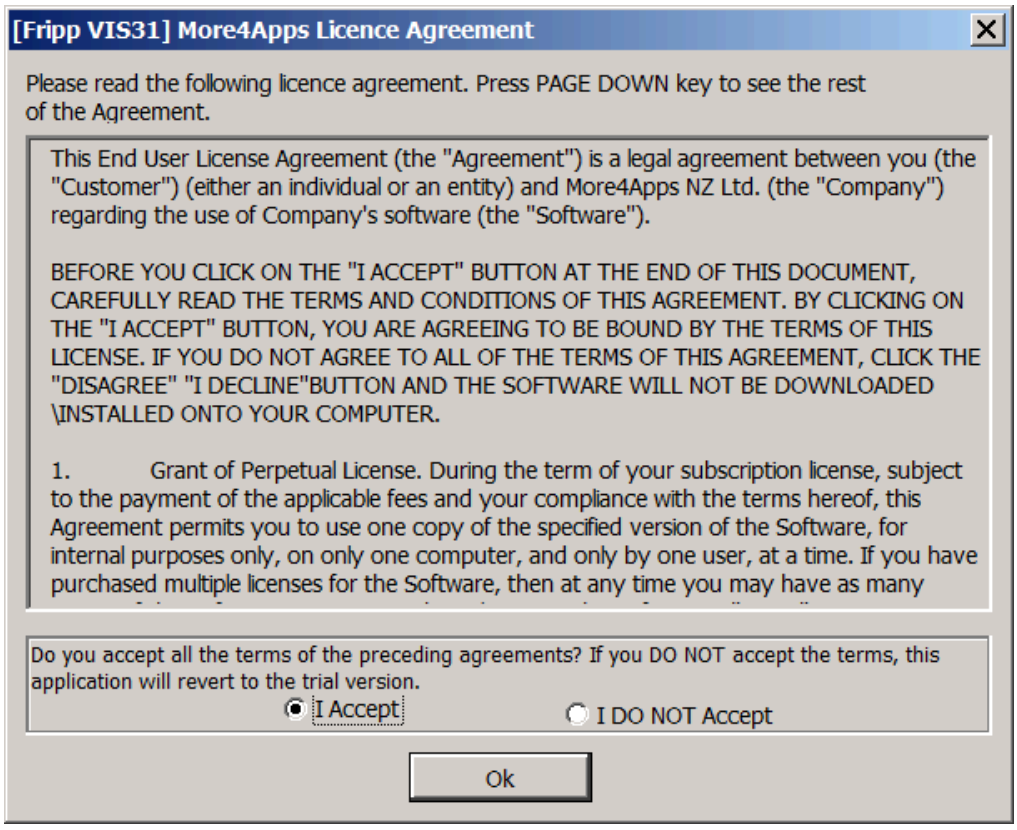
(c) Click the 'Enter Licence Key' button and paste the supplied key into the field and click 'Ok'.



(d) Select 'Yes' to save the Licence Key in the database.

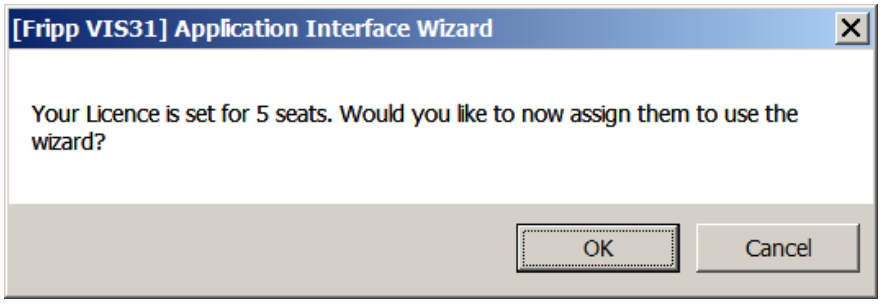


The Licence Agreement is displayed. To activate the loader, read and accept the terms by clicking in the 'I Accept' radio button and then click 'OK'.

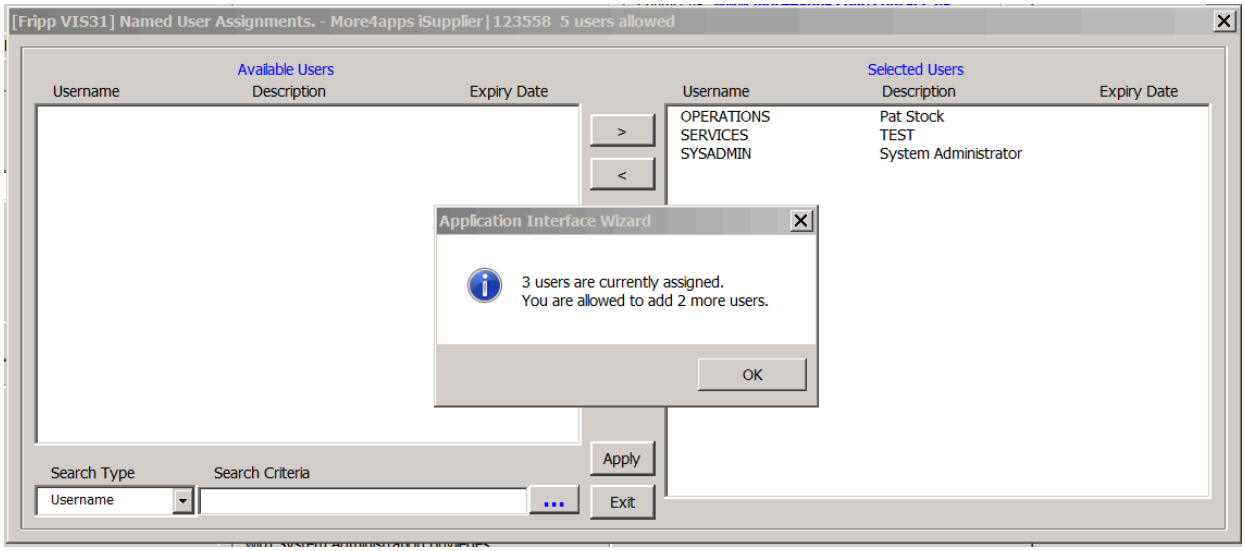


(e) A message is displayed advising that the Licence Key has been stored. Click 'OK'.

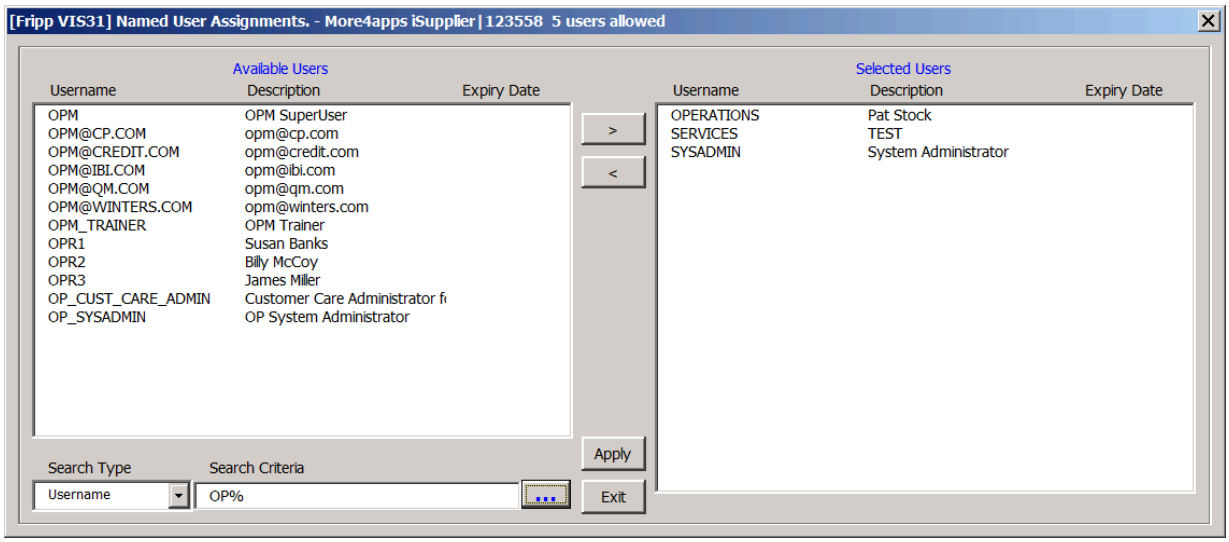
(f) Then it will ask to assign the users for the selected template/loader. Click 'Ok' to assign user(s).



(g) If any existing users are already assigned then it will be shown in the right-hand pane. The message will also state how many users' slots are remaining. Click 'Ok'.

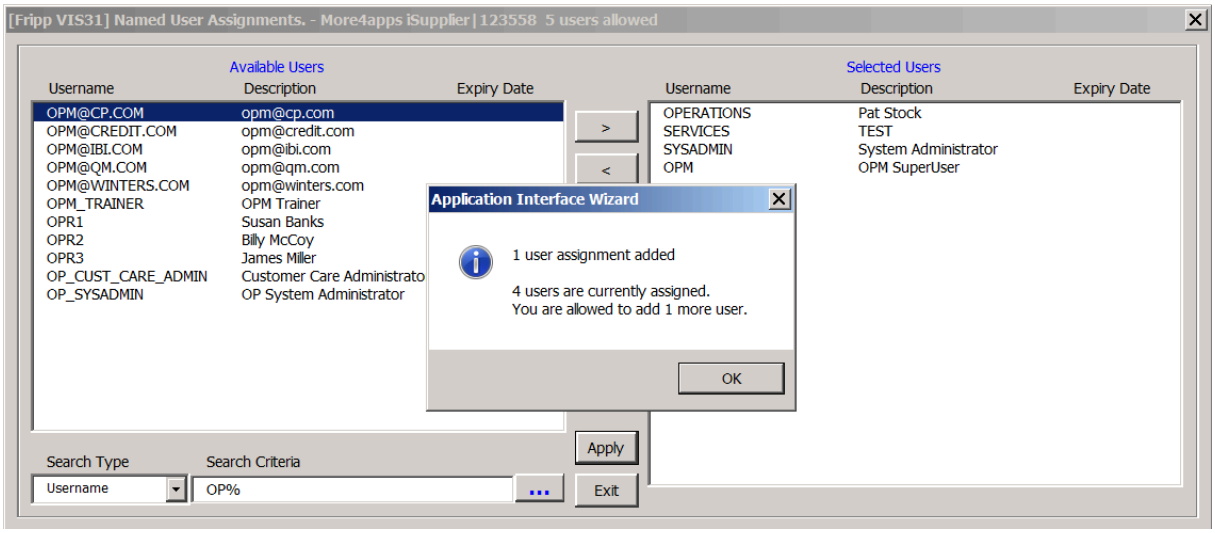


(h) Enter the full username or part of the username proceeding with a wild card (%) sign at the end and click three dot icon next to 'Search Criteria' text field to see the list of available users.



(i) Select the username from the left pane to assign or the right pane to remove the loader access followed by clicking the respective buttons below.

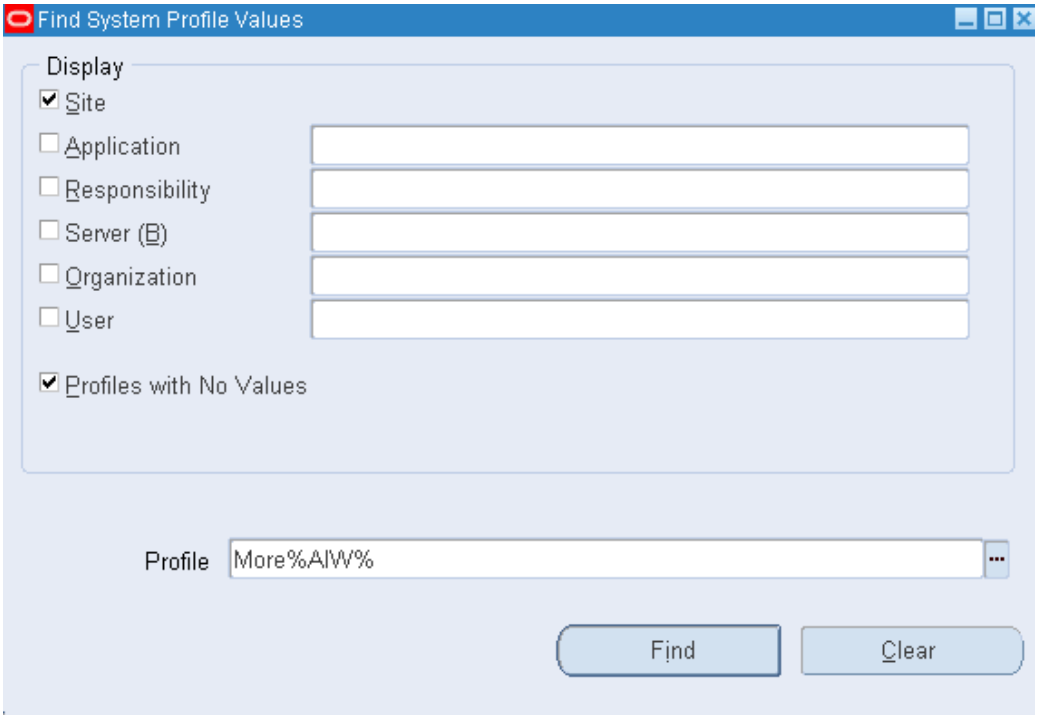
(j) Once all users have been specified or removed, click the 'Apply' button. A window will appear with new user assignment setup.



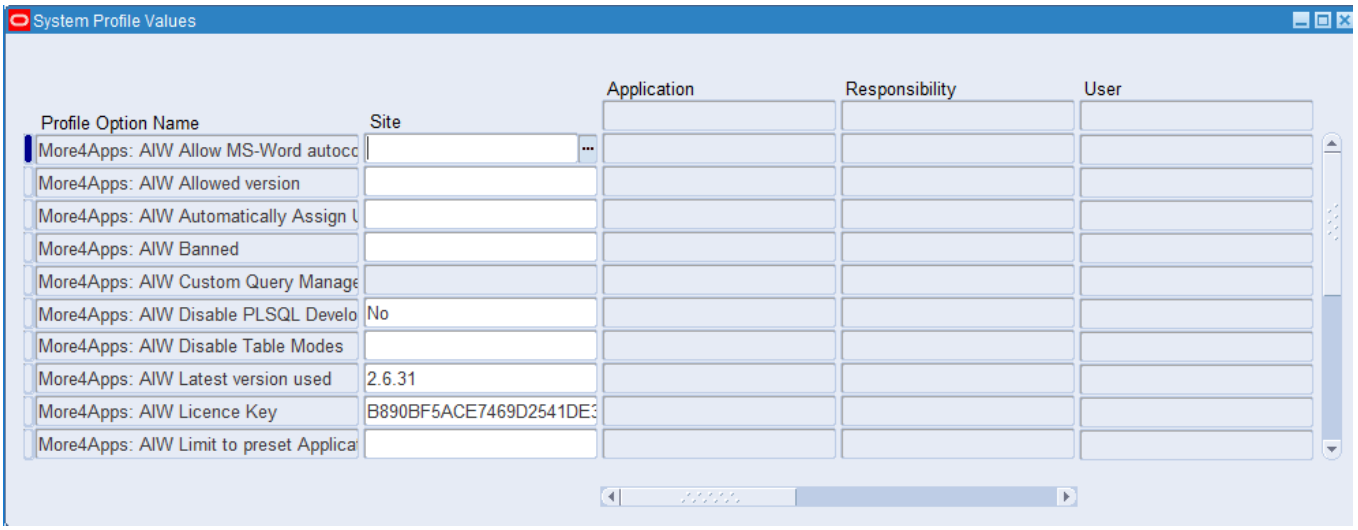
(k) Click 'Ok' and click 'Exit' to leave this form.

Profile Options

Profile options are common to all of our tools. To find the profile options applicable for this Wizard, navigate to System Profile and enter **More%AIW%** as shown in the screenshot below, then select the 'Find' button.



The profile options applicable for this Wizard will show on the 'System Profile Values' form as shown below.



Many profile options can be set at the Site, Application, Responsibility or User level. Modify your search in the 'Find System Profile Options' form to set options at the appropriate level.

Profile Options are installed when you log into an instance (from the Wizard) for the first time. If a profile option is not available then you will need to reinstall your profile options. To do this, refer to the Reinstall Profile Options available in the Common User Topics documentation.

Common Profile Options

The following profile options are created for each of our products:

- *More4Apps: AIW Allow MS-Word autocorrect characters* - This profile can be set to prevent MS-Word AutoCorrect characters (which are non-ASCII) from being loaded – If these are loaded it can cause data corruption.

- *More4Apps: AIW Allowed version* - This profile option is similar to 'More4Apps: AIW Minimum version' but is more restrictive. It will stop any version other than the listed version from operating.

- *More4Apps: AIW Automatically Assign User* - If this is set to 'Yes' new users will be automatically assigned up to the maximum allowed by the licence key.

- *More4Apps: AIW Banned* - If this profile option is set to 'Yes' the Wizard will not operate. This can be set at the site, responsibility or user level.

- *More4Apps: AIW Custom Query Manager* – This profile allows customers to nominate a user to manage download queries. This is useful if the user lacks sysadmin access and cannot be granted sysadmin. The profile is set up at the user level.

- *More4Apps: AIW Latest version used* - This profile option is for internal use only. It records the highest version number used.

- *More4Apps: AIW Licence Key* - This profile holds the licence key for the Wizard. If you have no licence key then this field will hold 'Trial Version' as the licence key and some functionality will be limited. For more details on licence keys refer to the following Installation and Configuration Guide: Enter the Licence Key.

- *More4Apps: AIW Limit to preset Application ID* - This profile option is NOT applicable to Application Interface Wizard. Please ignore it.

- *More4Apps: AIW Minimum version* - This profile option allows you to record the minimum spreadsheet version number allowed. If the version number of the spreadsheet being used is less than the recorded value then the Wizard will not operate.

- *More4Apps: AIW Named User* - This profile enables you to control the named users for Application Interface Wizard and is set at the user level in Oracle. For more details on how to set up named users refer to the following Installation and Configuration Guide: Add Named Users.

- *More4Apps: AIW Nominated Sysadmin for user maintenance* - This profile can be set to allow a nominated Oracle user name which doesn't have System Administrator profile value access to maintain users for this product. Other users that do have System Administrator profile access can still maintain product users if this profile is set to a different person.
  Note - This profile option is only available for clients with Wizard versions containing 'Common Modules' version 204 and greater.

- *More4Apps: AIW Suppress Custom Query Functionality* - This functionality can be suppressed by setting the profile option to 'Yes' at the user, responsibility or site level. When this profile option is set (at the appropriate level) selection of any custom query icon will produce a message of 'The custom query functionality has been disabled by your System Administrator. Please contact your System Administrator'.

- *More4Apps: AIW Template Manager* – This profile allows customers to nominate a user to manage templates. This is useful if the user lacks sysadmin access and cannot be granted sysadmin. The profile is set up at the user level. To enable sysadmin access for templating select 'Yes' for the current user.

- *More4Apps: AIW Unicode Encoding Option* - This profile determines if character sets used by different applications are converted. By default this is set to 'Automatic' and should not be changed unless advised by More4Apps Support.

- *More4Apps: AIW Validate Only* - This profile option is NOT applicable to Application Interface Wizard. Setting it will have no effect.

Application Interface Wizard - Specific Profile Options

There are no Application Interface Wizard specific profile options; however there may be encoded licence key profiles for specific loaders which look similar to the example profile option below:

- *More4Apps: AIW Disable PLSQL Development* - This profile option disabled the use of PLSQL validation in the edit by sheet and manage templates forms. If set to 'Yes' this will disable the functionality. If set to 'No' or blank this will enable the functionality.

- *More4Apps: AIW|XXXXX Licence Key* - This is the specific licence key for a More4Apps supplied template loader.

- *More4Apps: AIW|XXXXX Named User* - This profile enables you to control the named users for Application Interface Wizard and is set at the user level in Oracle. For more details on how to set up named users refer to the following Installation and Configuration Guide: Add Named Users. The XXXXX in this profile refers to the MD5 template number generated when creating a loader.

- *More4Apps: AIW Disable Table Modes* - This profile option is used to disable the ability to create a loader that is a direct table insert/update or delete, therefore limiting the loader creation options to an EBS Attachment or Procedure. When set to 'Yes' this will remove the 'Insert' and 'Delete' from the table modes list of values. If set to 'No' or blank this will not limit the list of values for table modes.

- *More4Apps: AIW User Password Update Allowed* - A password can be assigned to any new user. The update to any password is regulated by the profile "More4Apps: AIW User Password Update Allowed". This must be set to 'Yes' before you can update a user's password. This is a free text field.

Troubleshooting

If you are having any problems using the Wizard, click on this link to search our online resources https://community.more4apps.com/s/

If you are unable to find a solution or have any suggestions, please contact us at https://community.more4apps.com/s/ebs-contact-support/

Note: Due to the complex nature of this Wizard and how it can be used, we would prefer to also receive the Wizard workbook containing data and all error messages incurred at the time the issue presents. This will help us work towards solutions much quicker as the Workbook contains valuable version and instance information to assist our support team.

Licence Keys, Named Users & Installation

Pre-built Loaders vs Full Licence
Application Interface Wizard is a rich loaders platform aimed at empowering System Administrators to create their own Loaders for their user community.

There may be times however where More4apps has been commissioned to provide a pre-built supported Loader to meet unique business requirements. If this is the case the Wizard licence will only allow those specific loaders to be used and developers will be unable to create their own Templates without a full licence.

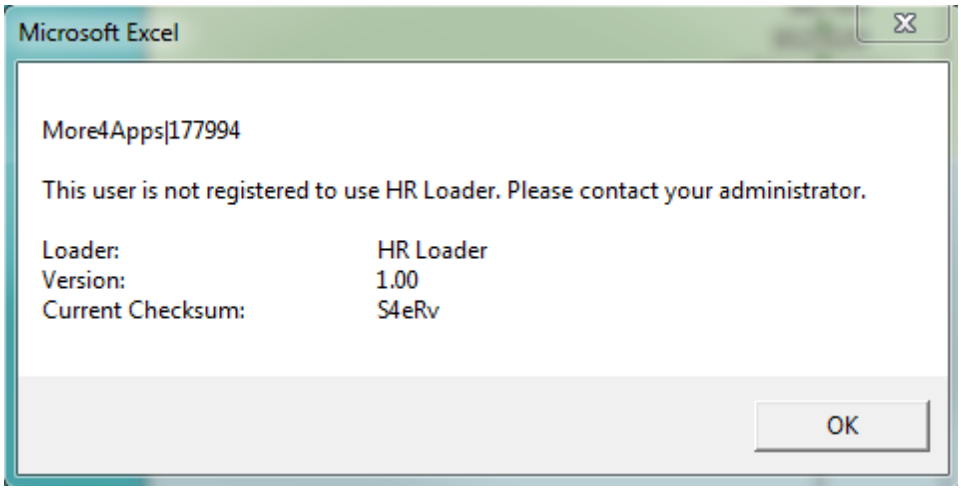
Contact More4apps for more information should your licence requirements change.

Licence Key and Named Users
The instructions for setting up the list of instances, using licence keys and adding named users for the full licence version of Application Interface Wizard are in the Installation and Configuration Guide.

If More4apps has supplied Specific Loaders for your use, please refer to the detailed process for creating these in this section: Installing More4apps Specific Loaders

Specific Loader Licence Keys
The Loader Licences work in the same way as the Wizard Licence Keys. If the key has expired the loader reverts back to the limit of five records per upload, as if the key were a trial licence.

A Wizard licence will enable all Wizard features to the user, but will not include any additional custom loaders made by More4apps, unless the loader licence has been activated. For example:



If the Wizard licence is not active or has expired but the loader licence is still current, users are only able to access that specific loader, and the system administrator will not have the freedom of creating their own templates as they would with a full Wizard licence.
Refer to section: Installing More4apps Specific Loaders

Specific Loader Named Users
Licence keys contain a limited number of users that can be assigned to use the Wizard at any time. Users assigned to Specific Loaders my not have access to the full Wizard functionality. If they require access to the Full Licence and Specific Loaders supplied by More4apps it is important that users are assigned to use both by the System Administrator.
Refer to section: Installing More4apps Specific Loaders

The system administrator must also make sure that the user has access to at least one of the responsibilities assigned to the template that the user will be expected to work in.

Oracle EBS Connection Method
This connection method is recommended for use with Oracle E-Business Suite (EBS).
This connection method is not available for use with non-EBS databases.
This method connects to the Oracle database via the intranet, using a similar web connection technology to that used in the Oracle EBS forms. Two packages will be installed into the database, and for Release 12 a Java Servlet will also be required.
Users will log in using their Oracle EBS Username, Password and Responsibility.
Click on this link to navigate to the Installation and Configuration Guide

Release Notes

Click on this link to navigate to the Application Interface Wizard Release Notes

This user documentation is based on Application Interface Wizard Release Version 3.02.19