Project and Portfolio Management Center

Software Version: 9.40

Open Interface Guide and Reference

Document Release Date: September 2016
Software Release Date: September 2016

Hewlett Packard Enterprise

Legal Notices

Warranty

The only warranties for Hewlett Packard Enterprise products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Hewlett Packard Enterprise shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from Hewlett Packard Enterprise required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe® is a trademark of Adobe Systems Incorporated.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

UNIX® is a registered trademark of The Open Group.

Documentation Updates

To check for recent updates or to verify that you are using the most recent edition of a document, go to: https://softwaresupport.hpe.com/.

This site requires that you register for an HP Passport and to sign in. To register for an HP Passport ID, click **Register** on the HPE Software Support site or click **Create an Account** on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HPE sales representative for details.

The following table indicates changes made to this document since the last released edition.

Support

Visit the HPE Software Support site at: https://softwaresupport.hpe.com/.

Most of the support areas require that you register as an HP Passport user and to sign in. Many also require a support contract. To register for an HP Passport ID, click **Register** on the HPE Support site or click **Create an Account** on the HP Passport login page.

To find more information about access levels, go to: https://softwaresupport.hpe.com/web/softwaresupport/access-levels.

HPE Software Solutions Now accesses the HPSW Solution and Integration Portal website. This site enables you to explore HPE Product Solutions to meet your business needs, includes a full list of Integrations between HPE Products, as well as a listing of ITIL Processes. The URL for this website is https://softwaresupport.hpe.com/km/KM01702731.

Contents

Chapter 1: Getting Started with the PPM Center Open Interface

Introduction to the Open Interface

The Open Interface allows integration of data from third-party products with key Project and Portfolio Management Center (PPM Center) entities. Relevant information from these products can be used for:

- Generating or updating users from a database or LDAP server
- Importing an organization model into PPM Center
- Generating requests and packages
- Performing workflow transactions for package lines and requests

The application program interface (API) described in this document uses interface tables within the PPM Center database. Data added to these interface tables is validated and eventually imported into standard PPM Center tables. This generates entities that can be processed using PPM Center.

Related Documentation

The following is a list of documentation that provides additional information about using PPM Center:

- *What's New and What's Changed*
- Getting Started
- Web Services Guide
- *Program Management User's Guide*
- *Portfolio Management User's Guide*
- Project Management User's Guide
- Resource Management User's Guide
- *Time Management User's Guide*

- *Demand Management User's Guide*
- *Demand Management Configuration Guide*
- *Financial Management User's Guide*
- *Reports Guide and Reference*
- *Commands, Tokens, and Validations Guide and Reference*
- *Installation and Administration Guide*
- *Security Model Guide and Reference*

Chapter 2: User Open Interface

Overview of the User Open Interface

PPM Center includes an open interface for importing information. This open interface can import user models from third-party systems, including LDAP databases, internally-developed systems, or human resources systems.

> **Note:** When a user submits a request through the open interface, PPM Center saves values for fields in CODE parameter (PARAMETER COLUMN), not in MEANING (VISIBLE_PARAMETER column). This pertains to fields of the following types: Currency, Date (Long, Medium, and Short formats), Numeric, Percentage, and Radio Button.

You should periodically synchronize the user model in PPM Center with the authoritative data source within your company. The synchronization process involves importing user attributes of the various users into the following tables:

- "KNTA_USERS_INT" on page 150
- "KNTA_USER_SECURITY_INT" on page 157
- "RSC_RESOURCES_INT" on page 160

These interface tables are described in "Open Interface Data Models" on page 104. The columns that can be used when importing users are detailed in the applicable step within this section.

The User Open Interface supports:

- Simple imports
- LDAP imports

For information on mapping your user model, see the *Resource Management User's Guide*. For information on user report types and running reports, see the *Reports Guide and Reference*.

To review the LDAP authentication process, see "LDAP Authentication" on page 181.

The open interface functionality expects users to input the **Currency**, **Date**, and **Number** fields in the standard formats (as described below) while populating the interface tables with the transactional data, regardless of the language or user locale.

Standard formats include:

- **Date**: dd-mon-yyyy (for example, 15-Nov-2009)
- **Numbers**: Decimal point separator is "." and thousands separator is ","
- **Currency**: Enter as Standard Text Currency/Number format

Performing a Simple Import

Step 1: Determine the Security Groups

Decide which users should have which security groups linked to them.

When importing users, it is possible to specify how the user is assigned to specific security groups. This is accomplished using a combination of the following fields from the Import Users report:

- **Security Groups**
- **User Security Group Action**
- **Add Missing Security Groups**

When you reach "Step 5: Start the Import" on page 27, you must have done the research so that you can specify values for the following:

- Add selected security groups to the group of users.
- Drop selected security groups from the user definitions.
- Add some security groups to the user definitions while dropping others. When using this ADD/DROP option, the KNTA_USER_SECURITY_INT table must also be populated as described in "Step 2: Add and Drop Security Groups" on the next page.
- Overwrite the security group specification to include only the specified security groups. This deletes all references to the user's security groups and replaces them with the selected ones.
- Add missing security groups. This creates a new security group, but does not link the user to that security group.

Step 2: Add and Drop Security Groups

If you decide that you need to use the ADD/DROP option, populate the KNTA_USER_SECURITY_INT interface table and then specify ADD or DROP for the USER_SECURITY_ACTION column in the interface table.

1. Using SQL*Loader, your favorite tool, or direct Oracle database-to-database communication, load your data into the required input columns in the following table.

   - "KNTA_USER_SECURITY_INT" on page 157

   See "Table 2-1. KNTA_USER_SECURITY_INT interface table" below.

2. Include a record for each desired security group action for each user.
3. Specify ADD or DROP for the USER_SECURITY_ACTION column.

Example of a Change in Security Groups

User A and User B exist as users of PPM Center and are linked to the following security groups:

- User A => security group X
- User B => security group Y

Using a single User Open Interface transaction, you want to change the users' security groups to the following:

- User A => security group Y
- User B => security group X

To do this, populate the KNTA_USER_SECURITY_INT table with the following records:

```
GROUP_ID USER_ID SECURITY_GROUP_NAME USER_SECURITY_ACTION
100 USER A GROUP X DROP
100 USER A GROUP Y ADD
100 USER B GROUP X ADD
100 USER B GROUP Y DROP
```

Table 2-1. KNTA_USER_SECURITY_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	Required	NUMBER	Uniquely identifies each transaction.

Table 2-1. KNTA_USER_SECURITY_INT interface table, continued

Column	Usage	Data Type	Description
PARENT_ TRANSACTION_ ID	Required	NUMBER	Provides the transaction ID (from KNTA_USERS_ INT) of the parent table being imported. If any child table is being used, set the TRANSACTION_ID in KNTA_USERS_INT to this value.
PARENT_ TABLE_NAME	Required	VARCHAR2	Identifies the table associated with this entity. The parent_table should be derived from KNTA_ USERS_INT.
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KNTA_USERS_INT.
EXISTS_FLAG	Optional	VARCHAR2	Indicates whether the user already exists.
PROCESS_ PHASE	Optional	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_ STATUS	Optional	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_ BY_USERNAME.
CREATED_BY_ USERNAME	Optional	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_ DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.

Table 2-1. KNTA_USER_SECURITY_INT interface table, continued

Column	Usage	Data Type	Description
DEST_ CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_ BY_USERNAME If both are left blank, the value is set to the user currently running the report.
DEST_ CREATION_ DATE	Optional	DATE	Indicates the date the record is created in the destination (PPM Center instance). If left blank, the value is derived from CREATION_ DATE.
DEST_LAST_ UPDATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the set to the user currently running the report.
DEST_LAST_ UPDATE_DATE	Optional	DATE	Indicates the date that the security data was last updated. If left blank, the current date is used.
DEST_ENTITY_ UPD_DATE	Optional	DATE	Indicates the date that either the user data or security data was last updated. If left blank, the current date is used.
USER_ SECURITY_ID	Optional	NUMBER	Identifies a user security when removing a user from a security group. This is normally left blank. This is normally left blank and is derived from the KNTA_USER_SECURITY_S sequence.
DEST_USER_ SECURITY_ID	Optional	NUMBER	Identifies a user security. This is normally left blank. This is normally left blank and is derived from the KNTA_USER_SECURITY_S sequence.
USER_ID	Optional	NUMBER	Identifies the user. When creating users, this is left blank and the value is derived from the KNTA_USERS_S sequence. For existing users, this refers to the USER_ID column in KNTA_USERS.

Table 2-1. KNTA_USER_SECURITY_INT interface table, continued

Column	Usage	Data Type	Description
DEST_USER_ID	Optional	NUMBER	Identifies the user. For existing users, this refers to the USER_ID column in KNTA_USERS. This is normally left blank and is derived from the KNTA_USERS_S sequence.
SECURITY_ GROUP_ID	Required	NUMBER	Indicates the security group for the user. Required for ADD; not required for DROP.
SOURCE_ TYPE_CODE	Required	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_WF.
SOURCE	Required	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
LOGON_ IDENTIFIER	Required	VARCHAR2	Identifies the ID used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ ID, the LOGON_IDENTIFIER column must be populated. Otherwise, populate the USERNAME column.
USERNAME	Required	VARCHAR2	Identifies the name used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_ NAME, the USERNAME column must be populated. Otherwise, populate the LOGON_IDENTIFIER column.
SECURITY_ GROUP_NAME	Required	VARCHAR2	Specifies the SECURITY_GROUP_NAME in KNTA_SECURITY_GROUPS.
USER_ SECURITY_ ACTION	Required	VARCHAR2	Indicates the action for user security. Valid values are ADD or DROP.

Step 3: Populate the User Interface Table

1. Using SQL*Loader, your favorite tool, or direct Oracle database-to-database communication, load your data into the required input columns in the following table.

   - "KNTA_USERS_INT" on page 150

   See "Table 2-2. KNTA_USERS_INT interface table" below.

2. Additional columns in KNTA_USER_SECURITY_INT must be populated when using the ADD/DROP security group action. For more information, see "Step 2: Add and Drop Security Groups" on page 11.

   > **Caution:** User data is not validated during import.

Table 2-2. KNTA_USERS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	Required	NUMBER	Uniquely identifies each transaction. See also PARENT_TRANSACTION_ID in KNTA_USER_SECURITY_INT.
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language. If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
EXISTS_FLAG	Optional	VARCHAR2	Indicates whether the user already exists.
PROCESS_PHASE	Optional	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182

Table 2-2. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
			for details.
PROCESS_STATUS	Optional	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME.
CREATED_BY_USERNAME	Optional	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_DATE	Optional	DATE	Indicates the date that the record was created. If left blank, the current date is used.
DEST_CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME. If both are left blank, the value is set to the user currently running the report.
DEST_CREATION_DATE	Optional	DATE	Indicates the date the record is created in the destination (PPM Center instance). If left blank, the value is derived from CREATION_DATE.
DEST_LAST_UPDATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data.

Table 2-2. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
			If left blank, the value is set to the user currently running the report.
DEST_LAST_ UPDATE_DATE	Optional	DATE	Indicates the date that the user data was last updated. If left blank, the current date is used.
DEST_ENTITY_ UPD_DATE	Optional	DATE	Indicates the date that either the user data or security data was last updated. If left blank, the current date is used.
USER_ID	Optional	NUMBER	Identifies the user. When creating users, this is left blank and the value is derived from the KNTA_USERS_S sequence. For existing users, this can be left blank or a valid USER_ID (from KNTA_USERS) be provided.
DEST_USER_ID	Optional	NUMBER	Identifies the user. This is normally left blank and is derived from the KNTA_USERS_S sequence.
USERNAME	Required	VARCHAR2	Identifies the name used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_ METHOD = USER_NAME, the USERNAME column must be populated for the user import. Otherwise, populate the LOGON_IDENTIFIER column.
DEST_USERNAME	Optional	NUMBER	Identifies the username. If left blank, the value is derived from USERNAME.
PASSWORD	Optional	VARCHAR2	Specifies the password for the user. If left blank, the value is set to the password of the user currently running the report.
PASSWORD_	Required	NUMBER	Specifies the number of days before the

Table 2-2. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
EXPIRATION_DAYS			current password expires.
PASSWORD_ EXPIRATION_DATE	Required	DATE	Specifies the date when the password should expire.
EMAIL_ADDRESS	Required	VARCHAR2	Specifies the email address of the user.
FIRST_NAME	Required	VARCHAR2	Specifies the user's first name. This is required only if creating a new user. It is not required when re-importing an existing user.
LAST_NAME	Required	VARCHAR2	Specifies the user's last name. This is required only if creating a new user. It is not required when re-importing an existing user.
START_DATE	Required	DATE	Specifies the user's start date.
END_DATE	Required	DATE	Specifies the user's end date.
DEFAULT_ ACCELERATOR_ID	Required	NUMBER	Sets the context identifier for the USER_ DATA fields.
SOURCE_TYPE_ CODE	Required	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_WF.
SOURCE	Required	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
USER_DATA_SET_ CONTEXT_ID	Required	NUMBER	Sets the context identifier for the USER_ DATA fields. Supply this or USERNAME.
USER_DATA1 VISIBLE_USER_ DATA1 through USER_DATA20	Required	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined. This information is not validated nor does it have a default value.

Table 2-2. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
VISIBLE_USER_DATA20			
AUTHENTICATION_MODE	Required	VARCHAR2	Specifies the user's authentication mode. If the user is being imported from a LDAP server, then this is automatically set to LDAP. Otherwise it is set to KINTANA. For custom implementations, other values can be used.
SCREEN_ID	Optional	NUMBER	Specifies the first screen shown after logon. If left blank, the default value is supplied.
SHORTCUT_BAR_FLAG	Optional	VARCHAR2	Option to show the shortcut bar in the screen manager. If left blank, the default value is supplied.
SHORTCUT_BAR_LOC_CODE	Optional	VARCHAR2	Specifies the position where the shortcut bar is displayed. If left blank, the default value is supplied.
SAVE_WINDOW_BOUNDS_FLAG	Optional	VARCHAR2	Option to save the size and location of the screen manager window after logoff. If they are saved, the settings are the default at the next logon. If left blank, the default value is supplied.
WINDOW_HEIGHT	Optional	NUMBER	Specifies the default height of the screen manager window. If left blank, the default value is supplied.
WINDOW_WIDTH	Optional	NUMBER	Specifies the default width of the screen manager window. If left blank, the default value is supplied.
WINDOW_X_LOCATION	Optional	NUMBER	Specifies the horizontal position of the screen manager window. If left blank, the default value is supplied.
WINDOW_Y_LOCATION	Optional	NUMBER	Specifies the vertical position of the screen manager window.

Table 2-2. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
			If left blank, the default value is supplied.
REUSE_ INTERNAL_ FRAME_FLAG	Optional	VARCHAR2	Option to open multiple internal frames within each screen. If left blank, the default value is supplied.
SHOW_ALL_ WORKFLOW_ STEPS_FLAG	Optional	VARCHAR2	Option to show all workflow steps within workflow status panels. If left blank, the default value is supplied.
SHOW_ TRAVERSED_ STEPS_FLAG	Optional	VARCHAR2	Option to show steps that have been traversed and are no longer active in the workflow status panels. If left blank, the default value is supplied.
NUM_BRANCH_ STEPS_TO_SHOW	Optional	NUMBER	If a currently active workflow step leads to several branches, specifies how many steps of each branch are shown within workflow status panels. If left blank, the default value is supplied.
NUM_KNOWN_ REACH_STEPS_ TO_SHOW	Optional	NUMBER	Specifies the number of steps of a non-branching path that are shown within workflow status panels. If left blank, the default value is supplied.
HIDE_IMMEDIATE_ STEPS_FLAG	Optional	VARCHAR2	Option to show workflow steps, based upon immediate executions and conditions, in workflow status panels. If left blank, the default value is supplied.
SHOW_CHANGE_ WARNINGS_FLAG	Optional	VARCHAR2	Option to display warning messages when a business entity that is used by another entity is updated. For example, when a workflow is updated that is used by a package line. If left blank, the default value is supplied.
HIDE_ CANCELLED_CRL_ FLAG	Optional	VARCHAR2	Option to display cancelled package lines in the packages screen. If left blank, the default value is supplied.

Table 2-2. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
DEFAULT_ BROWSER	Required	VARCHAR2	Specifies the default browser for the user.
DEST_USER_ PROFILE_ID	Optional	NUMBER	Specifies the user profile ID for the user.
COMPANY	Required	VARCHAR2	Identifies the company. This should be a valid LOOKUP_CODE from KNTA_LOOKUPS where LOOKUP_TYPE = `COMPANY'.
DOMAIN	Required	VARCHAR2	Identifies the Windows domain. Used for Exchange server (NTLM) authentication.
LOGON_ IDENTIFIER	Required	VARCHAR2	Identifies the ID used for the logon. The value should be a valid USERNAME in KNTA_ USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_ METHOD = LOGON_ID, the LOGON_ IDENTIFIER column must be populated. Otherwise, populate the USERNAME column.
PHONE_NUMBER	Required	VARCHAR2	Specifies the user's phone number on the resource page.
COST_RATE	Required	NUMBER	Specifies the user's cost rate.
WORKLOAD_ CAPACITY	Required	NUMBER	Specifies the user's workload capacity (in percentage) on the resource page.
MAX_ROWS_ PORTLETS	Required	NUMBER	Specifies the maximum number of results to be displayed on the maximized portlet.
DEPARTMENT_ CODE	Optional	VARCHAR2	Specifies the code for the department.
DEPARTMENT_ MEANING	Required	VARCHAR2	Specifies the description of the department.
LOCATION_CODE	Optional	VARCHAR2	Specifies the code for the location.
LOCATION_ MEANING	Required	VARCHAR2	Specifies the description of the location.

Table 2-2. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
MANAGER_USER_ID	Required	NUMBER	Specifies the user ID of the manager. Used if both MANAGER_USERNAME and MANAGER_LOGON_IDENTIFIER are left blank.
MANAGER_USERNAME	Required	VARCHAR2	Specifies the name of the manager. Used if MANAGER_LOGON_IDENTIFIER is left blank.
MANAGER_LOGON_IDENTIFIER	Required	VARCHAR2	Specifies the ID of the manager. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, the LOGON_IDENTIFIER column must be populated. Otherwise, populate the MANAGER_USERNAME column.
RESOURCE_CATEGORY_CODE	Optional	VARCHAR2	Specifies the code for the user's category.
RESOURCE_CATEGORY_MEANING	Required	VARCHAR2	Specifies the description of the user's category.
RESOURCE_TITLE_CODE	Optional	VARCHAR2	Specifies the code for the user's title.
RESOURCE_TITLE_MEANING	Required	VARCHAR2	Specifies the description of the user's title.
PRODUCT_ID_LIST	Optional	VARCHAR2	Indicates the user's license.

Step 4: Populate the Resource Interface Table

Using SQL*Loader, your favorite tool, or direct Oracle database-to-database communication, load your data into the required input columns in the following table.

- "RSC_RESOURCES_INT" on page 160

See "Table 2-3. RSC_RESOURCES_INT interface table" on the next page.

Caution: The source data is not validated during import.

Table 2-3. RSC_RESOURCES_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ ID	Optional	Number	Uniquely identifies each transaction. Initialized when interface loading start
PARENT_ TRANSACTION_ ID	Optional	Number	Provides the transaction ID (from KNTA_USERS_INT) of the parent table being imported. If any child table is being used, set the TRANSACTION_ ID in KNTA_USERS_INT to this value. KNTA_USERS_INT.TRANSACTION_ID
PARENT_ TABLE_NAME	Optional	Varchar2 (30)	Identifies the table associated with this entity.
GROUP_ID	Required	Number	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_ GROUPS_S sequence. This value should be the same as the parent's GROUP_ ID in KNTA_USERS_INT.
EXISTS_FLAG	Optional	Varchar2 (1)	Indicates whether or not the user already exists.
PROCESS_ PHASE	Optional	Number	Indicates the current stage of the record as it is being processed.
PROCESS_ STATUS	Optional	Number	Indicates the current disposition of the record.
CREATED_BY	Optional	Number	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME.
CREATED_BY_ USERNAME	Optional	Varchar2 (200)	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to

Table 2-3. RSC_RESOURCES_INT interface table, continued

Column	Usage	Data Type	Description
			the user currently running the report.
CREATION_ DATE	Optional	Date	Indicates the transaction date. If left blank, the current date is used.
DEST_ CREATED_BY	Optional	Number	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME. If both are left blank, the value is set to the user currently running the report.
DEST_ CREATION_ DATE	Optional	Date	Indicates the date the record is created in the destination (PPM Center instance). If left blank, the value is derived from CREATION_ DATE.
DEST_LAST_ UPDATED_BY	Optional	Number	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the user currently running the report.
DEST_LAST_ UPDATE_DATE	Optional	Date	Indicates the date that the user data was last updated. If left blank, the current date is used.
DEST_ENTITY_ UPD_DATE	Optional	Date	Indicates the date that the user data was last updated. If left blank, the current date is used.
SOURCE_ TYPE_CODE	Optional	Varchar2 (30)	Specifies the type of external update. It is initialized when interface loading starts.
SOURCE	Optional	Varchar2 (100)	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
LOGON_ IDENTIFIER	Required	Varchar2 (200)	Identifies the ID used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the

Table 2-3. RSC_RESOURCES_INT interface table, continued

Column	Usage	Data Type	Description
			server.conf file. If LOGON_METHOD = LOGON_ID, the LOGON_IDENTIFIER column must be populated. Otherwise, populate the USERNAME column.
USERNAME	Required	Varchar2 (200)	Identifies the name used for the logon. The value should be a valid USERNAME in KNTA_ USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_ NAME, the USERNAME column must be populated. Otherwise, populate the LOGON_IDENTIFIER column.
RESOURCE_ID	Optional	Number	Id of the resource derived from RSC_RESOURCES. This Id refers to the RESOURCE_ID column.
USER_ID	Optional	Number	Identifies the user. Derived from valid username or LOGON_IDENTIFIER.
PRIMARY_ ROLE_ID	Optional	Number	Derived by valid PRIMARY_ROLE_NAME.
PRIMARY_ ROLE_NAME	Required	Varchar2 (200)	Existing role name in PPM Center.
TIME_SHEET_ POLICY_ID	Optional	Number	Derived from valid TIME_SHEET_POLICY_NAME.
TIME_SHEET_ POLICY_NAME	Required	Varchar2 (200)	Existing time sheet policy name in PPM Center.
TM_ APPROVER_ID	Optional	Number	Derived from valid TM_APPROVER_USERNAME or TM_APPROVER_IDENT
TM_ APPROVER_ USERNAME	Optional	Varchar2 (200)	Username of time sheet approver. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_NAME, the TM_ APPROVER_USERNAME column must be populated for the user import. Otherwise, populate the TM_ APPROVER_IDENT column.
TM_ APPROVER_ IDENT	Optional	Varchar2 (200)	LOGON_IDENTIFIER of time sheet approver. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, this column should be populated. Otherwise, populate the TM_

Table 2-3. RSC_RESOURCES_INT interface table, continued

Column	Usage	Data Type	Description
			APPROVER_USERNAME column.
TM_BILLING_ APPROVER_ID	Optional	Number	Derived from valid TM_BILLING_APPROVER_ USERNAME or TM_BILLING_APPROVER_IDENT.
TM_BILLING_ APPROVER_ USERNAME	Optional	Varchar2 (200)	Username of time sheet billing approver. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_ NAME, the TM_BILLING_APPROVER_USERNAME column must be populated for the user import. Otherwise, populate the TM_BILLING_APPROVER_ IDENT column.
TM_BILLING_ APPROVER_ IDENT	Optional	Varchar2 (200)	LOGON_IDENTIFIER of time sheet billing approver. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, this column should be populated. Otherwise, populate the TM_BILLING_APPROVER_USERNAME column.
TM_ENABLED_ FLAG	Optional	char	Indicates if the time management is enabled for this resource.
TM_NOTIFS_ ENABLED_FLAG	Optional	char	Indicates if this resource should recieve time management notifications.
TIME_SHEET_ APPROVER_ SEC_GRP_ID	Optional	Number	Derived by correct TM_APPROVER_SEC_GRP_ NAME
TM_ APPROVER_ SEC_GRP_ NAME	Optional	Varchar2 (200)	Existing security group name in PPM Center.
BILLING_ APPROVER_ SEC_GRP_ID	Optional	Number	Derived from valid BILLING_APPROVER_SEC_GRP_ NAME.
BILLING_ APPROVER_ SEC_GRP_ NAME	Optional	Varchar2 (200)	Existing security group name in PPM Center.
USER_DATA(N)	Optional	Varchar2 (200)	User data segment. N is 1 to 100.

Table 2-3. RSC_RESOURCES_INT interface table, continued

Column	Usage	Data Type	Description
VISIBLE_ USERDATA(N)	Optional	Varchar2 (200)	User data segment. N is 1 to 100

Step 5: Start the Import

To import data from the interface tables, the Import Users report is used.

The Import Users report:

- Queries the KNTA_USERS_INT interface table for active records matching the given selection criteria.
- Queries the KNTA_USER_SECURITY_INT table.
- Validates the user information.
- Imports validated users into PPM Center tables. Partial imports are not allowed. Users with one or more failed fields are not imported.
- Reports on the results of the execution, listing the specified users that failed validation and the specific validation errors they encountered.

To run the Import Users report:

1. Log on to PPM Center.
2. From the menu bar, select **Open > Reports > Create Report**.

   The Submit New Report page appears.
3. Select **Administrative** from the Report Category.

   The page is updated showing the list of administrative reports.
4. Select **Import Users**.

   The Submit Report: Import Users window appears.
5. Complete the fields, as described in the following table.

   Required fields are denoted with a red asterisk next to the field on the screen. Depending on your selections, the required fields may vary.

The Import Users report has several parameters for controlling the behavior of the program execution. Pay special attention to:

- **Product Licenses**
- **LDAP Import** - Set this field to **No**
- **Search Filter**
- **User Authentication Mode**
- **Link User Security Groups from LDAP Groups**
- **Import Modified**

> **Note:** Although security groups can be different, all users imported in a single execution of the Import Users report must have the same user privileges. To set different attributes (security groups or product permissions) for imported users, it is necessary to run the report multiple times.

> **Caution:** USER_DATA for users is not validated as part of an import.

Field Name	Description
Group Id	Specifies the group ID for which the interface program should be run. The interface program will only look for records with this value in the GROUP_ID column. This is useful when importing a batch of packages.
Source Code	Indicates whether or not to set the SOURCE_CODE column of the final requests created with a free-form text code. This is used as an indicator of how the request was created for auditing or testing purposes. For an LDAP import, set to LDAP_IMPORT.
Run Import?	◦ **If set to Yes.** Indicates that the program will process the records in the interface table and try to import them. ◦ **If set to No.** Indicates that the program will report on the records in the interface table. This option is useful when auditing prior executions of the interface.
Show Successful Transactions?	Option to show users that were successfully imported.
Show Failed Transactions?	Option to show users that were not successfully imported.
Default	Specifies a default password.

Field Name	Description
Password	
Security Groups	Specifies security groups that have the right to access this group of users.
User Security Group Action	Selects action to perform (Add/Drop, Add, Drop, or Overwrite).
Add Missing Security Groups?	Option to add missing security groups.
Disable Users Not Imported	Option to disable users left in PPM Center after the import. For example, user A and B exist in PPM Center, and you want to import C and D using this report. When you specify Yes for this option, then A and B are disabled after you run the report.
Keep existing values for empty columns?	Option to keep existing values stored for empty columns.
Product Licenses	Selects the product license the imported users will have.
Region for Resource (Override default region)	Selects the region for the imported users. If no region is specified, the system default region is used.
LDAP Import	Option to perform LDAP import. Set this to **Yes** if the authentication mode in the `server.conf` file contains LDAP or an Exchange server (NTLM).
LDAP Import PPM User Only	Option to perform LDAP import of only PPM Center users.
Search Filter	Specifies the search filter using syntax of the conditions on PPM Center commands. See "Examples of Search Filter Values" on the next page.
Search DN	Specifies the search filter using syntax of the conditions on PPM Center commands.
User Authentication	Selects a user authentication mode. (LDAP or NTLM only)

Field Name	Description
Mode	
Link User Security Groups from LDAP Groups	Option to link security groups from LDAP Groups. (LDAP or NTLM only)
Import Modified	Option to modify the import. (LDAP or NTLM only)

6. (Optional) To test the process, set the **Run Import** field to **No.**
7. Click **Submit.**

For more information on the `server.conf` file, see the *Installation and Administration Guide*.

Examples of Search Filter Values

The following filter returns objects that match "Babs Jensen".

(cn=Babs Jensen)

The following filter returns objects that do not match "Tim Howes".

(!(cn=Tim Howes))

The following filter returns all people who's names match Jensen or whose name contains the string "Babs J".

(&(objectClass=Person)(|(sn=Jensen)(cn=Babs J*)))

The following filter returns objects that would help you find variations on the spelling of University of Michigan.

(o=univ*of*mich*)

This filter returns all Development objects that match: test1, test2, or test3.

(&(ou=Development)(|(uid=test1)(uid=test2)(uid=test3)))

Step 6: Verify Successful Completion

If any customizations to the import process have been made, it is extremely important to confirm that the import was successful.

To confirm that the import process completed successfully:

1. Click **View Report** to review the results of the import.

   This report identifies any errors with the import.

2. If errors are present, start your troubleshooting by referring to "Correcting Failures" on page 36.

All interface tables are automatically cleared by the purge service. The purging process depends on the following parameters in the server.conf file:

- **ENABLE_INTERFACE_CLEANUP.** Enables or disables the purge process.
- **DAYS_TO_KEEP_INTERFACE_ROWS.** Determines the number of days that records are retained in the interface tables.

For more information on the `server.conf` file, see the *Installation and Administration Guide*.

Performing an LDAP Import

Step 1: Determine the Security Groups

Decide which users should have which security groups linked to them.

When importing users, it is possible to specify how the user is assigned to specific security groups. This is accomplished using a combination of the following fields from the Import Users report:

- **Security Groups**
- **Security Groups Action**
- **Link Security Groups from LDAP Groups**

When you reach "Step 6: Start the Import" on page 34, you need to be able to specify the following:

- Add selected security groups to the group of users.
- Drop selected security groups from the user definitions.
- Add some security groups to the user definitions while dropping others. When using this ADD/DROP option, the KNTA_USER_SECURITY_INT table must also be populated. However, security information can be directly obtained from the LDAP server.
- Overwrite the security group specification to include only the specified security groups. This deletes all references to the user's security groups and replaces them with the selected ones.
- Add missing security groups. This creates a new security group, but does not link the user to that security group.

Step 2: Add/Drop Security Groups

This is an optional step. If you decide that you want to use the ADD/DROP option, see "Step 2: Add and Drop Security Groups" on page 11.

Step 3: Add KNTAUser Attribute

Adding the KNTAUser attribute to users on an LDAP server is a convenient way to mark users for importing, when **LDAP Import PPM User Only** is set to **Yes.** It is not a required step. **LDAP Import PPM User Only** can be set to **No,** and the **Search Filter** field used to query for the attribute of your choice. If **LDAP Import PPM User Only** is set to **Yes** on the Import Users report, only the LDAP users with the KNTAUser attribute are imported. To apply the KNTAUser attribute to users on an LDAP server, it is necessary to run a command locally on the server machine.

1. Log on to the PPM Center server machine.
2. Navigate to <*PPM_Home*>`/bin` directory, where <*PPM_Home*> represents the installation path for PPM Center.
3. Open a bash shell.
4. Log on as an LDAP user who has privileges to modify the LDAP schema.
5. To execute the `kLdap.sh` command, either:
   - Type `kLdap.sh`

     A prompt for a number of LDAP server parameters appears. Provide the requested information.

- Type `kLdap.sh -s`

  The LDAP parameters are read from the `server.conf` file and no additional information are requested.

> **Note:** LDAP users can only logon in PPM Center-only mode if they have a password defined in PPM Center. Also, if the server is in PPM Center-only mode, PPM Center passwords can be set for LDAP users. These passwords are not required.

For more information on the `server.conf` file, see the *Installation and Administration Guide*.

Step 4: Map LDAP Attribute

You can map the attributes on the LDAP server to attributes used by the PPM Server. Some of this mapping occurs by default, but it can also be controlled using the procedure in this section.

To map LDAP attributes:

1. Navigate to the following file:

   `<`*PPM_Home*`>/integration/ldap/LdapAttribute.conf`

   where <*PPM_Home*> represents the installation path for PPM Center.
2. (Optionally) Refer to the Sample files for a sample showing mapping to a Red Hat Directory Server and an Active Directory Server. The default mapping is for a Red Hat Directory Server.
3. Edit the file using the tool of your choice.
4. Map the attribute according to your needs.

The `LdapAttribute.conf` file is described in the *Installation and Administration Guide*.

Step 5: Configure the PPM Server

Several PPM Server parameters need to be considered when performing a user import from an LDAP server.

To set parameters in the server.conf file:

1. Navigate to the following:

   <*PPM_Home*>/server.conf

   where <*PPM_Home*> represents the installation path for PPM Center.
2. Changing the parameters as necessary.
3. Stop and restart the PPM Server.

For more information on the server.conf file, see the *Installation and Administration Guide*.

Step 6: Start the Import

To import data from the interface tables, the Import Users report is used.

The Import User report:

- Populates the interface tables with records from the LDAP server.
- Validates the user information.
- Imports validated users into PPM Center tables. Partial imports are not allowed. Users with one or more failed fields are not imported.
- Reports on the results of the execution, listing the specified users that failed validation and the specific validation errors they encountered.

> **Note:** You can import users from Org Units that do not have unique names but are of different hierarchical levels. A **Hierarchy** column or option is added to pages or popup windows that are related to Org Units to help differentiate the hierarchical levels of the Org Units you import.

To run the Import Users report:

1. Log on to PPM Center.
2. From the menu bar, select **Open > Reports > Create Report**.

   The Submit Report page appears.
3. Select **Administrative** from the Report Category.

   The page is updated showing the list of administrative reports.
4. Select **Import Users**.

The Submit Report: Import Users window appears.

5. Complete the fields as described in "Complete the fields, as described in the following table. " on page 27

   The Import Users report has several parameters for controlling the behavior of the program execution. Pay special attention to:

   - **Product Licenses**
   - **LDAP Import** - Set this field to **Yes**
   - **Search Filter**
   - **User Authentication Mode**
   - **Link User Security Groups from LDAP Groups**
   - **Import Modified**

> **Note:** All users imported using the Import Users report have the same user privileges. To set different attributes (security groups or product permissions) for imported users, it is necessary to run the report multiple times.

> **Caution:** USER_DATA for users is not validated as part of an import.

> **Note:** Remember that test the process by setting the **Run Import** field to **No.**

Step 7: Verify Successful Completion

If any customizations to the import process have been made, it is extremely important to confirm that the import was successful.

To confirm that the import process completed successfully:

1. Click **View Report** to review the results of the import.

   This report identifies any errors with the import.

2. If errors are present, start your troubleshooting by referring to "Correcting Failures" on the next page.

All interface tables are automatically cleared by the purge service. The purging process depends on the following parameters in the server.conf file:

- **ENABLE_INTERFACE_CLEANUP.** Enables or disables the purge process.
- **DAYS_TO_KEEP_INTERFACE_ROWS.** Determines the number of days that records are retained in the interface tables.

For more information on the `server.conf` file, see the *Installation and Administration Guide*.

Correcting Failures

When a user is successfully imported, information stored in the interface tables is not deleted, and no additional action is required. It is possible to view and process the user with the standard interface.

For users that fail to import, you should:

1. Examine the audit report from the open interface report.
2. Identify the failed records and the specific reasons for each failure.
3. (Optional) Depending on the reasons, correct the problem as necessary.

Failures may be due to:

- A mapping problem between the source data and existing data in PPM Center.
- Missing information that cannot be defaulted.

  For example, users require a username. If the username columns are left blank for records in the user interface table, the records fail validation. To correct this, the custom program or procedure that inserts records into the interface table needs to be modified to include this required data.
- A large volume of data being imported. If you suspect that this is the problem, confirm this hypothesis by importing a smaller number of records, then checking to see if the error message persists.

> **Note:** During the initial implementation of the open interface, the mapping between the third-party source and PPM Center should be thoroughly reviewed and the load programs thoroughly tested in a testing instance.
>
> It is good practice to monitor executions of the open interface and periodically monitor the import of desired data into PPM Center.

Chapter 3: Organization Unit Open Interface

Organization Unit Open Interface Overview

PPM Center includes an interface for importing information. This open interface can import organizational models from third-party systems including LDAP databases, internally developed organization modeling systems, or human resources systems.

You should periodically synchronize the organizational model in PPM Center with the authoritative data source within your company. The synchronization process involves importing organization unit attributes of the various resources into the following interface tables:

- "KRSC_ORG_UNITS_INT" on page 165
- "KRSC_ORG_UNIT_MEMBERS_INT" on page 168
- "KNTA_USERS_INT" on page 150

These interface tables are described in "Open Interface Data Models" on page 104. The columns that can be used when importing organizational models are detailed in this section.

The Organization Unit Open Interface supports:

- Simple imports
- LDAP imports

For information on mapping your organization model, see the *Resource Management User's Guide*. For information on user report types and running reports, see the *Reports Guide and Reference*. To review the LDAP authentication process, see "LDAP Authentication" on page 181.

Performing a Simple Import

Step 1: Load the Users

Load the users or resources into PPM Center by either:

- Running the Import Users report.

  For details, see "Performing a Simple Import" on page 10

- Using SQL*Loader, your favorite tool, or direct Oracle database-to-database communication, load your data into the KNTA_USERS_INT interface table.

  The specific fields in the KNTA_USERS_INT interface table that need to be populated are specified in "Table 2-2. KNTA_USERS_INT interface table" on page 15.

Step 2: Populate the Interface Tables

Using SQL*Loader, your favorite tool, or direct Oracle database-to-database communication, load your data into the required input columns in the following tables:

- "KRSC_ORG_UNITS_INT" on page 165
- "KRSC_ORG_UNIT_MEMBERS_INT" on page 168

See "Table 3-1. KRSC_ORG_UNITS_INT interface table" below and "Table 3-2. KRSC_ORG_UNIT_MEMBERS_INT interface table" on page 42.

> **Caution:** User data is not validated during import.

Table 3-1. KRSC_ORG_UNITS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	Required	NUMBER	Uniquely identifies each transaction.
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language. If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.

Table 3-1. KRSC_ORG_UNITS_INT interface table, continued

Column	Usage	Data Type	Description
EXISTS_FLAG	Optional	VARCHAR2	Indicates whether or not the organization unit already exists.
PROCESS_ PHASE	Optional	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_ STATUS	Optional	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_ BY_USERNAME.
CREATED_BY_ USERNAME	Optional	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_ DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.
DEST_ CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_ BY_USERNAME. Ig both are left blank, the value is set to the user currently running the report.
DEST_ CREATION_ DATE	Optional	DATE	Indicates the date the record is created in the destination (PPM Center instance). If left blank, the value is derived from CREATION_ DATE.
DEST_LAST_ UPDATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the set to the user currently running the report.

Table 3-1. KRSC_ORG_UNITS_INT interface table, continued

Column	Usage	Data Type	Description
DEST_LAST_ UPDATE_DATE	Optional	DATE	Indicates the date that the organization data was last updated. If left blank, the current date is used.
DEST_ENTITY_ UPD_DATE	Optional	DATE	Indicates the date that either the organization or membership data was last updated. If left blank, the current date is used.
SOURCE	Required	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
SOURCE_ TYPE_CODE	Required	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_WF.
ORG_UNIT_ID	Optional	NUMBER	Identifies the organization unit ID. For new organization units, the value is derived from the KRSC_ORG_UNITS_S sequence. For existing organization units, if left blank, the value is derived from ORG_UNIT_NAME.
ORG_UNIT_ NAME	Required	VARCHAR2	Identifies the organization unit name.
PARENT_ORG_ UNIT_ID	Optional	NUMBER	Identifies the parent unit ID for the organization unit. If left blank, the value is derived from PARENT_ ORG_UNIT_NAME.
PARENT_ORG_ UNIT_NAME	Required	VARCHAR2	Identifies the parent unit name for the organization unit. If left blank, then the organization unit appears as a top level unit in the organization model.
MANAGER_ID	Optional	NUMBER	Identifies the manager associated with the organization unit. If left blank, the value is derived from MANAGER_ USERNAME.
MANAGER_ USERNAME	Required	VARCHAR2	Specifies the name of the manager.

Table 3-1. KRSC_ORG_UNITS_INT interface table, continued

Column	Usage	Data Type	Description
MANAGER_ LOGON_ IDENTIFIER	Required	VARCHAR2	Specifies the ID of the manager. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ ID, the MANAGER_LOGON_IDENTIFIER column must be populated. Otherwise, the MANAGER_ USERNAME column must be populated.
DEPARTMENT_ CODE	Optional	VARCHAR2	Specifies the code for the department.
DEPARTMENT_ MEANING	Required	VARCHAR2	Specifies the description of the department.
LOCATION_ CODE	Optional	VARCHAR2	Specifies the code for the location.
LOCATION_ MEANING	Required	VARCHAR2	Specifies the description of the location.
CATEGORY_ CODE	Optional	VARCHAR2	Specifies the code for the category.
CATEGORY_ MEANING	Required	VARCHAR2	Specifies the description of the category.
ENABLED_FLAG	Optional	VARCHAR2	Indicates whether or not the organization unit is enabled upon import.
USER_DATA_ SET_CONTEXT_ ID	Required	NUMBER	Sets the context identifier for the USER_DATA fields. Supply this or ORG_UNIT_USERNAME.

DISTINGUISH_ NAME	Required	VARCHAR2	Specifies the distinguished name for the organization unit in the following format:
			`OU=>SubOU1=>SubSubOU1`
			`OU=>SubOU2`

PARENT_ DISTINGUISH_ NAME	Required	VARCHAR2	Specifies the distinguished name for the parent organization unit in the following format:
			`OU=>SubOU1`
			`OU`

USER_DATA1 VISIBLE_USER_ DATA1	Required	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.

Table 3-1. KRSC_ORG_UNITS_INT interface table, continued

Column	Usage	Data Type	Description
through USER_DATA20 VISIBLE_USER_DATA20			This information is not validated nor does it have a default value.

Table 3-2. KRSC_ORG_UNIT_MEMBERS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
EXISTS_FLAG	O	VARCHAR2	Indicates whether or not the organization unit already exists.
PROCESS_PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME.
CREATED_BY_USERNAME	I/O	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_DATE	I	DATE	Indicates the transaction date. If left blank, the current date is used.

Table 3-2. KRSC_ORG_UNIT_MEMBERS_INT interface table, continued

Column	Usage	Data Type	Description
DEST_ CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME. If both are left blank, the value is set to the user currently running the report.
DEST_ CREATION_ DATE	I/O	DATE	Indicates the date the record is created in the destination (PPM Center instance). If left blank, the value is derived from CREATION_ DATE.
DEST_LAST_ UPDATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the user currently running the report.
DEST_LAST_ UPDATE_DATE	I/O	DATE	Indicates the date that the membership data was last updated. If left blank, the current date is used.
DEST_ENTITY_ UPD_DATE	I/O	DATE	Indicates the date that either the organization or membership data was last updated. If left blank, the current date is used.
SOURCE	I	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
SOURCE_ TYPE_CODE	I	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_WF.
ORG_UNIT_ MEMBER_ID	I/O	NUMBER	Identifies the organization unit member. This is normally left blank and is derived from the KRSC_ORG_UNIT_MEMBER_S sequence.
ORG_UNIT_ID	I/O	NUMBER	Identifies the organization unit ID. This is normally left blank and is derived from KRSC_ ORG_UNITS.
ORG_UNIT_	I	VARCHAR2	Identifies the parent unit name for the organization unit.

Table 3-2. KRSC_ORG_UNIT_MEMBERS_INT interface table, continued

Column	Usage	Data Type	Description
NAME			
USER_ID	I/O	NUMBER	Identifies the user. For existing users, this refers to the USER_ID column in KNTA_USERS. This is normally left blank and is derived from the KNTA_USERS_S sequence.
USERNAME	I	VARCHAR2	Identifies the name used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_NAME, the USERNAME column must be populated for the user import. Otherwise, populate the LOGON_ID column.
LOGON_IDENTIFIER	I	VARCHAR2	Identifies the ID used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, the LOGON_ID column must be populated. Otherwise, populate the USERNAME column.
ORG_UNIT_DISTINGUISH_NAME	I	VARCHAR2	Specifies the distinguished name for the organization unit.

Step 3: Start the Import

To import data from the interface tables, the Run PPM Organization Unit Interface report is used.

The Run PPM Organization Unit Interface report:

- Queries the KRSC_ORG_UNITS_INT interface table for active records matching the given selection criteria
- Queries the KRSC_ORG_UNIT_MEMBERS_INT interface table
- Queries the KNTA_USERS_INT interface table
- Validates the organization information

- Imports validated organization units, organization unit members, and any new users into PPM Center tables
- Updates the KNTA_SECURITY_GROUPS table with information derived from the import
- Reports on the results of the execution, listing the specified organization units and organization members that failed validation, and the specific validation errors were encountered

To run the Run PPM Organization Unit Interface report:

1. Log on to PPM Center.
2. From the menu bar, select **Open > Reports > Create Report**.

   The Submit New Report page appears.
3. Select **Administrative** from the Report Category.

   The page is updated showing the list of administrative reports.
4. Select **Run PPM Organization Unit Interface**.

   The Submit Report: Run PPM Organization Unit Interface window appears.
5. Complete the fields as described in the following table.

   Required fields are denoted with a red asterisk next to the field on the screen. Depending on your selections, the required fields may vary.

   Control the behavior of the program execution, with the following fields:

   - **LDAP Import** - Set this field to **No**
   - **Extensible Search Filter**
   - **User Authentication Mode**
   - **Import Modified**
   - **Product Licenses**

Field Name	Description
Group Id	Specifies the group ID for which the interface program should be run. The interface program will only look for records with this value in the GROUP_ID column. This is useful when importing a batch of packages.
Source Code	Indicates whether or not to set the SOURCE_CODE column of the final requests created with a free-form text code. This is used as an indicator of how the request was created for auditing or testing purposes.

Field Name	Description
	For an LDAP import, set to LDAP_IMPORT.
Run Import	• **If set to Yes.** Indicates that the program will process the records in the interface table and try to import them. • **If set to No.** Indicates that the program will simply report on the records in the interface table. This option is useful when auditing prior executions of the interface.
Show Successful Transactions	Indicates whether or not to show packages and package lines that were successfully imported.
Show Failed Transactions	Indicates whether or not to show packages and package lines that were not successfully imported.
Default Password	Specifies a default password. For an LDAP import, this field is disabled and the passwords are automatically fetched from the LDAP server.
Org Unit Member Action	Specifies how the organization unit membership is managed during the import for existing organization units. Select one of the following options: • **No Changes to Existing Members.** The import does not add or remove any members in an existing organization unit. • **Replace All Existing Members.** Removes all members of the organization unit and replaces them with the members specified in the KRSC_ORG_UNIT_MEMBERS_INT interface table. • **Replace LDAP Imported Members.** Removes all members of the organization unit who are associated using LDAP and replaces them with members associated with the organization unit on the LDAP server. Other members, who have been added manually using the standard interface or by a separate open interface import, are not altered.
Add Missing Security Groups	Indicates whether or not to add missing security groups.
Disable Users Not Imported	Indicates whether or not to import users who have been disabled.
Keep existing values for empty columns	Indicates whether or not to keep existing values stored for empty columns.

Field Name	Description
Region for Org Unit	Specifies the regional calendar for imported organizational units. If no regional calendar is specified, the system default calendar is used.
LDAP Import	Indicates whether or not to perform LDAP import. Set this to **Yes** if the authentication mode in the `server.conf` file contains LDAP or an Exchange server (NTLM).
Extensible Search Filter	Specifies the search filter using syntax of the conditions on PPM Center commands.
User Authentication Mode	Selects a user authentication mode. (LDAP or NTLM only)
Import Modified	Indicates whether or not the import can be modified. (LDAP or NTLM only)
Product Licenses	Establishes the licensing for the imported users.

6. (Optional) To test the process, set the **Run Import** field to **No**.

For information about extensible search filters, see "Examples of Search Filter Values" on page 30

For more information on the `server.conf` file, see the *Installation and Administration Guide*.

Step 4: Verify Successful Completion

If any customizations to the import process have been made, it is extremely important to confirm that the import was successful.

To confirm that the import process completed successfully:

1. Review the Import Summary results of the import.

   This report identifies any errors with the import.

2. If errors are present, start your troubleshooting by referring to "Correcting Failures" on page 50.

All interface tables are automatically cleared by the purge service. The purging process depends on the following parameters in the `server.conf` file:

- **ENABLE_INTERFACE_CLEANUP.** Enables or disables the purge process.
- **DAYS_TO_KEEP_INTERFACE_ROWS.** Determines the number of days that records are retained in the interface tables.

For more information on the `server.conf` file, see the *Installation and Administration Guide*.

Performing an LDAP Import

Step 1: Map the LDAP Attributes

You can map the attributes on the LDAP server to attributes used by the PPM Server. Some of this mapping occurs by default, but it can also be controlled using the procedure in this section.

To map LDAP attributes:

1. Navigate to the following file:

   `<PPM_Home>/integration/ldap/LdapAttribute.conf`

   where `<PPM_Home>` represents the installation path for PPM Center.
2. (Optionally) Refer to the Sample files for a sample showing mapping to a Red Hat Directory Server and an Active Directory Server. The default mapping is for a Red Hat Directory Server.
3. Edit the file using the tool of your choice.
4. Map the attribute according to your needs.

> **Caution:** It is recommended that you verify the mappings for USERNAME, FIRST_NAME, and LAST_NAME before proceeding.

The `LdapAttribute.conf` file is described in the *Installation and Administration Guide*.

Step 2: Configure the PPM Server

Several PPM Server parameters need to be considered when performing an import from an LDAP server.

To set parameters in the server.conf file:

1. Navigate to the following:

   `<PPM_Home>/server.conf`

   where <*PPM_Home*> represents the installation path for PPM Center.
2. Changing the parameters as necessary.
3. Stop and restart the PPM Server.

For more information on the `server.conf` file, see the *Installation and Administration Guide*.

Step 3: Start the Import

To import data from the interface tables, the Run PPM Organization Unit Interface report is used.

The Run PPM Organization Unit Interface report:

- Populates the interface tables with records from the LDAP server
- Validates the user information
- Imports validated organization units and organization unit members into PPM Center tables
- Reports on the results of the execution, listing the specified users that failed validation and the specific validation errors they encountered

> **Note:** You can import users from Org Units that do not have unique names but are of different hierarchical levels. A Hierarchy column or option is added to pages or popup windows that are related to Org Units to help differentiate the hierarchical levels of the Org Units you import.

To run the Run PPM Organization Unit Interface report:

1. Log on to PPM Center.
2. From the menu bar, select **Open > Reports > Create Report**.

   The Submit New Report page appears.
3. Select **Administrative** from the Report Category.

   The page is updated showing the list of administrative reports.
4. Select **Run PPM Organization Unit Interface**.

   The Submit Report: Run PPM Organization Unit Interface window appears.

5. Complete the fields as described in the "Complete the fields as described in the following table. " on page 45

   Control the behavior of the program execution, with the following fields:

   - **LDAP Import** - Set this field to **Yes**
   - **Extensible Search Filter**
   - **User Authentication Mode**
   - **Import Modified**
   - **Product Licenses**

6. (Optional) To test the process, set the **Run Import** field to **No**.

Step 4: Verify Successful Completion

If any customizations to the import process have been made, it is extremely important to confirm that the import was successful.

To confirm that the import process completed successfully:

1. Review the results of the Import Summary at the end of the report.

   This report identifies any errors with the import.

2. If errors are present, start your troubleshooting by referring to "Correcting Failures" below

All interface tables are automatically cleared by the purge service. The purging process depends on the following parameters in the server.conf file:

- **ENABLE_INTERFACE_CLEANUP.** Enables or disables the purge process.
- **DAYS_TO_KEEP_INTERFACE_ROWS.** Determines the number of days that records are retained in the interface tables.

For more information on the `server.conf` file, see the *Installation and Administration Guide*.

Correcting Failures

When a user is successfully imported, information stored in the interface tables is not deleted, and no additional action is required. It is possible to view and process the user with the standard interface.

For users that fail to import, you should:

1. Examine the audit report from the open interface report.
2. Identify the failed records and the specific reasons for each failure.
3. (Optional) Depending on the reasons, correct the problem as necessary.

Failures may be due to:

- A mapping problem between the source data and existing data in PPM Center.
- Missing information that cannot be defaulted.

  For example, users require a username. If the username columns are left blank for records in the user interface table, the records fail validation. To correct this, the custom program or procedure that inserts records into the interface table needs to be modified to include this required data.
- A large volume of data being imported. If you suspect that this is the problem, confirm this hypothesis by importing a smaller number of records, then checking to see if the error message persists.

> **Note:** During the initial implementation of the open interface, the mapping between the third-party source and PPM Center should be thoroughly reviewed and the load programs thoroughly tested in a testing instance.
>
> It is good practice to monitor executions of the open interface and periodically monitor the import of desired data into PPM Center.

Chapter 4: Request Open Interface

Request Open Interface Overview

The open interface supports request creation. This open interface enables integration with non-PPM Center products. Relevant information from these products can be used to generate the appropriate request using the open interface. The open interface can also be used as a conversion mechanism to convert data from a legacy system into Demand Management during initial implementation.

> **Note:** The Request Open Interface cannot be used to update existing requests in the PPM Center.

The synchronization process involves importing request attributes to the following interface tables:

- "KCRT_REQUESTS_INT" on page 128
- "KCRT_REQUEST_DETAILS_INT" on page 133
- " KCRT_REQ_HEADER_DETAILS_INT" on page 135
- "KCRT_TABLE_ENTRIES_INT" on page 136

The following field group interface tables may also be needed:

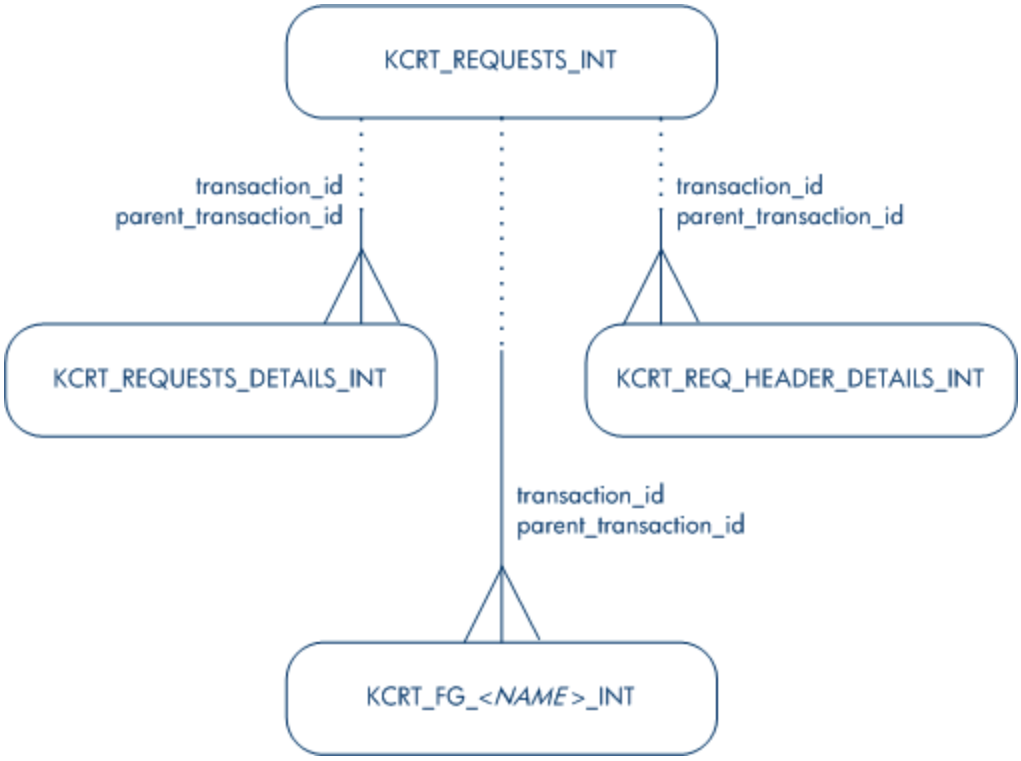
- "KCRT_FG_DEMAND_SCHEDULE_INT" on page 104
- "KCRT_FG_MASTER_PROJ_REF_INT" on page 106
- "KCRT_FG_PFM_ASSET_INT" on page 107
- "KCRT_FG_PFM_PROJECT_INT" on page 109
- "KCRT_FG_PFM_PROPOSAL_INT" on page 112
- "KCRT_FG_PROG_ISSUE_INT" on page 115
- "KCRT_FG_PROG_REFERENCE_INT" on page 116
- "KCRT_FG_PROG_RESOURCE_REQ_INT" on page 117
- "KCRT_FG_PROG_ISSUE_INT" on page 115
- "KCRT_FG_PROJ_RISK_INT" on page 119
- "KCRT_FG_PROJ_SCOPE_CHANGE_INT" on page 120

- "KCRT_FG_SLA_INT" on page 124
- "KCRT_FG_WORK_ITEMS_INT" on page 125

These interface tables are described in "Open Interface Data Models" on page 104. With the exception of the field group interface tables, the columns that can be used when importing requests are detailed in the appropriate step within this section.

The following figure displays the relationships between the various Request Open Interface tables. Note that the placeholder (<*NAME*>) represents any of the field group interface tables.

Figure 4-1. Request interface and supporting tables



For information on requests, see the *Demand Management Configuration Guide* and *Demand Management User's Guide* documents.

For information on user report types and running reports, see the *Reports Guide and Reference*.

Performing an Import

Step 1: Populate the Request Interface Tables

1. Using SQL*Loader, your favorite tool, or direct Oracle database-to-database communication, load your data into the required input columns in the following tables:
   - "KCRT_REQUESTS_INT" on page 128
   - "KCRT_REQUEST_DETAILS_INT" on page 133
   - " KCRT_REQ_HEADER_DETAILS_INT" on page 135
   - "KCRT_TABLE_ENTRIES_INT" on page 136

   See "Table 4-1. KCRT_REQUESTS_INT interface table" below through "Table 4-4. KCRT_TABLE_ENTRIES_INT interface table" on page 62.
2. (Optional) Use the applicable field group interface tables from "Open Interface Data Models" on page 104.
3. (Optional) Modify of the table records after they have been brought into the interface table.

   This can include the:
   - Setting of ID columns, such as GROUP_ID and TRANSACTION_ID
   - Defaulting of specific data not available in the source of the request, such as the third-party application or the ASCII file

   > **Caution:** User data is not validated during import.

"Table 4-1. KCRT_REQUESTS_INT interface table" below through "Table 4-4. KCRT_TABLE_ENTRIES_INT interface table" on page 62 describe the interface tables that must be populated with data.

Table 4-1. KCRT_REQUESTS_INT interface table

Column	Usage	Data Type	Description
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language.

Table 4-1. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
			If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
TRANSACTION_ ID	Required	NUMBER	Uniquely identifies each transaction. If any detail table is being used, set the PARENT_ TRANSACTION_ID in the detail interface tables to this value.
PROCESS_ PHASE	Optional	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_ STATUS	Optional	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
REQUEST_ID	Optional	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
CREATION_ DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.
CREATED_ USERNAME	Optional	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_ USERNAME.
LAST_UPDATE_	Optional	DATE	Indicates the transaction date.

Table 4-1. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
DATE			If left blank, the current date is used.
LAST_ UPDATED_ USERNAME	Required	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction.
LAST_ UPDATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. This is normally left blank and is derived from LAST_ UPDATED_USERNAME.
ENTITY_LAST_ UPDATE_DATE	Optional	DATE	Indicates the transaction date. This is normally left blank and the current date is used.
REQUEST_ NUMBER	Optional	VARCHAR2	Identifies the request. This is normally left blank and is derived from REQUEST_ID. If a value is entered, it should be unique and should match the value in the REQUEST_ID field.
REQUEST_ TYPE_NAME	Required	VARCHAR2	Identifies the request type. Derived from KCRT_REQUESTS_TYPES.
REQUEST_ TYPE_ID	Optional	NUMBER	Identifies the request type. If left blank, the value is derived from REQUEST_ TYPE_NAME.
REQUEST_ SUBTYPE_ NAME	Required	VARCHAR2	Identifies the request subtype. If a value is entered, it should be a valid subtype from KCRT_REQUEST_SUB_TYPES.
REQUEST_ SUBTYPE_ID	Optional	NUMBER	Identifies the request subtype. If left blank, the value is derived from REQUEST_ SUBTYPE_NAME.
DESCRIPTION	Required	VARCHAR2	Specifies a user-visible description of the request.
RELEASE_ DATE	Optional	DATE	Indicates when the request first became active. For new requests, this should be left blank and the current date is used. When converting existing requests from a third-party system, enter the initial creation date of the request

Table 4-1. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
			in the remote system.
STATUS_NAME	Optional	VARCHAR2	Indicates the current status of the request. This should be a valid status for the given request. This should be a request status for at least one workflow step of the workflow. If left blank, the new request will get the initial status indicated on the request type definition.
STATUS_ID	Optional	NUMBER	Indicates the current status of the request. If left blank, the value is derived from STATUS_NAME.
WORKFLOW_NAME	Optional	VARCHAR2	Specifies the workflow that the request should follow. This is normally left blank and its value is based on the values for request type, department, and application for the request.
WORKFLOW_ID	Optional	NUMBER	Specifies the workflow that the request should follow. This is normally left blank and the value is derived from WORKFLOW_NAME.
DEPARTMENT_CODE	Optional	VARCHAR2	Specifies the code for the department.
DEPARTMENT_NAME	Required	VARCHAR2	Specifies the name of the department. This should be a valid MEANING from KNTA_LOOKUPS where `LOOKUP_TYPE = 'DEPARTMENT_CODE'`.
PRIORITY_CODE	Optional	VARCHAR2	Specifies the user-defined priority for the request.
PRIORITY_NAME	Required	VARCHAR2	Specifies the user-defined priority name for the request. If entered, this should be a valid MEANING from KNTA_LOOKUPS where `LOOKUP_TYPE = 'REQUEST_PRIORITY'`.
APPLICATION	Required	VARCHAR2	Indicates the user-defined application for the request. This should be a valid LOOKUP_CODE from KNTA_LOOKUPS where `LOOKUP_TYPE = 'APPLICATION'`.

Table 4-1. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
ASSIGNED_TO_USERNAME	Required	VARCHAR2	Specifies the USERNAME (from KNTA_USERS) that should initially be assigned the request.
ASSIGNED_TO_USER_ID	Optional	NUMBER	Specifies the USER_ID (from KNTA_USERS) that should initially be assigned the request. If left blank, the value is derived from ASSIGNED_TO_USERNAME.
ASSIGNED_TO_GROUP_NAME	Required	VARCHAR2	Specifies the SECURITY_GROUP_ID (from KNTA_SECURITY_GROUPS) that should initially be assigned the request.
ASSIGNED_TO_GROUP_ID	Optional	NUMBER	Specifies the SECURITY_GROUP_ID that should initially be assigned to the request. This is normally left blank and the value is derived from ASSIGNED_TO_GROUP_NAME.
PROJECT_CODE	Required	VARCHAR2	Indicates the user-defined project for the request. This should be a valid value from KNTA_LOOKUPS where `LOOKUP_TYPE = 'PROJECT'`.
CONTACT_FIRST_NAME	Required	VARCHAR2	Specifies the first name of the contact for the request. This should be a valid value from FIRST_NAME in KCRT_CONTACTS. If a value is entered, CONTACT_LAST_NAME must also be populated.
CONTACT_LAST_NAME	Required	VARCHAR2	Specifies the last name of the contact for the request. This should be a valid value from LAST_NAME in KCRT_CONTACTS. If a value is entered, CONTACT_FIRST_NAME must also be populated.
CONTACT_ID	Optional	NUMBER	Specifies the ID of the contact for the request. This is derived from the CONTACT_FIRST_NAME and CONTACT_LAST_NAME.
RELEASED_FLAG	Required	VARCHAR2	Indicates whether or not the request should be released after import. Valid values are:

Table 4-1. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
			• Y • N The default value is N.
USER_DATA_ SET_CONTEXT_ ID	Obsolete	NUMBER	No longer used.
USER_DATA1 VISIBLE_USER_ DATA1 through USER_DATA20 VISIBLE_ USERS_DATA20	Optional	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined. This information is not validated nor does it have a default value.
PARAMETER_ SET_CONTEXT_ ID	Required	NUMBER	Sets the context identifier for the detail fields. Either this or REQUEST_TYPE_NAME must be populated.
NOTES	Required	LONG	Optional. Provides free-form note text that is visible in the **Notes** tab of the request window. Carriage returns should be represented as {\n} and is replaced with actual carriage returns when the note is moved into the notes table. This can be helpful when the interface table is populated through SQL*Loader.
SOURCE_ TYPE_CODE	Required	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_RI.
SOURCE	Required	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
WORKFLOW_ STEP_ID	Optional	NUMBER	Identifies the workflow step that becomes eligible for user processing.
COMPANY	Required	VARCHAR2	Identifies the name of the company associated with this request.

Table 4-1. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
			This should be a valid LOOKUP_CODE from KNTA_LOOKUPS where LOOKUP_TYPE = `COMPANY'.

Table 4-2. KCRT_REQUEST_DETAILS_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
TRANSACTION_ID	Required	NUMBER	Uniquely identifies each transaction.
PARENT_TRANSACTION_ID	Required	NUMBER	Provides the transaction ID (from KCRT_REQUESTS_INT) of the parent table being imported.
REQUEST_DETAIL_ID	Optional	NUMBER	Identifies the detail ID of the request (from KCRT_REQUEST_DETAILS).
REQUEST_ID	Optional	NUMBER	Identifies the request. If left blank, the value is derived from the KCRT_REQUESTS_S sequence.
REQUEST_TYPE_ID	Optional	NUMBER	Identifies the request type. If left blank, the value is derived from REQUEST_TYPE_NAME.
PARAMETER_SET_CONTEXT_ID	Optional	NUMBER	Sets the context identifier for the detail fields. If left blank, the value is derived from the REQUEST_TYPE_NAME.
BATCH_NUMBER	Required	NUMBER	Specifies the batch number for the custom fields. This corresponds to the **Storage** tab in the **field** definition window on the request type.
PARAMETER1 VISIBLE_PARAMETER1 through	Required	VARCHAR2	Specifies the values for all the custom fields defined in the request.

Table 4-2. KCRT_REQUEST_DETAILS_INT interface table, continued

Column	Usage	Data Type	Description
PARAMETER50 VISIBLE_ PARAMETER50			
LOOKUP_TYPE1 VALIDATION_ TYPE_CODE1 through LOOKUP_TYPE50 VALIDATION_ TYPE_CODE50	Optional	VARCHAR2 VARCHAR2	Identifies the lookup type for each PARAMETER and the validation type code for each PARAMETER. This is required only if custom data is defined.

Table 4-3. KCRT_REQ_HEADER_DETAILS_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	Required	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	Required	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQ_HEADER_ DETAIL_ID	Optional	NUMBER	Identifies the header detail ID for the request. If left blank, the value is derived from the KCRT_ REQ_HEADER_DETAILS_S sequence.
REQUEST_ID	Optional	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_ TYPE_ID	Optional	NUMBER	Identifies the request type. This is normally left blank and is derived from

Table 4-3. KCRT_REQ_HEADER_DETAILS_INT interface table, continued

Column	Usage	Data Type	Description
			REQUEST_TYPE_NAME.
BATCH_ NUMBER	Required	NUMBER	Specifies the batch number for the custom fields. This corresponds to the **Storage** tab in the **field** definition window on the request type.
PARAMETER1 VISIBLE_ PARAMETER1 through PARAMETER50 VISIBLE_ PARAMETER50	Required	VARCHAR2	Specifies the values for all the custom fields defined in the request.
LOOKUP_TYPE1 VALIDATION_ TYPE_CODE1 through LOOKUP_ TYPE50 VALIDATION_ TYPE_CODE50	Required	VARCHAR2 VARCHAR2	Identifies the lookup type for each PARAMETER and the validation type code for each PARAMETER. This is required only if custom data is defined.

Table 4-4. KCRT_TABLE_ENTRIES_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_ S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	Required	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_	Required	NUMBER	Provides the transaction ID (from KCRT_ REQUEST_DETAILS_INT) of the parent table

Table 4-4. KCRT_TABLE_ENTRIES_INT interface table, continued

Column	Usage	Data Type	Description
ID			being imported.
PARENT_FIELD_TOKEN	Required	VARCHAR2	Specifies the token.
TABLE_ENTRY_ID	Optional	NUMBER	Identifies the table entry record. If left blank, the value is derived from the KCRT_TABLE_ENTRIES_S sequence.
REQUEST_ID	Optional	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
PARAMETER_SET_FIELD_ID	Optional	NUMBER	Specifies the field in the table to which this entry belongs.
SEQ	Required	NUMBER	Provides a user-visible sequence number for the package line. This must be a unique, positive integer that does not conflict with other records being imported.
PARAMETER_SET_CONTEXT_ID	Optional	NUMBER	Sets the context identifier for the detail fields. If left blank, the value is derived from the REQUEST_TYPE_NAME.
VISIBLE_PARAMETER1 PARAMETER1 through VISIBLE_PARAMETER50 PARAMETER50	Required	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.
LOOKUP_TYPE1 through LOOKUP_TYPE50	Required	VARCHAR2	Identifies the lookup type for each PARAMETER. This is required only if user data is defined.
VALIDATION_TYPE_CODE1 through VALIDATION_TYPE_CODE50	Optional	VARCHAR2	Identifies the validation type code for each PARAMETER. This is required only if user data is defined.

Step 2: Start the Import

To import data from the interface tables, the Import Requests report is used.

The Import Requests report:

- Queries the KCRT_REQUESTS_INT interface table for active records matching the given selection criteria.
- Defaults any information that has defaulting rules in Demand Management but has not been specified in the interface table records. For example, if the REQUEST_ID column is left blank, it is defaulted from a sequence.
- Validates request header and detail data for both referential and data integrity. This validation is based on the logic used when entering or updating data through the standard interface. Information in User Data fields is not validated.
- Imports validated requests into the Demand Management request tables. Partial imports are not allowed. Requests with one or more failed fields will not be imported.
- Moves the request to the appropriate request status and moves the request to the first workflow step corresponding to the specific request status, if indicated.
- Reports on the results of the execution, listing the specified requests that failed validation and the specific validation errors they encountered.

To run the Import Requests report:

1. Log on to PPM Center.
2. From the menu bar, select **Open > Reports > Create Report**.

   The Submit New Report page appears.
3. Select **Administrative** from the Report Category.

   The page is updated showing the list of administrative reports.
4. Select **Import Requests.**

   The Submit Report: Import Request window appears.
5. Complete the fields, as described in the following table.

Required fields are denoted with a red asterisk next to the field on the screen. Depending on your selections, the required fields may vary.

Field Name	Description
Group Id	Specifies the group ID for which the interface program should be run. The interface program will only look for records with this value in the GROUP_ID column. This is useful when importing a batch of Requests.
Run Import	◦ **If set to Yes.** Indicates that the program will process the records in the interface table and try to import them. ◦ **If set to No.** Indicates that the program will simply report on the records in the interface table. This option is useful when auditing prior executions of the open interface.
Show Successful Transactions	Indicates whether or not to show requests that were successfully imported.
Source Code	Indicates whether or not to set the SOURCE_CODE column of the final requests created with a free-form text code. This is used as an indicator of how the request was created for auditing or testing purposes.

6. (Optional) To test the process, set the **Run Import** field to **No**.

Step 3: Verify Successful Completion

If any customizations to the import process have been made, it is extremely important to confirm that the import was successful.

To confirm that the import process completed successfully:

1. Review the results of the Import Summary.

   This report identifies any errors with the import.

2. If errors are present, start your troubleshooting by referring to "Correcting Failures" on the next page.

All interface tables are automatically cleared by the purge service. The purging process depends on the following parameters in the server.conf file:

- **ENABLE_INTERFACE_CLEANUP.** Enables or disables the purge process.
- **DAYS_TO_KEEP_INTERFACE_ROWS.** Determines the number of days that records are

retained in the interface tables.

For more information on the `server.conf` file, see the *Installation and Administration Guide*.

Correcting Failures

When a request is successfully imported, information stored in the interface tables is not deleted, and no additional action is required. It is possible to view and process the request using Demand Management.

> **Note:** During the initial implementation of the open interface, the mapping between the third-party source and PPM Center should be thoroughly reviewed and the load programs thoroughly tested in a testing instance.
>
> It is good practice to monitor executions of the open interface and periodically monitor the import of desired data into Demand Management.

For users that fail to import, you should:

1. Examine the audit report from the open interface report.
2. Identify the failed records and the specific reasons for each failure.
3. (Optional) Depending on the reasons, correct the problem as necessary.

Failures may be due to:

- A mapping problem between the source data and existing Demand Management data.

  You may need to add the specific project in Demand Management, or map the source project to a project name that already exists in Demand Management.

- Missing information that cannot be defaulted.

  For example, requests require a request type. If the request type columns are left blank for records in the requests interface table, the records will fail validation.

  To correct this, the custom program or procedure that inserts records into the interface table needs to be modified to include this required data.

- A large volume of data being imported.

  If you suspect that this is the problem, confirm this hypothesis by importing a smaller number of records, then checking to see if the error message persists.

- Other configuration and mapping problems in either the source or in Demand Management, or could be the result of errors in the custom loading program.

Chapter 5: Package Open Interface

Package Open Interface Overview

PPM Center includes an open interface for package creation and the creation of new package lines.

The Package Open Interface can be used:

- To support site-specific customizations such as the automatic addition of package lines based on the processing of a package, or the spawning of child packages from other packages
- As a conversion mechanism to convert data from a legacy system into Deployment Management during initial implementation

Data added to the interface tables is validated and eventually imported into standard Deployment Management tables. This generates packages and package lines that can be processed using Deployment Management.

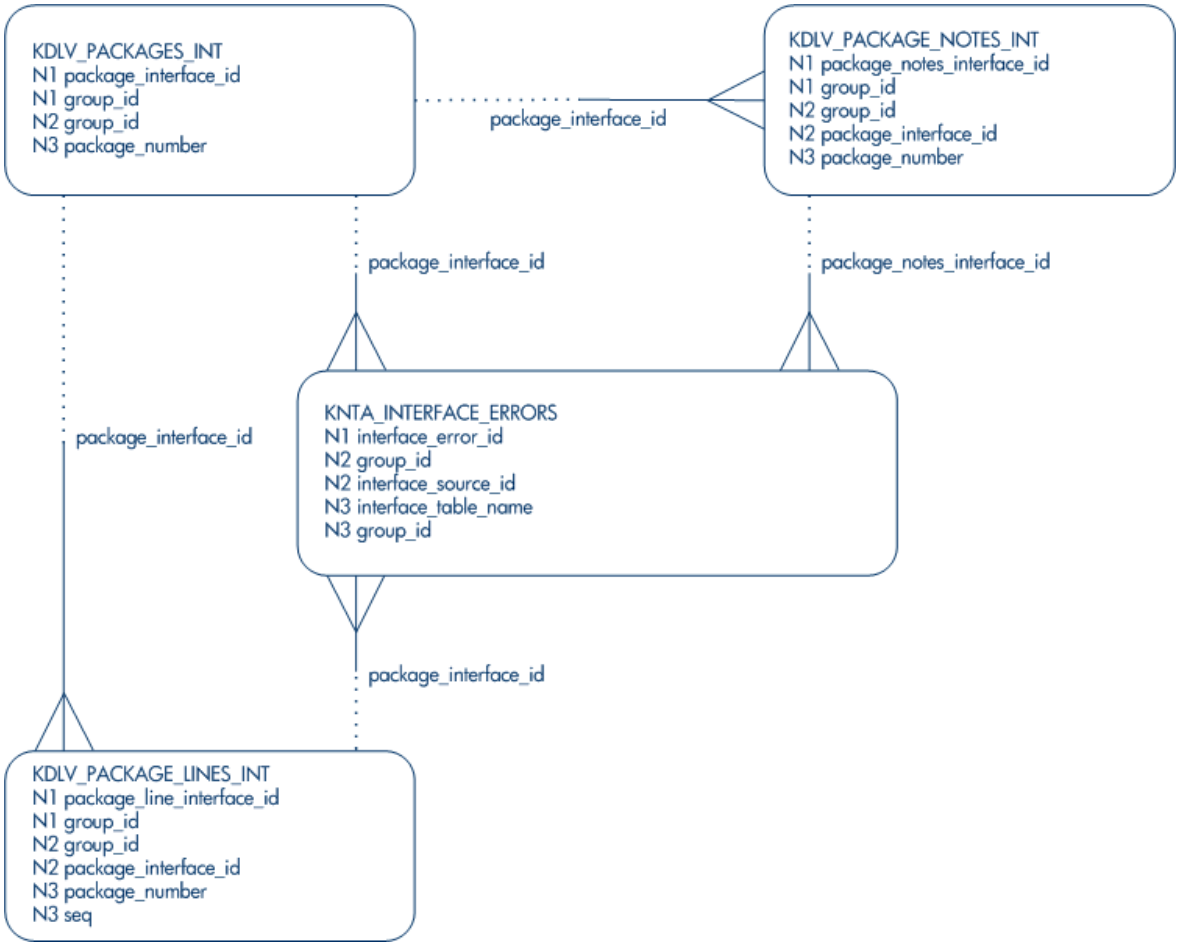
The synchronization process involves importing package attributes into the following interface tables:

- "KDLV_PACKAGES_INT" on page 138
- "KDLV_PACKAGE_LINES_INT" on page 142
- "KDLV_PACKAGE_NOTES_INT" on page 146

These interface tables are described in "Open Interface Data Models" on page 104. The columns that can be used when importing are detailed in the appropriate step within this section.

The following figure displays the relationships between the various Package Open Interface tables.

Figure 5-1. Package interface and supporting tables



For information on requests, see the *Demand Management Configuration Guide* and *Demand Management User's Guide*.

> **Note:** Integration between the products in PPM Center is automatic and does not require user development or user customization involving the open interface. For example, no customization work needs to be done to support the creation of Deployment Management packages from Demand Management requests.
>
> This does assume that the relevant workflows and request types exist.

Performing an Import

Step 1: Load the Interface Tables

Using SQL*Loader, your favorite tool, or direct Oracle database-to-database communication, load your data into the required input columns in the following tables:

- "KDLV_PACKAGES_INT" on page 138
- "KDLV_PACKAGE_LINES_INT" on page 142
- "KDLV_PACKAGE_NOTES_INT" on page 146

See "Table 5-1. KDLV_PACKAGES_INT interface table" below, "Table 5-2. KDLV_PACKAGE_LINES_INT interface table" on page 75 and "Table 5-3. KDLV_PACKAGE_NOTES_INT interface table" on page 78.

> **Caution:** User data is not validated during import.

Table 5-1. KDLV_PACKAGES_INT interface table

Column	Usage	Data Type	Description
PACKAGE_INTERFACE_ID	Required	NUMBER	Provides a unique identifier for the each record. Derived from the KDLV_INTERFACES_S sequence. For lines tied to a new package, this can be used to tie the line record to the parent record in KDLV_PACKAGES_INT. The PACKAGE_NUMBER and PACKAGE_ID columns can be used for this tie as well. This is required if package lines exist. For new lines, this should be left blank.
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language. If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.

Table 5-1. KDLV_PACKAGES_INT interface table, continued

Column	Usage	Data Type	Description
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
PROCESS_ PHASE	Optional	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_ STATUS	Optional	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_ BY_USERNAME. If both are left blank, the value is set to the user currently running the report.
CREATED_BY_ USERNAME	Required	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank.
CREATION_ DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.
SOURCE_CODE	Required	VARCHAR2	Provides the identify of the source of the record. This value is not validated and is for informational purposes only.
PACKAGE_ID	Required	NUMBER	Provides an identifier for a package and makes the association between the package and package lines. Derived from the KDLV_PACKAGES_S sequence. For lines tied to a new package, this column can be used to tie the line record to the parent record in KDLV_PACKAGES_INT. Either PACKAGE_ INTERFACE_ID and PACKAGE_NUMBER can be used to tie the records.

Table 5-1. KDLV_PACKAGES_INT interface table, continued

Column	Usage	Data Type	Description
			For new lines to be imported into existing packages, this column should refer to the PACKAGE_ID of the existing package.
REQUESTED_BY	Required	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user requesting the package. If left blank, the value is derived from REQUESTED_BY_USERNAME. If both are left blank, the value is set to the user currently running the report.
REQUESTED_BY_USERNAME	Required	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) of the user requesting the package. This is used only if REQUESTED_BY is left blank.
PACKAGE_NUMBER	Required	VARCHAR2	Identifies the package number. This must use either the same value as PACKAGE_ID or a unique string.
ASSIGNED_TO_USER_ID	Optional	NUMBER	Specifies the USER_ID (from KNTA_USERS) that should initially be assigned the request. If left blank, the value is derived from ASSIGNED_TO_USERNAME. If both are left blank, the package will not have an initial value.
ASSIGNED_TO_USERNAME	Required	VARCHAR2	Specifies the USERNAME (from KNTA_USERS) that should initially be assigned the request. This is used only if ASSIGNED_TO_USER_ID is left blank.
ASSIGNED_TO_GROUP_ID	Optional	NUMBER	Specifies the SECURITY_GROUP_ID (from KNTA_SECURITY_GROUPS) that should initially be assigned to the package. If left blank, this value is derived from ASSIGNED_TO_GROUP_NAME. If both are left blank, the package will not have an initial value.
ASSIGNED_TO_GROUP_NAME	Required	VARCHAR2	Specifies the SECURITY_GROUP_ID (from KNTA_SECURITY_GROUPS) that should initially be assigned the package.

Table 5-1. KDLV_PACKAGES_INT interface table, continued

Column	Usage	Data Type	Description
			This is used only if ASSIGNED_TO_GROUP_ID is left blank.
DESCRIPTION	Required	VARCHAR2	Specifies a user-visible description of the package.
PACKAGE_ TYPE_CODE	Required	VARCHAR2	Provides a user-defined categorization of the package. Must be a valid LOOKUP_CODE from KNTA_ LOOKUPS where `LOOKUP_TYPE = 'PACKAGE_ TYPE'`.
PRIORITY_ CODE	Required	VARCHAR2	Indicates the user-defined priority for the package. Must be a valid LOOKUP_CODE from KNTA_ LOOKUPS where `LOOKUP_TYPE = 'PACKAGE_ PRIORITY'`.
STATUS_CODE	Optional	VARCHAR2	Indicates the status of the package.
PROJECT_ CODE	Required	VARCHAR2	Indicates the user-defined project for the package. This should be a valid value from KNTA_LOOKUPS where `LOOKUP_TYPE = 'PROJECT'`.
WORKFLOW_ID	Required	NUMBER	Specifies the workflow that the package should follow. Derived from WORKFLOW_NAME. Either WORKFLOW_ID or WORKFLOW_NAME must be entered.
WORKFLOW_ NAME	Required	VARCHAR2	Specifies the workflow that the package should follow. This is used only if WORKFLOW_ID is left blank.
PRIORITY_SEQ	Optional	NUMBER	Provides a sequence number used to determine the relative priority of packages that are scheduled to process at the same time. If left blank, the value is set to 10.
RELEASE_FLAG	Required	VARCHAR2	Indicates whether or not the interface program will release the package once it imports in into the standard Deployment Management tables. Valid values are: • Yes • No

Table 5-1. KDLV_PACKAGES_INT interface table, continued

Column	Usage	Data Type	Description
			The default is No.
USER_DATA_SET_CONTEXT_ID	Optional	NUMBER	Sets the context identifier for the USER_DATA fields. If left blank, the value is set to 1202.
USER_DATA1 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USER_DATA20	Required	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined. This information is not validated nor does it have a default value.
SOURCE_PACKAGE_ID	Required	NUMBER	Identifies the original package for this distribution package.
DISTPKG_STATUS_MEANING	Required	VARCHAR2	Provides a user-visible status for this distribution package.
RUN_GROUP	Required	NUMBER	Provides a run group number of a specific distribution package.
DISTRIBUTION_ID	Required	NUMBER	Identifies the distribution associated with the package.
ENABLED_FLAG	Required	VARCHAR2	Indicates whether or not the distribution package is enabled upon import. (Applies to distribution packages only.) Valid values are: • Y • N The default values is Y.
DIST_STEP_TRANSACTION_ID	Required	NUMBER	Specifies the path of the distribution workflow step that was run in the transaction with DIST_STEP_TRANSACTION_ID.

Table 5-2. KDLV_PACKAGE_LINES_INT interface table

Column	Usage	Data Type	Description
PACKAGE_ LINE_ INTERFACE_ID	Optional	NUMBER	Provides a unique identifier for the record. If left blank, the value is derived from the KDLV_ INTERFACES_S sequence.
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
PACKAGE_ INTERFACE_ID	Required	NUMBER	Provides a unique identifier for the each record. Derived from the KDLV_INTERFACES_S sequence. For lines tied to a new package, this can be used to tie the line record to the parent record in KDLV_ PACKAGES_INT. The PACKAGE_NUMBER and PACKAGE_ID columns can be used for this tie as well. This is required if package lines exist. For new lines, this should be left blank.
PACKAGE_ID	Required	NUMBER	Provides an identifier for a package and makes the association between the package and package lines. Derived from the KDLV_PACKAGES_S sequence. For new lines to be imported into existing packages, this column should refer to the PACKAGE_ID of the existing package. For lines tied to a new package, this column can be used to tie the line record to the parent record in KDLV_PACKAGES_INT. Either PACKAGE_ INTERFACE_ID and PACKAGE_NUMBER can be used to tie the records.
PACKAGE_ NUMBER	Required	VARCHAR2	Identifies the package number. This must use either the same value as PACKAGE_ ID or a unique string.
PROCESS_ PHASE	Optional	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.

Table 5-2. KDLV_PACKAGE_LINES_INT interface table, continued

Column	Usage	Data Type	Description
PROCESS_ STATUS	Optional	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME. If both are left blank, the value is set to the user currently running the report.
CREATED_BY_ USERNAME	Required	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank.
CREATION_ DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.
SOURCE_ CODE	Required	VARCHAR2	Provides the identify of the source of the record. This value is not validated and is for informational purposes only.
SEQ	Required	NUMBER	Provides a user-visible sequence number for the package line. This must be a unique, positive integer and not conflict with other package lines in the interface table or existing lines if importing lines to an existing packages.
PACKAGE_ LINE_ID	Optional	NUMBER	Provides the identifier for a package line. This is normally left blank and the value is derived from the KDLV_PACKAGE_LINES_S sequence.
OBJECT_ TYPE_ID	Required	NUMBER	Provides the object type ID attached to the package line. Derived from OBJECT_TYPE_ID (in KDLV_ OBJECT_TYPES). Either OBJECT_TYPE_ID or OBJECT_TYPE_ NAME must be entered.
OBJECT_ TYPE_NAME	Required	VARCHAR2	Provides the object type name attached to the package line.

Table 5-2. KDLV_PACKAGE_LINES_INT interface table, continued

Column	Usage	Data Type	Description
			Derived from OBJECT_TYPE_NAME (in KDLV_OBJECT_TYPES). This is used only if OBJECT_TYPE_ID is left blank.
OBJECT_NAME	Required	VARCHAR2	Specifies the name of the object to be processed. This value is not validated.
APP_CODE	Optional	VARCHAR2	Specifies the application category for the package line. Derived from KDLV_ENVIRONMENT_APPS. The APP_CODE must exist for all environments in the workflow attached to the package. APP_CODE can be used as information and can sometimes determine migration behavior.
PARAMETER_SET_CONTEXT_ID	Optional	NUMBER	Sets the context identifier for the detail fields. This is normally left blank and is derived from OBJECT_TYPE_ID.
PARAMETER1 VISIBLE_PARAMETER1 through PARAMETER30 VISIBLE_PARAMETER30	Required	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.
RELEASE_FLAG	Optional	VARCHAR2	Indicates whether or not the interface program will release the package once it imports in into the standard Deployment Management tables. Valid values are: • Yes • No The default is No.
USER_DATA_SET_CONTEXT_ID	Optional	NUMBER	Sets the context identifier for the USER_DATA fields. If left blank, the value is set to 1203.

Table 5-2. KDLV_PACKAGE_LINES_INT interface table, continued

Column	Usage	Data Type	Description
USER_DATA1 VISIBLE_ USER_DATA1 through USER_DATA20 VISIBLE_ USER_DATA20	Required	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined. This information is not validated nor does it have a default value.
OBJECT_ REVISION	Required	VARCHAR2	Specifies the denormalized object_revision of the object entered on this line.
SOURCE_ PACKAGE_ LINE_ID	Required	NUMBER	Identifies the original package line for this distribution package line.
ENABLED_ FLAG	Required	VARCHAR2	Indicates whether or not the distribution package is enabled upon import. (Applies to distribution packages only.) Valid values are: • Y • N The default is Y.

Table 5-3. KDLV_PACKAGE_NOTES_INT interface table

Column	Usage	Data Type	Description
PACKAGE_ NOTE_ INTERFACE_ ID	Optional	NUMBER	Provides a unique identifier for the record. If left blank, the value is derived from the KDLV_ INTERFACES_S sequence.
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
PACKAGE_ INTERFACE_ ID	Required	NUMBER	Provides a unique identifier for the each record. Derived from the KDLV_INTERFACES_S sequence. This is required if package lines exist. For new lines, this

Table 5-3. KDLV_PACKAGE_NOTES_INT interface table, continued

Column	Usage	Data Type	Description
			should be left blank. For lines tied to a new package, this can be used to tie the line record to the parent record in KDLV_PACKAGES_INT. The PACKAGE_NUMBER and PACKAGE_ID columns can be used for this tie as well.
PACKAGE_ID	Required	NUMBER	Provides an identifier for a package and makes the association between the package and note. Derived from the KDLV_PACKAGES_S sequence. Identifies the package ID. This can be used to tie the note record to the parent record in KDLV_PACKAGES_INT. Either PACKAGE_INTERFACE_ID and PACKAGE_NUMBER can be used to tie the records.
PACKAGE_NUMBER	Required	VARCHAR2	Identifies the package number. This must use either the same value as PACKAGE_ID or a unique string. This can be used to tie the note record to the parent record in KDLV_PACKAGES_INT. The PACKAGE_INTERFACE_ID and PACKAGE_ID can be used for this tie as well.
PROCESS_PHASE	Optional	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	Optional	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME. If both are left blank, the value is set to the user currently running the report.
CREATED_BY_USERNAME	Required	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank.

Table 5-3. KDLV_PACKAGE_NOTES_INT interface table, continued

Column	Usage	Data Type	Description
CREATION_ DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.
SOURCE_ CODE	Required	VARCHAR2	Provides the identify of the source of the record. This value is not validated and is for informational purposes only.
NOTE	Required	CLOB	Specifies the full text of the note.
REPLACE_ NOTE_FLAG	Obsolete	VARCHAR2	No longer used.

Step 2: Start the Import

To import data from the interface tables, the Run PPM Package Interface report is used.

The Run PPM Package Interface report:

- Queries the interface tables for active records matching the given selection criteria.
- Defaults any information that has defaulting rules in Deployment Management but has not been specified in the interface table records.
- Validates package header data and package line object type information for referential and data integrity. This validation is based on the logic used when entering or updating data through the standard interface. Information in User Data fields and in-line parameters is not validated.
- Imports packages and package lines passing validation into the standard package tables. Partial imports are not allowed. packages with one or more failed lines will not be imported.
- Can be used to submit new packages.
- Reports on the results of the execution, listing both the packages and package lines that passed validation and were imported, and those that failed validation and the specific validation errors they encountered.

To run the Run PPM Package Interface report:

1. Log on to PPM Center.
2. From the menu bar, select **Open > Reports > Create Report**.

The Submit New Report page appears.

3. Select **Administrative** from the Report Category.

   The page is updated showing the list of administrative reports.

4. Select **Run PPM Package Interface.**

   The Submit Report: Run PPM Package Interface window appears.

5. Complete the fields, as described in the following table.

   Required fields are denoted with a red asterisk next to the field on the screen. Depending on your selections, the required fields may vary.

Field Name	Definition
Group Id	Specifies the group ID for which the interface program should be run. The interface program will only look for records with this value in the GROUP_ID column. This is useful when importing a batch of packages.
Package No.	Specifies the package number for which the interface program should be run. The interface program will only look for records with this value in the PACKAGE_NUMBER column. This is useful when importing a specific package.
Package Id	Specifies the package ID for which the interface program should be run. The interface program will only look for records with this value in the PACKAGE_ID calumniation is useful when importing a specific package.
Source Code	Indicates whether or not to set the SOURCE_TYPE_CODE column of the final requests created with a free-form text code. This is used as an indicator of how the request was created for auditing or testing purposes.
Run Import	◦ **If set to Yes.** Indicates that the program will process the records in the interface table and try to import them. ◦ **If set to No.** Indicates that the program will simply report on the records in the interface table. This option is useful when auditing prior executions of the interface.
Show Successful Transactions	Indicates whether or not to show packages and package lines that were successfully imported.
Show Failed Transactions	Indicates whether or not to show packages and package lines that were not successfully imported.

6. (Optional) To test the process, set the **Run Import** field to **No**.

Step 3: Verify Successful Completion

If any customizations to the import process have been made, it is extremely important to confirm that the import was successful.

To confirm that the import process completed successfully:

1. Review the results of the Import Summary.

   This report identifies any errors with the import.

2. If errors are present, start your troubleshooting by referring to "Correcting Failures" below.

All interface tables are automatically cleared by the purge service. The purging process depends on the following parameters in the server.conf file:

- **ENABLE_INTERFACE_CLEANUP.** Enables or disables the purge process.
- **DAYS_TO_KEEP_INTERFACE_ROWS.** Determines the number of days that records are retained in the interface tables.

For more information on the `server.conf` file, see the *Installation and Administration Guide.*

Correcting Failures

When a package is successfully imported, information stored in the interface tables is not deleted, and no additional action is required. The package can be viewed and processed using Deployment Management.

For packages and package lines that fail to import, corrective actions are required. The first step is examining the audit report from the open interface program to identify the failed records and the specific reasons for each failure.

Depending on the reasons, it may be necessary to correct the problem through a variety of means. Some failure might occur due to a mapping problem between the source data and existing data.

> **Note:** The source data might use a project name that does not exist in Deployment Management. Corrective measures for this specific problem would include adding the specific project in Deployment Management, or mapping the source project to a project name that already exists in Deployment Management.

Other failures might be due to missing required information that cannot be defaulted.

> **Note:** Package lines require an object type. If the object type columns were left blank for records in the package lines interface table, the records will fail validation. To correct this, the custom program or procedure that inserts records into the interface table needs to be modified to include this required data.

Failures could occur due to other configuration and mapping problems in either the source or in Deployment Management, or could be the result of errors in the custom loading program.

> **Note:** During initial implementation of the open interface, the mapping between the third-party source and Deployment Management should be thoroughly reviewed and the load programs thoroughly tested.
>
> Additionally, it is good practice to monitor executions of the open interface and periodically monitor that the desired data is being imported into Deployment Management.

Chapter 6: Workflow Transaction Open Interface

Workflow Transaction Open Interface Overview

PPM Center includes an open interface for importing workflow transactions. Workflow transactions are all of the actions that can be performed at a workflow step for a package line or request, such as a file migration or a design approval. The open interface supports the following workflow transactions:

- **Submit.** A user can submit a package (and all of its lines) or a request.
- **Decision.** A user can make a choice at a decision workflow step. For example, a user could decide to approve a workflow step (that has choices Approved and Not Approved).
- **Delegation.** A user can delegate the choice at a Decision step to another user.
- **Execution.** A user can perform an execution at a workflow step. This execution could be object type or request type command execution, a SQL statement, a PL/SQL function, a token evaluation, or a workflow step command.
- **Schedule execution.** A user can schedule an execution to be performed at a later date or time.
- **Bypass execution.** A user can bypass an execution and manually provide the result instead. For example, if a file did not need to be migrated to an environment, a user could bypass the migration and supply the result Succeeded instead.
- **Override result.** A user can override the result at any non-eligible step that is still active. For example, if a migration failed and there is no transition defined from the step on the Failure result, a user could override the Failure with another result.
- **Cancel.** A user can cancel a package line or a request.
- **Force transition.** A user can force a transition from one workflow step to another, even if there is no standard transition between the two steps defined in the workflow.

> **Caution:** The force transition feature is not supported through the standard interface.

The Workflow Transaction Open Interface is a set of tables within the PPM Center database. Data added to these tables is validated and workflow steps within package lines and requests are acted upon based upon the information.

The primary purpose of the Workflow Transaction Open Interface is to allow integration with products other than PPM Center. You can use relevant information from these products to perform workflow transactions for package lines and requests. You can also use the open interface as a mechanism to convert data from a legacy system into PPM Center during initial implementation.

> **Note:** Currently, the Workflow Transaction Open Interface does not support the creation of packages from requests (create_package and create_package_and_wait). Also, it does not support the Ready for Release command or the creation of requests from requests.
>
> The Workflow Transaction Open Interface does support the use of subworkflows. When dealing with subworkflows, keep the following in mind:
>
> - The workflow step sequence should be the same as the step sequence visible on the screen. For example, 2.4.5.
> - Force transition can only be performed to the same level.
> - When needed, it is necessary to pass in the workflow step sequence and not the workflow step ID or the workflow step name. The same applies to TO_WORKFLOW_STEP_SEQUENCE.

The Workflow Transaction Open Interface Data Model

The following interface table is used by the Workflow Transaction Open Interface:

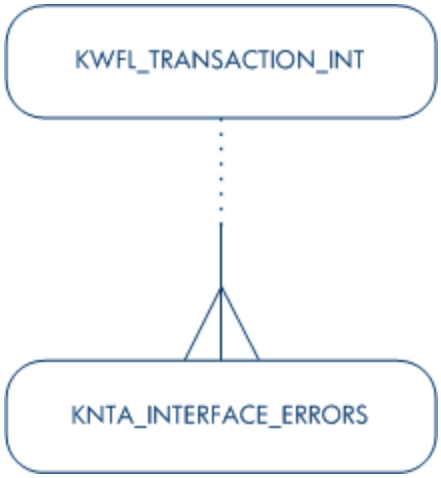
- "KWFL_TRANSACTIONS_INT" on page 174

This interface table is described in "Open Interface Data Models" on page 104. The columns to use when importing workflows are detailed in the appropriate step within this section.

Parameters are required or optional, depending upon the type of transaction you are importing. All of these parameters are described in "Open Interface Data Models" on page 104. The parameters used for each type of transaction are detailed where appropriate.

The following figure illustrates the relationship between the KWFL_TRANSACTIONS_INT and KNTA_INTERFACE_ERRORS table:

Figure 6-1. Workflow transaction interface and supporting tables



Performing an Import

Step 1: Load the Interface Tables

Using SQL*Loader, your favorite tool, or direct Oracle database-to-database communication, load your data into the required input columns in the following table:

- "KWFL_TRANSACTIONS_INT" on page 174

See "Table 6-1. KWFL_TRANSACTIONS_INT interface table" below

Table 6-1. KWFL_TRANSACTIONS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	Required	NUMBER	Uniquely identifies each transaction.
CREATION_DATE	Optional	DATE	The date/time for the step transaction cannot be set via this interface table. Instead, the value is taken from the LAST_UPDATE_DATE.
CREATED_USERNAME	Required	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. Supply this or CREATED_BY.
CREATED_BY	Required	NUMBER	Identifies the USER_ID (from KNTA_USERS)

Table 6-1. KWFL_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
			for the user performing the transaction. Supply this or CREATED_USERNAME.
LAST_UPDATE_DATE	Optional	DATE	This value is taken from the report execution date/time, which is transferred to the date/time for the step transaction. If left blank, the current date is used.
LAST_UPDATED_USERNAME	Required	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. Supply this or LAST_UPDATED_BY.
LAST_UPDATED_BY	Optional	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. Supply this or LAST_UPDATED_USERNAME. If both are left blank, the value is derived from CREATED_USERNAME.
EVENT	Required	VARCHAR2	Specifies the type of workflow transaction.
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language. If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.
GROUP_ID	Required	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
WORKFLOW_ENGINE_BATCH_ID	Optional	NUMBER	Specifies the batch in which the workflow engine carried out this transaction.
PROCESS_PHASE	Optional	NUMBER	Indicates the current stage of the record as it is being processed.

Table 6-1. KWFL_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
			See "Process State Information" on page 182 for details.
PROCESS_ STATUS	Optional	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
SOURCE_ TYPE_CODE	Required	VARCHAR2	Specifies the type of external update. This should be a left blank or have a value of INTERFACE_WF.
SOURCE	Required	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
INSTANCE_ SOURCE_ TYPE_CODE	Required	VARCHAR2	Indicates whether or not the transaction is for a package line (CR) or a request (IR).
INSTANCE_ SOURCE_SET_ NUMBER	Required	VARCHAR2	Specifies the package number (PACKAGE_ NUMBER from KDLV_PACKAGES) or request number (REQUEST_NUMBER from KCRT_ REQUESTS). Supply this or INSTANCE_SOURCE_SET_ID.
INSTANCE_ SOURCE_SET_ ID	Required	NUMBER	Specifies the package ID (PACKAGE_ID from KDLV_PACKAGES) or request ID (REQUEST_ ID from KCRT_REQUESTS). Supply this or INSTANCE_SOURCE_SET_ NUMBER.
INSTANCE_ SOURCE_LINE_ SEQ	Required	NUMBER	Specifies the package line sequence number (SEQ from KDLV_PACKAGE_LINES). Supply this or INSTANCE_SOURCE_ID.
INSTANCE_ SOURCE_ID	Required	NUMBER	Specifies the package line ID (PACKAGE_LINE_ ID from KDLV_PACKAGE_LINES) or request ID (REQUEST_ID from KCRT_REQUESTS). Supply this or INSTANCE_SOURE_LINE_SEQ (for package lines) or INSTANCE_SOURCE_ SET_NUMBER (for requests).

Table 6-1. KWFL_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
WORKFLOW_ STEP_NAME	Required	VARCHAR2	Specifies the name of the workflow step (STEP_ NAME from KWFL_WORKFLOW_STEPS). Supply this or WORKFLOW_STEP_ID.
WORKFLOW_ STEP_SEQ	Required	VARCHAR2	Specifies the sequence number of the workflow step. Supply this or WORKFLOW_STEP_ID. For subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth.
RESULT_VALUE	Required	VARCHAR2	Indicates the result of the step. This is normally not displayed to the user; therefore it may be an ID or internal code.
VISIBLE_ RESULT_VALUE	Required	VARCHAR2	Indicates the result of the step. This is the result value that a user normally sees.
USER_ COMMENTS	Required	VARCHAR2	Specifies comments for the transaction. Any comments are appended to the notes for the package or request.
DELEGATED_ TO_USERNAME	Required	VARCHAR2	Specifies the USERNAME (from KNTA_ USERS) for the user that the decision is being delegated to. Supply this or DELEGATED_TO_USER_ID.
DELEGATED_ TO_USER_ID	Required	NUMBER	Specifies the USER_ID (from KNTA_USERS) for the user that the decision is being delegated to. Supply this or DELEGATED_TO_USERNAME.
SCHEDULE_ DATE	Required	DATE	Indicates the date that the execution step is scheduled to run.
WORKFLOW_ID	Optional	NUMBER	Specifies the workflow that the package should follow.
WORKFLOW_ INSTANCE_ID	Optional	NUMBER	Specifies the instance ID.
WORKFLOW_ STEP_ID	Required	NUMBER	Specifies the workflow step ID (WORKFLOW_ STEP_ID from KWFL_WORKFLOW_STEPS).

Table 6-1. KWFL_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
			Supply this, WORKFLOW_STEP_NAME, or WORKFLOW_STEP_SEQ.
WORKFLOW_ INSTANCE_ STEP_ID	Optional	NUMBER	Specifies the instance step ID.
CURRENT_ STEP_ TRANSACTION_ ID	Optional	NUMBER	Specifies the current step transaction ID.
APPROVALS_ REQUIRED_ CODE	Optional	NUMBER	Specifies the code for the required approvals.
EVENT_ GROUP_ID	Optional	NUMBER	Specifies the group ID for the event.
CMD_ EXECUTION_ SCHD_TASK_ID	Optional	NUMBER	Specifies the execution step and the command that has been scheduled. This will specify the identified for the scheduled task.
TO_ WORKFLOW_ STEP_SEQ	Required	VARCHAR2	Specifies the sequence number of the workflow step for the step that the package line or request should transition to. Supply this, TO_WORKFLOW_STEP_ID, or TO_WORKFLOW_STEP_NAME.
TO_ WORKFLOW_ STEP_NAME	Required	VARCHAR2	Specifies the name of the workflow step (STEP_ NAME from KWFL_WORKFLOW_STEPS) for the step that the package line or request should transition to. Supply this, TO_WORKFLOW_STEP_SEQ, or TO_WORKFLOW_STEP_ID.
TO_ WORKFLOW_ STEP_ID	Required	NUMBER	Specifies the workflow step ID (WORKFLOW_ STEP_ID from KWFL_WORKFLOW_STEPS) for the step that the package line or request should transition to. Supply this, TO_WORKFLOW_STEP_NAME, or TO_WORKFLOW_STEP_SEQ.

Step 2: Load the Parameters

The Workflow Transaction Open Interface can be used for different types of transactions. Parameters are required or optional depending upon the type of transaction you are importing.

To populate the Workflow Transaction Open Interface tables:

1. Define a PL/SQL procedure, function, or anonymous block to call the following procedure:
   - "KWFL_TXN_INT.INSERT_ROW" on page 178
2. See the following sections for the list of parameters used for each type of transaction (or event):
   - "Parameters Used For All Events" on the next page
   - "Parameters for Package or Request Status" on page 94
   - "Parameters for Decision Step Results" on page 94
   - "Parameters for Decision Step Delegation" on page 95
   - "Parameters for Execution Steps" on page 96
   - "Parameters for Execution Step Schedule" on page 96
   - "Parameters for Execution Step Bypass" on page 97
   - "Parameters for Changing Step Result" on page 98
   - "Parameters for Forced Workflow Step Transition" on page 99
   - "Parameters for Package Line or Request Cancellation" on page 100

The following is an example of an anonymous PL/SQL block used to insert rows into the interface table for transactions for decisions steps for requests. Note that some optional parameters are not used.

```
set serveroutput on;
set verify off;

define p_created_username = '&1';
define p_request_number = '&2';
define p_workflow_step_seq = '&3';
define p_visible_result_value = '&4';

declare
      x_message_type number;
```

```
      x_message_name VARCHAR2;
      x_message VARCHAR2(1000);

begin
   kwfl_txn_int.insert_row
      (p_event => 'APPROVAL_VOTE',
      p_group_id => left blank,
      p_created_username => '&p_created_username',
      p_source => left blank,
      p_request_number => '&p_request_number',
      p_package_number => left blank,
      p_package_line_seq => left blank,
      Workflow Transaction Open Interface 125
      p_workflow_step_name => left blank,
      p_workflow_step_seq => '&p_workflow_step_seq',
      p_visible_result_value => '&p_visible_result_value',
      p_user_comments => left blank,
      p_delegated_to_username => left blank,
      p_schedule_date => left blank,
      p_to_workflow_step_name => left blank,
      p_to_workflow_step_seq => left blank,
      o_message_type => x_message_type,
      o_message_name => x_message_name,
      o_message => x_message);
   if (x_message_type != KNTA_Constant.SUCCESS) then
      dbms_output.put_line(x_message_name);
      dbms_output.put_line(x_message);
      end if;
end;
/
```

If the previous code is located in a file called `run_interface.sql`, the following command would be used (from the command line) to run the code.

sqlplus <*username*>/<*password*> @run_interface.sql 'jsmith' '12345' '1' 'Approved'

where <*username*> and <*password*> represent the user ID and password for the appropriate PPM Center database

Parameters Used For All Events

The following parameters should be used (or considered) for all transactions.

Table 6-2. Parameters used for all events

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. The value depends on the type of transaction.
P_GROUP_ID	Optional	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. If left blank, the value is generated by the system.
P_CREATED_USERNAME	Required	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction.
P_SOURCE	Optional	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
P_REQUEST_NUMBER	Requirement varies	VARCHAR2	Identifies the request. Supply this or P_PACKAGE_NUMBER.
P_PACKAGE_NUMBER	Requirement varies	VARCHAR2	Identifies the package number. Supply this or P_REQUEST_NUMBER.
P_USER_COMMENTS	Optional	VARCHAR2	Specifies comments for the transaction. Any comments are appended to the notes for the package or request.
O_MESSAGE_TYPE	Leave blank	NUMBER	Indicates what type of error occurred. Valid values (from KNTA_Constant) are: • SUCCESS - No error occurred • USER_ERR - User error • INTERNAL_ERR - An internal error occurred • WARNING - A non-fatal warning is returned
O_MESSAGE_NAME	Leave blank	VARCHAR2	Specifies the internal message name of the error that was returned. This is used mainly for debugging purposes.

Table 6-2. Parameters used for all events, continued

Parameter	Usage	Data Type	Description
O_MESSAGE	Leave blank	VARCHAR2 (1000)	Provides the error message.

Parameters for Package or Request Status

"Table 6-3. Parameters for package or request status" below lists the parameters to use for the status for packages or requests.

Table 6-3. Parameters for package or request status

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. Set to INSTANCE_SET_CREATE.

Parameters for Decision Step Results

"Table 6-4. Parameters for decision step results" below lists the parameters to use for the status for decision steps.

Table 6-4. Parameters for decision step results

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. Set to APPROVAL_VOTE.
P_PACKAGE_LINE_SEQ	Requirement varies	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_LINES_S sequence. Use this parameter if the transaction is for a package line.
P_WORKFLOW_STEP_NAME	Requirement varies	VARCHAR2	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS). Supply this or P_WORKFLOW_STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2	Specifies the sequence number of the workflow step. For subworkflows, the sequence numbers of the

Table 6-4. Parameters for decision step results, continued

Parameter	Usage	Data Type	Description
			workflow steps could be in the form of 2.4.5 and so forth. Supply this or P_WORKFLOW_STEP_NAME.
P_VISIBLE_ RESULT_ VALUE	Required	VARCHAR2	Indicates the result of the step. This is the result value that a user normally sees.

Parameters for Decision Step Delegation

The following parameters should be used for the delegation of decision steps.

Table 6-5. Parameters for decision step delegation

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. Set to APPROVAL_DELEGATE.
P_PACKAGE_ LINE_SEQ	Requirement varies	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_LINES_S sequence. Use if the transaction is for a package line.
P_WORKFLOW_ STEP_NAME	Requirement varies	VARCHAR2	Specifies the name of the workflow step (STEP_ NAME from KWFL_WORKFLOW_STEPS). Supply this or P_WORKFLOW_STEP_SEQ.
P_WORKFLOW_ STEP_SEQ	Requirement varies	VARCHAR2	Specifies the sequence number of the workflow step. For subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply this or P_WORKFLOW_STEP_NAME.
P_DELEGATED_ TO_USERNAME	Required	VARCHAR2	Specifies the USERNAME (from KNTA_ USERS) for the user that the decision is being delegated to.

Parameters for Execution Steps

The following parameters should be used for execution steps.

Table 6-6. Parameters for execution step

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. Set to EXECUTION_EXECUTE.
P_PACKAGE_LINE_SEQ	Requirement varies	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_LINES_S sequence. Use if the transaction is for a package line.
P_WORKFLOW_STEP_NAME	Requirement varies	VARCHAR2	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS). Supply this or P_WORKFLOW_STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2	Specifies the sequence number of the workflow step. For subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply this or P_WORKFLOW_STEP_NAME.

Parameters for Execution Step Schedule

The following parameters should be used for execution step schedules.

Table 6-7. Parameters for execution step schedule

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. Set to EXECUTION_SCHEDULE.
P_PACKAGE_LINE_SEQ	Requirement varies	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_LINES_S sequence.

Table 6-7. Parameters for execution step schedule, continued

Parameter	Usage	Data Type	Description
			Use if the transaction is for a package line.
P_WORKFLOW_STEP_NAME	Requirement varies	VARCHAR2	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS). Supply this or P_WORKFLOW_STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2	Specifies the sequence number of the workflow step. For subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply this or P_WORKFLOW_STEP_NAME.
P_SCHEDULE_DATE	Required	DATE	Indicates the date that the execution step is scheduled to run.

Parameters for Execution Step Bypass

The following parameters should be used for execution step bypass.

Table 6-8. Parameters for execution step bypass

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. Set to BYPASS_EXECUTION.
P_PACKAGE_LINE_SEQ	Requirement varies	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_LINES_S sequence. Use if the transaction is for a package line.
P_WORKFLOW_STEP_NAME	Requirement varies	VARCHAR2	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS). Supply this or P_WORKFLOW_STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2	Specifies the sequence number of the workflow step. For subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so

Table 6-8. Parameters for execution step bypass, continued

Parameter	Usage	Data Type	Description
			forth. Supply this or P_WORKFLOW_STEP_NAME.
P_VISIBLE_ RESULT_ VALUE	Required	VARCHAR2	Indicates the result of the step. This is the result value that a user normally sees.

Parameters for Changing Step Result

The following parameters should be used for changing a step result.

Table 6-9. Parameters for changing step result

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. Set to RESULT_OVERRIDE.
P_PACKAGE_ LINE_SEQ	Requirement varies	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_LINES_S sequence. Use if the transaction is for a package line.
P_ WORKFLOW_ STEP_NAME	Requirement varies	VARCHAR2	Specifies the name of the workflow step (STEP_ NAME from KWFL_WORKFLOW_STEPS). Supply this or P_WORKFLOW_STEP_SEQ.
P_ WORKFLOW_ STEP_SEQ	Requirement varies	VARCHAR2	Specifies the sequence number of the workflow step. For subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply this or P_WORKFLOW_STEP_NAME.
P_VISIBLE_ RESULT_ VALUE	Required	VARCHAR2	Indicates the result of the step. This is the result value that a user normally sees.

Parameters for Forced Workflow Step Transition

The following parameters should be used for a workflow step transition.

Table 6-10. Parameters for forced workflow step transition

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. Set to FORCE_TRANSITION. Note that this does not work between a subworkflow and its parent workflow.
P_ PACKAGE_ LINE_SEQ	Requirement varies	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_LINES_S sequence. Use if the transaction is for a package line.
P_ WORKFLOW_ STEP_NAME	Requirement varies	VARCHAR2	Specifies the name of the workflow step (STEP_ NAME from KWFL_WORKFLOW_STEPS). Supply this or P_WORKFLOW_STEP_SEQ.
P_ WORKFLOW_ STEP_SEQ	Requirement varies	VARCHAR2	Specifies the sequence number of the workflow step. For subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply this or P_WORKFLOW_STEP_NAME.
P_VISIBLE_ RESULT_ VALUE	Required	VARCHAR2	Indicates the result of the step. This is the result value that a user normally sees.
P_TO_ WORKFLOW_ STEP_NAME	Requirement varies	VARCHAR2	Specifies the name of the workflow step (STEP_ NAME from KWFL_WORKFLOW_STEPS) for the step that the package line or request should transition to. Supply this or P_TO_WORKFLOW_STEP_SEQ.
P_TO_ WORKFLOW_ STEP_SEQ	Requirement varies	VARCHAR2	Specifies the sequence number of the workflow step for the step that the package line or request should transition to. Supply this or P_TO_WORKFLOW_STEP_NAME.

Parameters for Package Line or Request Cancellation

The following parameters should be used for canceling a package line or request.

Table 6-11. Parameters for package line or request cancellation

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2	Specifies the type of workflow transaction. Set to INSTANCE_SET_CANCEL.
P_PACKAGE_ LINE_SEQ	Requirement varies	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_ LINES_S sequence. Use if the transaction is for a package line.

Step 3: Start the Import

To import data from the interface tables, the Run Workflow Transaction Interface report is used.

The Run Workflow Transaction Interface report:

- Queries the interface table for active records matching the given selection criteria.
- Derives all missing information in the interface table. For example, CREATED_BY is derived from CREATED_BY_USERNAME.
- Validates all data in the interface table, according to the same rules used when entering or updating data through the standard interface. For example, the CREATED_BY_USERNAME must exist in PPM Center and must be enabled.
- Performs the workflow transactions for all records that pass validation. This generates or updates records in the standard workflow tables, and this may update information in the standard package or request tables.
- Schedules executions. For any object type or request type commands that need to be run, scheduled tasks are generated.

> **Note:** For these types of executions, the interface table will not correctly reflect the final results. However, workflow step commands can be scheduled.

The report shows all transactions that were processed by the Workflow Transaction Open Interface report. If desired, successful transactions can be eliminated from the report, so that only errors are displayed.

To run the Run PPM Package Interface report:

1. Log on to PPM Center.
2. From the menu bar, select **Open > Reports > Create Report.**

   The Submit New Report page appears.
3. Select **Administrative** from the Report Category.

   The page is updated showing the list of administrative reports.
4. Select **Run Workflow Transaction Interface.**

   The Submit Report: Run Workflow Transaction Interface window appears.
5. Complete the fields, as described in the following table.

   Required fields are denoted with a red asterisk next to the field on the screen. Depending on your selections, the required fields may vary.

Field Name	Definition
Group ID	Specifies the group ID for which the interface program should be run. The interface program will only look for records with this value in the GROUP_ ID column. This is useful when importing a batch of packages.
Source Code	Indicates whether or not to set the SOURCE_TYPE_CODE column of the final requests created with a free-form text code. This is used as an indicator of how the request was created for auditing or testing purposes.
Run Import	◦ **If set to Yes.** Indicates that the program will process the records in the interface table and try to import them. ◦ **If set to No.** Indicates that the program will simply report on the records in the interface table. This option is useful when auditing prior executions of the interface.
Resubmit	**If set to Yes.** Specifies that the program will reset the appropriate values for the records in the interface table, remove any previous errors, and rerun the interface for the records. To resubmit failed transactions, it is necessary to provide a Group ID and optionally a Source Code.
Show Successful Transactions	Indicates whether or not to show packages and package lines that were successfully imported.

6. (Optional) To test the process, set the **Run Import** field to **No**.
7. Click **Submit**.

Step 4: Verify Successful Completion

If any customizations to the import process have been made, it is extremely important to confirm that the import was successful.

To confirm that the import process completed successfully:

1. Click **View Report** to review the results of the import.

   This report identifies any errors with the import.
2. If errors are present, start your troubleshooting by referring to "Correcting Failures" below.

All interface tables are automatically cleared by the purge service. The purging process depends on the following parameters in the server.conf file:

- **ENABLE_INTERFACE_CLEANUP.** Enables or disables the purge process.
- **DAYS_TO_KEEP_INTERFACE_ROWS.** Determines the number of days that records are retained in the interface tables.

For more information on the `server.conf` file, see the *Installation and Administration Guide.*

Correcting Failures

When a workflow transaction is successfully processed, information stored in the interface table is not deleted, and no additional action is required. Users can view the results of the transaction through the workflow transaction interface report. Successful transactions are deleted from the interface table daily.

For users that fail to import, you should:

1. Examine the audit report from the open interface report.
2. Identify the failed records and the specific reasons for each failure.
3. (Optionally) Depending on the reasons, correct the problem as necessary.

Failures may be due to:

- A mapping problem between the source data and existing data in PPM Center.
- Missing information that cannot be defaulted.

For example, if a workflow step is not provided for an execution, the records will fail validation. To correct this, the custom program or procedure that inserts records into the interface table needs to be modified to include this required data.

- Other configuration and mapping problems in either the source or in PPM Center or could be the result of errors in the custom loading program.

> **Note:** During the initial implementation of the open interface, the mapping between the third-party source and PPM Center should be thoroughly reviewed and the load programs thoroughly tested in a testing instance.
>
> It is good practice to monitor executions of the open interface and periodically monitor the import of desired data into PPM Center.

Appendix A: Open Interface Data Models

This appendix describes all the data models used in the open interface. Each section details a single interface table and provides the following information:

- **Column.** Provides the name of the column.
- **Usage.** Indicates whether or not the data is used exclusively for input (I) or exclusively for output (O). Columns specified as both (I/O) allow for input or provide (output) data if a value is not provided.
- **Data Type.** Specifies the data type of the column.
- **Description.** Provides a description of the data and related information, dependencies, and any pertinent tips.

> **Caution:** You should only provide data for columns specified with input (I) or input and output (I/O) usage. Unexpected results may occur if you provide data for columns that are to be used exclusively for output (O).

This appendix describes the `INSERT_ROW` parameters used with the `KWFL_TXN_INT` package, that are used only with the Workflow Transaction Open Interface.

KCRT_CONTACTS_INT

This interface table is reserved for future use. Data in this table is ignored.

KCRT_FG_CMDB_APPLICATION_INT

This table has been deprecated. It exists in the database to support legacy customizations.

KCRT_FG_DEMAND_SCHEDULE_INT

The KCRT_FG_DEMAND_SCHEDULE_INT interface table stores validation information, for each request, that is related to the Demand Management Scheduling Fields field group.

Table A-1. KCRT_FG_DEMAND_SCHEDULE_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_TRANSACTION_ID	I	NUMBER	Provides the transaction ID (from KCRT_REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_TYPE_ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
SCHEDULE_DATE	I	DATE	Indicates the date that the demand was scheduled.
REJECT_DATE	I	DATE	Indicates the date that the demand was rejected.
EFFORT	I	NUMBER	Specifies the effort associated with the satisfied demand (in hours).
DEMAND_SATISFIED_DATE	I	DATE	Indicates the date that the demand was satisfied.

KCRT_FG_IMPACT_ANALYSIS_INT

Table for importing MAM-integrated requests.

Table A-2. KCRT_FG_IMPACT_ANALYSIS_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the

Table A-2. KCRT_FG_IMPACT_ANALYSIS_INT interface table, continued

Column	Usage	Data Type	Description
			same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction. If left blank, the value is generated by the system.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_ TYPE_ID	I	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
MAM_RFC_ID	O	VARCHAR2	Identifies the MAM RFC.
IMPACT_ RESULT_VALUE	O	NUMBER	Indicates the result of the MAM Impact analysis.
IMPACT_ RESULT_ MEANING	O	VARCHAR2	Indicates the meaning of the MAM Impact analysis.

KCRT_FG_MASTER_PROJ_REF_INT

The KCRT_FG_MASTER_PROJ_REF_INT interface table stores validation information, for each request, that is related to the Master Project Reference on Request field group.

Table A-3. KCRT_FG_MASTER_PROJ_REF_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time.

Table A-3. KCRT_FG_MASTER_PROJ_REF_INT interface table, continued

Column	Usage	Data Type	Description
			Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_TRANSACTION_ID	I	NUMBER	Provides the transaction ID (from KCRT_REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_TYPE_ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
REF_MASTER_PROJECT_ID	I	VARCHAR2	Creates a reference to the specified master project ID.
REF_MASTER_PROJECT_NAME	I	VARCHAR2	Creates a reference to the specified master project name.

KCRT_FG_PFM_ASSET_INT

The KCRT_PFM_ASSET_INT interface table is used to store asset information.

Table A-4. KCRT_FG_PFM_ASSET_INT interface table

Column	Usage	Data Type	Description
PFM_ASSET_INTERFACE_ID	I	NUMBER	System-generated identifier.
GROUP_ID	I/O	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report.

Table A-4. KCRT_FG_PFM_ASSET_INT interface table, continued

Column	Usage	Data Type	Description
			Derived from the KNTA_INTERFACE_ GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ID	I/O	NUMBER	Uniquely identifies each transaction. If left blank, the value is generated by the system.
PARENT_ TRANSACTION_ID	I/O	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
PROCESS_PHASE	I	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182.
PROCESS_STATUS	I	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182.
REQUEST_ID	I	NUMBER	Indicates the request ID of the PFM asset.
REQUEST_TYPE_ID	I	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
ASSET_NAME	I	VARCHAR2	Indicates the name of the asset.
ASSET_HEALTH_ CODE	I	VARCHAR2	Indicates the asset health information code generated by the system.
ASSET_HEALTH_ MEANING	I	VARCHAR2	Indicates the asset health information meaning generated by the system
ASSET_BUSINESS_ UNIT_CODE	I	VARCHAR2	Indicates the business unit code for asset.
ASSET_BUSINESS_ UNIT_MEANING	I	VARCHAR2	Indicates the business unit Meaning for asset.
ASSET_BUSINESS_ OBJECTIVE_ID	I	NUMBER	Indicates the business objective ID for asset.
ASSET_BUSINESS_ OBJECTIVE_NAME	I	VARCHAR2	Indicates the business objective name for asset.
ASSET_PROJECT_ CLASS_CODE	I	VARCHAR2	Indicates the project class code for asset.

Table A-4. KCRT_FG_PFM_ASSET_INT interface table, continued

Column	Usage	Data Type	Description
ASSET_PROJECT_CLASS_MEANING	I	VARCHAR2	Indicates the project class meaning for asset.
ASSET_ASSET_CLASS_CODE	I	VARCHAR2	Indicates the asset class code for asset.
ASSET_ASSET_CLASS_MEANING	I	VARCHAR2	Indicates the asset class meaning for asset.
ASSET_PROJECT_PLAN_ID	I	NUMBER	Indicates the unique identifier of the project plan.
ASSET_PROJECT_PLAN_NAME	I	VARCHAR2	Indicates the name of the project plan.
ASSET_PROJECT_PLAN_URL	I	VARCHAR2	Indicates the Web address or URL associated with the project plan.
ASSET_STAFF_PROF_ID	I	NUMBER	Indicates Uniquely identifies the staff profile associated with the asset.
ASSET_STAFF_PROF_NAME	I	VARCHAR2	Indicates the name of the staff profile associated with the asset.
ASSET_RETURN_ON_INVESTMENT	I	NUMBER	Indicates the calculated value of the return on investment (ROI) of the asset.
ASSET_NET_PRESENT_VALUE	I	NUMBER	Indicates the NPV for asset.
ASSET_CUSTOM_FIELD_VALUE	I	NUMBER	Indicates the custom financial metric value for asset.
ASSET_VALUE_RATING	I	NUMBER	Indicates the value for score of score domain "Value".
ASSET_RISK_RATING	I	NUMBER	Indicates the value for score of score domain "Risk".
ASSET_TOTAL_SCORE	I	NUMBER	Indicates the total score for asset.
ASSET_DISCOUNT_RATE	I	NUMBER	Indicates the discount rate for asset.

KCRT_FG_PFM_PROJECT_INT

Table for importing PFM projects into KCRT_FG_PFM_PROJECT.

Table A-5. KCRT_FG_PFM_PROJECT_INT interface table

Column	Usage	Data Type	Description
PFM_PROJECT_INTERFACE_ID	I	NUMBER	System-generated identifier.
GROUP_ID	I/O	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ID	I/O	NUMBER	Uniquely identifies each transaction. If left blank, the value is generated by the system.
PARENT_TRANSACTION_ID	I/O	NUMBER	Provides the transaction ID (from KCRT_REQUESTS_INT) of the parent table being imported.
PROCESS_PHASE	I	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	I	NUMBER	Indicates the current disposition of the record. See "LDAP Authentication" on page 181 for details.
REQUEST_ID	I	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_TYPE_ID	I	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
PROJECT_NAME	I	VARCHAR2	Indicates the name of the project.
PROJECT_HEALTH_CODE	I	VARCHAR2	Identifies the project health information code. Derived from PRJ_CALCULATED_HEALTH_CODE and PRJ_OVERRIDDEN_HEALTH_CODE.
PROJECT_HEALTH_	I	VARCHAR2	Identifies the project health information meaning. Derived from PRJ_CALCULATED_HEALTH_

Table A-5. KCRT_FG_PFM_PROJECT_INT interface table, continued

Column	Usage	Data Type	Description
MEANING			MEANING and PRJ_OVERRIDDEN_HEALTH_ CODE.
PRJ_ BUSINESS_ UNIT_CODE	I	VARCHAR2	Indicates the business unit code for project.
PRJ_ BUSINESS_ UNIT_MEANING	I	VARCHAR2	Indicates the business unit meaning for project.
PRJ_ BUSINESS_ OBJECTIVE_ID	I	NUMBER	Indicates the business objective ID for project.
PRJ_ BUSINESS_ OBJECTIVE_ NAME	I	VARCHAR2	Indicates the business objective name for project.
PRJ_PROJECT_ CLASS_CODE	I	VARCHAR2	Indicates the project class code for project.
PRJ_PROJECT_ CLASS_ MEANING	I	VARCHAR2	Indicates the project class meaning for project.
PRJ_ASSET_ CLASS_CODE	I	VARCHAR2	Asset class code for project.
PRJ_ASSET_ CLASS_ MEANING	I	VARCHAR2	Asset class meaning for project.
PRJ_PROJECT_ MANAGER_ USER_ID	I	NUMBER	Code that uniquely identifies the user that is the project manager.
PRJ_PROJECT_ MANAGER_ USERNAME	I	VARCHAR2	Username of the project manager.
PRJ_PROJECT_ PLAN_ID	I	NUMBER	Unique identifier for the project plan.
PRJ_PROJECT_ PLAN_NAME	I	VARCHAR2	Name of the project plan.
PRJ_PROJECT_ PLAN_URL	I	VARCHAR2	Web address or URL associated with the project plan.

Table A-5. KCRT_FG_PFM_PROJECT_INT interface table, continued

Column	Usage	Data Type	Description
PRJ_STAFF_ PROF_ID	I	NUMBER	Unique identifier of the staff profile associated with the PFM project.
PRJ_STAFF_ PROF_NAME	I	VARCHAR2	Name of the staff profile associated with the PFM project.
PRJ_RETURN_ ON_ INVESTMENT	I	NUMBER	Calculated value indicating the return on investment (ROI) of the PFM project.
PRJ_NET_ PRESENT_ VALUE	I	NUMBER	NPV for project.
PRJ_CUSTOM_ FIELD_VALUE	I	NUMBER	Financial metric for project.
PRJ_VALUE_ RATING	I	NUMBER	The score for score domain "Value".
PRJ_RISK_ RATING	I	NUMBER	This score for score domain "Risk".
PRJ_TOTAL_ SCORE	I	NUMBER	Total score.
PRJ_ DISCOUNT_ RATE	I	NUMBER	Discount rate for project.

KCRT_FG_PFM_PROPOSAL_INT

Table for importing PFM proposals into KCRT_FG_PFM_PROPOSAL.

Table A-6. KCRT_FG_PFM_PROPOSAL_INT interface table

Column	Usage	Data Type	Description
PFM_PROPOSAL_ INTERFACE_ID	I	NUMBER	System-generated unique identifier for the proposal.
GROUP_ID	I/O	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report.

Table A-6. KCRT_FG_PFM_PROPOSAL_INT interface table, continued

Column	Usage	Data Type	Description
			Derived from the KNTA_INTERFACE_ GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ID	I/O	NUMBER	Uniquely identifies each transaction. If left blank, the value is generated by the system.
PARENT_ TRANSACTION_ID	I/O	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
PROCESS_PHASE	I	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	I	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
REQUEST_ID	I	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_TYPE_ID	I	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
PROPOSAL_NAME	I	VARCHAR2	Name of the proposal.
PROP_BUSINESS_ UNIT_CODE	I	VARCHAR2	Indicates the business unit code for proposal.
PROP_BUSINESS_ UNIT_MEANING	I	VARCHAR2	Indicates the business unit meaning for proposal.
PROP_BUSINESS_ OBJECTIVE_ID	I	NUMBER	Unique identifier for the proposed business objective.
PROP_BUSINESS_ OBJECTIVE_NAME	I	VARCHAR2	Name of the proposed business objective.
PROP_PROJECT_ CLASS_CODE	I	VARCHAR2	Indicates the project class code for proposal.

Table A-6. KCRT_FG_PFM_PROPOSAL_INT interface table, continued

Column	Usage	Data Type	Description
PROP_PROJECT_CLASS_MEANING	I	VARCHAR2	Indicates the project class meaning for proposal.
PROP_ASSET_CLASS_CODE	I	VARCHAR2	Asset class code for proposal.
PROP_ASSET_CLASS_MEANING	I	VARCHAR2	Asset class meaning for proposal.
PROP_PROJECT_MANAGER_USER_ID	I	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the project manager.
PROP_PROJECT_MANAGER_USERNAME	I	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction.
PROP_PROJECT_TEMPLATE_ID	I	NUMBER	Unique identifier for the proposed project template.
PROP_PROJECT_TEMPLATE_NAME	I	VARCHAR2	Name of the proposed project template.
PROP_STAFF_PROF_ID	I	NUMBER	Unique identifier of the staff profile associated with the proposal.
PROP_STAFF_PROF_NAME	I	VARCHAR2	Name of the staff profile associated with the proposal.
PROP_RETURN_ON_INVESTMENT	I	NUMBER	Indicates the ROI for a proposal.
PROP_NET_PRESENT_VALUE	I	NUMBER	Indicates the NPV for a proposal.
PROP_CUSTOM_FIELD_VALUE	I	NUMBER	Indicates the financial metric for a proposal.
PROP_VALUE_RATING	I	NUMBER	Indicates the score for score domain of "Value".
PROP_RISK_RATING	I	NUMBER	indicates the score for score domain of "Risk".
PROP_TOTAL_SCORE	I	NUMBER	Indicates the total score for a proposal.
PROP_DISCOUNT_RATE	I	NUMBER	Indicates the discount rate for a proposal.
PROP_PLAN_START_PERIOD_ID	I	NUMBER	The period ID of the Planned Start Date for the proposal.

Table A-6. KCRT_FG_PFM_PROPOSAL_INT interface table, continued

Column	Usage	Data Type	Description
PROP_PLAN_FINISH_PERIOD_ID	I	NUMBER	The period ID of the Planned Finish Date for the proposal.
PROP_PLAN_START_PERIOD_NAME	I	NUMBER	The period name of the Planned Start Date for the proposal.
PROP_PLAN_START_PERIOD_NAME	I	VARCHAR2	The period name of the Planned Start Date for the proposal.
PROP_PLAN_FINISH_PERIOD_NAME	I	VARCHAR2	The period name of the Planned Finish Date for the proposal
PROP_PROJECT_PLAN_ID	I	NUMBER	Unique identifier for the proposed project plan.
PROP_PROJECT_PLAN_NAME	I	VARCHAR2	Name of the proposed project plan.

KCRT_FG_PROG_ISSUE_INT

The KCRT_FG_PROG_ISSUE_INT interface table stores validation information, for each request, that is related to the Program Issue field group.

Table A-7. KCRT_FG_PROG_ISSUE_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_TRANSACTION_ID	I	NUMBER	Provides the transaction ID (from KCRT_REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I/O	NUMBER	Identifies the request.

Table A-7. KCRT_FG_PROG_ISSUE_INT interface table, continued

Column	Usage	Data Type	Description
			This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_TYPE_ ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
ESCALATION_ LEVEL_CODE	I	VARCHAR2	Specifies the code for the escalation level.
ESCALATION_ LEVEL_MEANING	I	VARCHAR2	Specifies the description of the escalation level.

KCRT_FG_PROG_REFERENCE_INT

The KCRT_FG_PROG_REFERENCE_INT interface table stores validation information, for each request, that is related to the Program Reference on Request field group.

Table A-8. KCRT_FG_PROG_REFERENCE_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_	I/O	NUMBER	Identifies the request type.

Table A-8. KCRT_FG_PROG_REFERENCE_INT interface table, continued

Column	Usage	Data Type	Description
TYPE_ID			This is normally left blank and is derived from REQUEST_TYPE_NAME.
REF_ PROGRAM_ID	I	VARCHAR2	Creates a reference to the specified program ID.
REF_ PROGRAM_ NAME	I	VARCHAR2	Creates a reference to the specified program name.

KCRT_FG_PROG_RESOURCE_REQ_INT

The KCRT_FG_PROG_RESOURCE_REQ_INT interface table stores validation information, for each request, that is related to the Program Resource Request field group.

Table A-9. KCRT_FG_PROG_RESOURCE_REQ_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_ TYPE_ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.

Table A-9. KCRT_FG_PROG_RESOURCE_REQ_INT interface table, continued

Column	Usage	Data Type	Description
ROLE_ DESCRIPTION_ CODE	I	VARCHAR2	Provides a description of the resource's role.

KCRT_FG_PROJ_ISSUE_INT

The KCRT_FG_PROJ_ISSUE_INT interface table stores validation information, for each request, that is related to the Project Issue field group.

Table A-10. KCRT_FG_PROJ_ISSUE_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_TYPE_ ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
ESCALATION_ LEVEL_CODE	I	VARCHAR2	Specifies the code for the escalation level.
ESCALATION_ LEVEL_MEANING	I	VARCHAR2	Specifies the description of the escalation level.

KCRT_FG_PROJ_RISK_INT

The KCRT_FG_PROJ_RISK_INT interface table stores validation information, for each request, that is related to the Project Risk field group.

Table A-11. KCRT_FG_PROJ_RISK_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_TYPE_ ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
PROBABILITY_ CODE	I	VARCHAR2	Specifies the probability code of the program risk.
PROBABILITY_ MEANING	I	VARCHAR2	Specifies the description of the program risk.
RISK_IMPACT_ LEVEL_CODE	I	VARCHAR2	Specifies the code for the impact level of the program's risk.
RISK_IMPACT_ LEVEL_MEANING	I	VARCHAR2	Specifies the description of the impact level of the program's risk.

KCRT_FG_PROJ_SCOPE_CHANGE_INT

The KCRT_FG_PROJ_SCOPE_CHANGE_INT interface table stores validation information, for each request, that is related to the Project Scope Change field group.

Table A-12. KCRT_FG_PROJ_SCOPE_CHANGE_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_ TYPE_ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
IMPACT_ SEVERITY_ CODE	I	VARCHAR2	Specifies the severity code for the impact of the scope change.
IMPACT_ SEVERITY_ MEANING	I	VARCHAR2	Specifies the description of the severity impact of the scope change.
CR_LEVEL_ CODE	I	VARCHAR2	Specifies the code for the change request importance level of the scope change.
CR_LEVEL_ MEANING	I	VARCHAR2	Specifies the description of the change request importance level of the scope change.

KCRT_FG_QC_DEFECT_INFO_INT

Table for importing requests that have information about a corresponding Quality Center defect into KCRT_FG_QC_DEFECT_INFO.

Table A-13. KCRT_FG_QC_DEFECT_INFO_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I/O	NUMBER	System-generated identifier
TRANSACTION_ID	I/O	NUMBER	Provides a unique identifier for each transaction
PARENT_TRANSACTION_ID	I/O	NUMBER	Provides the transaction id (from KCRT_REQUESTS_INT) of the parent table being imported
REQUEST_ID	I	NUMBER	Identifies the request.
REQUEST_TYPE_ID	I	NUMBER	Identifies the request type.
QC_DEFECT_DOMAIN	I/O	VARCHAR2	Identifies the Quality Center Domain name.
QC_DEFECT_PROJECT	I/O	VARCHAR2	Identifies the Quality Center Project name.
QC_DEFECT_ASSIGN_TO_USER	I/O	VARCHAR2	Identifies the Quality Center user name of the assigned to user.
QC_DEFECT_NO	I/O	NUMBER	Identifies the Quality Center Defect ID.
QC_DEFECT_STATUS	I/O	VARCHAR2	Identifies the Quality Center Defect status (BUG.BG_STATUS).
QC_DEFECT_ATTACHMENT_URL	I/O	VARCHAR2	Identifies the link to Quality Center defect attachments.
QC_DEFECT_INT_MSG	I/O	VARCHAR2	A message informing you whether the Quality Center integration is working properly or whether there are errors. The message changes when one of the following occurs: a new entity (a requirement or defect) is created in Quality Center by PPM Center, the request is updated in PPM Center by Quality Center, or PPM Center attempts to update Quality Center but

Table A-13. KCRT_FG_QC_DEFECT_INFO_INT interface table, continued

Column	Usage	Data Type	Description
			encounters an error.
QC_DEFECT_ DOMAIN_ID	I/O	NUMBER	Identifies the Quality Center Domain ID.
QC_DEFECT_ PROJECT_ID	I/O	NUMBER	Identifies the Quality Center Project ID.
QC_DEFECT_ ASSIGN_TO_ USER_ID	I/O	VARCHAR2	Identifies the Quality Center assigned to user ID.
QC_DEFECT_ INSTANCE	I/O	VARCHAR2	Identifies the URL of the Quality Center server (instance).
QC_DEFECT_ INSTANCE_ID	I/O	NUMBER	Identifies the Quality Center instance.
QC_DEFECT_ ATT_URL_ID	I/O	VARCHAR2	Identifies the Quality Center attachments URL.

KCRT_FG_QC_INFO_INT

Table for importing requests that are integrated with Quality Center server into KCRT_FG_QC_INFO.

Table A-14. KCRT_FG_QC_INFO_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction. If left blank, the value is generated by the system.
PARENT_ TRANSACTION_ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.

Table A-14. KCRT_FG_QC_INFO_INT interface table, continued

Column	Usage	Data Type	Description
REQUEST_ID	I	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_TYPE_ID	I	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
QC_DOMAIN	I	VARCHAR2	Identifies the Quality Center Domain name.
QC_PROJECT	I	VARCHAR2	Identifies the Quality Center Project name.
QC_USER	I	VARCHAR2	Identifies the Quality Center user name.
QC_PASSWORD	I	VARCHAR2	Identifies the Quality Center password associated with the user.
QC_TESTSET_EXEC_ID	I	NUMBER	Uniquely identifies the Quality Center test set execution.
QC_TESTSET_STATUS	I	VARCHAR2	Identifies the status value for the Quality Center test set execution.
QC_REQUIREMENTS_COVERAGE	I	NUMBER	Identifies the percentage of tests planned for this requirement that have been run.
QC_OPEN_DEFECTS	I	NUMBER	Identifies the number of open defects in the project.
QC_DOMAIN_ID	I	NUMBER	Identifies the Quality Center domain.
QC_PROJECT_ID	I	NUMBER	Uniquely identifies the Quality Center project.

KCRT_FG_SERVICE_REQUEST_INT

The KCRT_FG_SERVICE_REQUEST_INT interface table is deprecated. Do not import data into it. It exists solely to support customizations made in earlier versions of PPM Center.

KCRT_FG_SLA_INT

The KCRT_FG_SLA_INT interface table stores validation information, for each request, that is related to the Demand Management SLA Fields field group.

Table A-15. KCRT_FG_SLA_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_ TYPE_ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
SERVICE_ REQUESTED_ DATE	I	DATE	Indicates the date that the service was requested.
SLA_LEVEL_ CODE	I	VARCHAR2	Specifies the code for the service level agreement level. If both SLA_LEVEL_CODE and SLA_LEVEL data types are specified, PPM Center uses the value specified in SLA_LEVEL.
SLA_LEVEL	I	VARCHAR2	Specifies the description for the service level agreement level.

Table A-15. KCRT_FG_SLA_INT interface table, continued

Column	Usage	Data Type	Description
			If both SLA_LEVEL_CODE and SLA_LEVEL data types are specified, PPM Center uses the value specified in SLA_LEVEL.
VIOLATION_ DATE	I	DATE	Indicates the date that the SLA rule was violated.
SERVICE_ SATISFIED_ DATE	I	DATE	Indicates the date that the service was satisfied.

KCRT_FG_WORK_ITEMS_INT

The KCRT_FG_WORK_ITEMS_INT interface table stores validation information, for each request, that are related to the Work Item Fields field group.

Table A-16. KCRT_FG_WORK_ITEMS_INT interface table

Column	Usage	Data Type	Description
WORK_ITEM_ INTERFACE_ID	Obsolete	NUMBER	No longer used.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
PROCESS_ PHASE	Obsolete	NUMBER	No longer used.
PROCESS_ STATUS	Obsolete	NUMBER	No longer used.

Table A-16. KCRT_FG_WORK_ITEMS_INT interface table, continued

Column	Usage	Data Type	Description
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
REQUEST_ TYPE_ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
WORKLOAD_ FLAG	I	VARCHAR2	Indicates whether or not this request should count as workload against resource capacity. Valid values are: • Y • N The default value is Y.
WORKLOAD_ FLAG_MEANING	I	VARCHAR2	Indicates whether or not there is a description associated with WORKLOAD_FLAG. Valid values are: • Yes • No The default value is Yes.
WORKLOAD_ CATEGORY_ CODE	O	VARCHAR2	Specifies the code for the category if the workload represented by this request falls under a category.
WORKLOAD_ CATEGORY_ MEANING	I	VARCHAR2	Specifies the description for the category if the workload represented by this request falls under a category.
ALLOW_ EXTERNAL_ UPDATE_FLAG	I	VARCHAR2	Indicates whether or not the actuals can be updated by an external system (such as Time Management time sheets). Valid values are: • Y • N The default value is N.
USR_	I	DATE	Specifies the date when the work item is scheduled

Table A-16. KCRT_FG_WORK_ITEMS_INT interface table, continued

Column	Usage	Data Type	Description
SCHEDULED_ START_DATE			to start. This is the same day as SCHEDULED_START_ DATE, but the time may not be at 8:00 a.m.
USR_ SCHEDULED_ FINISH_DATE	I	DATE	Specifies the date when the work item is scheduled to finish. This is the same day as SCHEDULED_FINISH_ DATE, but the time may not be the end of the work day.
SCHEDULED_ START_DATE	I	DATE	Specifies the date that the work item is scheduled to start. The starting time is at 8:00 a.m. on that day.
SCHEDULED_ FINISH_DATE	I	DATE	Specifies the date when the work item is scheduled to finish. The ending time is at the end of the working day.
SCHEDULED_ EFFORT	I	NUMBER	Specifies the effort (in hours). Usually equal to (duration) x (hours/day).
SCHEDULED_ DURATION	I	NUMBER	Specifies the number of working days between USR_SCHEDULED_START_DATE and USR_ SCHEDULED_FINISH_DATE.
SCHED_EFF_ OVER_DUR	I	NUMBER	Provides a helper column to be used when calculation actuals (no units).
USR_ACTUAL_ START_DATE	I	DATE	Indicates the date when the work item is scheduled to start. This is the same day as ACTUAL_START_DATE, but the time may not be 8:00 a.m.
USR_ACTUAL_ FINISH_DATE	I	DATE	Indicates the date when the work item is scheduled to finish. This is the same day as ACTUAL_FINISH_DATE, but the time may not be the end of the work day.
ACTUAL_ START_DATE	I	DATE	Indicates the date that the work item actually starts. The starting time is at 8:00 a.m. on that day.
ACTUAL_ FINISH_DATE	I	DATE	Indicates the date that the work item actually finishes. This occurs at the end of that day.

Table A-16. KCRT_FG_WORK_ITEMS_INT interface table, continued

Column	Usage	Data Type	Description
ACTUAL_ EFFORT	I	NUMBER	Specifies the effort (in hours). Usually equal to (duration) x (hours/day).
ACTUAL_ DURATION	I	NUMBER	Indicates the number of working days between USR_ACTUAL_START_DATE and USR_ ACTUAL_FINISH_DATE.
ACTUAL_EFF_ OVER_DUR	I	NUMBER	Provides a helper column used when calculation actuals (no units).
BOOKED_SKILL_ ID	I	NUMBER	Provides the ID of the booked skill. This must match a SKILL_ID in KRSC_SKILLS.
BOOKED_SKILL_ NAME	I	VARCHAR2	Provides the skill name booked on this request. This must match a SKILL_NAME in KRSC_ SKILLS.

KCRT_REQUESTS_INT

The KCRT_REQUESTS_INT interface table stores request header and detail information for each new request generated. This includes information such as request number, priority, project name, description, and attached notes. This table also holds columns to import user-defined detail fields (user data) determined by the request type for each specific request.

Table A-17. KCRT_REQUESTS_INT interface table

Column	Usage	Data Type	Description
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language. If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
TRANSACTION_	I	NUMBER	Uniquely identifies each transaction.

Table A-17. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
ID			If any detail table is being used, set the PARENT_TRANSACTION_ID in the detail interface tables to this value.
PROCESS_PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
CREATION_DATE	I/O	DATE	Indicates the transaction date. If left blank, the current date is used.
CREATED_USERNAME	I/O	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_USERNAME.
LAST_UPDATE_DATE	I/O	DATE	Indicates the transaction date. If left blank, the current date is used.
LAST_UPDATED_USERNAME	I	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction.
LAST_UPDATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. This is normally left blank and is derived from LAST_UPDATED_USERNAME.

Table A-17. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
ENTITY_LAST_ UPDATE_DATE	I/O	DATE	Indicates the transaction date. This is normally left blank and the current date is used.
REQUEST_ NUMBER	I/O	VARCHAR2	Identifies the request. This is normally left blank and is derived from REQUEST_ID. If a value is entered, it should be unique and should match the value in the REQUEST_ID field.
REQUEST_ TYPE_NAME	I	VARCHAR2	Identifies the request type. Derived from KCRT_REQUESTS_TYPES.
REQUEST_ TYPE_ID	I/O	NUMBER	Identifies the request type. If left blank, the value is derived from REQUEST_ TYPE_NAME.
REQUEST_ SUBTYPE_ NAME	I	VARCHAR2	Identifies the request subtype. If a value is entered, it should be a valid subtype from KCRT_REQUEST_SUB_TYPES.
REQUEST_ SUBTYPE_ID	I/O	NUMBER	Identifies the request subtype. If left blank, the value is derived from REQUEST_ SUBTYPE_NAME.
DESCRIPTION	I	VARCHAR2	Specifies a user-visible description of the request.
RELEASE_ DATE	I/O	DATE	Indicates when the request first became active. For new requests, this should be left blank and the current date is used. When converting existing requests from a third-party system, enter the initial creation date of the request in the remote system.
STATUS_NAME	I/O	VARCHAR2	Indicates the current status of the request. This should be a valid status for the given request. This should be a request status for at least one workflow step of the workflow. If left blank, the new request will get the initial status indicated on the request type definition.
STATUS_ID	I/O	NUMBER	Indicates the current status of the request.

Table A-17. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
			If left blank, the value is derived from STATUS_NAME.
WORKFLOW_NAME	I/O	VARCHAR2	Specifies the workflow that the request should follow. This is normally left blank and its value is based on the values for request type, department, and application for the request.
WORKFLOW_ID	I/O	NUMBER	Specifies the workflow that the request should follow. This is normally left blank andthe value is derived from WORKFLOW_NAME.
DEPARTMENT_CODE	O	VARCHAR2	Specifies the code for the department.
DEPARTMENT_NAME	I	VARCHAR2	Specifies the name of the department. This should be a valid MEANING from KNTA_LOOKUPS where `LOOKUP_TYPE = 'DEPARTMENT_CODE'`.
PRIORITY_CODE	O	VARCHAR2	Specifies the user-defined priority for the request.
PRIORITY_NAME	I	VARCHAR2	Specifies the user-defined priority name for the request. If entered, this should be a valid MEANING from KNTA_LOOKUPS where `LOOKUP_TYPE = 'REQUEST_PRIORITY'`.
APPLICATION	I	VARCHAR2	Indicates the user-defined application for the request. This should be a valid LOOKUP_CODE from KNTA_LOOKUPS where `LOOKUP_TYPE = 'APPLICATION'`.
ASSIGNED_TO_USERNAME	I	VARCHAR2	Specifies the USERNAME (from KNTA_USERS) that should initially be assigned the request.
ASSIGNED_TO_USER_ID	I/O	NUMBER	Specifies the USER_ID (from KNTA_USERS) that should initially be assigned the request. If left blank, the value is derived from ASSIGNED_TO_USERNAME.
ASSIGNED_TO_GROUP_NAME	I	VARCHAR2	Specifies the SECURITY_GROUP_ID (from KNTA_SECURITY_GROUPS) that should initially be

Table A-17. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
			assigned the request.
ASSIGNED_TO_GROUP_ID	I/O	NUMBER	Specifies the SECURITY_GROUP_ID that should initially be assigned to the request. This is normally left blank and the value is derived from ASSIGNED_TO_GROUP_NAME.
PROJECT_CODE	I	VARCHAR2	Indicates the user-defined project for the request. This should be a valid value from KNTA_LOOKUPS where `LOOKUP_TYPE = 'PROJECT'`.
CONTACT_FIRST_NAME	I	VARCHAR2	Specifies the first name of the contact for the request. This should be a valid value from FIRST_NAME in KCRT_CONTACTS. If a value is entered, CONTACT_LAST_NAME must also be populated.
CONTACT_LAST_NAME	I	VARCHAR2	Specifies the last name of the contact for the request. This should be a valid value from LAST_NAME in KCRT_CONTACTS. If a value is entered, CONTACT_FIRST_NAME must also be populated.
CONTACT_ID	O	NUMBER	Specifies the ID of the contact for the request. This is derived from the CONTACT_FIRST_NAME and CONTACT_LAST_NAME.
RELEASED_FLAG	I	VARCHAR2	Indicates whether or not the request should be released after import. Valid values are: • Y • N The default value is N.
USER_DATA_SET_CONTEXT_ID	Obsolete	NUMBER	No longer used.
USER_DATA1	I	VARCHAR2	Specifies the user-defined fields attached to the user screen.

Table A-17. KCRT_REQUESTS_INT interface table, continued

Column	Usage	Data Type	Description
VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USERS_DATA20			This is required only if user data is defined. This information is not validated nor does it have a default value.
PARAMETER_SET_CONTEXT_ID	I	NUMBER	Sets the context identifier for the detail fields. Either this or REQUEST_TYPE_NAME must be populated.
NOTES	I	LONG	Optional. Provides free-form note text that is visible in the **Notes** tab of the request window. Carriage returns should be represented as {\n} and is replaced with actual carriage returns when the note is moved into the notes table. This can be helpful when the interface table is populated through SQL*Loader.
SOURCE_TYPE_CODE	I	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_RI.
SOURCE	I	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
WORKFLOW_STEP_ID	O	NUMBER	Identifies the workflow step that becomes eligible for user processing.
COMPANY	I	VARCHAR2	Identifies the name of the company associated with this request. This should be a valid LOOKUP_CODE from KNTA_LOOKUPS where LOOKUP_TYPE = `COMPANY'.

KCRT_REQUEST_DETAILS_INT

The KCRT_REQUESTS_INT interface table is used to store validation information related to the user-defined custom fields for each request.

Table A-18. KCRT_REQUEST_DETAILS_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQUEST_ DETAIL_ID	I/O	NUMBER	Identifies the detail ID of the request (from KCRT_ REQUEST_DETAILS).
REQUEST_ID	I/O	NUMBER	Identifies the request. If left blank, the value is derived from the KCRT_ REQUESTS_S sequence.
REQUEST_TYPE_ ID	I/O	NUMBER	Identifies the request type. If left blank, the value is derived from REQUEST_ TYPE_NAME.
PARAMETER_ SET_CONTEXT_ID	I/O	NUMBER	Sets the context identifier for the detail fields. If left blank, the value is derived from the REQUEST_TYPE_NAME.
BATCH_NUMBER	I	NUMBER	Specifies the batch number for the custom fields. This corresponds to the **Storage** tab in the **field** definition window on the request type.
PARAMETER1 VISIBLE_ PARAMETER1 through PARAMETER50 VISIBLE_ PARAMETER50	I	VARCHAR2	Specifies the values for all the custom fields defined in the request. Values do not need to be entered.
LOOKUP_TYPE1 VALIDATION_	I	VARCHAR2 VARCHAR2	Identifies the lookup type for each PARAMETER and the validation type code for each PARAMETER.

Table A-18. KCRT_REQUEST_DETAILS_INT interface table, continued

Column	Usage	Data Type	Description
TYPE_CODE1 through LOOKUP_TYPE50 VALIDATION_ TYPE_CODE50			This is required only if custom data is defined.

KCRT_REQ_HEADER_DETAILS_INT

The KCRT_REQ_HEADER_DETAILS_INT interface table stores data for custom fields that are defined in the request header.

> **Note:** Standard request header type fields (such as request number and priority) are stored in KCRT_REQUESTS_INT.

Table A-19. KCRT_REQ_HEADER_DETAILS_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUESTS_INT) of the parent table being imported.
REQ_HEADER_ DETAIL_ID	I/O	NUMBER	Identifies the header detail ID for the request. If left blank, the value is derived from the KCRT_REQ_ HEADER_DETAILS_S sequence.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.

Table A-19. KCRT_REQ_HEADER_DETAILS_INT interface table, continued

Column	Usage	Data Type	Description
REQUEST_ TYPE_ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.
BATCH_ NUMBER	I	NUMBER	Specifies the batch number for the custom fields. This corresponds to the **Storage** tab in the **field** definition window on the request type.
PARAMETER1 VISIBLE_ PARAMETER1 through PARAMETER50 VISIBLE_ PARAMETER50	I	VARCHAR2	Specifies the values for all the custom fields defined in the request.
LOOKUP_TYPE1 VALIDATION_ TYPE_CODE1 through LOOKUP_ TYPE50 VALIDATION_ TYPE_CODE50	I	VARCHAR2 VARCHAR2	Identifies the lookup type for each PARAMETER and the validation type code for each PARAMETER. This is required only if custom data is defined.

KCRT_TABLE_ENTRIES_INT

The KCRT_TABLE_ENTRIES_INT interface table specifies the table fields defined in the request type for the request.

Table A-20. KCRT_TABLE_ENTRIES_INT interface table

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S

Table A-20. KCRT_TABLE_ENTRIES_INT interface table, continued

Column	Usage	Data Type	Description
			sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KCRT_ REQUEST_DETAILS_INT) of the parent table being imported.
PARENT_FIELD_ TOKEN	I	VARCHAR2	Specifies the token.
TABLE_ENTRY_ID	I/O	NUMBER	Identifies the table entry record. If left blank, the value is derived from the KCRT_ TABLE_ENTRIES_S sequence.
REQUEST_ID	I/O	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.
PARAMETER_ SET_FIELD_ID	I/O	NUMBER	Specifies the field in the table to which this entry belongs.
SEQ	I	NUMBER	Provides a user-visible sequence number for the package line. This must be a unique, positive integer that does not conflict with other records being imported.
PARAMETER_ SET_CONTEXT_ ID	I/O	NUMBER	Sets the context identifier for the detail fields. If left blank, the value is derived from the REQUEST_TYPE_NAME.
VISIBLE_ PARAMETER1 PARAMETER1 through VISIBLE_ PARAMETER50 PARAMETER50	I	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.
LOOKUP_TYPE1	I	VARCHAR2	Identifies the lookup type for each PARAMETER.

Table A-20. KCRT_TABLE_ENTRIES_INT interface table, continued

Column	Usage	Data Type	Description
through LOOKUP_TYPE50			This is required only if user data is defined.
VALIDATION_ TYPE_CODE1 through VALIDATION_ TYPE_CODE50	I	VARCHAR2	Identifies the validation type code for each PARAMETER. This is required only if user data is defined.

KDLV_PACKAGES_INT

The KDLV_PACKAGES_INT interface table is used to define header information for each new package. This interface table stores package header information for new packages to be generated. This includes information such as package number, priority, project name, and description. This table also holds columns to import user data information (custom fields attached to the packages).

Table A-21. KDLV_PACKAGES_INT interface table

Column	Usage	Data Type	Description
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language. If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.
PACKAGE_ INTERFACE_ID	I	NUMBER	Provides a unique identifier for the each record. Derived from the KDLV_INTERFACES_S sequence. For lines tied to a new package, this can be used to tie the line record to the parent record in KDLV_ PACKAGES_INT. The PACKAGE_NUMBER and PACKAGE_ID columns can be used for this tie as well. This is required if package lines exist. For new lines, this should be left blank.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S

Table A-21. KDLV_PACKAGES_INT interface table, continued

Column	Usage	Data Type	Description
			sequence.
PROCESS_ PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_ STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME. If both are left blank, the value is set to the user currently running the report.
CREATED_BY_ USERNAME	I	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank.
CREATION_ DATE	I/O	DATE	Indicates the transaction date. If left blank, the current date is used.
SOURCE_CODE	I	VARCHAR2	Provides the identify of the source of the record. This value is not validated and is for informational purposes only.
PACKAGE_ID	I	NUMBER	Provides an identifier for a package and makes the association between the package and package lines. Derived from the KDLV_PACKAGES_S sequence. For lines tied to a new package, this column can be used to tie the line record to the parent record in KDLV_ PACKAGES_INT. Either PACKAGE_INTERFACE_ ID and PACKAGE_NUMBER can be used to tie the records. For new lines to be imported into existing packages, this column should refer to the PACKAGE_ID of the existing package.
REQUESTED_	I	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the

Table A-21. KDLV_PACKAGES_INT interface table, continued

Column	Usage	Data Type	Description
BY			user requesting the package. If left blank, the value is derived from REQUESTED_BY_USERNAME. If both are left blank, the value is set to the user currently running the report.
REQUESTED_BY_USERNAME	I	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) of the user requesting the package. This is used only if REQUESTED_BY is left blank.
PACKAGE_NUMBER	I	VARCHAR2	Identifies the package number. This must use either the same value as PACKAGE_ID or a unique string.
ASSIGNED_TO_USER_ID	I/O	NUMBER	Specifies the USER_ID (from KNTA_USERS) that should initially be assigned the request. If left blank, the value is derived from ASSIGNED_TO_USERNAME. If both are left blank, the package will not have an initial value.
ASSIGNED_TO_USERNAME	I	VARCHAR2	Specifies the USERNAME (from KNTA_USERS) that should initially be assigned the request. This is used only if ASSIGNED_TO_USER_ID is left blank.
ASSIGNED_TO_GROUP_ID	I/O	NUMBER	Specifies the SECURITY_GROUP_ID (from KNTA_SECURITY_GROUPS) that should initially be assigned to the package. If left blank, this value is derived from ASSIGNED_TO_GROUP_NAME. If both are left blank, the package will not have an initial value.
ASSIGNED_TO_GROUP_NAME	I	VARCHAR2	Specifies the SECURITY_GROUP_ID (from KNTA_SECURITY_GROUPS) that should initially be assigned the package. This is used only if ASSIGNED_TO_GROUP_ID is left blank.
DESCRIPTION	I	VARCHAR2	Specifies a user-visible description of the package.

Table A-21. KDLV_PACKAGES_INT interface table, continued

Column	Usage	Data Type	Description
PACKAGE_ TYPE_CODE	I	VARCHAR2	Provides a user-defined categorization of the package. Must be a valid LOOKUP_CODE from KNTA_ LOOKUPS where `LOOKUP_TYPE = 'PACKAGE_TYPE'`.
PRIORITY_ CODE	I	VARCHAR2	Indicates the user-defined priority for the package. Must be a valid LOOKUP_CODE from KNTA_ LOOKUPS where `LOOKUP_TYPE = 'PACKAGE_ PRIORITY'`.
STATUS_CODE	O	VARCHAR2	Indicates the status of the package.
PROJECT_ CODE	I	VARCHAR2	Indicates the user-defined project for the package. This should be a valid value from KNTA_LOOKUPS where `LOOKUP_TYPE = 'PROJECT'`.
WORKFLOW_ID	I	NUMBER	Specifies the workflow that the package should follow. Derived from WORKFLOW_NAME. Either WORKFLOW_ID or WORKFLOW_NAME must be entered.
WORKFLOW_ NAME	I	VARCHAR2	Specifies the workflow that the package should follow. This is used only if WORKFLOW_ID is left blank.
PRIORITY_SEQ	I/O	NUMBER	Provides a sequence number used to determine the relative priority of packages that are scheduled to process at the same time. If left blank, the value is set to 10.
RELEASE_FLAG	I	VARCHAR2	Indicates whether or not the interface program will release the package once it imports in into the standard Deployment Management tables. Valid values are: • Yes • No The default is No.
USER_DATA_ SET_CONTEXT_ ID	I/O	NUMBER	Sets the context identifier for the USER_DATA fields. If left blank, the value is set to 1202.
USER_DATA1	I	VARCHAR2	Specifies the user-defined fields attached to the user screen.

Table A-21. KDLV_PACKAGES_INT interface table, continued

Column	Usage	Data Type	Description
VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USER_DATA20			This is required only if user data is defined. This information is not validated nor does it have a default value.
SOURCE_PACKAGE_ID	I	NUMBER	Identifies the original package for this distribution package.
DISTPKG_STATUS_MEANING	I	VARCHAR2	Provides a user-visible status for this distribution package.
RUN_GROUP	I	NUMBER	Provides a run group number of a specific distribution package.
DISTRIBUTION_ID	I	NUMBER	Identifies the distribution associated with the package.
ENABLED_FLAG	I	VARCHAR2	Indicates whether or not the distribution package is enabled upon import. (Applies to distribution packages only.) Valid values are: • Y • N The default values is Y.
DIST_STEP_TRANSACTION_ID	I	NUMBER	Specifies the path of the distribution workflow step that was run in the transaction with DIST_STEP_TRANSACTION_ID.

KDLV_PACKAGE_LINES_INT

The KDLV_PACKAGE_LINES_INT interface table defines each package line for a new package, or each package line added to an existing package. This interface table holds information for each package line on the new package, or for each new package line to be added to an existing package. This includes information of the specific object type and application code for the package line, and parameter information for the specified object type and user data for the package line.

Table A-22. KDLV_PACKAGE_LINES_INT interface table

Column	Usage	Data Type	Description
PACKAGE_ LINE_ INTERFACE_ID	I/O	NUMBER	Provides a unique identifier for the record. If left blank, the value is derived from the KDLV_ INTERFACES_S sequence.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
PACKAGE_ INTERFACE_ID	I	NUMBER	Provides a unique identifier for the each record. Derived from the KDLV_INTERFACES_S sequence. For lines tied to a new package, this can be used to tie the line record to the parent record in KDLV_ PACKAGES_INT. The PACKAGE_NUMBER and PACKAGE_ID columns can be used for this tie as well. This is required if package lines exist. For new lines, this should be left blank.
PACKAGE_ID	I	NUMBER	Provides an identifier for a package and makes the association between the package and package lines. Derived from the KDLV_PACKAGES_S sequence. For new lines to be imported into existing packages, this column should refer to the PACKAGE_ID of the existing package. For lines tied to a new package, this column can be used to tie the line record to the parent record in KDLV_ PACKAGES_INT. Either PACKAGE_INTERFACE_ID and PACKAGE_NUMBER can be used to tie the records.
PACKAGE_ NUMBER	I	VARCHAR2	Identifies the package number. This must use either the same value as PACKAGE_ID or a unique string.
PROCESS_ PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_ STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.

Table A-22. KDLV_PACKAGE_LINES_INT interface table, continued

Column	Usage	Data Type	Description
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME. If both are left blank, the value is set to the user currently running the report.
CREATED_BY_USERNAME	I	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank.
CREATION_DATE	I/O	DATE	Indicates the transaction date. If left blank, the current date is used.
SOURCE_CODE	I	VARCHAR2	Provides the identify of the source of the record. This value is not validated and is for informational purposes only.
SEQ	I	NUMBER	Provides a user-visible sequence number for the package line. This must be a unique, positive integer and not conflict with other package lines in the interface table or existing lines if importing lines to an existing packages.
PACKAGE_LINE_ID	I/O	NUMBER	Provides the identifier for a package line. This is normally left blank and the value is derived from the KDLV_PACKAGE_LINES_S sequence.
OBJECT_TYPE_ID	I	NUMBER	Provides the object type ID attached to the package line. Derived from OBJECT_TYPE_ID (in KDLV_OBJECT_TYPES). Either OBJECT_TYPE_ID or OBJECT_TYPE_NAME must be entered.
OBJECT_TYPE_NAME	I	VARCHAR2	Provides the object type name attached to the package line. Derived from OBJECT_TYPE_NAME (in KDLV_OBJECT_TYPES). This is used only if OBJECT_TYPE_ID is left blank.
OBJECT_NAME	I	VARCHAR2	Specifies the name of the object to be processed.

Table A-22. KDLV_PACKAGE_LINES_INT interface table, continued

Column	Usage	Data Type	Description
			This value is not validated.
APP_CODE	I/O	VARCHAR2	Specifies the application category for the package line. Derived from KDLV_ENVIRONMENT_APPS. The APP_CODE must exist for all environments in the workflow attached to the package. APP_CODE can be used as information and can sometimes determine migration behavior.
PARAMETER_ SET_ CONTEXT_ID	I/O	NUMBER	Sets the context identifier for the detail fields. This is normally left blank and is derived from OBJECT_ TYPE_ID.
PARAMETER1 VISIBLE_ PARAMETER1 through PARAMETER30 VISIBLE_ PARAMETER30	I	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.
RELEASE_ FLAG	I/O	VARCHAR2	Indicates whether or not the interface program will release the package once it imports in into the standard Deployment Management tables. Valid values are: • Yes • No The default is No.
USER_DATA_ SET_ CONTEXT_ID	I/O	NUMBER	Sets the context identifier for the USER_DATA fields. If left blank, the value is set to 1203.
USER_DATA1 VISIBLE_ USER_DATA1 through USER_DATA20	I	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined. This information is not validated nor does it have a default value.

Table A-22. KDLV_PACKAGE_LINES_INT interface table, continued

Column	Usage	Data Type	Description
VISIBLE_ USER_DATA20			
OBJECT_ REVISION	I	VARCHAR2	Specifies the denormalized object_revision of the object entered on this line.
SOURCE_ PACKAGE_ LINE_ID	I	NUMBER	Identifies the original package line for this distribution package line.
ENABLED_ FLAG	I	VARCHAR2	Indicates whether or not the distribution package is enabled upon import. (Applies to distribution packages only.) Valid values are: • Y • N The default is Y.

KDLV_PACKAGE_NOTES_INT

The KDLV_PACKAGE_NOTES_INT interface table defines the notes attached to the new package. It can only be used when importing a new package and cannot be used to update the notes of an existing package.

Table A-23. KDLV_PACKAGE_NOTES_INT interface table

Column	Usage	Data Type	Description
PACKAGE_ NOTE_ INTERFACE_ ID	I/O	NUMBER	Provides a unique identifier for the record. If left blank, the value is derived from the KDLV_ INTERFACES_S sequence.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
PACKAGE_ INTERFACE_	I	NUMBER	Provides a unique identifier for the each record. Derived from the KDLV_INTERFACES_S sequence.

Table A-23. KDLV_PACKAGE_NOTES_INT interface table, continued

Column	Usage	Data Type	Description
ID			This is required if package lines exist. For new lines, this should be left blank. For lines tied to a new package, this can be used to tie the line record to the parent record in KDLV_PACKAGES_INT. The PACKAGE_NUMBER and PACKAGE_ID columns can be used for this tie as well.
PACKAGE_ID	I	NUMBER	Provides an identifier for a package and makes the association between the package and note. Derived from the KDLV_PACKAGES_S sequence. Identifies the package ID. This can be used to tie the note record to the parent record in KDLV_PACKAGES_INT. Either PACKAGE_INTERFACE_ID and PACKAGE_NUMBER can be used to tie the records.
PACKAGE_NUMBER	I	VARCHAR2	Identifies the package number. This must use either the same value as PACKAGE_ID or a unique string. This can be used to tie the note record to the parent record in KDLV_PACKAGES_INT. The PACKAGE_INTERFACE_ID and PACKAGE_ID can be used for this tie as well.
PROCESS_PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME. If both are left blank, the value is set to the user currently running the report.
CREATED_BY_USERNAME	I	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank.

Table A-23. KDLV_PACKAGE_NOTES_INT interface table, continued

Column	Usage	Data Type	Description
CREATION_DATE	I/O	DATE	Indicates the transaction date. If left blank, the current date is used.
SOURCE_CODE	I	VARCHAR2	Provides the identify of the source of the record. This value is not validated and is for informational purposes only.
NOTE	I	CLOB	Specifies the full text of the note.
REPLACE_NOTE_FLAG	Obsolete	VARCHAR2	No longer used.

KDLV_TRANSACTIONS_INT

The KDLV_TRANSACTIONS_INT interface table is used to provide transaction information.

This table and other interface tables will be processed through the Java™ report that calls a PL/SQL package that can be run either from the command line or through the report job scheduler. This allows the user to immediately run the interface but also allow for scheduled processing.

Table A-24. KDLV_TRANSACTIONS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_INTERFACE_ID	I/O	NUMBER	System-generated identifier
GROUP_ID	I/O	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
PROCESS_PHASE	I/O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	I/O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.

Table A-24. KDLV_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME.
CREATED_BY_USERNAME	I/O	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_DATE	I/O	DATE	Indicates the date that the record was created. If left blank, the current date is used.
SOURCE_CODE	I/O	VARCHAR2	Provides the identify of the source of the record. This value is not validated and is for informational purposes only.
PACKAGE_ID	I/O	NUMBER	Provides an identifier for a package and makes the association between the package and package lines. Derived from the KDLV_PACKAGES_S sequence. For lines tied to a new package, this column can be used to tie the line record to the parent record in KDLV_PACKAGES_INT. Either PACKAGE_INTERFACE_ID and PACKAGE_NUMBER can be used to tie the records. For new lines to be imported into existing packages, this column should refer to the PACKAGE_ID of the existing package.
PACKAGE_NUMBER	I/O	VARCHAR2	Identifies the package number. This must use either the same value as PACKAGE_ID or a unique string.
PACKAGE_LINE_ID	I/O	NUMBER	Provides the identifier for a package line. This is normally left blank and the value is derived from the KDLV_PACKAGE_LINES_S sequence.
PACKAGE_LINE_SEQ	I/O	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_LINES_S sequence.

Table A-24. KDLV_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
			Use if the transaction is for a package line.
WORKFLOW_ INSTANCE_ STEP_ID	I/O	NUMBER	Specifies the instance step ID.
WORKFLOW_ STEP_ID	I/O	NUMBER	Specifies the workflow step ID (WORKFLOW_STEP_ ID from KWFL_WORKFLOW_STEPS). Supply this, WORKFLOW_STEP_NAME, or WORKFLOW_STEP_SEQ.
WORKFLOW_ STEP_NAME	I/O	VARCHAR2	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS). Supply this or P_WORKFLOW_STEP_SEQ.
RESULT_VALUE	I/O	VARCHAR2	Indicates the result of the step. This is normally not displayed to the user; therefore it may be an ID or internal code.
VISIBLE_ RESULT_VALUE	I/O	VARCHAR2	Indicates the result of the step. This is the result value that a user normally sees.
EXECUTE_ FLAG	I/O	VARCHAR2	Indicates whether or not a specific step is executable.
COMMENTS	I/O	VARCHAR2	Specifies comments for the transaction. Any comments are appended to the notes for the package or request.
SCHEDULE_ DATE	I/O	DATE	Indicates the date that the execution step is scheduled to run.

KNTA_USERS_INT

The KNTA_USERS_INT interface table is used to provide user attributes for new or existing users. It is also used to link the users to various products in the PPM Center.

Table A-25. KNTA_USERS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction. See also PARENT_TRANSACTION_ID in KNTA_ USER_SECURITY_INT.

Table A-25. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language. If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
EXISTS_FLAG	O	VARCHAR2	Indicates whether or not the user already exists.
PROCESS_PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME.
CREATED_BY_USERNAME	I/O	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_DATE	I/O	DATE	Indicates the date that the record was created. If left blank, the current date is used.
DEST_CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction.

Table A-25. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
			If left blank, the value is derived from CREATED_BY_USERNAME. If both are left blank, the value is set to the user currently running the report.
DEST_CREATION_DATE	I/O	DATE	Indicates the date the record is created in the destination (PPM Center instance). If left blank, the value is derived from CREATION_DATE.
DEST_LAST_UPDATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the user currently running the report.
DEST_LAST_UPDATE_DATE	I/O	DATE	Indicates the date that the user data was last updated. If left blank, the current date is used.
DEST_ENTITY_UPD_DATE	I/O	DATE	Indicates the date that either the user data or security data was last updated. If left blank, the current date is used.
USER_ID	I/O	NUMBER	Identifies the user. When creating users, this is left blank and the value is derived from the KNTA_USERS_S sequence. For existing users, this can be left blank or a valid USER_ID (from KNTA_USERS) be provided.
DEST_USER_ID	I/O	NUMBER	Identifies the user. This is normally left blank and is derived from the KNTA_USERS_S sequence.
USERNAME	I	VARCHAR2	Identifies the name used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_NAME, the USERNAME column must be populated for the user import. Otherwise, populate the LOGON_IDENTIFIER column.
DEST_USERNAME	I/O	NUMBER	Identifies the username.

Table A-25. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
			If left blank, the value is derived from USERNAME.
PASSWORD	I/O	VARCHAR2	Specifies the password for the user. If left blank, the value is set to the password of the user currently running the report.
PASSWORD_ EXPIRATION_DAYS	I	NUMBER	Specifies the number of days before the current password expires.
PASSWORD_ EXPIRATION_DATE	I	DATE	Specifies the date when the password should expire.
EMAIL_ADDRESS	I	VARCHAR2	Specifies the email address of the user.
FIRST_NAME	I	VARCHAR2	Specifies the user's first name. This is required only if creating a new user. It is not required when re-importing an existing user.
LAST_NAME	I	VARCHAR2	Specifies the user's last name. This is required only if creating a new user. It is not required when re-importing an existing user.
START_DATE	I	DATE	Specifies the user's start date.
END_DATE	I	DATE	Specifies the user's end date.
DEFAULT_ ACCELERATOR_ID	I	NUMBER	Sets the context identifier for the USER_DATA fields.
SOURCE_TYPE_ CODE	I	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_ WF.
SOURCE	I	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
USER_DATA_SET_ CONTEXT_ID	I	NUMBER	Sets the context identifier for the USER_DATA fields. Supply this or USERNAME.
USER_DATA1 VISIBLE_USER_ DATA1	I	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.

Table A-25. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
through USER_DATA20 VISIBLE_USER_DATA20			This information is not validated nor does it have a default value.
AUTHENTICATION_MODE	I	VARCHAR2	Specifies the user's authentication mode. If the user is being imported from a LDAP server, then this is automatically set to LDAP. Otherwise it is set to KINTANA. For custom implementations, other values can be used.
SCREEN_ID	I/O	NUMBER	Specifies the first screen shown after logon. If left blank, the default value is supplied.
SHORTCUT_BAR_FLAG	I/O	VARCHAR2	Indicates whether or not the shortcut bar is shown in the screen manager. If left blank, the default value is supplied.
SHORTCUT_BAR_LOC_CODE	I/O	VARCHAR2	Specifies the position where the shortcut bar is displayed. If left blank, the default value is supplied.
SAVE_WINDOW_BOUNDS_FLAG	I/O	VARCHAR2	Indicates whether or not the size and location of the screen manager window are saved after logoff. If they are saved, the settings are the default at the next logon. If left blank, the default value is supplied.
WINDOW_HEIGHT	I/O	NUMBER	Specifies the default height of the screen manager window. If left blank, the default value is supplied.
WINDOW_WIDTH	I/O	NUMBER	Specifies the default width of the screen manager window. If left blank, the default value is supplied.
WINDOW_X_LOCATION	I/O	NUMBER	Specifies the horizontal position of the screen manager window. If left blank, the default value is supplied.
WINDOW_Y_LOCATION	I/O	NUMBER	Specifies the vertical position of the screen manager window.

Table A-25. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
			If left blank, the default value is supplied.
REUSE_ INTERNAL_ FRAME_FLAG	I/O	VARCHAR2	Indicates whether or not multiple internal frames can be opened within each screen. If left blank, the default value is supplied.
SHOW_ALL_ WORKFLOW_ STEPS_FLAG	I/O	VARCHAR2	Indicates whether or not all workflow steps are shown within workflow status panels. If left blank, the default value is supplied.
SHOW_ TRAVERSED_ STEPS_FLAG	I/O	VARCHAR2	Indicates whether or not steps that have been traversed and are no longer active are shown within workflow status panels. If left blank, the default value is supplied.
NUM_BRANCH_ STEPS_TO_SHOW	I/O	NUMBER	If a currently active workflow step leads to several branches, specifies how many steps of each branch are shown within workflow status panels. If left blank, the default value is supplied.
NUM_KNOWN_ REACH_STEPS_ TO_SHOW	I/O	NUMBER	Specifies the number of steps of a non-branching path that are shown within workflow status panels. If left blank, the default value is supplied.
HIDE_IMMEDIATE_ STEPS_FLAG	I/O	VARCHAR2	Indicates whether or not workflow steps based upon immediate executions and conditions are shown within workflow status panels. If left blank, the default value is supplied.
SHOW_CHANGE_ WARNINGS_FLAG	I/O	VARCHAR2	Indicates whether or not warning messages are displayed when a business entity that is used by another entity is updated. For example, when a workflow is updated that is used by a package line. If left blank, the default value is supplied.
HIDE_ CANCELLED_CRL_ FLAG	I/O	VARCHAR2	Indicates whether or not cancelled package lines are displayed in the packages screen. If left blank, the default value is supplied.
DEFAULT_ BROWSER	I	VARCHAR2	Specifies the default browser for the user.

Table A-25. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
DEST_USER_ PROFILE_ID	O	NUMBER	Specifies the user profile ID for the user.
COMPANY	I	VARCHAR2	Identifies the company. This should be a valid LOOKUP_CODE from KNTA_LOOKUPS where LOOKUP_TYPE = `COMPANY'.
DOMAIN	I	VARCHAR2	Identifies the Windows domain. Used for Exchange server (NTLM) authentication.
LOGON_ IDENTIFIER	I	VARCHAR2	Identifies the ID used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, the LOGON_IDENTIFIER column must be populated. Otherwise, populate the USERNAME column.
PHONE_NUMBER	I	VARCHAR2	Specifies the user's phone number on the resource page.
COST_RATE	I	NUMBER	Specifies the user's cost rate.
WORKLOAD_ CAPACITY	I	NUMBER	Specifies the user's workload capacity (in percentage) on the resource page.
MAX_ROWS_ PORTLETS	I	NUMBER	Specifies the maximum number of results to be displayed on the maximized portlet.
DEPARTMENT_ CODE	O	VARCHAR2	Specifies the code for the department.
DEPARTMENT_ MEANING	I	VARCHAR2	Specifies the description of the department.
LOCATION_CODE	O	VARCHAR2	Specifies the code for the location.
LOCATION_ MEANING	I	VARCHAR2	Specifies the description of the location.
MANAGER_USER_ ID	I	NUMBER	Specifies the user ID of the manager. Used if both MANAGER_USERNAME and MANAGER_LOGON_IDENTIFIER are left blank.
MANAGER_ USERNAME	I	VARCHAR2	Specifies the name of the manager. Used if MANAGER_LOGON_IDENTIFIER is left

Table A-25. KNTA_USERS_INT interface table, continued

Column	Usage	Data Type	Description
			blank.
MANAGER_ LOGON_ IDENTIFIER	I	VARCHAR2	Specifies the ID of the manager. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, the LOGON_IDENTIFIER column must be populated. Otherwise, populate the MANAGER_USERNAME column.
RESOURCE_ CATEGORY_CODE	O	VARCHAR2	Specifies the code for the user's category.
RESOURCE_ CATEGORY_ MEANING	I	VARCHAR2	Specifies the description of the user's category.
RESOURCE_ TITLE_CODE	O	VARCHAR2	Specifies the code for the user's title.
RESOURCE_ TITLE_MEANING	I	VARCHAR2	Specifies the description of the user's title.
PRODUCT_ID_LIST	O	VARCHAR2	Indicates the user's license.

KNTA_USER_SECURITY_INT

The KNTA_USER_SECURITY_INT interface table is used to define the user security information.

Table A-26. KNTA_USER_SECURITY_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.
PARENT_ TRANSACTION_ ID	I	NUMBER	Provides the transaction ID (from KNTA_USERS_INT) of the parent table being imported. If any child table is being used, set the TRANSACTION_ID in KNTA_USERS_INT to this value.
PARENT_ TABLE_NAME	I	VARCHAR2	Identifies the table associated with this entity. The parent_table should be derived from KNTA_ USERS_INT.

Table A-26. KNTA_USER_SECURITY_INT interface table, continued

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KNTA_USERS_INT.
EXISTS_FLAG	O	VARCHAR2	Indicates whether or not the user already exists.
PROCESS_ PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_ STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME.
CREATED_BY_ USERNAME	I/O	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_ DATE	I/O	DATE	Indicates the transaction date. If left blank, the current date is used.
DEST_ CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME If both are left blank, the value is set to the user currently running the report.
DEST_ CREATION_	I/O	DATE	Indicates the date the record is created in the destination (PPM Center instance).

Table A-26. KNTA_USER_SECURITY_INT interface table, continued

Column	Usage	Data Type	Description
DATE			If left blank, the value is derived from CREATION_ DATE.
DEST_LAST_ UPDATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the set to the user currently running the report.
DEST_LAST_ UPDATE_DATE	I/O	DATE	Indicates the date that the security data was last updated. If left blank, the current date is used.
DEST_ENTITY_ UPD_DATE	I/O	DATE	Indicates the date that either the user data or security data was last updated. If left blank, the current date is used.
USER_ SECURITY_ID	I/O	NUMBER	Identifies a user security when removing a user from a security group. This is normally left blank. This is normally left blank and is derived from the KNTA_USER_SECURITY_S sequence.
DEST_USER_ SECURITY_ID	I/O	NUMBER	Identifies a user security. This is normally left blank. This is normally left blank and is derived from the KNTA_USER_SECURITY_S sequence.
USER_ID	I/O	NUMBER	Identifies the user. When creating users, this is left blank and the value is derived from the KNTA_USERS_S sequence. For existing users, this refers to the USER_ID column in KNTA_USERS.
DEST_USER_ID	I/O	NUMBER	Identifies the user. For existing users, this refers to the USER_ID column in KNTA_USERS. This is normally left blank and is derived from the KNTA_USERS_S sequence.
SECURITY_ GROUP_ID	I	NUMBER	Indicates the security group for the user. Required for ADD; not required for DROP.

Table A-26. KNTA_USER_SECURITY_INT interface table, continued

Column	Usage	Data Type	Description
SOURCE_ TYPE_CODE	I	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_WF.
SOURCE	I	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
LOGON_ IDENTIFIER	I	VARCHAR2	Identifies the ID used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, the LOGON_IDENTIFIER column must be populated. Otherwise, populate the USERNAME column.
USERNAME	I	VARCHAR2	Identifies the name used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_ NAME, the USERNAME column must be populated. Otherwise, populate the LOGON_IDENTIFIER column.
SECURITY_ GROUP_NAME	I	VARCHAR2	Specifies the SECURITY_GROUP_NAME in KNTA_ SECURITY_GROUPS.
USER_ SECURITY_ ACTION	I	VARCHAR2	Indicates the action for user security. Valid values are ADD or DROP.

RSC_RESOURCES_INT

The RSC_RESOURCES_INT interface table is used to provide resource attributes for new or existing resources.

Table A-27. RSC_RESOURCES_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ ID	I	Number	Uniquely identifies each transaction.

Table A-27. RSC_RESOURCES_INT interface table, continued

Column	Usage	Data Type	Description
			Initialized when interface loading start
PARENT_ TRANSACTION_ ID	I	Number	Provides the transaction ID (from KNTA_USERS_INT) of the parent table being imported. If any child table is being used, set the TRANSACTION_ID in KNTA_USERS_INT to this value. KNTA_USERS_INT.TRANSACTION_ID
PARENT_ TABLE_NAME	I	Varchar2 (30)	Identifies the table associated with this entity.
GROUP_ID	I	Number	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_ GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KNTA_USERS_INT.
EXISTS_FLAG	O	Varchar2 (1)	Indicates whether or not the user already exists.
PROCESS_ PHASE	O	Number	Indicates the current stage of the record as it is being processed.
PROCESS_ STATUS	O	Number	Indicates the current disposition of the record.
CREATED_BY	I/O	Number	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME.
CREATED_BY_ USERNAME	I/O	Varchar2 (200)	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_ DATE	I/O	Date	Indicates the transaction date. If left blank, the current date is used.

Table A-27. RSC_RESOURCES_INT interface table, continued

Column	Usage	Data Type	Description
DEST_ CREATED_BY	I/O	Number	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME. If both are left blank, the value is set to the user currently running the report.
DEST_ CREATION_ DATE	I/O	Date	Indicates the date the record is created in the destination (PPM Center instance). If left blank, the value is derived from CREATION_DATE.
DEST_LAST_ UPDATED_BY	I/O	Number	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the user currently running the report.
DEST_LAST_ UPDATE_DATE	I/O	Date	Indicates the date that the user data was last updated. If left blank, the current date is used.
DEST_ENTITY_ UPD_DATE	I/O	Date	Indicates the date that the user data was last updated. If left blank, the current date is used.
SOURCE_ TYPE_CODE	I	Varchar2 (30)	Specifies the type of external update. It is initialized when interface loading starts.
SOURCE	I	Varchar2 (100)	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
LOGON_ IDENTIFIER	I	Varchar2 (200)	Identifies the ID used for the logon. Thevalue should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the server.conf file. If LOGON_METHOD = LOGON_ID, the LOGON_IDENTIFIER column must be populated. Otherwise, populate the USERNAME column.
USERNAME	I	Varchar2 (200)	Identifies the name used for the logon. The value should be a valid USERNAME in KNTA_ USERS.

Table A-27. RSC_RESOURCES_INT interface table, continued

Column	Usage	Data Type	Description
			Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_NAME, the USERNAME column must be populated. Otherwise, populate the LOGON_IDENTIFIER column.
RESOURCE_ID	O	Number	Id of the resource derived from RSC_RESOURCES. This Id refers to the RESOURCE_ID column.
USER_ID	O	Number	Identifies the user. Derived from valid username or LOGON_IDENTIFIER.
PRIMARY_ ROLE_ID	O	Number	Derived by valid PRIMARY_ROLE_NAME.
PRIMARY_ ROLE_NAME	I	Varchar2 (200)	Existing role name in PPM Center.
TIME_SHEET_ POLICY_ID	O	Number	Derived from valid TIME_SHEET_POLICY_NAME.
TIME_SHEET_ POLICY_NAME	I	Varchar2 (200)	Existing time sheet policy name in PPM Center.
TM_ APPROVER_ID	O	Number	Derived from valid TM_APPROVER_USERNAME or TM_ APPROVER_IDENT
TM_ APPROVER_ USERNAME	I	Varchar2 (200)	Username of time sheet approver. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_NAME, the TM_ APPROVER_USERNAME column must be populated for the user import. Otherwise, populate the TM_ APPROVER_IDENT column.
TM_ APPROVER_ IDENT	I	Varchar2 (200)	LOGON_IDENTIFIER of time sheet approver. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, this column should be populated. Otherwise, populate the TM_APPROVER_ USERNAME column.
TM_BILLING_ APPROVER_ID	O	Number	Derived from valid TM_BILLING_APPROVER_ USERNAME or TM_BILLING_APPROVER_IDENT.
TM_BILLING_ APPROVER_ USERNAME	I	Varchar2 (200)	Username of time sheet billing approver. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_NAME,

Table A-27. RSC_RESOURCES_INT interface table, continued

Column	Usage	Data Type	Description
			the TM_BILLING_APPROVER_USERNAME column must be populated for the user import. Otherwise, populate the TM_BILLING_APPROVER_IDENT column.
TM_BILLING_APPROVER_IDENT	I	Varchar2 (200)	LOGON_IDENTIFIER of time sheet billing approver. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, this column should be populated. Otherwise, populate the TM_BILLING_APPROVER_USERNAME column.
TM_ENABLED_FLAG	I	char	Indicates if the time management is enabled for this resource.
TM_NOTIFS_ENABLED_FLAG	I	char	Indicates if this resource should recieve time management notifications.
TIME_SHEET_APPROVER_SEC_GRP_ID	O	Number	Derived by correct TM_APPROVER_SEC_GRP_NAME
TM_APPROVER_SEC_GRP_NAME	I	Varchar2 (200)	Existing security group name in PPM Center.
BILLING_APPROVER_SEC_GRP_ID	O	Number	Derived from valid BILLING_APPROVER_SEC_GRP_NAME.
BILLING_APPROVER_SEC_GRP_NAME	I	Varchar2 (200)	Existing security group name in PPM Center.
USER_DATA(N)	I	Varchar2 (200)	User data segment. N is 1 to 100.
VISIBLE_USERDATA(N)	I	Varchar2 (200)	User data segment. N is 1 to 100

KRSC_ORG_UNITS_INT

The KRSC_ORG_UNITS_INT interface table is used to define the attributes of the organization unit records being imported.

Table A-28. KRSC_ORG_UNITS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction.
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language. If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
EXISTS_FLAG	O	VARCHAR2	Indicates whether or not the organization unit already exists.
PROCESS_PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_USERNAME.
CREATED_BY_USERNAME	I/O	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction.

Table A-28. KRSC_ORG_UNITS_INT interface table, continued

Column	Usage	Data Type	Description
			This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_ DATE	I/O	DATE	Indicates the transaction date. If left blank, the current date is used.
DEST_ CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_ BY_USERNAME. Ig both are left blank, the value is set to the user currently running the report.
DEST_ CREATION_ DATE	I/O	DATE	Indicates the date the record is created in the destination (PPM Center instance). If left blank, the value is derived from CREATION_ DATE.
DEST_LAST_ UPDATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the set to the user currently running the report.
DEST_LAST_ UPDATE_DATE	I/O	DATE	Indicates the date that the organization data was last updated. If left blank, the current date is used.
DEST_ENTITY_ UPD_DATE	I/O	DATE	Indicates the date that either the organization or membership data was last updated. If left blank, the current date is used.
SOURCE	I	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
SOURCE_ TYPE_CODE	I	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_WF.
ORG_UNIT_ID	I/O	NUMBER	Identifies the organization unit ID.

Table A-28. KRSC_ORG_UNITS_INT interface table, continued

Column	Usage	Data Type	Description
			For new organization units, the value is derived from the KRSC_ORG_UNITS_S sequence. For existing organization units, if left blank, the value is derived from ORG_UNIT_NAME.
ORG_UNIT_NAME	I	VARCHAR2	Identifies the organization unit name.
PARENT_ORG_UNIT_ID	I/O	NUMBER	Identifies the parent unit ID for the organization unit. If left blank, the value is derived from PARENT_ORG_UNIT_NAME.
PARENT_ORG_UNIT_NAME	I	VARCHAR2	Identifies the parent unit name for the organization unit. If left blank, then the organization unit appears as a top level unit in the organization model.
MANAGER_ID	I/O	NUMBER	Identifies the manager associated with the organization unit. If left blank, the value is derived from MANAGER_USERNAME.
MANAGER_USERNAME	I	VARCHAR2	Specifies the name of the manager.
MANAGER_LOGON_IDENTIFIER	I	VARCHAR2	Specifies the ID of the manager. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, the MANAGER_LOGON_IDENTIFIER column must be populated. Otherwise, the MANAGER_USERNAME column must be populated.
DEPARTMENT_CODE	O	VARCHAR2	Specifies the code for the department.
DEPARTMENT_MEANING	I	VARCHAR2	Specifies the description of the department.
LOCATION_CODE	O	VARCHAR2	Specifies the code for the location.
LOCATION_MEANING	I	VARCHAR2	Specifies the description of the location.
CATEGORY_CODE	O	VARCHAR2	Specifies the code for the category.

Table A-28. KRSC_ORG_UNITS_INT interface table, continued

Column	Usage	Data Type	Description
CATEGORY_ MEANING	I	VARCHAR2	Specifies the description of the category.
ENABLED_FLAG	O	VARCHAR2	Indicates whether or not the organization unit is enabled upon import.
USER_DATA_ SET_CONTEXT_ ID	I	NUMBER	Sets the context identifier for the USER_DATA fields. Supply this or ORG_UNIT_USERNAME.

DISTINGUISH_ NAME	Required	VARCHAR2	Specifies the distinguished name for the organization unit in the following format:
			`OU=>SubOU1=>SubSubOU1`
			`OU=>SubOU2`

PARENT_ DISTINGUISH_ NAME	Required	VARCHAR2	Specifies the distinguished name for the parent organization unit in the following format:
			`OU=>SubOU1`
			`OU`

USER_DATA1 VISIBLE_USER_ DATA1 through USER_DATA20 VISIBLE_USER_ DATA20	I	VARCHAR2	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined. This information is not validated nor does it have a default value.

KRSC_ORG_UNIT_MEMBERS_INT

The KRSC_ORG_UNIT_MEMBERS_INT interface table is used to specify members for the organization units which were created through the organization unit interface tables.

Table A-29. KRSC_ORG_UNIT_MEMBERS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ ID	I	NUMBER	Uniquely identifies each transaction.

Table A-29. KRSC_ORG_UNIT_MEMBERS_INT interface table, continued

Column	Usage	Data Type	Description
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
EXISTS_FLAG	O	VARCHAR2	Indicates whether or not the organization unit already exists.
PROCESS_ PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_ STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME.
CREATED_BY_ USERNAME	I/O	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_ DATE	I	DATE	Indicates the transaction date. If left blank, the current date is used.
DEST_ CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME. If both are left blank, the value is set to the user currently running the report.
DEST_ CREATION_ DATE	I/O	DATE	Indicates the date the record is created in the destination (PPM Center instance). If left blank, the value is derived from CREATION_

Table A-29. KRSC_ORG_UNIT_MEMBERS_INT interface table, continued

Column	Usage	Data Type	Description
			DATE.
DEST_LAST_ UPDATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the user currently running the report.
DEST_LAST_ UPDATE_DATE	I/O	DATE	Indicates the date that the membership data was last updated. If left blank, the current date is used.
DEST_ENTITY_ UPD_DATE	I/O	DATE	Indicates the date that either the organization or membership data was last updated. If left blank, the current date is used.
SOURCE	I	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
SOURCE_ TYPE_CODE	I	VARCHAR2	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_WF.
ORG_UNIT_ MEMBER_ID	I/O	NUMBER	Identifies the organization unit member. This is normally left blank and is derived from the KRSC_ORG_UNIT_MEMBER_S sequence.
ORG_UNIT_ID	I/O	NUMBER	Identifies the organization unit ID. This is normally left blank and is derived from KRSC_ ORG_UNITS.
ORG_UNIT_ NAME	I	VARCHAR2	Identifies the parent unit name for the organization unit.
USER_ID	I/O	NUMBER	Identifies the user. For existing users, this refers to the USER_ID column in KNTA_USERS. This is normally left blank and is derived from the KNTA_USERS_S sequence.
USERNAME	I	VARCHAR2	Identifies the name used for the logon. The value should be a valid USERNAME in KNTA_USERS.

Table A-29. KRSC_ORG_UNIT_MEMBERS_INT interface table, continued

Column	Usage	Data Type	Description
			Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = USER_NAME, the USERNAME column must be populated for the user import. Otherwise, populate the LOGON_ID column.
LOGON_IDENTIFIER	I	VARCHAR2	Identifies the ID used for the logon. The value should be a valid USERNAME in KNTA_USERS. Depends on the LOGON_METHOD setting in the `server.conf` file. If LOGON_METHOD = LOGON_ID, the LOGON_ID column must be populated. Otherwise, populate the USERNAME column.
ORG_UNIT_DISTINGUISH_NAME	I	VARCHAR2	Specifies the distinguished name for the organization unit.

KWFL_STEP_TRANSITIONS_INT

This table is used to migrate the workflow definition between instances.

Table A-30. KWFL_STEP_TRANSITIONS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction. If left blank, the value is generated by the system.
PARENT_TRANSACTION_ID	I	NUMBER	Provides the transaction ID (from KCRT_REQUESTS_INT) of the parent table being imported.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID for each batch of imported users when running the User Open Interface report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. This value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT.
EXISTS_FLAG	I/O	VARCHAR2	If set to 'Y' then the record already exists in the database.

Table A-30. KWFL_STEP_TRANSITIONS_INT interface table, continued

Column	Usage	Data Type	Description
PROCESS_ PHASE	I/O	NUMBER	Indicates the current stage of the record as it is being processed. Indicates which phase a record is at: Pending, Deriving, Validating, Importing or Completed See "Process State Information" on page 182 for details.
PROCESS_ STATUS	I/O	NUMBER	Indicates the current status of the interface record: Pending, In Process, Errored or Completed See "Process State Information" on page 182 for details.
CREATED_BY_ USERNAME	I/O	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. This is used only if CREATED_BY is left blank. If both are left blank, the value is set to the user currently running the report.
CREATION_ DATE	I/O	DATE	Indicates the date that the record was created. If left blank, the current date is used.
DEST_ CREATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME. If both are left blank, the value is set to the user currently running the report.
DEST_ CREATION_ DATE	I/O	DATE	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. If left blank, the value is derived from CREATED_BY_ USERNAME. If both are left blank, the value is set to the user currently running the report.
DEST_LAST_ UPDATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user that last updated the data. If left blank, the value is set to the user currently running the report.
DEST_LAST_	I/O	DATE	Indicates the date that the user data was last updated.

Table A-30. KWFL_STEP_TRANSITIONS_INT interface table, continued

Column	Usage	Data Type	Description
UPDATE_DATE			If left blank, the current date is used.
STEP_ TRANSITION_ID	I/O	NUMBER	Identifier for the source, system-generated.
DEST_STEP_ TRANSITION_ID	I/O	NUMBER	Identifier for destination, system-generated.
TRANSITION_ TYPE_CODE	I/O	VARCHAR2	Defined by the lookup type WF_TRANSITION_TYPE. One of the following values: "SPECIFIC_VALUES", "OTHER_VALUES", "ALL_VALUES", "SPECIFIC_ ERROR", "OTHER_ERRORS", "ALL_ERRORS".
OPERATOR_ CODE	I/O	VARCHAR2	Defined by the lookup type WF_TRANS_OPERATOR. One of the following "=", "!=". If the validation is a text field with the numeric mask set, or if it is a date field, then the following values are also possible: "<", ">", "<=", ">=".
RESULT_VALUE	I/O	VARCHAR2	Indicates the result of the step. This is normally not displayed to the user; therefore it may be an ID or internal code.
VISIBLE_ RESULT_VALUE	I/O	VARCHAR2	Indicates the result of the step. This is the result value that a user normally sees.
FROM_ WORKFLOW_ STEP_ID	I/O	NUMBER	Identifies the source workflow step.
FROM_ WORKFLOW_ STEP_NAME	I/O	VARCHAR2	Provides the name of the source workflow step.
DEST_FROM_ WORKFLOW_ STEP_ID	I/O	NUMBER	Identifies of the source workflow step in destination.
TO_ WORKFLOW_ STEP_ID	I/O	NUMBER	Identifies the target workflow step.
TO_ WORKFLOW_ STEP_NAME	I/O	VARCHAR2	Provides the name of the target workflow step.
DEST_TO_ WORKFLOW_ STEP_ID	I/O	NUMBER	Identifies the target workflow step in destination.

Table A-30. KWFL_STEP_TRANSITIONS_INT interface table, continued

Column	Usage	Data Type	Description
DRAW_LABEL_POS	I/O	VARCHAR2	The relative position of where the label is displayed in the workflow layout. It is a real number between 0.0 and 1.0.
DRAW_JOINT_COORDS	I/O	VARCHAR2	A list of coordinates separated by ";". Each coordinate is a pair of integers, separated by a comma. Example: "100,200;300,400" defines two joints at positions (100,200) and (300,400). If NULL, a straight line is drawn from the source to the target step figure.
SOURCE_TYPE_CODE	I/O	VARCHAR2	Specifies the type of external update. This should be a left blank or have a value of INTERFACE_WF.
SOURCE	I/O	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.

KWFL_TRANSACTIONS_INT

The KWFL_TRANSACTIONS_INT interface table is used to set workflow decision steps and store the specific transaction that is to be performed at a workflow step for a package line or a request. This information includes the transaction type, package number, request number, and the workflow step.

Table A-31. KWFL_TRANSACTIONS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Uniquely identifies each transaction.
CREATION_DATE	I/O	DATE	The date/time for the step transaction cannot be set via this interface table. Instead, the value is taken from the LAST_UPDATE_DATE.
CREATED_USERNAME	I	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. Supply this or CREATED_BY.
CREATED_BY	I	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. Supply this or CREATED_USERNAME.

Table A-31. KWFL_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
LAST_UPDATE_DATE	I/O	DATE	This value is taken from the report execution date/time, which is transferred to the date/time for the step transaction. If left blank, the current date is used.
LAST_UPDATED_USERNAME	I	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. Supply this or LAST_UPDATED_BY.
LAST_UPDATED_BY	I/O	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. Supply this or LAST_UPDATED_USERNAME. If both are left blank, the value is derived from CREATED_USERNAME.
EVENT	I	VARCHAR2	Specifies the type of workflow transaction.
DATA_LANG	I	VARCHAR2	Specifies the language of the data being imported, so it can be properly validated in the respective language. If no value is provided, then the language context is the same as the system language of the PPM Center instance into which the data is being imported.
GROUP_ID	I	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence.
WORKFLOW_ENGINE_BATCH_ID	O	NUMBER	Specifies the batch in which the workflow engine carried out this transaction.
PROCESS_PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See "Process State Information" on page 182 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See "Process State Information" on page 182 for details.
SOURCE_	I	VARCHAR2	Specifies the type of external update.

Table A-31. KWFL_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
TYPE_CODE			This should be a left blank or have a value of INTERFACE_WF.
SOURCE	I	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
INSTANCE_ SOURCE_ TYPE_CODE	I	VARCHAR2	Indicates whether or not the transaction is for a package line (CR) or a request (IR).
INSTANCE_ SOURCE_SET_ NUMBER	I	VARCHAR2	Specifies the package number (PACKAGE_NUMBER from KDLV_PACKAGES) or request number (REQUEST_NUMBER from KCRT_REQUESTS). Supply this or INSTANCE_SOURCE_SET_ID.
INSTANCE_ SOURCE_SET_ ID	I	NUMBER	Specifies the package ID (PACKAGE_ID from KDLV_ PACKAGES) or request ID (REQUEST_ID from KCRT_REQUESTS). Supply this or INSTANCE_SOURCE_SET_ NUMBER.
INSTANCE_ SOURCE_LINE_ SEQ	I	NUMBER	Specifies the package line sequence number (SEQ from KDLV_PACKAGE_LINES). Supply this or INSTANCE_SOURCE_ID.
INSTANCE_ SOURCE_ID	I	NUMBER	Specifies the package line ID (PACKAGE_LINE_ID from KDLV_PACKAGE_LINES) or request ID (REQUEST_ID from KCRT_REQUESTS). Supply this or INSTANCE_SOURE_LINE_SEQ (for package lines) or INSTANCE_SOURCE_SET_ NUMBER (for requests).
WORKFLOW_ STEP_NAME	I	VARCHAR2	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS). Supply this or WORKFLOW_STEP_ID.
WORKFLOW_ STEP_SEQ	I	VARCHAR2	Specifies the sequence number of the workflow step. Supply this or WORKFLOW_STEP_ID. For subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth.

Table A-31. KWFL_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
RESULT_VALUE	I	VARCHAR2	Indicates the result of the step. This is normally not displayed to the user; therefore it may be an ID or internal code.
VISIBLE_ RESULT_VALUE	I	VARCHAR2	Indicates the result of the step. This is the result value that a user normally sees.
USER_ COMMENTS	I	VARCHAR2	Specifies comments for the transaction. Any comments are appended to the notes for the package or request.
DELEGATED_ TO_USERNAME	I	VARCHAR2	Specifies the USERNAME (from KNTA_USERS) for the user that the decision is being delegated to. Supply this or DELEGATED_TO_USER_ID.
DELEGATED_ TO_USER_ID	I	NUMBER	Specifies the USER_ID (from KNTA_USERS) for the user that the decision is being delegated to. Supply this or DELEGATED_TO_USERNAME.
SCHEDULE_ DATE	I	DATE	Indicates the date that the execution step is scheduled to run.
WORKFLOW_ID	O	NUMBER	Specifies the workflow that the package should follow.
WORKFLOW_ INSTANCE_ID	O	NUMBER	Specifies the instance ID.
WORKFLOW_ STEP_ID	I	NUMBER	Specifies the workflow step ID (WORKFLOW_STEP_ ID from KWFL_WORKFLOW_STEPS). Supply this, WORKFLOW_STEP_NAME, or WORKFLOW_STEP_SEQ.
WORKFLOW_ INSTANCE_ STEP_ID	O	NUMBER	Specifies the instance step ID.
CURRENT_ STEP_ TRANSACTION_ ID	O	NUMBER	Specifies the current step transaction ID.
APPROVALS_ REQUIRED_ CODE	O	NUMBER	Specifies the code for the required approvals.
EVENT_ GROUP_ID	O	NUMBER	Specifies the group ID for the event.

Table A-31. KWFL_TRANSACTIONS_INT interface table, continued

Column	Usage	Data Type	Description
CMD_ EXECUTION_ SCHD_TASK_ID	O	NUMBER	Specifies the execution step and the command that has been scheduled. This will specify the identified for the scheduled task.
TO_ WORKFLOW_ STEP_SEQ	I	VARCHAR2	Specifies the sequence number of the workflow step for the step that the package line or request should transition to. Supply this, TO_WORKFLOW_STEP_ID, or TO_ WORKFLOW_STEP_NAME.
TO_ WORKFLOW_ STEP_NAME	I	VARCHAR2	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS) for the step that the package line or request should transition to. Supply this, TO_WORKFLOW_STEP_SEQ, or TO_ WORKFLOW_STEP_ID.
TO_ WORKFLOW_ STEP_ID	I	NUMBER	Specifies the workflow step ID (WORKFLOW_STEP_ ID from KWFL_WORKFLOW_STEPS) for the step that the package line or request should transition to. Supply this, TO_WORKFLOW_STEP_NAME, or TO_ WORKFLOW_STEP_SEQ.

KWFL_TXN_INT.INSERT_ROW

The Workflow Transaction Open Interface can be used for different types of transactions. Different parameters are required or optional depending upon the type of transaction. The values are established using the `INSERT_ROW` procedure in the `KWFL_TXN_INT` package.

Table A-32. KWFL_TXN_INT.INSERT_ROW parameters

Parameter	Usage	Data Type	Description
P_EVENT	I	VARCHAR2	Specifies the type of workflow transaction. The value depends on the type of transaction.
P_GROUP_ID	I/O	NUMBER	Groups all the records that should be processed at the same time. Use only one GROUP_ID each time you run a report. Derived from the KNTA_INTERFACE_GROUPS_S sequence. If left blank, the value is generated by the system.

Table A-32. KWFL_TXN_INT.INSERT_ROW parameters, continued

Parameter	Usage	Data Type	Description
P_CREATED_USERNAME	I	VARCHAR2	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction.
P_SOURCE	I	VARCHAR2	Specifies the source of the information. This information is not validated during an import. For example, the name of the third-party application or a value of CONVERSION.
P_REQUEST_NUMBER	I	VARCHAR2	Identifies the request.
P_PACKAGE_NUMBER	I	VARCHAR2	Identifies the package number.
P_PACKAGE_LINE_SEQ	I	NUMBER	Provides the identifier for a package line. Derived from the KDLV_PACKAGE_LINES_S sequence.
P_WORKFLOW_STEP_NAME	I	VARCHAR2	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS).
P_WORKFLOW_STEP_SEQ	I	VARCHAR2	Specifies the sequence number of the workflow step. For subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth.
P_VISIBLE_RESULT_VALUE	I	VARCHAR2	Indicates the result of the step. This is the result value that a user normally sees.
P_USER_COMMENTS	I	VARCHAR2	Specifies comments for the transaction. Any comments are appended to the notes for the package or request.
P_DELEGATED_TO_USERNAME	I	VARCHAR2	Specifies the USERNAME (from KNTA_USERS) for the user that the decision is being delegated to.
P_SCHEDULE_DATE	I	DATE	Indicates the date that the execution step is scheduled to run.
P_TO_WORKFLOW_STEP_NAME	I	VARCHAR2	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS) for the step that the package line or request should transition to.
P_TO_WORKFLOW_STEP_SEQ	I	VARCHAR2	Specifies the sequence number of the workflow step for the step that the package line or request should transition to.

Table A-32. KWFL_TXN_INT.INSERT_ROW parameters, continued

Parameter	Usage	Data Type	Description
O_ MESSAGE_ TYPE	O	NUMBER	Indicates what type of error occurred. Valid values (from KNTA_Constant) are: • SUCCESS - No error occurred • USER_ERR - User error • INTERNAL_ERR - An internal error occurred • WARNING - A non-fatal warning is returned
O_ MESSAGE_ NAME	O	VARCHAR2	Specifies the internal message name of the error that was returned. This is used mainly for debugging purposes.
O_MESSAGE	O	VARCHAR2	Provides the error message.

Appendix B: LDAP Authentication

PPM Center uses simple authentication to authenticate against any LDAP v.3 (or later) compliant LDAP server.

The authentication steps involve:

1. The PPM Server binds to the LDAP server using the credentials supplied in the KINTANA_LDAP_ID and KINTANA_LDAP_PASSWORD server attributes.

   This step is optional. PPM Center does an anonymous authentication if a password is not supplied in `server.conf`.

   For more information on the `server.conf` file, see the *Installation and Administration Guide*.

2. PPM Center tries to obtain the distinguished name of the user by supplying a search filter to the LDAP server in the form uid=<*username*> (where <*username*> is the user ID on the LDAP server).

   Here the attribute uid could vary from one LDAP server to another depending on the information supplied in the `LdapAttribute.conf` file.

3. If PPM Center obtains a unique distinguished name, then it tries to rebind to the LDAP server using the distinguished name and the password supplied by the user.

If more than one LDAP server has been specified in the LDAP_URL server attribute, PPM Center tries to authenticate against all of them until it succeeds. If the referral option has been enabled, then PPM Center also queries the referral server for authentication if the user is not present in primary server.

PPM Center also supports LDAP authentication over SSL by using passwords. To enable the SSL option, set the LDAP_SSL_PORT server attribute to the SSL port of the LDAP server.

Appendix C: Process State Information

Overview

As the reports are run, the program processes the interface tables and provides information on both the phase and status (state) of the execution.

PROCESS_PHASE

The PROCESS_PHASE column indicates the current phase of the record as it is being processed.

A record goes through the following phases as it is processed. The initial value should be set at one.

- 1 - Pending
- 2 - Derivation
- 3 - Validation
- 6 - Final Validation
- 7 - Batch Processing
- 5 - Completed

PROCESS_STATUS

The PROCESS_STATUS column indicates the current status of the record as it is being processed.

A record could have the following statuses as it is processed. The initial value should be set at one.

- 1 - Pending
- 2 - In Process
- 3 - Error
- 7 - Completed

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Open Interface Guide and Reference (Project and Portfolio Management Center 9.40)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to your_IE_team_PDL@hpe.com.

We appreciate your feedback!