Excel Interface (XLI) For Micro Focus

PPMetrics – Request Management Configuration

Document Version: 3.5
Document Release Date: March 2022

Table of Contents

1. Introduction

➤ **In this document, for brevity, the term " XLI" is used to refer to the Excel Interface Add-on for Micro Focus**

This document provides information about configuring the request management module of the XLI. It was written for:

- System and PPM instance administrators
- PPM technical support personnel

2. XLI Request Management Overview

The goal of the XLI's request management functionality is to provide an alternative user experience for updating OOTB or custom PPM requests and projects. It is normally used for use cases such as mass creation, mass updates, team collaboration, and replacement of back-end updates by admin. It is not intended to replace the PPM request interface completely as it lacks workflow capabilities, attachments management etc.

➤ It is also feasible to link any custom Excel template to PPM Demand Management using the XLI Request Management module. Please contact your PPMetrics representative for details.

3. XLI Request Configuration Overview

At a high-level, the process of configuring PPM request types to be available through the XLI consists of the following steps:

1. Determining the request types that should be supported
2. Updating a single XLI database table (XL_SUPP_REQ_TYPES) with information about these request types
3. As needed, updating several additional XLI configuration tables in situations where the XLI's default retrieval of request type fields need to be overridden or supplemented
4. Downloading and updating requests through the XLI. Note that whenever an XLI configuration change is made to a request type the option "Refresh Fields Configuration" on the request search must be checked for the changes to become visible (it is checked by default when the XLI workbook it opened, or when the user switches request type selection).
5. Configuring custom filters (please refer to the installation and administration guide).

The list below summarizes request type field attributes that get automatically detected and downloaded by the XLI (without any field-level or rules configuration):

1. All enabled request type fields are brought over, except for attachments, passwords, staffing profiles, and financial data tables. Hidden fields may or may not be brought over, depending on the value of the parameter "Download Hidden Request Type Fields" in the USER_SETTINGS tab.
2. Read-only PPM fields are made read-only in XLI as well
3. Character limit automatically set
4. Values of drop-down and auto-complete-list fields with list based validations are brought over.
5. Field change rules do not get automatically inherited by the XLI but PPM does trigger them when updates are sent back.
6. Editing rights based on field-level-security.

The following field attributes and capabilities require XLI configuration:

1. Mandatory fields: Fields defined as mandatory are always made required when downloaded to the XLI, regardless of their request statuses.
2. Field Change Rules: When requests are updated or downloaded through XLI, PPM does fire field change rules but at times there is a need to trigger them locally in XLI. Note that when table components are updated using XLI, field change rules are not fired by PPM.
3. UI Rules: When requests are updated or downloaded through XLI, field attributes at the request level may be set using XLI UI rules. 4. Default column widths
5. Special numbers/dates formats
6. SQL based drop downs and auto-complete-lists
7. Partial download of fields: It is sometimes desired to only download a subset of the request type fields to the XLI
8. Editable PPM fields that need to be made read-only in XLI
9. Specific order of processing fields as they are sent to PPM 10. Field validations not inherited by default (e.g. MIN/MAX values)
11. Viewing rights based on field-level-security.

4. Table Structure

In order to perform the XLI request management configuration, several XLI database tables residing in the PPM database schema have to be updated.

- XL_SUPP_REQ_TYPES: This table contains one or more records for each PPM request type that needs to have a corresponding XLI version.

Field Name	Description	Data Type and Valid Values	Required	Default	Example
SUPP_REQ_TYPE_ID	Unique record identifier	Number	Y		
PPM_REQ_TYPE_ID	PPM request type ID as in the KCRT_REQUEST_TYPES table. Note that this column is not unique, allowing you to configure multiple XLI versions of a given PPM request type	Number	Y		
DISPLAYED_REQ_TYPE_NAME	The name of the request type as displayed to XLI users	Text (80)	Y		
ADMIN_ONLY_FLAG	Determines whether the request type should or should not be visible to XLI users who do not have the PPM user administration or configuration license	Text (1) Y/N	Y		
PROJECT_FLAG	Determines whether the request type is a PPM project (uses the PFM – Project field group).	Text (1) Y/N	Y		
DESC_FIELD_SOURCE_SQL	The data source of the default “Description” field on the XLI request search form. Use only if the “Description” field on this form needs to be replaced with a different filter.	Text (4000)	N		(SELECT prj_project_name FROM kcrt_fg_pfm_project WHERE request_id = req.request_id)

DESC_FIELD_PROMPT	The prompt of the filter field to replace the default “Description”. Use only if the “Description” field needs to be replaced with a different filter.	Text (200)	N		Project
DESC_FIELD_FILTER_SQL	The condition that is to be applied when the user performs a search using the overwritten “Description” field. Use only if the “Description” field needs to be replaced with a different filter.	Text (4000)	N		AND EXISTS(SELECT 'Y' FROM kcrt_fg_pfm_project WHERE request_id = req.request_id AND UPPER(prj_project_name) LIKE UPPER('%#?#%'))
GEN_REQ_FILTER_SQL	General filter that is applied to all searches performed (hidden to the user).	Text (4000)	N		AND req.priority_code = 'HIGH'
FREEZE_COL_LETTER	Determines the Excel column that should be frozen when requests of the type are downloaded. If left blank, the XLI uses column “E” as a system default.	Text (2)	N		F
CREATION_ONLY_FLAG	Determines whether to prevent users from creating/deleting requests through XLI. If set to “Y”, the “Search” button becomes disabled as well as all filters, and the “Delete” column is renamed to “Ignore”.	Text (1)	N		N

TABLE_COMP_ID_FIELD_SQL	For table components only, specifies the database column use to identify each table record downloaded by users. May reference the following aliases: kcrt_requests req, kcrt_req_header_details rhd, kcrt_request_details rd If left blank, the XLI will use the field req.description as the identifier.	Text (4000)	N		rhd.visible_parameter8
CUST_TEMP_FILTER_SQL	For custom Excel templates only, the condition that is to be applied when the user performs a search. Use only if the “Description” field needs to be replaced with a different filter.	Text (4000)	N		AND (EXISTS (SELECT 'Y' FROM kcrt_fg_master_proj_ref WHERE request_id = req.request_id AND UPPER(ref_master_project_name) LIKE UPPER('%#?#%')) OR req.request_id LIKE '%#?#%')
CUST_TEMP_FILTER_PROMPT	For custom Excel templates only, the value of this parameter replaces the default label on the search text box	Text (200)	N		(Enter any part of request ID or project)
CUST_TEMP_SOURCE_SQL	For custom Excel templates only, the data source of the default “Description” field on the XLI. Use only if the “Description” field needs to be replaced with a different filter.	Text (4000)	N		(SELECT ref_master_project_name FROM kcrt_fg_master_proj_ref WHERE request_id = req.request_id)
EDIT_ONLY_FLAG	Determines whether request creation should be allowed for the request type	Text (1)	N		

- XL_REQ_TYPE_FIELDS: This table contains information about field attributes that supplement or override the PPM fields configuration

Field Name	Description	Data Type and Valid Values	Required	Default	Example
SUPP_REQ_TYPE_ID	Foreign key to the field SUPP_REQ_TYPE_ID in XL_SUPP_REQ_TYPES	Number	Y		30122
PARAMETER_TOKEN	Full PPM token of the field. Used to join records in this table to PPM request type fields	Text (200)	Y		REQ.VP.P_RISK_AREA For table components, use the following format: REQD.VP.CI_TBL.CI_TYPE
REQUIRED_FLAG	Determine whether the field should be mandatory to the user	Text (1) Y/N	Y	N	N
VALIDATION_SQL_HIDDEN	Not Currently used	Text (4000)	N		

VALIDATION_SQL_VISIBLE	Populate for pop-up fields (POPUP_FLAG = 'Y') with the query behind each field's popup form. Must return a single field value and reference the user's search string using #?#. For fields returning a large number of records it is critical to limit the result set with the clause WHERE ROWNUM < ([MAX_MULT_ROWS] + 1)
	
	Alternatively, populate this field for drop down fields with SQL queries as their source (POPUP_FLAG = 'N'). In these cases, the query should return a single record with the values delimited by $@$.
	
	Note that these queries may contain references to other fields or even various Excel cells (similar to the concept of tokens in PPM). Please refer to the tokens section of the installation and administration guide.
		Text (4000)	N		Pop-up field:
					
					SELECT flds FROM (SELECT usr.full_name flds FROM knta_users usr WHERE (usr.end_date IS NULL OR usr.end_date > SYSDATE) AND UPPER(usr.full_name) LIKE UPPER('%#?#%') ORDER BY 1) WHERE ROWNUM < ([MAX_MULT_ROWS] + 1)
					
					Note that for table component fields only, you must format the return value as visible_value (#hidden_value) Example:
					
					SELECT *meaning || ' (#' || lookup_code || ')' FROM knta_lookups where lookup_type = 'ALM - Non Release Sub WF'*
					
					Drop-down:
					
					SELECT RTRIM (XMLAGG (XMLELEMENT (e, workplan_template_name || '$@$') ORDER BY workplan_template_name).EXTRACT ('//text()').getClobVal(), ',') FROM pm_workplan_templates

					WHERE is_enabled_flag='Y'
					ORDER BY workplan_template_name

POPUP_FLAG	Determines whether the field value may be populated using a pop-up form (similar to PPM's auto-complete-lists).	Text (1) Y/N	Y	N	N
NUMBER_FORMAT	The way by which Excel should format the value. Must be a valid Excel number format	Text (30)	N		m/d/yyyy = use to format date based on the user's locale
COL_WIDTH	The default width of the Excel column used to display the field	Number	N		30
DOWNLOAD_FLAG	Determines whether the field should be downloaded to the XLI template or ignored all together. If DOWNLOAD_FLAG = 'N', other column values become irrelevant	Text (1) Y/N	Y	Y	Y
HIDDEN_FLAG	Determines whether the field should be visible or hidden to the user.	Text (1) Y/N	Y	N	N
PROCESSING_ORDER	Determines the order by which the XLI should send updates to this field to PPM when other fields are updated at the same time. Note that PPM processes field-level request type rules in the order by which they're sent by the XLI, therefore only change the default value of 0 in situations where the request type's rule logic may be broken otherwise.	Number	Y	0	0

EXCLUSIVE_PROCESSING_FLAG	Determines whether field updates should be sent to PPM separately and prior to nonexclusive fields. Useful in situations where it is important to ensure that certain fields get updated and the request saved before others.	Text (1) Y/N	Y	N	N
DISPLAY_ONLY_FLAG	Determines whether a PPM editable field should be made read-only in XLI	Text (1) Y/N	N	N	N
VALIDATION_FORMULA1	Serves the same purpose as the first fields at the bottom of Excel's data validation settings screen	NUMBER	N		0 Another example: '=TODAY()
VALIDATION_FORMULA2	Serves the same purpose as the second field at the bottom of Excel's data validation settings screen	NUMBER	N		1000
LOCATION_SEQ	Determines the location of the field in XLI. If left blank, the XLI displays fields based on their order in PPM.	NUMBER	N		1
PROMPT_OVERRIDE	If filled out, replace the field's PPM prompt with the value entered	Text (200)	N		Project Description

VALIDATION_OPERATOR	If the request type column needs to have a special Excel data validation, this field should be used and corresponds to "data" field in Excel's data validation settings screen. The valid numeric values are listed here: https://msdn.microsoft.com/enus/library/office/ff840923.aspx	NUMBER	N		6
VALIDATION_ERR_MSG	Custom error message that is displayed to the user when the data validation fails. If left empty, the XLI uses Excel's default.	Text (400)	N		Please enter a value between 100 – 500
CUST_TEMP_XL_RANGE	For custom Excel templates only, this field is used to map a PPM field to a specific cell on the sheet of the custom template.	Text (200)	N		B11

➤ It is highly recommended to populate this table with all the fields of the PPM request types that need to have a XLI user interface as a starting point for the configuration work. The scripts below will populate this table with all the fields of the request type of choice (including table component fields). Since it only adds missing fields, you may safely run it whenever a field is added to the PPM request type.

```
-- standard fields
INSERT INTO xl_req_type_fields fld
SELECT [replace with XL_SUPP_REQ_TYPES.SUPP_REQ_TYPE_ID] supp_req_type_id,fldv.token,'N',NULL, NULL,
'N',NULL,20,'Y','N',0,'N','N', NULL, NULL, NULL, NULL, NULL, NULL, NULL
FROM xl_req_type_fields_v fldv
WHERE fldv.request_type_id = [replace with KCRT_REQUEST_TYPES.REQUEST_TYPE_ID]
AND fldv.component_type_code    IN (1,2,3,4,5,6,7,8)
AND NOT EXISTS (SELECT 'Y' FROM xl_req_type_fields WHERE supp_req_type_id = fldv.supp_req_type_id AND parameter_token =
fldv.token);

-- table component fields
INSERT INTO xl_req_type_fields fld
SELECT [replace with XL_SUPP_REQ_TYPES.SUPP_REQ_TYPE_ID]  supp_req_type_id,fldv.token,'N',NULL, NULL,
'N',NULL,20,'Y','N',0,'N','N', NULL, NULL, NULL, NULL, NULL, NULL, NULL
FROM xl_table_comp_fields_v fldv
WHERE fldv.request_type_id = [replace with KCRT_REQUEST_TYPES.REQUEST_TYPE_ID]
AND fldv.component_type_code    IN (1,2,3,4,5,6,7,8)
AND NOT EXISTS (SELECT 'Y' FROM xl_req_type_fields WHERE supp_req_type_id = fldv.supp_req_type_id AND parameter_token =
fldv.token);
```

• XL_REQ_TYPE_RULES: This table contains a list of field change rules that have to be replicated in the XLI environment

Field Name	Description	Data Type and Valid Values	Required	Default	Example
PPM_REQ_TYPE_ID	The ID of the PPM request type from KCRT_REQUEST_TYPES table	Number	Y		30220
RULE_ID	Unique identifier of the rule	Number	Y		60
RULE_NAME	Brief description of the rule	Text (150)	Y		Set Project ID
DESCRIPTION	Detailed description of the rule	Text (300)	N		Set project ID upon selection of a project
SEQ_NUMBER	The sequence by which the rule should be triggered relative to other rules of the request type	Number	Y		1

RULE_SQL	The SQL statement which should be evaluated and potentially triggered. Every result field must be enclosed by #@# and the WHERE clause of the statement represents the condition(s) for triggering of the rule. Dependent fields should be represented using their full token, including brackets. An alternative syntax for UI rules only is supported as well, which gets processed by reading data from the Excel file rather than the database, hence faster but more limited in functionality. Format: #@#[TOKEN]#@#[OPERATOR]#@#[VALUE]#@#[Y for AND N for OR]	Text (4000)	Y		SQL rule example: Populate a “Project ID” field based on the value of a “Project Name” field: *SELECT ‘#@#’ || request_id || ‘#@#’ FROM kcrt_fg_pfm_project fg WHERE ‘[REQ.VP.KNTA_MASTER_PROJECT_REF]’ = fg.prj_project_name* Alternative syntax example (Excel level processing): Trigger when the status of the request is ‘New’ *#@#[REQ.STATUS_NAME]#@#=#@#New#@#Y*
ENABLED_FLAG	Determines whether the rule is active. Disabled rules are completely ignored by the processing engine.	Text (1) Y/N	Y		Y

MAKE_EDITABLE_FLAG	Determines whether all the result fields should be made editable when the rule fires and returns a record. Note that fields which are defined as display only, based on PPM or based on the table XL_REQ_TYPE_FIELDS, can never be made editable using rules. Also, if the MAKE_REQUIRED_FLAG field below is enabled, there is no need to set this field (required fields are always made editable).	Text (1) Y/N			Y
MAKE_DISPLAY_ONLY_FLAG	Determines whether all the result fields should be made non-editable (if not already noneditable) when the rule fires and returns a record. Note that fields which are required, whether based on rules or based on the table XL_REQ_TYPE_FIELDS, can never be made non-editable	Text (1) Y/N			Y

MAKE_REQUIRED_FLAG	Determines whether all the result fields should be made required when the rule fires (if not already required) and returns a record. Note that fields which are defined as display only, whether based on PPM or based on the table XL_REQ_TYPE_FIELDS, can never be made required using rules	Text (1) Y/N			Y
MAKE_OPTIONAL_FLAG	Determines whether all the result fields should be made optional when the rule fires and	Text (1) Y/N			Y
	returns a record. Note that fields which are required or display only, based on the table XL_REQ_TYPE_FIELDS or PPM, cannot be made optional using rules				
PROCESS_SQL_RULE	Determines whether to populate the rule's result columns (as stored in the table XL_RULE_RESULTS) with the output of the RULE_SQL	Text (1) Y/N			Y
PROCESS_UI_RULE	Determines whether to apply the rule's UI instructions (e.g. make editable) if the RULE_SQL returns a record	Text (1) Y/N			Y
ON_LOAD_FLAG	Determines whether to process the rule when requests get downloaded from PPM (equivalent to PPM's "page load" rules)	Text (1) Y/N			Y

FIELD_CHANGE_FLAG	Determines whether to process the rule upon changes to one of its dependent fields (as defined in the table XL_ RULE_DEPENDENCIES)	Text (1) Y/N			Y

- XL_ RULE_DEPENDENCIES: This table contains a list of fields which changes to their values may trigger one or more rules.

Field Name	Description	Data Type and Valid Values	Required	Default	Example
DEPENDENCY_ID	Unique record identifier	Number	Y		30213
RULE_ID	Foreign key to the field RULE_ID in the table XL_REQ_TYPE_RULES	Number	Y		32343
PARAMETER_TOKEN	Full token of the field for which	Text (200)	Y		REQ.PRIORITY_ NAME
	the rule needs to evaluated upon field change				

- XL_ RULE_RESULTS: This table contains a list of fields which their values may be updated through one or more rules

Field Name	Description	Data Type and Valid Values	Required	Default	Example
RESULT_ID	Unique record identifier	Number			30213
RULE_ID	Foreign key to the XLI rule in the table XL_REQ_TYPE_RULES	Number			32343

PARAMETER_TOKEN	Full token of the field which may be updated if the rule is triggered				REQ.VP.DETAILED_DESC
SEQ_NUMBER	The sequence by which the field should be updated relative to other fields that may be updated by the same rule. All the result fields of the same rule must have a unique SEQ_NUMBER value	Number			1

5. Project Creation

Projects are unique request types in PPM and therefore treated differently in XLI as well. In order to enable project creation, run the following script to add three artificial fields to the table XL_REQ_TYPE_FIELDS: Project Type, Region, and Work Plan Template. The first two, alongside the project's start/finish dates, project name, and project manager are all required fields for project creation, as in the core UI. Additionally, before projects may be created, set the XL_SUPP_REQ_TYPES.EDIT_ONLY_FLAG of the request type to “N”.

```
 INSERT INTO XL_REQ_TYPE_FIELDS
(SUPP_REQ_TYPE_ID,PARAMETER_TOKEN,REQUIRED_FLAG,VALIDATION_SQL_HIDDEN,VALIDATION_SQL_VISIBLE,POPUP_FLAG,NU
MBER_FORMAT,COL_WIDTH,DOWNLOAD_FLAG,HIDDEN_FLAG,PROCESSING_ORDER,EXCLUSIVE_PROCESSING_FLAG,DISPLAY_ON
LY_FLAG,VALIDATION_FORMULA1,VALIDATION_FORMULA2,LOCATION_SEQ,PROMPT_OVERRIDE,VALIDATION_OPERATOR,VALIDATI
ON_ERR_MSG,CUST_TEMP_XL_RANGE) values ([replace with
XL_SUPP_REQ_TYPES.SUPP_REQ_TYPE_ID],'XL.PROJECT_TYPE','N',null,'SELECT RTRIM (XMLAGG (XMLELEMENT (e,
pt.project_type_name  || ''$@$'') ORDER BY  pt.project_type_name).EXTRACT (''//text()'').getClobVal(), '','')
FROM pm_project_types pt
          WHERE pt.enabled_flag = ''Y''
            AND pt.project_id IS NULL
          ORDER BY project_type_name
          ','N',null,25,'Y','N',0,'N','N',null,null,null,null,null,null,null);

 INSERT INTO XL_REQ_TYPE_FIELDS
(SUPP_REQ_TYPE_ID,PARAMETER_TOKEN,REQUIRED_FLAG,VALIDATION_SQL_HIDDEN,VALIDATION_SQL_VISIBLE,POPUP_FLAG,NU
MBER_FORMAT,COL_WIDTH,DOWNLOAD_FLAG,HIDDEN_FLAG,PROCESSING_ORDER,EXCLUSIVE_PROCESSING_FLAG,DISPLAY_ON
LY_FLAG,VALIDATION_FORMULA1,VALIDATION_FORMULA2,LOCATION_SEQ,PROMPT_OVERRIDE,VALIDATION_OPERATOR,VALIDATI
ON_ERR_MSG,CUST_TEMP_XL_RANGE) values ([replace with XL_SUPP_REQ_TYPES.SUPP_REQ_TYPE_ID],'XL.REGION','N',null,'SELECT
RTRIM (XMLAGG (XMLELEMENT (e, rgn.region_name  || ''$@$'') ORDER BY  rgn.region_name).EXTRACT (''//text()'').getClobVal(), '','')
FROM knta_regions rgn
          WHERE rgn.enabled_flag = ''Y''
          ','N',null,14,'Y','N',0,'N','N',null,null,null,null,null,null,null);

 INSERT INTO XL_REQ_TYPE_FIELDS
(SUPP_REQ_TYPE_ID,PARAMETER_TOKEN,REQUIRED_FLAG,VALIDATION_SQL_HIDDEN,VALIDATION_SQL_VISIBLE,POPUP_FLAG,NU
MBER_FORMAT,COL_WIDTH,DOWNLOAD_FLAG,HIDDEN_FLAG,PROCESSING_ORDER,EXCLUSIVE_PROCESSING_FLAG,DISPLAY_ON
LY_FLAG,VALIDATION_FORMULA1,VALIDATION_FORMULA2,LOCATION_SEQ,PROMPT_OVERRIDE,VALIDATION_OPERATOR,VALIDATI
ON_ERR_MSG,CUST_TEMP_XL_RANGE) values ([replace with
XL_SUPP_REQ_TYPES.SUPP_REQ_TYPE_ID],'XL.WP_TEMPLATE','N',null,'SELECT RTRIM (XMLAGG (XMLELEMENT (e,
workplan_template_name  || ''$@$'') ORDER BY  workplan_template_name).EXTRACT (''//text()'').getClobVal(), '','')
FROM pm_workplan_templates
          WHERE is_enabled_flag=''Y''
          ORDER BY workplan_template_name','N',null,44,'Y','N',0,'N','N',null,null,null,null,null,null,null);
```

Afterwards, run the following script to add request types rules to enforce the desired functionality:

```
DECLARE
          l_rule_id NUMBER;
        l_dep_id NUMBER;   l_rslt_id
        NUMBER;

BEGIN
          SELECT MAX(rule_id) + 1 INTO l_rule_id FROM xl_req_type_rules;
          SELECT MAX(dependency_id) + 1 INTO l_dep_id FROM xl_rule_dependencies;
        SELECT MAX(result_id) + 1 INTO l_rslt_id FROM xl_rule_results;

          INSERT INTO xl_req_type_rules
```

```
(PPM_REQ_TYPE_ID,RULE_ID,RULE_NAME,DESCRIPTION,SEQ_NUMBER,RULE_SQL,ENABLED_FLAG,MAKE_EDITABLE
_FLAG,MAKE_DISPLAY_ONLY_FLAG,MAKE_REQUIRED_FLAG,MAKE_OPTIONAL_FLAG,PROCESS_SQL_RULE_FLAG,PR
OCESS_UI_RULE_FLAG,ON_LOAD_FLAG,FIELD_CHANGE_FLAG) VALUES ([replace with
XL_SUPP_REQ_TYPES.PPM_REQ_TYPE_ID],l_rule_id,'Disable Project Creation Fields',NULL,(SELECT
NVL(MAX(seq_number),0) + 1 FROM xl_req_type_rules WHERE ppm_req_type_id = l_req_type),'SELECT ''#@#'' FROM dual
WHERE ''[REQ.VP.KNTA_PROJECT_NAME]'' IS NOT NULL','Y','N','Y','N','N','N','Y','Y','N');

  INSERT INTO xl_req_type_rules
(PPM_REQ_TYPE_ID,RULE_ID,RULE_NAME,DESCRIPTION,SEQ_NUMBER,RULE_SQL,ENABLED_FLAG,MAKE_EDITABLE
_FLAG,MAKE_DISPLAY_ONLY_FLAG,MAKE_REQUIRED_FLAG,MAKE_OPTIONAL_FLAG,PROCESS_SQL_RULE_FLAG,PR
OCESS_UI_RULE_FLAG,ON_LOAD_FLAG,FIELD_CHANGE_FLAG) VALUES ([replace with
XL_SUPP_REQ_TYPES.PPM_REQ_TYPE_ID],l_rule_id + 1,'Make Project Creation Fields Required',NULL,(SELECT
NVL(MAX(seq_number),0) + 1 FROM xl_req_type_rules WHERE ppm_req_type_id = l_req_type),'SELECT ''#@##@#'' FROM
dual WHERE ''[XL.PROJECT_TYPE]'' IS NOT NULL','Y','N','N','Y','N','N','Y','N','Y');

   INSERT INTO xl_req_type_rules
(PPM_REQ_TYPE_ID,RULE_ID,RULE_NAME,DESCRIPTION,SEQ_NUMBER,RULE_SQL,ENABLED_FLAG,MAKE_EDITABLE
_FLAG,MAKE_DISPLAY_ONLY_FLAG,MAKE_REQUIRED_FLAG,MAKE_OPTIONAL_FLAG,PROCESS_SQL_RULE_FLAG,PR
OCESS_UI_RULE_FLAG,ON_LOAD_FLAG,FIELD_CHANGE_FLAG) values ([replace with
XL_SUPP_REQ_TYPES.PPM_REQ_TYPE_ID],l_rule_id + 2,'Clearing Finish Date if Invalid',null,(SELECT
NVL(MAX(seq_number),0) + 1 FROM xl_req_type_rules WHERE ppm_req_type_id = l_req_type),'SELECT ''#@#'' FROM dual
WHERE ''[REQ.VP.KNTA_PLAN_START_DATE]'' IS NOT NULL AND  ''[REQ.VP.KNTA_PLAN_FINISH_DATE]'' IS NOT NULL
            AND TO_DATE(''[REQ.VP.KNTA_PLAN_START_DATE]'',''MONTH YYYY'') >
TO_DATE(''[REQ.VP.KNTA_PLAN_FINISH_DATE]'',''MONTH YYYY'')','Y',null,null,null,null,'Y',null,null,'Y');
   INSERT INTO XL_RULE_DEPENDENCIES (DEPENDENCY_ID,RULE_ID,PARAMETER_TOKEN) VALUES
(l_dep_id,l_rule_id,'REQ.VP.KNTA_PROJECT_NAME');

  INSERT INTO XL_RULE_DEPENDENCIES (DEPENDENCY_ID,RULE_ID,PARAMETER_TOKEN) VALUES (l_dep_id +
1,l_rule_id + 1,'XL.PROJECT_TYPE');

   INSERT INTO XL_RULE_DEPENDENCIES (DEPENDENCY_ID,RULE_ID,PARAMETER_TOKEN) VALUES (l_dep_id +
2,l_rule_id + 2,'REQ.VP.KNTA_PLAN_FINISH_DATE');
```

```
        INSERT INTO XL_RULE_DEPENDENCIES (DEPENDENCY_ID,RULE_ID,PARAMETER_TOKEN) values (l_dep_id +
    3,l_rule_id + 2,'REQ.VP.KNTA_PLAN_START_DATE');

        INSERT INTO XL_RULE_RESULTS (RESULT_ID,RULE_ID,PARAMETER_TOKEN,SEQ_NUMBER) VALUES
    (l_rslt_id,l_rule_id,'XL.PROJECT_TYPE',2);

        INSERT INTO XL_RULE_RESULTS (RESULT_ID,RULE_ID,PARAMETER_TOKEN,SEQ_NUMBER) VALUES (l_rslt_id +
    1,l_rule_id,'XL.REGION',1);

        INSERT INTO XL_RULE_RESULTS (RESULT_ID,RULE_ID,PARAMETER_TOKEN,SEQ_NUMBER) VALUES (l_rslt_id +
    2,l_rule_id,'XL.WP_TEMPLATE',3);

        INSERT INTO XL_RULE_RESULTS (RESULT_ID,RULE_ID,PARAMETER_TOKEN,SEQ_NUMBER) VALUES (l_rslt_id +
    3,l_rule_id + 1,'XL.REGION',2);

        INSERT INTO XL_RULE_RESULTS (RESULT_ID,RULE_ID,PARAMETER_TOKEN,SEQ_NUMBER) VALUES (l_rslt_id +
    4,l_rule_id + 2,'REQ.VP.KNTA_PLAN_FINISH_DATE',1);
END;
```

6. PowerPoint Generation

The XLI supports generating a PowerPoint presentation, based on any customer-defined template, while embedding PPM data.
Every XLI instance may only have a single PPM request type, enabled for use through XLI, for which PowerPoint decks may be generated by users, always at the individual request level. At a high-level, there are two steps for the process of enabling this feature:

1. Preparing the customer defined PowerPoint template by embedding substitution variables.
2. Populating various XLI server settings, such as the PowerPoint file information, and the PPM data extraction query.

PowerPoint template preparation steps, which are to be applied to the desired customer-defined template:

1. Every text field which has to be replaced with PPM data upon deck generation should be populated with #field sequence in data export query#. For example, if the intent is to export the request description and it is the first field in the query, replace all locations where the description should be visible on the deck with #1#.
2. PowerPoint tables with a *static* number of cells, which are to be populated with PPM data, must be named REPLACE (in PowerPoint, select the table, the click on Select menu item -> *Selection Pane*). Then, follow the naming convention in the previous step for each cell.
3. PowerPoint tables with a *dynamic* number of rows, which are to be populated with PPM data, keep their caption row(s) and one empty data row below. Additionally, name each one of these tables in a unique and meaningful way (e.g. ASSUMPTIONS).
4. Place the file on the PPM application server(s) under server_common\deploy\itg.war\download or a sub-path.

XLI server settings steps:

1. Populate the SETTING_VALUE column of the setting PP_FILE_NAME in the table XL_SETTINGS with the name of the PowerPoint file on the PPM application server(s) used as a template for deck generation by users. The XLI will start searching from the /download directory (e.g. Project_Status_Report.pptx).
2. Populate the SETTING_VALUE column of the setting PP_REQ_TYPE_NAME in the table XL_SETTINGS with the name of the request type which should support this feature. This value must match one of the enabled request types in SUPP_REQ_TYPES.DISPLAYED_REQ_TYPE_NAME.
3. Populate the SETTING_VALUE column of the setting PP_QUERY1 with the SQL query used as the data source of the PowerPoint deck generated out of XLI. The query must reference the supplied request ID with #?# and should follow the format:

   ```
   SELECT  '#@#' || field_a || '#@#' || field_b
   FROM kcrt_requests
   WHERE request_id = #?#
   ```
   Note that additional PP_QUERY[X] settings (e.g. PP_QUERY2) may be added to the table, and the XLI will process them in sequential order during PowerPoint generation.
4. Populate the SETTING_VALUE column of the setting PP_TBL1 with the SQL query used as the data source of one of the dynamic tables

   $@$PowerPoint table name$@$First row of data (not captions) in the table$@$The query of the format SELECT '#@#' || field_a || '#@#' || field_b
   in the PowerPoint deck generated out of XLI. The setting_value should adhere to the following format:

   For example:

$@$ASSUMPTIONS$@$2$@$SELECT '#@#' || req.created_by || '#@#' || req.description FROM kcrt_requests WHERE request_id = #?#

Note that additional PP_TBL[X] may be added to the setting table, for each dynamic table in the PowerPoint template.

In order to test this feature, perform the following steps:

1. Login to the XLI workbook.
2. Download any request of the type enabled for PowerPoint generation.
3. Highlight a single cell on the request row.
4. The PowerPoint generation button should then become enabled . Click on it.