Mercury IT Governance Center™

Open Interface Guide and Reference

Version: 6.0

MERCURY™

Mercury
379 North Whisman Road
Mountain View, CA 94043
Tel: (650) 603-5200
Toll Free: (800) TEST-911
Customer Support: (877) TEST-HLP
Fax: (650) 603-5300

If you have any comments or suggestions regarding this document, please send email to documentation@mercury.com.

Publication Number: ITG60OpenInterface1104A

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction

In This Chapter:

About This Document

In addition to the standard interface for processing requests, packages, workflows, users, organizations, and other transactions, Mercury IT Governance Center includes this open interface for performing a limited set of data integration with third-party products.

This document describes the open interface. Each chapter addresses the data model and processes used to import the following entities from an existing database:

- Chapter 1, *Introduction,* on page 11

  Describes the document and provides an overview of the open interface.

- Chapter 2, *User Open Interface,* on page 15

  Describes generating or updating users from a database or Lightweight Directory Access Protocol (LDAP) server.

- Chapter 3, *Organization Unit Open Interface,* on page 45

  Describes importing an organization model into Mercury IT Governance Center.

- Chapter 4, *Request Open Interface,* on page 67

  Describes request generation.

- Chapter 5, *Package Open Interface,* on page 87

  Describes package and package line generation.

- Chapter 6, *Workflow Transaction Open Interface,* on page 107

  Describes workflow transactions for package lines and requests.

- Appendix A: *Open Interface Data Models* on page 133

  Describes the interface tables used by the open interface.

- Appendix B: *LDAP Authentication* on page 195

  Describes the steps involved in authentication on an LDAP server.

- Appendix C: *Process State Information* on page 197

  Describes the states associated with the PROCESS_PHASE and PROCESS_STATUS columns used in many of the interface tables.

Who Should Read This Document

This document is intended for the following audiences:

- User administrators
- System or instance administrators
- Application developers or configurators

For More Information

For information about audience types, see the *Guide to Documentation*.

Prerequisite and Related Documents

Prerequisite documents include:

- *Getting Started*
- *Key Concepts*
- *Using the Workbench*
- *Reports Guide and Reference*

Supplemental documentation includes:

- *Mercury Demand Management User's Guide*
- *Mercury Change Management User's Guide*
- *Mercury Resource Management User's Guide*
- *Mercury Demand Management: Configuring a Request Resolution System*
- *System Administration Guide and Reference*
- *Security Guide and Reference*
- *Commands, Tokens, and Validation Guide and Reference*

For More Information

For information about these documents and how to access them, see the *Guide to Documentation*.

Overview of the Open Interface

The open interface allows integration of data from third-party products with key Mercury IT Governance 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 Mercury IT Governance 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 Mercury IT Governance Center database. Data added to these interface tables is validated and eventually imported into standard Mercury IT Governance Center tables. This generates entities that can be processed using Mercury IT Governance Center.

Chapter 2 User Open Interface

In This Chapter:

Overview

Mercury IT Governance Center™ includes an open interface for importing user information. This open interface can import user models from third-party systems including LDAP databases, internally developed systems, or human resources systems. Using this interface, you can periodically synchronize the user model in Mercury IT Governance Center with the authoritative data source within their company.

In general, the synchronization process involves importing user attributes of the various users. Using the interface tables listed in the following section (*The Data Model*), the User Open Interface supports:

- Simple imports
- LDAP imports

For More Information

For information on mapping your user model, see the *Mercury Resource Management User's Guide*.

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

See Appendix B: *LDAP Authentication* on page 195 to review the LDAP authentication process.

The Data Model

The following interface tables are used by the User Open Interface.

- *KNTA_USERS_INT*
- *KNTA_USER_SECURITY_INT*

These interface tables are described, in their entirety, in Appendix A: *Open Interface Data Models* on page 133. The columns that can be used when importing users are detailed in the appropriate step within this chapter.

Performing a Simple Import

Step One: 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 Four: Start the Import* on page 29, you can 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 as described in *Step Two: Add/Drop Security Groups* on page 18.
- 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 Two: Add/Drop Security Groups

If you decide that you need to use the ADD/DROP option, first populate the KNTA_USER_SECURITY_INT interface table. This table needs to include a record for each desired security group action for each user. The input columns for this interface table are listed in *Table 2-1* on page 18.

The population can be done through any means supported by the Oracle database. Standard mechanisms include the use of SQL*Loader to load the contents of an ASCII file or direct Oracle database-to-database communication through database links.

Example of a Change in Security Groups

User A and User B exist as users of Mercury IT Governance 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, first 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 input options

Column	Usage	Data Type	Description
TRANSACTION_ID	Required	NUMBER	Provides a unique identifier for each transaction.
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.

Table 2-1. KNTA_USER_SECURITY_INT interface table input options [continued]

Column	Usage	Data Type	Description
PARENT_TABLE_NAME	Optional	VARCHAR2 (30)	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.
CREATED_BY	Option varies	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	Option varies	VARCHAR2 (30)	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 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 (Mercury IT Governance Center instance). If left blank, the value is derived from CREATION_DATE.

Table 2-1. KNTA_USER_SECURITY_INT interface table input options [continued]

Column	Usage	Data Type	Description
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.
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	Requirement varies	NUMBER	Indicates the security group for the user. Required for ADD; not required for DROP.

Table 2-1. KNTA_USER_SECURITY_INT interface table input options [continued]

Column	Usage	Data Type	Description
SOURCE_TYPE_CODE	Optional	VARCHAR2 (30)	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	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	Requirement varies	VARCHAR2 (30)	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	Requirement varies	VARCHAR2 (30)	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 (40)	Specifies the SECURITY_GROUP_ NAME in KNTA_SECURITY_GROUPS.
USER_SECURITY_ACTION	Required	VARCHAR2 (30)	Indicates the action for user security. Valid values are ADD or DROP.

Step Three: Populate the Security Data

During a simple import, specific columns in the KNTA_USERS_INT interface tables must be populated. The input columns for this interface table are listed in the following table (*Table 2-2*).

This population can be done through any means supported by the Oracle database (such as SQL*Loader or direct Oracle database-to-database communication).

Warning User data is not validated during import.

Table 2-2. KNTA_USERS_INT interface table input options

Column	Usage	Data Type	Description
TRANSACTION_ID	Required	NUMBER	Provides a unique identifier for each transaction. See also PARENT_TRANSACTION_ID in KNTA_USER_SECURITY_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.
CREATED_BY	Option varies	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	Option varies	VARCHAR2 (30)	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.

Table 2-2. KNTA_USERS_INT interface table input options [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 (Mercury IT Governance 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 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.

Table 2-2. KNTA_USERS_INT interface table input options [continued]

Column	Usage	Data Type	Description
USERNAME	Requirement varies	VARCHAR2 (30)	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 (40)	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	Optional	NUMBER	Specifies the number of days before the current password expires.
PASSWORD_EXPIRATION_ DATE	Optional	DATE	Specifies the date when the password should expire.
EMAIL_ADDRESS	Optional	VARCHAR2 (80)	Specifies the email address of the user.
FIRST_NAME	Requirement varies	VARCHAR2 (30)	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	Requirement varies	VARCHAR2 (30)	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	Optional	DATE	Specifies the user's start date.
END_DATE	Optional	DATE	Specifies the user's end date.
DEFAULT_ACCELERATOR_ ID	Optional	NUMBER	Sets the context identifier for the USER_ DATA fields.
SOURCE_TYPE_CODE	Optional	VARCHAR2 (30)	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_WF.

Table 2-2. KNTA_USERS_INT interface table input options [continued]

Column	Usage	Data Type	Description
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.
USER_DATA_SET_ CONTEXT_ID	Requirement varies	NUMBER	Sets the context identifier for the USER_ DATA fields. Supply either this or USERNAME.
USER_DATA1 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USER_DATA20	Requirement varies	VARCHAR2 (200)	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.
AUTHENTICATION_MODE	Requirement varies	VARCHAR2 (30)	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 (1)	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	Optional	VARCHAR2 (4)	Specifies the position where the shortcut bar is displayed. If left blank, the default value is supplied.
SAVE_WINDOW_BOUNDS_ FLAG	Optional	VARCHAR2 (1)	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	Optional	NUMBER	Specifies the default height of the screen manager window. If left blank, the default value is supplied.

Table 2-2. KNTA_USERS_INT interface table input options [continued]

Column	Usage	Data Type	Description
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. If left blank, the default value is supplied.
REUSE_INTERNAL_ FRAME_FLAG	Optional	VARCHAR2 (1)	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	Optional	VARCHAR2 (1)	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	Optional	VARCHAR2 (1)	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	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 (1)	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.

Table 2-2. KNTA_USERS_INT interface table input options [continued]

Column	Usage	Data Type	Description
SHOW_CHANGE_ WARNINGS_FLAG	Optional	VARCHAR2 (1)	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	Optional	VARCHAR2 (1)	Indicates whether or not cancelled package lines are displayed in the packages screen. If left blank, the default value is supplied.
DEFAULT_BROWSER	Optional	VARCHAR2 (300)	Specifies the default browser for the user.
COMPANY	Optional	VARCHAR2 (1)	Identifies the company.
DOMAIN	Optional	VARCHAR2 (80)	Identifies the Windows domain. Used for Exchange server (NTLM) authentication.
LOGON_IDENTIFIER	Requirement varies	VARCHAR2 (30)	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	Optional	VARCHAR2 (300)	Specifies the user’s phone number on the resource page.
COST_RATE	Optional	NUMBER	Specifies the user’s cost rate.
WORKLOAD_CAPACITY	Optional	NUMBER	Specifies the user’s workload capacity (in percentage) on the resource page.
MAX_ROWS_PORTLETS	Optional	NUMBER	Specifies the maximum number of results to be displayed on the maximized portlet.
DEPARTMENT_MEANING	Required	VARCHAR2 (80)	Specifies the description of the department.

Table 2-2. KNTA_USERS_INT interface table input options [continued]

Column	Usage	Data Type	Description
LOCATION_MEANING	Required	VARCHAR2 (80)	Specifies the description of the location.
MANAGER_USER_ID	Requirement varies	NUMBER	Specifies the user ID of the manager. Used if both MANAGER_USERNAME and MANAGER_LOGON_IDENTIFIER are left blank.
MANAGER_USERNAME	Requirement varies	VARCHAR2 (80)	Specifies the name of the manager. Used if MANAGER_LOGON_ IDENTIFIER is left blank.
MANAGER_LOGON_ IDENTIFIER	Requirement varies	VARCHAR2 (300)	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_ MEANING	Required	VARCHAR2 (80)	Specifies the description of the user's category.
RESOURCE_TITLE_ MEANING	Required	VARCHAR2 (80)	Specifies the description of the user's title.

Note

Additional columns in KNTA_USER_SECURITY_INT must be populated when using the ADD/DROP security group action. For more information, see *Step Two: Add/ Drop Security Groups* on page 18.

Step Four: 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 Mercury IT Governance 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. Expand **Reports** from the menu bar.

   The list of report options appears.

2. Select **Submit New Report.**

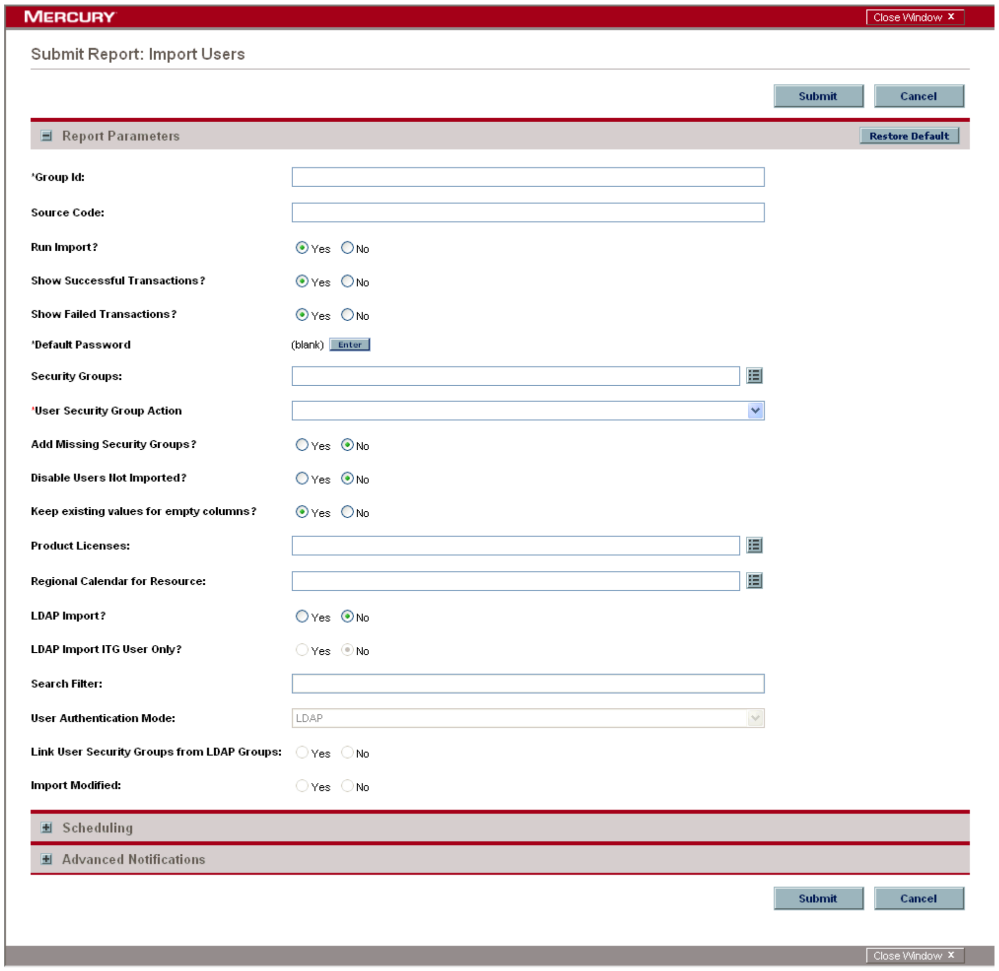
   The Submit New Report page appears.

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 Requests window appears.



Submit Report: Import Users

Report Parameters

*Group Id:	
Source Code:	
Run Import?	[✓ Yes] No
Show Successful Transactions?	[✓ Yes] No
Show Failed Transactions?	[✓ Yes] No
*Default Password	(blank)
Security Groups:	
*User Security Group Action	
Add Missing Security Groups?	Yes [✓ No]
Disable Users Not Imported?	Yes [✓ No]
Keep existing values for empty columns?	[✓ Yes] No
Product Licenses:	
Regional Calendar for Resource:	
LDAP Import?	Yes [✓ No]
LDAP Import ITG User Only?	Yes [✓ No]
Search Filter:	
User Authentication Mode:	LDAP
Link User Security Groups from LDAP Groups:	Yes No
Import Modified:	Yes No

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

   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 differents, 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.

Warning

USER_DATA for users is not validated as part of an import.

Note

Remember that you can test the process by setting the Run Import field to **No**.

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. In the case of 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 users that were successfully imported.
Show Failed Transactions	Indicates whether or not to show users that were not successfully imported.
Default Password	Specifies a default 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	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.
Product Licenses	Selects the product license the imported user(s) will have.

Field Name	Description
Regional Calendar For Resource	Selects the regional calendar for the imported users. 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).
LDAP Import ITG User Only	Indicates whether or not to perform LDAP import of only Mercury IT Governance Center users.
Search Filter	Specifies the search filter using syntax of the conditions on Mercury IT Governance Center commands.
User Authentication Mode	Selects a user authentication mode. (LDAP or NTLM only)
Link User Security Groups from LDAP Groups	Indicates whether or not to link security groups from LDAP Groups. (LDAP or NTLM only)
Import Modified	Indicates whether or not the import can be modified. (LDAP or NTLM only)

Note

Required fields are denoted with a red asterisk. These fields may vary based on your selections.

For More Information

The `server.conf` file is described in the *System Administration Guide and Reference*.

Examples of Search Filter Values

The following filter returns objects that have cn = Babs Jensen:

```
(cn=Babs Jensen)
```

The following filter returns objects that do not have cn = Tim Howes:

```
(!(cn=Tim Howes))
```

The following filter returns all objects belonging to object class Person and who are either sn = Jensen or whose cn contains Babs J (this would include Babs J or Babs Jen):

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

The following filter returns objects with o like univ%of%mich% (using sql concepts):

```
(o=univ*of*mich*)
```

This filter returns all objects with ou= Development and that have uid=test1 or uid=test2 or uid=test3

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

Step Five: Verify Successful Completion

Confirm that the import process completed successfully. Click **View Report** to review the results of the import.

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

This report identifies any errors with the import. If errors are present, start your troubleshooting by referring to *Correcting Failures* on page 44.

Keep in mind that all interface tables are automatically cleared by the purge service. The purging process depends on the following attributes 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

The `server.conf` file is described in the *System Administration Guide and Reference*.

Performing an LDAP Import

Step One: 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 Six: Start the Import* on page 40, you can 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 as described in somewhere.

  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 Two: Add/Drop Security Groups

This is an optional step. If you decide that you want to use the ADD/DROP option, follow the procedure detailed in *Step Two: Add/Drop Security Groups* on page 18.

Step Three: 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 ITG User Only is set to **Yes.** However, it is not a required step. LDAP Import ITG User Only can be set to **No,** and the Search Filter field used to query for the attribute of your choice.

If LDAP Import ITG 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 execute a command locally on the server machine. The command is located in the `<`*`ITG_Home`*`>/bin` directory (where `<`*`ITG_Home`*`>` represents the installation path for Mercury IT Governance Center) and should be run using a bash shell. This command needs to be run by an LDAP user who has privileges to modify the LDAP schema.

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 Mercury IT Governance Center-only mode if they have a password defined in Mercury IT Governance Center. Also, if the server is in Mercury IT Governance Center-only mode, Mercury IT Governance Center passwords can be set for LDAP users. These passwords are not required.

For More Information

The `server.conf` file is described in the *System Administration Guide and Reference*.

Step Four: Map LDAP Attribute

It is possible to map the attributes on the LDAP server to attributes used by the Mercury IT Governance Server. Some of this mapping occurs by default, but it is possible to exercise greater control by mapping the attributes in the following file (where `<ITG_Home>` represents the installation path for Mercury IT Governance Center):

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

Note

Sample files for mapping to a Netscape Directory Server and an Active Directory Server can be found in the same directory. The default mapping is for a Netscape Directory Server.

For More Information

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

Step Five: Configure the Mercury IT Governance Server

Several Mercury IT Governance Server parameters need to be considered when performing a user import from an LDAP server. These attributes are set in the `server.conf` file located in the following directory (where `<`*`ITG_Home`*`>` represents the installation path for Mercury IT Governance Center):

```
<ITG_Home>/server.conf
```

After changing these parameters, stop and restart the Mercury IT Governance Server.

For More Information

The `server.conf` file is described in the *System Administration Guide and Reference*.

Step Six: 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 Mercury IT Governance 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. Expand **Reports** from the menu bar.

   The list of report options appears.

2. Select **Submit New Report.**

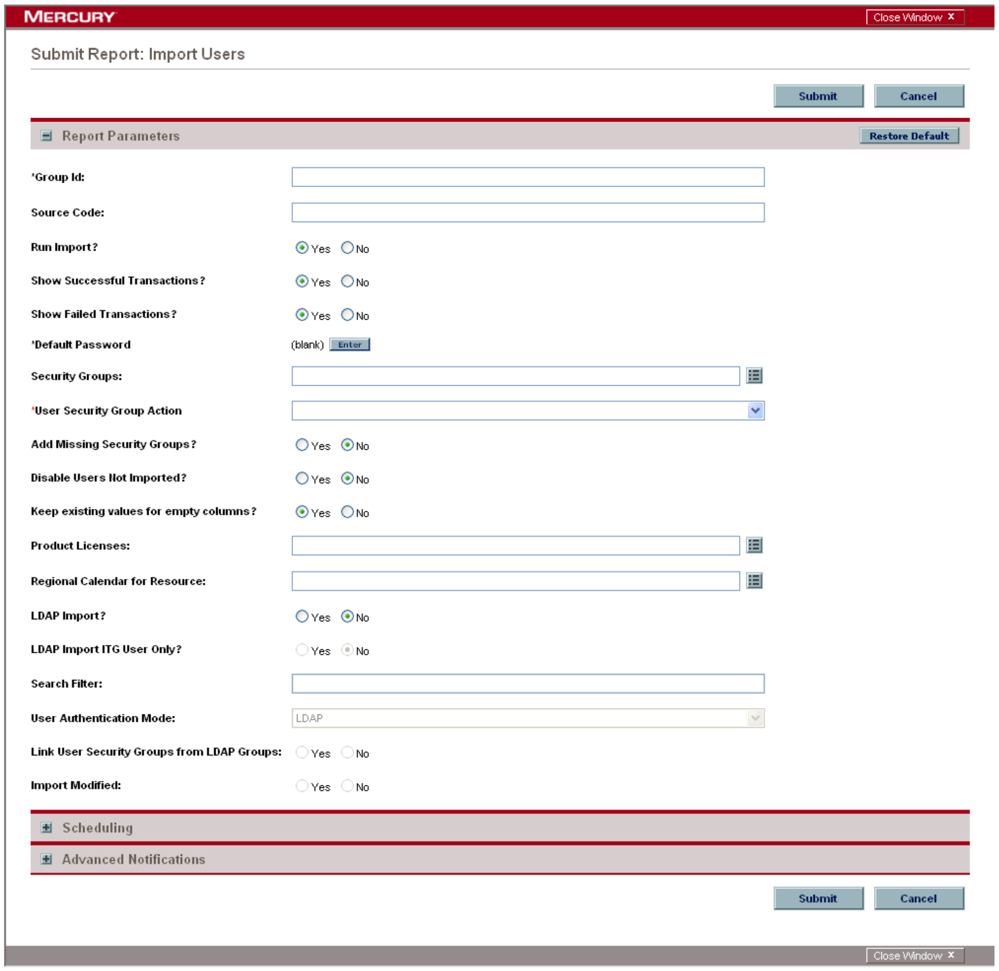
   The Submit New Report page appears.

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 Requests window appears.



Submit Report: Import Users

*Group Id:	
Source Code:	
Run Import?	[✓ Yes] No
Show Successful Transactions?	[✓ Yes] No
Show Failed Transactions?	[✓ Yes] No
*Default Password	(blank) Enter
Security Groups:	
*User Security Group Action	
Add Missing Security Groups?	Yes [✓ No]
Disable Users Not Imported?	Yes [✓ No]
Keep existing values for empty columns?	[✓ Yes] No
Product Licenses:	
Regional Calendar for Resource:	
LDAP Import?	Yes [✓ No]
LDAP Import ITG User Only?	Yes [✓ No]
Search Filter:	
User Authentication Mode:	LDAP
Link User Security Groups from LDAP Groups:	Yes No
Import Modified:	Yes No

5. Complete the fields as described in step 5 on page 31.

   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.

Warning

USER_DATA for users is not validated as part of an import.

Note

Remember that you can test the process by setting the Run Import field to **No**.

Step Seven: Verify Successful Completion

Confirm that the import process completed successfully. Click **View Report** to review the results of the import.

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

This report identifies any errors with the import. If errors are present, start your troubleshooting by referring to *Correcting Failures* on page 44.

Keep in mind that all interface tables are automatically cleared by the purge service. The purging process depends on the following attributes 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

The `server.conf` file is described in the *System Administration Guide and Reference*.

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, corrective actions are required. The first step involves examining the audit report from the open interface report 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 failures may occur due to a mapping problem between the source data and existing data in Mercury IT Governance Center.

Other failures may be due to 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.

Additionally, failures could sometimes be due to a large volume of data being imported. If you suspect that this is the problem, confirm this hyphothesis by importing a smaller number of records, then checking to see if the error messsage persists.

Note

During the initial implementation of the open interface, the mapping between the third-party source and Mercury IT Governance Center should be thoroughly reviewed and the load program(s) thoroughly tested in a testing instance.

Additionally, it is good practice to monitor executions of the open interface and periodically monitor the import of desired data into Mercury IT Governance Center.

Chapter

3

Organization Unit Open Interface

In This Chapter:

Overview

Mercury IT Governance Center includes an interface for importing organization information. This open interface can import organizational models from third-party systems including LDAP databases, internally developed organization modeling systems, or human resources systems. Using this interface, you can periodically synchronize the organizational model in Mercury IT Governance Center with the authoritative data source within your company.

In general, the synchronization process involves importing organization unit attributes of the various resources. Using the interface tables listed in the following section (*The Data Model*), the Organization Unit Open Interface supports:

- Simple imports
- LDAP imports

For More Information

For information on mapping your organization model, see the *Mercury Resource Management User's Guide*.

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

See Appendix B: *LDAP Authentication* on page 195 to review the LDAP authentication process.

The Data Model

The following interface tables are used by the Organization Unit Open Interface:

- *KRSC_ORG_UNITS_INT*
- *KRSC_ORG_UNIT_MEMBERS_INT*
- *KNTA_USERS_INT*

These interface tables are described, in their entirety, in Appendix A: *Open Interface Data Models* on page 133. The columns that can be used when importing organizational models are detailed in the appropriate step within this chapter.

Performing a Simple Import

Step One: Load the Users

Load the users or resources into Mercury IT Governance Center by either:

- Running the Import Users report.

  For details, see *Performing a Simple Import* on page 17.

- Populating 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* on page 22.

The population can be done through any means supported by the Oracle database. Standard mechanisms include the use of SQL*Loader to load the contents of an ASCII file or direct Oracle database-to-database communication through database links.

Step Two: Populate the Interface Tables

During a simple import, specific columns in the KRSC_ORG_UNITS_INT and KRSC_ORG_UNIT_MEMBERS_INT interface tables must be populated. The input columns for each are listed in the following table (*Table 3-1*) as well as *Table 3-2* on page 52.

This population can be done through any means supported by the Oracle database (such as the SQL*Loader or direct Oracle database-to-database communication).

Warning User data is not validated during import.

Table 3-1. KRSC_ORG_UNITS_INT interface table input options

Column	Usage	Data Type	Description
TRANSACTION_ID	Required	NUMBER	Provides a unique identifier for each transaction.
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.
CREATED_BY	Option varies	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	Option varies	VARCHAR2 (30)	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 3-1. KRSC_ORG_UNITS_INT interface table input options [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. 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 (Mercury IT Governance 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 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	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.
SOURCE_TYPE_CODE	Optional	VARCHAR2 (30)	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.

Table 3-1. KRSC_ORG_UNITS_INT interface table input options [continued]

Column	Usage	Data Type	Description
ORG_UNIT_NAME	Required	VARCHAR2 (30)	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	Optional	VARCHAR2 (30)	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	Requirement varies	VARCHAR2 (30)	Specifies the name of the manager.
MANAGER_LOGON_ IDENTIFIER	Requirement varies	VARCHAR2 (30)	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_MEANING	Optional	VARCHAR2 (80)	Specifies the description of the department.
LOCATION_MEANING	Optional	VARCHAR2 (80)	Specifies the description of the location.
CATEGORY_MEANING	Optional	VARCHAR2 (80)	Specifies the description of the category.

Table 3-1. KRSC_ORG_UNITS_INT interface table input options [continued]

Column	Usage	Data Type	Description
USER_DATA_SET_ CONTEXT_ID	Requirement varies	NUMBER	Sets the context identifier for the USER_ DATA fields. Supply either this or ORG_UNIT_ USERNAME.
USER_DATA1 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USER_DATA20	Requirement varies	VARCHAR2 (200)	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 3-2. KRSC_ORG_UNIT_MEMBERS_INT interface table input options

Column	Usage	Data Type	Description
TRANSACTION_ID	Required	NUMBER	Provides a unique identifier for each transaction.
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.
CREATED_BY	Option varies	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 (30)	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. 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 (Mercury IT Governance Center instance). If left blank, the value is derived from CREATION_DATE.

Table 3-2. KRSC_ORG_UNIT_MEMBERS_INT interface table input options [continued]

Column	Usage	Data Type	Description
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 membership 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	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.
SOURCE_TYPE_CODE	Optional	VARCHAR2 (30)	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	Optional	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	Optional	NUMBER	Identifies the organization unit ID. This is normally left blank and is derived from KRSC_ORG_UNITS.
ORG_UNIT_NAME	Required	VARCHAR2 (30)	Identifies the parent unit name for the organization unit.
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.

Table 3-2. KRSC_ORG_UNIT_MEMBERS_INT interface table input options [continued]

Column	Usage	Data Type	Description
USERNAME	Requirement varies	VARCHAR2 (30)	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	Requirement varies	VARCHAR2 (30)	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.

Step Three: Start the Import

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

The Run ITG 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 Mercury IT Governance 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 ITG Organization Unit Interface report:

1. Expand **Reports** from the menu bar.

   The list of report options appears.

2. Select **Submit New Report.**

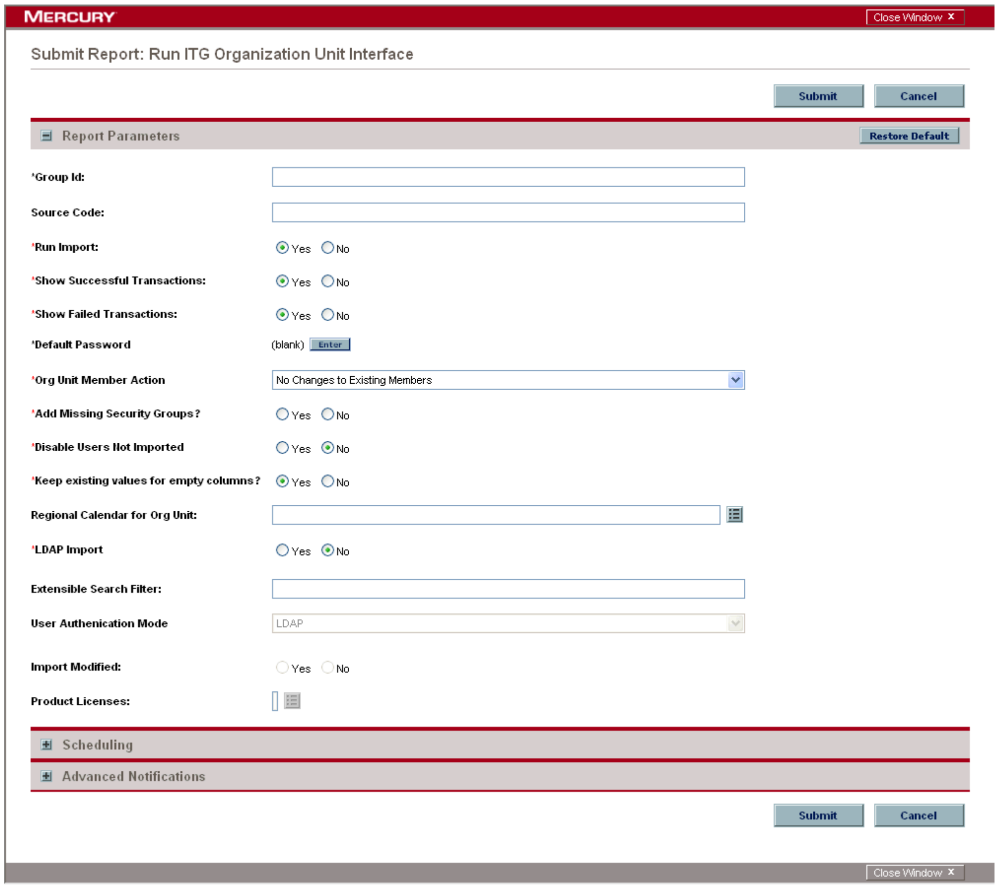
   The Submit New Report page appears.

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

   The page is updated showing the list of administrative reports.

4. Select **Run ITG Organization Unit Interface.**

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



Submit Report: Run ITG Organization Unit Interface

Report Parameters

*Group Id:	
Source Code:	
*Run Import:	[✓ Yes] No
*Show Successful Transactions:	[✓ Yes] No
*Show Failed Transactions:	[✓ Yes] No
*Default Password	(blank)
*Org Unit Member Action	No Changes to Existing Members
*Add Missing Security Groups?	Yes No
*Disable Users Not Imported	Yes [✓ No]
*Keep existing values for empty columns?	[✓ Yes] No
Regional Calendar for Org Unit:	
*LDAP Import	Yes [✓ No]
Extensible Search Filter:	
User Authenication Mode	LDAP
Import Modified:	Yes No
Product Licenses:	

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

   The Run ITG Organization Unit Interface report has several parameters for controlling the behavior of the program execution. Pay special attention to:

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

Note

Remember that you can test the process by setting the Run Import field to **No.**

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. In the case of 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.

Field Name	Description
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.
Regional Calendar 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 Mercury IT Governance Center commands.
User Authentication Mode	Selects a user authentication mode. (LDAP or NTLM only)

Field Name	Description
Import Modified	Indicates whether or not the import can be modified. (LDAP or NTLM only)
Product Licenses	Establishes the licensing for the imported users.

Note

Required fields are denoted with a red asterisk. These fields may vary based on your selections.

For More Information

For information about extensible search filters, see *Examples of Search Filter Values* on page 34.

The `server.conf` file is described in the *System Administration Guide and Reference*.

Step Four: Verify Successful Completion

Confirm that the import process completed successfully. Click **View Report** to review the results of the import.

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

This report identifies any errors with the import. If errors are present, start your troubleshooting by referring to *Correcting Failures* on page 65.

Keep in mind that all interface tables are automatically cleared by the purge service. The purging process depends on the following attributes 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

The `server.conf` file is described in the *System Administration Guide and Reference*.

Performing an LDAP Import

Step One: Map the LDAP Attributes

It is possible to map the attributes on the LDAP server to attributes used by the Mercury IT Governance Server. Some of this mapping occurs by default, but it is possible to exercise greater control by mapping the attributes in the following file (where `<`*`ITG_Home`*`>` represents the installation path for Mercury IT Governance Center):

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

Note

Mercury IT Governance Center also provides sample files (in the same directory) for mapping to a Netscape Directory Server and an Active Directory Server.

The default mapping is for a Netscape Directory Server.

Warning

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

For More Information

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

Step Two: Configure the Mercury IT Governance Server

Several Mercury IT Governance Server attributes need to be considered when performing a user import from an LDAP server. These attributes are set in the following file (where `<ITG_Home>` represents the installation path for Mercury IT Governance Center):

```
<ITG_Home>/server.conf
```

After changing any of these attributes, stop and restart the Mercury IT Governance Server.

For More Information

The `server.conf` file is described in the *System Administration Guide and Reference*.

Step Three: Start the Import

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

The Run ITG 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 Mercury IT Governance Center tables
- Reports on the results of the execution, listing the specified users that failed validation and the specific validation errors they encountered

To run the Run ITG Organization Unit Interface report:

1. Expand **Reports** from the menu bar.

   The list of report options appears.

2. Select **Submit New Report.**

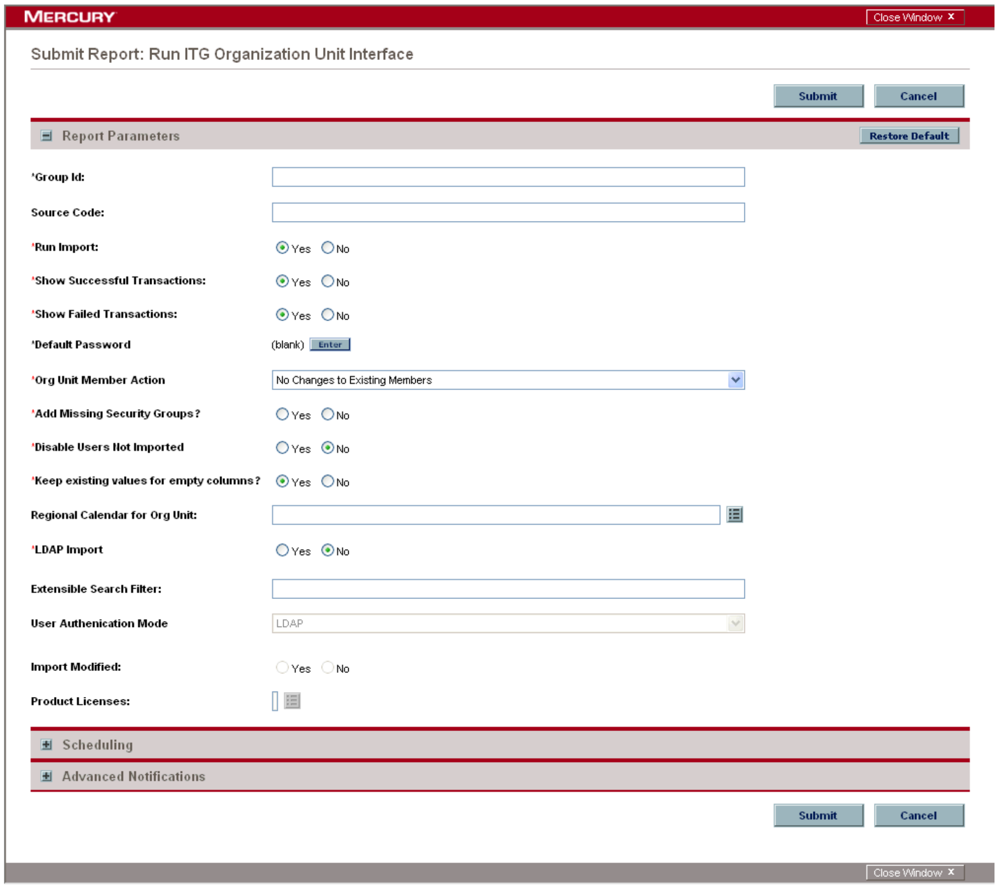
   The Submit New Report page appears.

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

   The page is updated showing the list of administrative reports.

4. Select **Run ITG Organization Unit Interface.**

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



Submit Report: Run ITG Organization Unit Interface

Report Parameters

*Group Id:	
Source Code:	
*Run Import:	[✓ Yes] No
*Show Successful Transactions:	[✓ Yes] No
*Show Failed Transactions:	[✓ Yes] No
*Default Password	(blank)
*Org Unit Member Action	No Changes to Existing Members
*Add Missing Security Groups?	Yes No
*Disable Users Not Imported	Yes [✓ No]
*Keep existing values for empty columns?	[✓ Yes] No
Regional Calendar for Org Unit:	
*LDAP Import	Yes [✓ No]
Extensible Search Filter:	
User Authenication Mode	LDAP
Import Modified:	Yes No
Product Licenses:	

5. Complete the fields as described in the previous step 5 on page 56.

   The Run ITG Organization Unit Interface report has several parameters for controlling the behavior of the program execution. Pay special attention to:

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

Note

Remember that you can test the process by setting the Run Import field to **No**.

Step Four: Verify Successful Completion

Confirm that the import process completed successfully. Click **View Report** to review the results of the import.

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

This report identifies any errors with the import. If errors are present, start your troubleshooting by referring to *Correcting Failures* on page 65.

Keep in mind that all interface tables are automatically cleared by the purge service. The purging process depends on the following attributes in the `server.conf` file:

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

For More Information

The `server.conf` file is described in the *System Administration Guide and Reference*.

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, corrective actions are required. The first step involves examining the audit report from the open interface report 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 failures may occur due to a mapping problem between the source data and existing data in Mercury IT Governance Center.

Other failures may be due to 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.

Note

During the initial implementation of the open interface, the mapping between the third-party source and Mercury IT Governance Center should be thoroughly reviewed and the load program(s) thoroughly tested in a testing instance.

Additionally, it is good practice to monitor executions of the open interface and periodically monitor the import of desired data into Mercury IT Governance Center.

Chapter

4

Request Open Interface

In This Chapter:

Overview

In addition to the standard interface for the entry of new requests, the open interface supports request creation. This open interface enables integration with non-Mercury IT Governance 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 Mercury Demand Management™ during initial implementation.

Note

The Request Open Interface cannot be used to update existing requests in the Mercury IT Governance Center.

In general, the synchronization process involves importing request attributes using the interface tables listed in the following section (*The Data Model*).

For More Information

For information on requests, see the *Mercury Demand Management: Configuring a Request Resolution System* document.

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

The Data Model

The following interface tables are used by the Request Open Interface.

- *KCRT_REQUESTS_INT*
- *KCRT_REQUEST_DETAILS_INT*
- *KCRT_REQ_HEADER_DETAILS_INT*
- *KCRT_TABLE_ENTRIES_INT*

The following field group interface tables may also be used by the Request Open Interface.

- *KCRT_FG_DEMAND_SCHEDULE_INT*
- *KCRT_FG_MASTER_PROJ_REF_INT*
- *KCRT_FG_PROG_ISSUE_INT*
- *KCRT_FG_PROG_REFERENCE_INT*
- *KCRT_FG_PROG_RESOURCE_REQ_INT*
- *KCRT_FG_PROG_RISK_INT*
- *KCRT_FG_PROG_SCOPE_CHANGE_INT*
- *KCRT_FG_SLA_INT*
- *KCRT_FG_WORK_ITEMS_INT*

These interface tables are described, in their entirety, in Appendix A: *Open Interface Data Models* on page 133. With the exception of the field group interface tables, the columns that can be used when importing request are detailed in the appropriate step within this chapter.

Figure 4-1 on page 70 displays the relationships between the various Request Open Interface tables. Note that the placeholder (<*NAME*>) represents any of the field group interface tables.

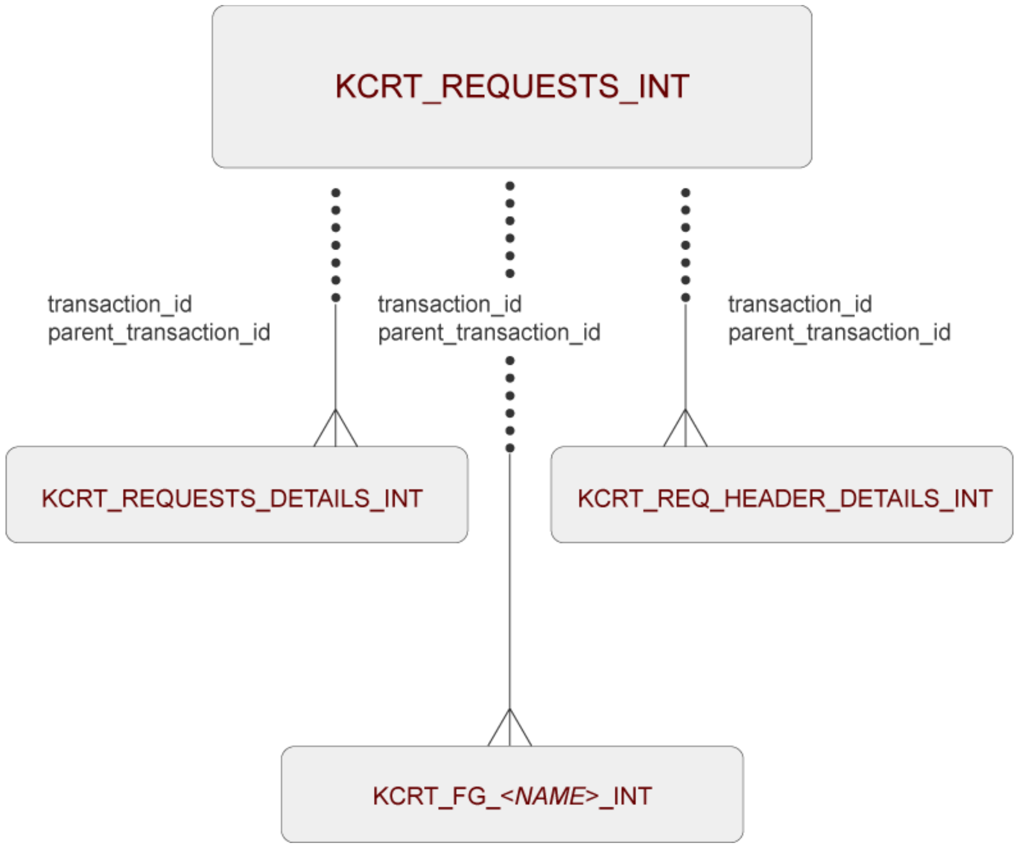


Figure 4-1. Request interface and supporting tables

Performing an Import

Step One: Populate the Request Interface Tables

During an import of requests, specific columns in the Request Open Interface tables must be populated. The input columns are listed in the following table (*Table 4-1*) as well as *Table 4-2* on page 77 through *Table 4-4* on page 81. Remember to also utilize the applicable field group interface table(s) as shown in Appendix A: *Open Interface Data Models* on page 133.

This population can be done through any means support by the Oracle database (such as SQL*Loader or direct Oracle database-to-database communication).

The population process can also involve the manipulation of the table records once they have been brought into the interface table. This can include the setting of ID columns (such as GROUP_ID and TRANSACTION_ID) and the defaulting of specific data not available in the source of the request (such as the third-party application or the ASCII file).

Warning User data is not validated during import.

Table 4-1. KCRT_REQUESTS_INT interface table input options

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	Provides a unique identifier for each transaction. If any detail table is being used, set the PARENT_TRANSACTION_ID in the detail interface tables to this value.
REQUEST_ID	Optional	NUMBER	Identifies the request. This is normally left blank and is derived from the KCRT_REQUESTS_S sequence.

Table 4-1. KCRT_REQUESTS_INT interface table input options [continued]

Column	Usage	Data Type	Description
CREATION_DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.
CREATED_USERNAME	Optional	VARCHAR2 (30)	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_DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.
LAST_UPDATED_ USERNAME	Required	VARCHAR2 (30)	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 (30)	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 (80)	Identifies the request type. Derived from KCRT_REQUESTS_ TYPES.

Table 4-1. KCRT_REQUESTS_INT interface table input options [continued]

Column	Usage	Data Type	Description
REQUEST_TYPE_ID	Optional	NUMBER	Identifies the request type. If left blank, the value is derived from REQUEST_TYPE_NAME.
REQUEST_SUBTYPE_ NAME	Optional	VARCHAR2 (80)	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	Optional	VARCHAR2 (240)	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 in the remote system.
STATUS_NAME	Optional	VARCHAR2 (80)	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 (80)	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.

Table 4-1. KCRT_REQUESTS_INT interface table input options [continued]

Column	Usage	Data Type	Description
WORKFLOW_ID	Optional	NUMBER	Specifies the workflow that the request should follow. This is normally left blank andthe value is derived from WORKFLOW_NAME.
DEPARTMENT_NAME	Optional	VARCHAR2 (80)	Specifies the name of the department. This should be a valid MEANING from KNTA_LOOKUPS.where `LOOKUP_TYPE = 'DEPARTMENT_CODE'`.
PRIORITY_NAME	Optional	VARCHAR2 (80)	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	Optional	VARCHAR2 (30)	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	Option varies	VARCHAR2 (30)	Specifies the USERNAME (from KNTA_USERS) that should initially be assigned the request.
ASSIGNED_TO_USER_ID	Option varies	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	Optional	VARCHAR2 (30)	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 initally be assigned to the request. This is normally left blank and the value is derived from ASSIGNED_TO_GROUP_NAME.

Table 4-1. KCRT_REQUESTS_INT interface table input options [continued]

Column	Usage	Data Type	Description
PROJECT_CODE	Optional	VARCHAR2 (30)	Indicates the user-defined project for the request. This should be a valid value from KNTA_ LOOKUPS where `LOOKUP_TYPE = 'PROJECT'`.
CONTACT_FIRST_NAME	Option varies	VARCHAR2 (30)	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	Option varies	VARCHAR2 (30)	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.
RELEASED_FLAG	Optional	VARCHAR2 (1)	Indicates whether or not the request should be released after import. Valid values are: • Y • N The default value is N.
USER_DATA1 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USERS_DATA20	Requirement varies	VARCHAR2 (200)	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	Requirement varies	NUMBER	Sets the context identifier for the detail fields. Either this or REQUEST_TYPE_NAME must be populated.

Table 4-1. KCRT_REQUESTS_INT interface table input options [continued]

Column	Usage	Data Type	Description
NOTES	Optional	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	Optional	VARCHAR2 (30)	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	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.
COMPANY	Optional	VARCHAR2 (30)	Identifies the name of the company associated with this request.

Table 4-2. KCRT_REQUEST_DETAILS_INT interface table input options

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	Provides a unique identifier for 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.

Table 4-2. KCRT_REQUEST_DETAILS_INT interface table input options [continued]

Column	Usage	Data Type	Description
PARAMETER1 VISIBLE_PARAMETER1 through PARAMETER50 VISIBLE_PARAMETER50	Requirement varies	VARCHAR2 (200)	Specifies the values for all the custom fields defined in the request.
LOOKUP_TYPE1 VALIDATION_TYPE_CODE1 through LOOKUP_TYPE50 VALIDATION_TYPE_ CODE50	Requirement varies	VARCHAR2 (80) VARCHAR2 (30)	Identifies the lookup type for each PARAMETER as well as 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 input options

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	Provides a unique identifier for 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 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.

Table 4-3. KCRT_REQ_HEADER_DETAILS_INT interface table input options [continued]

Column	Usage	Data Type	Description
PARAMETER1 VISIBLE_PARAMETER1 through PARAMETER50 VISIBLE_PARAMETER50	Requirement varies	VARCHAR2 (200)	Specifies the values for all the custom fields defined in the request.
LOOKUP_TYPE1 VALIDATION_TYPE_CODE1 through LOOKUP_TYPE50 VALIDATION_TYPE_ CODE50	Requirement varies	VARCHAR2 (80) VARCHAR2 (30)	Identifies the lookup type for each PARAMETER as well as 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 input options

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	Provides a unique identifier for each transaction.
PARENT_TRANSACTION_ID	Required	NUMBER	Provides the transaction ID (from KCRT_REQUESTS_INT) of the parent table being imported.
PARENT_FIELD_TOKEN	Required	VARCHAR2 (30)	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.

Table 4-4. KCRT_TABLE_ENTRIES_INT interface table input options [continued]

Column	Usage	Data Type	Description
VISIBLE_PARAMETER1 PARAMETER1 through VISIBLE_PARAMETER50 PARAMETER50	Requirement varies	VARCHAR2 (200)	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.
LOOKUP_TYPE1 through LOOKUP_TYPE50	Requirement varies	VARCHAR2 (80)	Identifies the lookup type for each PARAMETER. This is required only if user data is defined.
VALIDATION_TYPE_CODE1 through VALIDATION_TYPE_ CODE50	Requirement varies	VARCHAR2 (30)	Identifies the validation type code for each PARAMETER. This is required only if user data is defined.

Step Two: 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 Mercury 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 Mercury 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. Expand **Reports** from the menu bar.

   The list of report options appears.

2. Select **Submit New Report.**

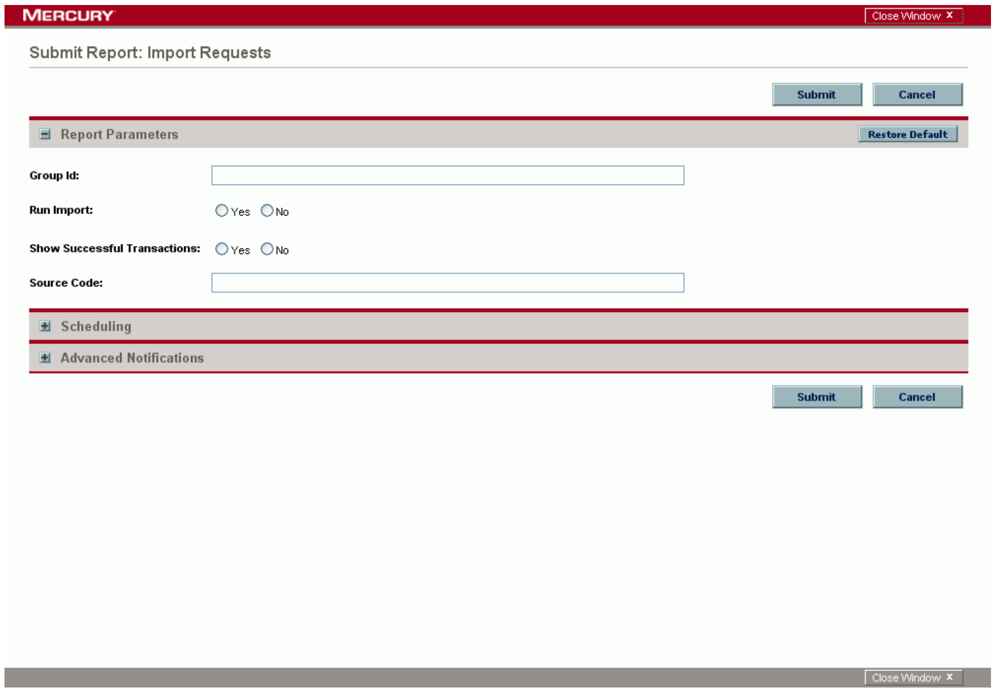
   The Submit New Report page appears.

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.



Group Id:	
Run Import:	Yes No
Show Successful Transactions:	Yes No
Source Code:	

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

Note

Remember that you can test the process by setting the Run Import field to **No**.

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.

Field Name	Description
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.

Note

Required fields are denoted with a red asterisk. These fields may vary based on your selections.

Step Three: Verify Successful Completion

Confirm that the import process completed successfully. Click **View Report** to review the results of the import.

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

This report identifies any errors with the import. If errors are present, start your troubleshooting by referring to *Correcting Failures* on page 86.

Keep in mind that all interface tables are automatically cleared by the purge service. The purging process depends on the following attributes 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

The `server.conf` file is described in the *System Administration Guide and Reference*.

Correcting Failures

When a request is successfully imported, information stored in the interface tables is not deleted, and no additional action is required. Users can view and process the request using the Mercury Demand Management standard interface.

For requests that fail to import, corrective actions are required. The first step is examining the audit report from the open interface report 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 failures might occur due to a mapping problem between the source data and existing Mercury Demand Management data. For example, the source data might use a project name that does not exist in Mercury Demand Management. Corrective measures for this specific problem would include adding the specific project in Mercury Demand Management, or mapping the source project to a project name that already exists in Mercury Demand Management.

Other failures might be due to missing required 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.

Failures could occur due to other configuration and mapping problems in either the source or in Mercury Demand 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 non-Mercury IT Governance Center source and Mercury Demand Management should be thoroughly reviewed and the load program(s) thoroughly tested in a testing instance. Additionally, it is good practice to monitor executions of the open interface and periodically monitor that the desired data is being imported into Mercury Demand Management.

Chapter

5

Package Open Interface

In This Chapter:

Overview

In addition to the standard interface for the entry of new packages and package lines, Mercury IT Governance Center includes an open interface for package creation and the creation of new package lines. The open interface uses interface tables within the Mercury Change Management™ database instance. Data added to these interface tables is validated and eventually imported into standard Mercury Change Management tables. This generates packages and package lines that can be processed using Mercury Change Management.

The primary purpose of the Package Open Interface is to allow integration with non-Mercury IT Governance Center products. Relevant information from these products can be used to generate the appropriate packages using the open interface. The Package Open Interface can also 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. The open interface can also be used as a conversion mechanism to convert data from a legacy system into Mercury Change Management during initial implementation.

In general, the synchronization process involves importing package attributes using the interface tables described in the following section (*The Data Model* on page 89).

For More Information

For information on requests, see the *Mercury Change Management User's Guide*. Additional insight might also be gleaned from the *Mercury Demand Management User's Guide*.
:

Note

Integration between the products in Mercury IT Governance 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 Mercury Change Management packages from Mercury Demand Management requests.

This does presuppose that the relevant workflows and request types exist.

The Data Model

The following interface tables are used by the Package Open Interface.

- *KDLV_PACKAGES_INT*
- *KDLV_PACKAGE_LINES_INT*
- *KDLV_PACKAGE_NOTES_INT*

These interface tables are described, in their entirety, in Appendix A: *Open Interface Data Models* on page 133.

Figure 5-1 displays the relationships between the various Package Open Interface tables.

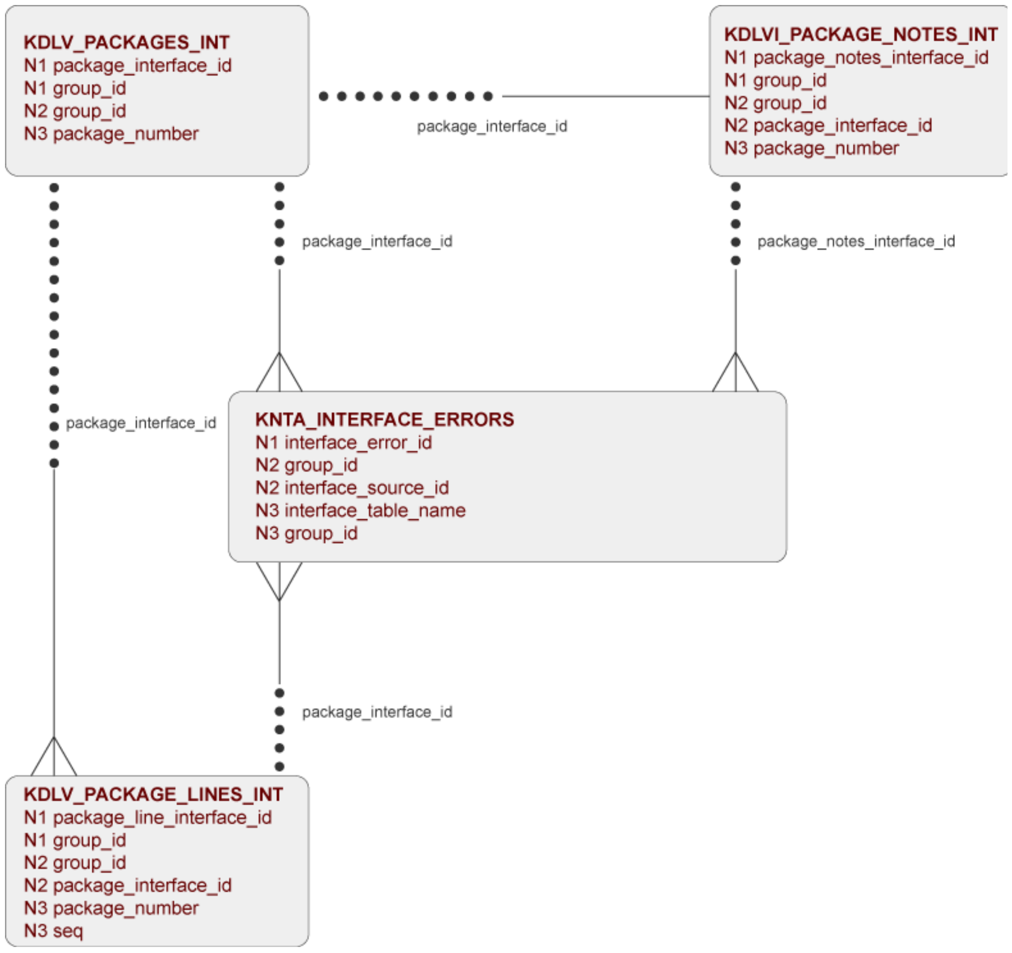


Figure 5-1. Package interface and supporting tables

Performing an Import

Step One: Load the Interface Tables

During an import of packages and package lines, specific columns in the Package Open Interface tables must be populated. The input columns are listed in the following table (*Table 5-1*) as well as *Table 5-2* on page 95 and *Table 5-3* on page 99.

The population can be done through any means supported by the Oracle database. Standard mechanisms include the use of SQL*Loader to load in the contents of an ASCII file or direct Oracle database to database-to-database communication through database links.

Warning

User data is not validated during import.

Table 5-1. KDLV_PACKAGES_INT interface table input options

Column	Usage	Data Type	Description
PACKAGE_INTERFACE_ID	Requirement varies	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	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 5-1. KDLV_PACKAGES_INT interface table input options [continued]

Column	Usage	Data Type	Description
CREATED_BY	Requirement varies	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	Requirement varies	VARCHAR2 (30)	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	Optional	VARCHAR2 (30)	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. For new lines to be imported into existing packages, this column should refer to the PACKAGE_ID of the existing package.
REQUESTED_BY	Requirement varies	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.

Table 5-1. KDLV_PACKAGES_INT interface table input options [continued]

Column	Usage	Data Type	Description
REQUESTED_BY_ USERNAME	Requirement varies	VARCHAR2 (30)	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 (40)	Identifies the package number. This must use either the same value as PACKAGE_ID or a unique string.
ASSIGNED_TO_USER_ID	Requirement varies	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	Requirement varies	VARCHAR2 (30)	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	Requirement varies	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	Requirement varies	VARCHAR2 (30)	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	Optional	VARCHAR2 (240)	Specifies a user-visible description of the package.
PACKAGE_TYPE_CODE	Optional	VARCHAR2 (30)	Provides a user-defined categorization of the package. Must be a valid LOOKUP_CODE from KNTA_LOOKUPS where `LOOKUP_TYPE = 'PACKAGE_TYPE'`.

Table 5-1. KDLV_PACKAGES_INT interface table input options [continued]

Column	Usage	Data Type	Description
PRIORITY_CODE	Optional	VARCHAR2 (30)	Indicates the user-defined priority for the package. Must be a valid LOOKUP_CODE from KNTA_LOOKUPS where `LOOKUP_TYPE = 'PACKAGE_PRIORITY'`.
PROJECT_CODE	Optional	VARCHAR2 (30)	Indicates the user-defined project for the package. This should be a valid value from KNTA_ LOOKUPS where `LOOKUP_TYPE = 'PROJECT'`.
WORKFLOW_ID	Requirement varies	NUMBER	Specifies the workflow that the package should follow. Derived from WORKFLOW_NAME. Either WORKFLOW_ID or WORKFLOW_NAME must be entered.
WORKFLOW_NAME	Requirement varies	VARCHAR2 (80)	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	Optional	VARCHAR2 (1)	Indicates whether or not the interface program will release the package once it imports in into the standard Mercury Change 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 1202.

Table 5-1. KDLV_PACKAGES_INT interface table input options [continued]

Column	Usage	Data Type	Description
USER_DATA1 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USER_DATA20	Requirement varies	VARCHAR2 (200)	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	Optional	NUMBER	Identifies the original package for this distribution package.
DISTPKG_STATUS_ MEANING	Optional	VARCHAR2 (80)	Provides a user-visible status for this distribution package.
RUN_GROUP	Optional	NUMBER	Provides a run group number of a specific distribution package.
DISTRIBUTION_ID	Optional	NUMBER	Identifies the distribution associated with the package.
ENABLED_FLAG	Optional	VARCHAR2 (1)	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	Optional	NUMBER	Specifies the path of the distribution workflow step that was executed in the transaction with DIST_STEP_ TRANSACTION_ID.

Table 5-2. KDLV_PACKAGE_LINES_INT interface table input options

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	Requirement varies	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	Optional	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	Optional	VARCHAR2 (40)	Identifies the package number. This must use either the same value as PACKAGE_ID or a unique string.

Table 5-2. KDLV_PACKAGE_LINES_INT interface table input options [continued]

Column	Usage	Data Type	Description
CREATED_BY	Requirement varies	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	Requirement varies	VARCHAR2 (30)	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	Optional	VARCHAR2 (30)	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	Requirement varies	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.

Table 5-2. KDLV_PACKAGE_LINES_INT interface table input options [continued]

Column	Usage	Data Type	Description
OBJECT_TYPE_NAME	Requirement varies	VARCHAR2 (80)	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	Required	VARCHAR2 (300)	Specifies the name of the object to be processed. This value is not validated.
APP_CODE	Optional	VARCHAR2 (30)	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	Requirement varies	VARCHAR2 (200)	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.
RELEASE_FLAG	Optional	VARCHAR2 (1)	Indicates whether or not the interface program will release the package once it imports in into the standard Mercury Change 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 input options [continued]

Column	Usage	Data Type	Description
USER_DATA1 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USER_DATA20	Requirement varies	VARCHAR2 (200)	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	Optional	VARCHAR2 (300)	Specifies the denormalized object_ revision of the object entered on this line.
SOURCE_PACKAGE_LINE_ ID	Optional	NUMBER	Identifies the original package line for this distribution package line.
ENABLED_FLAG	Optional	VARCHAR2 (1)	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 input options

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	Requirement varies	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 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	Optional	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.

Table 5-3. KDLV_PACKAGE_NOTES_INT interface table input options [continued]

Column	Usage	Data Type	Description
PACKAGE_NUMBER	Optional	VARCHAR2 (40)	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.
CREATED_BY	Requirement varies	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	Requirement varies	VARCHAR2 (30)	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	Optional	VARCHAR2 (30)	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.

Step Two: Start the Import

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

The Run ITG Package Interface report:

- Queries the interface tables for active records matching the given selection criteria.
- Defaults any information that has defaulting rules in Mercury Change 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, as well as those that failed validation and the specific validation errors they encountered.

To run the Run ITG Package Interface report:

1. Expand **Reports** from the menu bar.

   The list of report options appears.

2. Select **Submit New Report.**

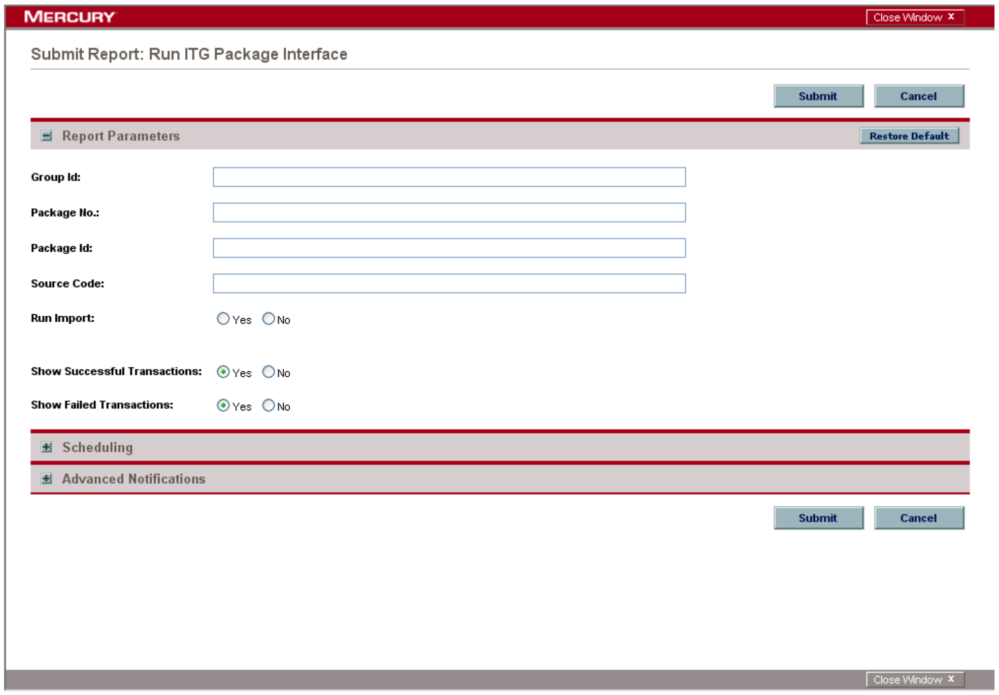
   The Submit New Report page appears.

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

   The page is updated showing the list of administrative reports.

4. Select **Run ITG Package Interface.**

   The Submit Report: Run ITG Package Interface window appears.



Submit Report: Run ITG Package Interface

Report Parameters

Group Id:	
Package No.:	
Package Id:	
Source Code:	
Run Import:	Yes No
Show Successful Transactions:	[✓ Yes] No
Show Failed Transactions:	[✓ Yes] No

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

Note

Remember that you can test the process by setting the Run Import field to **No**.

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.

Field Name	Definition
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 column.This 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.

Note

Required fields are denoted with a red asterisk. These fields may vary based on your selections.

Step Three: Verify Successful Completion

Confirm that the import process completed successfully. Click **View Report** to review the results of the import.

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

This report identifies any errors with the import. If error are present, start your troubleshooting by referring to *Correcting Failures* on page 105.

Keep in mind that all interfere tables are automatically cleared by the purge service. The purging process depends on the following attributes 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

The `server.conf` file is described in the *System Administration Guide and Reference*.

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 Mercury Change 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 Mercury Change Management. Corrective measures for this specific problem would include adding the specific project in Mercury Change Management, or mapping the source project to a project name that already exists in Mercury Change 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 Mercury Change 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 Mercury Change Management should be thoroughly reviewed and the load program(s) 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 Mercury Change Management.

Chapter

6

Workflow Transaction Open Interface

In This Chapter:

Overview

In addition to the standard interface for performing executions and approving workflow steps, Mercury IT Governance Center includes an open interface for performing these same 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.

Warning

The force transition feature is not supported through the standard interface.

The Workflow Transaction Open Interface is a set of tables within the Mercury IT Governance 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 non-Mercury IT Governance Center products. Relevant information from these products can be used to perform workflow transactions for package lines and requests. The open interface can also be used as a mechanism to convert data from a legacy system into Mercury IT Governance 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.

Note

The Workflow Transaction Open Interface does support the use of subworkflows. When dealing with subworkflows, it is important to remember that:

- The workflow step sequence should be the same as the one 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 Data Model

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

- *KWFL_TRANSACTIONS_INT*

This interface tables is described, in its entirety, in Appendix A: *Open Interface Data Models* on page 133. The columns that can be used when importing workflows are detailed in the appropriate step within this chapter.

Additionally, different parameters are required (or optional) depending upon the type of transaction. All of these parameters are described in Appendix A: *Open Interface Data Models* on page 133. The parameters used for each type of transaction is detailed where appropriate.

Figure 6-1 displays the relationship between the KWFL_TRANSACTIONS_INT and KNTA_INTERFACE_ERRORS table.

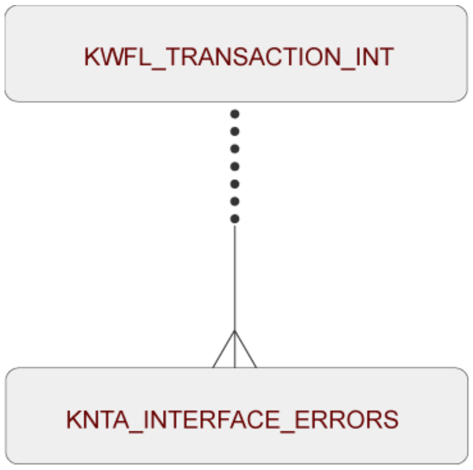


Figure 6-1. Workflow transaction interface and supporting tables

Performing an Import

Step One: Load the Interface Tables

During an import of workflows, specific columns in the Workflow Transaction Open Interface tables must be populated. The input columns are listed in the following table (*Table 6-1*).

The population can be done through any means supported by the Oracle database. Standard mechanisms include the use of SQL*Loader to load in the contents of an ASCII file or direct Oracle database to database-to-database communication through database links.

Table 6-1. KWFL_TRANSACTIONS_INT interface table input options

Column	Usage	Data Type	Description
TRANSACTION_ID	Required	NUMBER	Provides a unique identifier for each transaction.
CREATION_DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.
CREATED_USERNAME	Requirement varies	VARCHAR2 (30)	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. Supply either this or CREATED_BY.
CREATED_BY	Requirement varies	NUMBER	Identifies the USER_ID (from KNTA_USERS) for the user performing the transaction. Supply either this or CREATED_USERNAME.
LAST_UPDATE_DATE	Optional	DATE	Indicates the transaction date. If left blank, the current date is used.
LAST_UPDATED_USERNAME	Requirement varies	VARCHAR2 (30)	Identifies the USERNAME (from KNTA_USERS) for the user performing the transaction. Supply either this or LAST_UPDATED_BY.

Table 6-1. KWFL_TRANSACTIONS_INT interface table input options [continued]

Column	Usage	Data Type	Description
LAST_UPDATED_BY	Requirement varies	NUMBER	Identifies the USER_ID (from KNTA_ USERS) for the user performing the transaction. Supply either this or LAST_UPDATED_ USERNAME. If both are left blank, the value is derived from CREATED_USERNAME.
EVENT	Required	VARCHAR2 (40)	Specifies the type of workflow transaction.
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.
SOURCE_TYPE_CODE	Optional	VARCHAR2 (30)	Specifies the type of external update. This should be a left blank or have a value of INTERFACE_WF.
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.
INSTANCE_SOURCE_ TYPE_CODE	Required	VARCHAR2 (30)	Indicates whether or not the transaction is for a package line (‘CR’) or a request (‘IR').
INSTANCE_SOURCE_SET_ NUMBER	Option varies	VARCHAR2 (40)	Specifies the package number (PACKAGE_NUMBER from KDLV_ PACKAGES) or request number (REQUEST_NUMBER from KCRT_ REQUESTS). Supply either this or INSTANCE_ SOURCE_SET_ID.
INSTANCE_SOURCE_SET_ ID	Option varies	NUMBER	Specifies the package ID (PACKAGE_ID from KDLV_PACKAGES) or request ID (REQUEST_ID from KCRT_ REQUESTS). Supply either this or INSTANCE_ SOURCE_SET_NUMBER.

Table 6-1. KWFL_TRANSACTIONS_INT interface table input options [continued]

Column	Usage	Data Type	Description
INSTANCE_SOURCE_LINE_SEQ	Option varies	NUMBER	Specifies the package line sequence number (SEQ from KDLV_PACKAGE_LINES). Supply either this or INSTANCE_SOURCE_ID.
INSTANCE_SOURCE_ID	Option varies	NUMBER	Specifies the package line ID (PACKAGE_LINE_ID from KDLV_PACKAGE_LINES) or request ID (REQUEST_ID from KCRT_REQUESTS). Supply either this or INSTANCE_SOURE_LINE_SEQ (for package lines) or INSTANCE_SOURCE_SET_NUMBER (for requests).
WORKFLOW_STEP_NAME	Option varies	VARCHAR2 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS). Supply either this or WORKFLOW_STEP_ID.
WORKFLOW_STEP_SEQ	Option varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step. Supply either this or WORKFLOW_STEP_ID. In the case of subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth.
RESULT_VALUE	Optional	VARCHAR2 (200)	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	Optional	VARCHAR2 (200)	Indicates the result of the step. This is the result value that a user normally sees.
USER_COMMENTS	Optional	VARCHAR2 (200)	Specifies comments for the transaction. Any comments are appended to the notes for the package or request.
DELEGATED_TO_USERNAME	Option varies	VARCHAR2 (30)	Specifies the USERNAME (from KNTA_USERS) for the user that the decision is being delegated to. Supply either this or DELEGATED_TO_USER_ID.

Table 6-1. KWFL_TRANSACTIONS_INT interface table input options [continued]

Column	Usage	Data Type	Description
DELEGATED_TO_USER_ID	Option varies	NUMBER	Specifies the USER_ID (from KNTA_ USERS) for the user that the decision is being delegated to. Supply either this or DELEGATED_TO_ USERNAME.
SCHEDULE_DATE	Optional	DATE	Indicates the date that the execution step is scheduled to run.
WORKFLOW_STEP_ID	Option varies	NUMBER	Specifies the workflow step ID (WORKFLOW_STEP_ID from KWFL_ WORKFLOW_STEPS). Supply either this, WORKFLOW_STEP_ NAME, or WORKFLOW_STEP_SEQ.
TO_WORKFLOW_STEP_ SEQ	Option varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step for the step that the package line or request should transition to. Supply either this, TO_WORKFLOW_ STEP_ID, or TO_WORKFLOW_STEP_ NAME.
TO_WORKFLOW_STEP_ NAME	Option varies	VARCHAR2 (80)	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 either this, TO_WORKFLOW_ STEP_SEQ, or TO_WORKFLOW_ STEP_ID.
TO_WORKFLOW_STEP_ID	Option varies	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 either this, TO_WORKFLOW_ STEP_NAME, or TO_WORKFLOW_ STEP_SEQ.

Step Two: Load the Parameters

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.

In order to facilitate populating the Workflow Transaction Open Interface tables, we have provided the `KWFL_TXN_INT.INSERT_ROW` procedure. All of the parameters for the `INSERT_ROW` procedure in the `KWFL_TXN_INT` package are described in Appendix A: *Open Interface Data Models* on page 133.

The list of parameters used for each type of transaction (or event) is provided in the following tables.

- *Parameters Used For All Events* on page 117
- *Parameters for Package or Request Status* on page 118
- *Parameters for Decision Step Results* on page 119
- *Parameters for Decision Step Delegation* on page 120
- *Parameters for Execution Steps* on page 121
- *Parameters for Execution Step Schedule* on page 122
- *Parameters for Execution Step Bypass* on page 123
- *Parameters for Changing Step Result* on page 124
- *Parameters for Forced Workflow Step Transition* on page 125
- *Parameters for Package Line or Request Cancellation* on page 126

The INSERT_ROW procedure needs to be called by another PL/SQL procedure, function, or anonymous block.

The following example of an anonymous PL/SQL block could be 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(80);
   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,
     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 (the placeholders *<username>* and *<password>* represent the user ID and password for the appropriate Mercury IT Governance Center database).

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

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 (40)	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 (80)	Identifies the USERNAME (from KNTA_ USERS) for the user performing the transaction.
P_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.
P_REQUEST_NUMBER	Requirement varies	VARCHAR2 (40)	Identifies the request. Supply either this or P_PACKAGE_ NUMBER.
P_PACKAGE_NUMBER	Requirement varies	VARCHAR2 (40)	Identifies the package number. Supply either this or P_REQUEST_ NUMBER.
P_USER_COMMENTS	Optional	VARCHAR2 (200)	Specifies comments for the transaction. Any comments are appended to the notes for the package or request.

Table 6-2. Parameters used for all events [continued]

Parameter	Usage	Data Type	Description
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 (80)	Specifies the internal message name of the error that was returned. This is used mainly for debugging purposes.
O_MESSAGE	Leave blank	VARCHAR2 (1000)	Provides the error message.

Parameters for Package or Request Status

The following parameter should be used 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 (40)	Specifies the type of workflow transaction. Set to INSTANCE_SET_CREATE.

Parameters for Decision Step Results

The following parameters should be used for the status for decision steps.

Table 6-4. Parameters for decision step results

Parameter	Usage	Data Type	Description
P_EVENT	Required	VARCHAR2 (40)	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 if the transaction is for a package line.
P_WORKFLOW_STEP_ NAME	Requirement varies	VARCHAR2 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_ WORKFLOW_STEPS). Supply either this or P_WORKFLOW_ STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step. In the case of subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply either this or P_WORKFLOW_ STEP_NAME.
P_VISIBLE_RESULT_VALUE	Required	VARCHAR2 (200)	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 (40)	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 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_ WORKFLOW_STEPS). Supply either this or P_WORKFLOW_ STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step. In the case of subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply either this or P_WORKFLOW_ STEP_NAME.
P_DELEGATED_TO_ USERNAME	Required	VARCHAR2 (30)	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 (40)	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 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_ WORKFLOW_STEPS). Supply either this or P_WORKFLOW_ STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step. In the case of subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply either 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 (40)	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. Use if the transaction is for a package line.
P_WORKFLOW_STEP_ NAME	Requirement varies	VARCHAR2 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_ WORKFLOW_STEPS). Supply either this or P_WORKFLOW_ STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step. In the case of subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply either 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 (40)	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 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_ WORKFLOW_STEPS). Supply either this or P_WORKFLOW_ STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step. In the case of subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply either this or P_WORKFLOW_ STEP_NAME.
P_VISIBLE_RESULT_VALUE	Required	VARCHAR2 (200)	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 (40)	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 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_WORKFLOW_STEPS). Supply either this or P_WORKFLOW_STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step. In the case of subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply either this or P_WORKFLOW_STEP_NAME.
P_VISIBLE_RESULT_VALUE	Required	VARCHAR2 (200)	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 (40)	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 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_ WORKFLOW_STEPS). Supply either this or P_WORKFLOW_ STEP_SEQ.
P_WORKFLOW_STEP_SEQ	Requirement varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step. In the case of subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth. Supply either this or P_WORKFLOW_ STEP_NAME.
P_VISIBLE_RESULT_VALUE	Required	VARCHAR2 (200)	Indicates the result of the step. This is the result value that a user normally sees.

Table 6-10. Parameters for forced workflow step transition [continued]

Parameter	Usage	Data Type	Description
P_TO_WORKFLOW_STEP_ NAME	Requirement varies	VARCHAR2 (80)	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 either this or P_TO_ WORKFLOW_STEP_SEQ.
P_TO_WORKFLOW_STEP_ SEQ	Requirement varies	VARCHAR2 (30)	Specifies the sequence number of the workflow step for the step that the package line or request should transition to. Supply either 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 (40)	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 Three: 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 Mercury IT Governance 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 executed, scheduled tasks are generated to run.

Note

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

Additionally, 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 ITG Package Interface report:

1. Expand **Reports** from the menu bar.

   The list of report options appears.

2. Select **Submit New Report.**

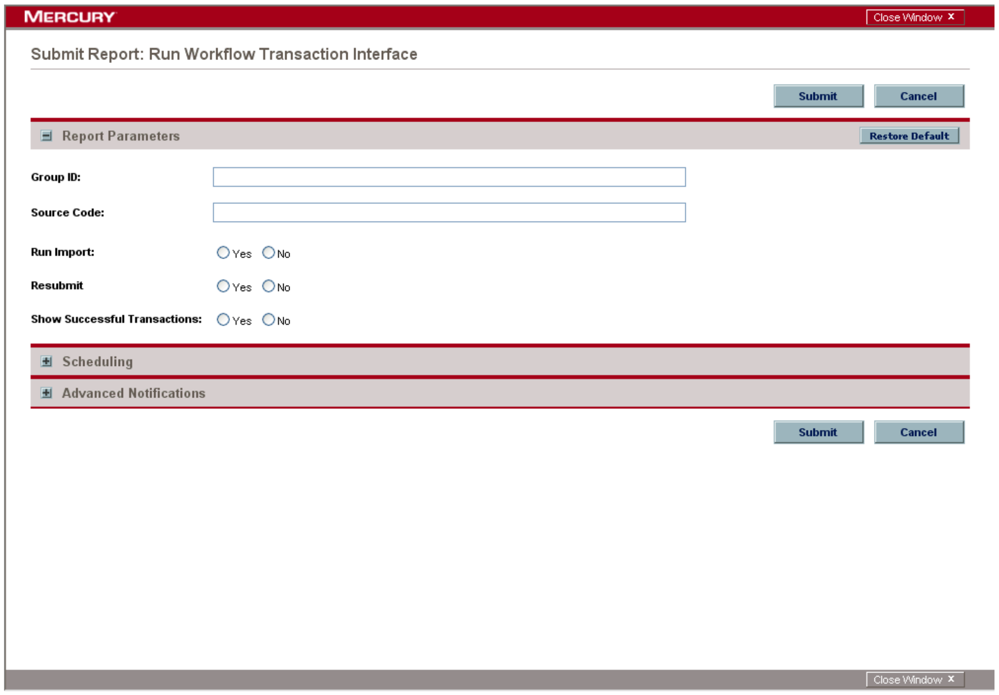
   The Submit New Report page appears.

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.



Submit Report: Run Workflow Transaction Interface

Group ID:	
Source Code:	
Run Import:	○ Yes ○ No
Resubmit	○ Yes ○ No
Show Successful Transactions:	○ Yes ○ No

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

Note

Remember that you can test the process by setting the Run Import field to **No**.

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.

Note

Required fields are denoted with a red asterisk. These fields may vary based on your selections.

Step Four: Verify Successful Completion

Confirm that the import process completed successfully. Click **View Report** to review the results of the import.

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

This report identifies any errors with the import. If error are present, start your troubleshooting by referring to *Correcting Failures* on page 130.

Keep in mind that all interface tables are automatically cleared by the purge service. The purging process depends on the following attributes 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

The `server.conf` file is described in the *System Administration Guide and Reference*.

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 workflow transactions 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 Mercury IT Governance Center data.

For example, the source data might use a result value that does not exist in Mercury IT Governance Center. Corrective measures for this specific problem would include adding the specific result to the validation for the workflow step or choosing a new result value.

Other failures might be due to missing required 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.

Failures could occur due to other configuration and mapping problems in either the source or in Mercury IT Governance Center or could be the result of errors in the custom loading program.

Note

During initial implementation of the open interface, the mapping between the non-Mercury IT Governance Center source and Mercury IT Governance Center should be thoroughly reviewed and the load program(s) thoroughly tested. Additionally, it is good practice to monitor executions of the open interface and periodically monitor that the desired transactions are being processed in Mercury IT Governance Center.

Appendix

A Open Interface Data Models

In This Appendix:

Overview

This appendix describes all of 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 yet will provide (output) data if a value is not provided.
- **Data Type.** Specifies the data type to be used for the column.
- **Description.** Provides a description of the data as well as related information, dependencies, and any pertinent tips.

Warning

You should only provide data for columns specified with input (I) or input and output (I/O) usage.

Unexpected result may occur if you provide data for columns that are to be used exclusively for output (O).

Additionally, this appendix describes the `INSERT_ROW` parameters used with the `KWFL_TXN_INT` package. These are used exclusively with the Workflow Transaction Open Interface. The following information is provided for these parameters.

- **Parameter.** Provides the name of the parameter.
- **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 yet will provide (output) data if a value is not provided.
- **Data Type.** Specifies the data type to be used for the parameter.
- **Description.** Provides a description of the data as well as related information, dependencies, and any pertinent tips.

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 Mercury IT Governance Center.

Table A-1. KNTA_USERS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Provides a unique identifier for each transaction. See also PARENT_TRANSACTION_ID in KNTA_USER_SECURITY_INT.
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 (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. See Appendix C: *Process State Information* on page 197 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See Appendix C: *Process State Information* on page 197 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 (30)	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.

Table A-1. KNTA_USERS_INT interface table [continued]

Column	Usage	Data Type	Description
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	Indicates the date the record is created in the destination (Mercury IT Governance 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.

Table A-1. KNTA_USERS_INT interface table [continued]

Column	Usage	Data Type	Description
USERNAME	I	VARCHAR2 (30)	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. If left blank, the value is derived from USERNAME.
PASSWORD	I/O	VARCHAR2 (40)	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 (80)	Specifies the email address of the user.
FIRST_NAME	I	VARCHAR2 (30)	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 (30)	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 (30)	Specifies the type of external update. This should be a specific interface or migrator name, left blank, or have a value of INTERFACE_WF.

Table A-1. KNTA_USERS_INT interface table [continued]

Column	Usage	Data Type	Description
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.
USER_DATA_SET_ CONTEXT_ID	I	NUMBER	Sets the context identifier for the USER_ DATA fields. Supply either this or USERNAME.
USER_DATA1 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USER_DATA20	I	VARCHAR2 (200)	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.
AUTHENTICATION_MODE	I	VARCHAR2 (30)	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 (1)	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 (4)	Specifies the position where the shortcut bar is displayed. If left blank, the default value is supplied.
SAVE_WINDOW_BOUNDS_ FLAG	I/O	VARCHAR2 (1)	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.

Table A-1. KNTA_USERS_INT interface table [continued]

Column	Usage	Data Type	Description
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. If left blank, the default value is supplied.
REUSE_INTERNAL_ FRAME_FLAG	I/O	VARCHAR2 (1)	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 (1)	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 (1)	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 (1)	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.

Table A-1. KNTA_USERS_INT interface table [continued]

Column	Usage	Data Type	Description
SHOW_CHANGE_ WARNINGS_FLAG	I/O	VARCHAR2 (1)	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 (1)	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 (300)	Specifies the default browser for the user.
DEST_USER_PROFILE_ID	O	NUMBER	Specifies the user profile ID for the user.
COMPANY	I	VARCHAR2 (1)	Identifies the company.
DOMAIN	I	VARCHAR2 (80)	Identifies the Windows domain. Used for Exchange server (NTLM) authentication.
LOGON_IDENTIFIER	I	VARCHAR2 (30)	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 (300)	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 (300)	Specifies the code for the department.

Table A-1. KNTA_USERS_INT interface table [continued]

Column	Usage	Data Type	Description
DEPARTMENT_MEANING	I	VARCHAR2 (80)	Specifies the description of the department.
LOCATION_CODE	O	VARCHAR2 (30)	Specifies the code for the location.
LOCATION_MEANING	I	VARCHAR2 (80)	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 (80)	Specifies the name of the manager. Used if MANAGER_LOGON_ IDENTIFIER is left blank.
MANAGER_LOGON_ IDENTIFIER	I	VARCHAR2 (300)	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 (30)	Specifies the code for the user's category.
RESOURCE_CATEGORY_ MEANING	I	VARCHAR2 (80)	Specifies the description of the user's category.
RESOURCE_TITLE_CODE	O	VARCHAR2 (30)	Specifies the code for the user's title.
RESOURCE_TITLE_ MEANING	I	VARCHAR2 (80)	Specifies the description of the user's title.
PRODUCT_ID_LIST	O	VARCHAR2 (200)	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-2. KNTA_USER_SECURITY_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Provides a unique identifier for 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 (30)	Identifies the table associated with this entity. The parent_table should be derived from KNTA_USERS_INT.
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. See Appendix C: *Process State Information* on page 197 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See Appendix C: *Process State Information* on page 197 for details.

Table A-2. KNTA_USER_SECURITY_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 (30)	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_DATE	I/O	DATE	Indicates the date the record is created in the destination (Mercury IT Governance 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 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.

Table A-2. KNTA_USER_SECURITY_INT interface table [continued]

Column	Usage	Data Type	Description
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.
SOURCE_TYPE_CODE	I	VARCHAR2 (30)	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 (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.

Table A-2. KNTA_USER_SECURITY_INT interface table [continued]

Column	Usage	Data Type	Description
LOGON_IDENTIFIER	I	VARCHAR2 (30)	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 (30)	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 (40)	Specifies the SECURITY_GROUP_ NAME in KNTA_SECURITY_GROUPS.
USER_SECURITY_ACTION	I	VARCHAR2 (30)	Indicates the action for user security. Valid values are ADD or DROP.

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-3. KRSC_ORG_UNITS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Provides a unique identifier for 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 (1)	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 Appendix C: *Process State Information* on page 197 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See Appendix C: *Process State Information* on page 197 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 (30)	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-3. KRSC_ORG_UNITS_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. 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 (Mercury IT Governance 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 (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.
SOURCE_TYPE_CODE	I	VARCHAR2 (30)	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. 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.

Table A-3. KRSC_ORG_UNITS_INT interface table [continued]

Column	Usage	Data Type	Description
ORG_UNIT_NAME	I	VARCHAR2 (30)	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 (30)	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 (30)	Specifies the name of the manager.
MANAGER_LOGON_ IDENTIFIER	I	VARCHAR2 (30)	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 (30)	Specifies the code for the department.
DEPARTMENT_MEANING	I	VARCHAR2 (80)	Specifies the description of the department.
LOCATION_CODE	O	VARCHAR2 (30)	Specifies the code for the location.
LOCATION_MEANING	I	VARCHAR2 (80)	Specifies the description of the location.
CATEGORY_CODE	O	VARCHAR2 (30)	Specifies the code for the category.
CATEGORY_MEANING	I	VARCHAR2 (80)	Specifies the description of the category.

Table A-3. KRSC_ORG_UNITS_INT interface table [continued]

Column	Usage	Data Type	Description
ENABLED_FLAG	O	VARCHAR2 (1)	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 either this or ORG_UNIT_ USERNAME.
USER_DATA1 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USER_DATA20	I	VARCHAR2 (200)	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-4. KRSC_ORG_UNIT_MEMBERS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Provides a unique identifier for 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 (1)	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 Appendix C: *Process State Information* on page 197 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See Appendix C: *Process State Information* on page 197 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 (30)	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 A-4. 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 (Mercury IT Governance 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 (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.
SOURCE_TYPE_CODE	I	VARCHAR2 (30)	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.

Table A-4. KRSC_ORG_UNIT_MEMBERS_INT interface table [continued]

Column	Usage	Data Type	Description
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 (30)	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 (30)	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 (30)	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.

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-5. KCRT_REQUESTS_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	Provides a unique identifier for each transaction. 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 Appendix C: *Process State Information* on page 197 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See Appendix C: *Process State Information* on page 197 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.

Table A-5. KCRT_REQUESTS_INT interface table [continued]

Column	Usage	Data Type	Description
CREATED_USERNAME	I/O	VARCHAR2 (30)	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 (30)	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.
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 (30)	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 (80)	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.

Table A-5. KCRT_REQUESTS_INT interface table [continued]

Column	Usage	Data Type	Description
REQUEST_SUBTYPE_ NAME	I	VARCHAR2 (80)	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 (240)	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 (80)	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. If left blank, the value is derived from STATUS_NAME.
WORKFLOW_NAME	I/O	VARCHAR2 (80)	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.

Table A-5. KCRT_REQUESTS_INT interface table [continued]

Column	Usage	Data Type	Description
DEPARTMENT_CODE	O	VARCHAR2 (30)	Specifies the code for the department.
DEPARTMENT_NAME	I	VARCHAR2 (80)	Specifies the name of the department. This should be a valid MEANING from KNTA_LOOKUPS.where `LOOKUP_TYPE = 'DEPARTMENT_CODE'`.
PRIORITY_CODE	O	VARCHAR2 (30)	Specifies the user-defined priority for the request.
PRIORITY_NAME	I	VARCHAR2 (80)	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 (30)	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 (30)	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 (30)	Specifies the SECURITY_GROUP_ID (from KNTA_SECURITY_GROUPS) that should initially be assigned the request.
ASSIGNED_TO_GROUP_ID	I/O	NUMBER	Specifies the SECURITY_GROUP_ID that should initally be assigned to the request. This is normally left blank and the value is derived from ASSIGNED_TO_ GROUP_NAME.

Table A-5. KCRT_REQUESTS_INT interface table [continued]

Column	Usage	Data Type	Description
PROJECT_CODE	I	VARCHAR2 (30)	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 (30)	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 (30)	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 (1)	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 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USERS_DATA20	I	VARCHAR2 (200)	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-5. KCRT_REQUESTS_INT interface table [continued]

Column	Usage	Data Type	Description
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 (30)	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 (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.
WORKFLOW_STEP_ID	O	NUMBER	Identifies the workflow step that becomes eligible for user processing.
COMPANY	I	VARCHAR2 (30)	Identifies the name of the company associated with this request.

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-6. 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	Provides a unique identifier for 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.

Table A-6. KCRT_REQUEST_DETAILS_INT interface table [continued]

Column	Usage	Data Type	Description
PARAMETER1 VISIBLE_PARAMETER1 through PARAMETER50 VISIBLE_PARAMETER50	I	VARCHAR2 (200)	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 (80) VARCHAR2 (30)	Identifies the lookup type for each PARAMETER as well as the validation type code for each PARAMETER. 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-7. 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	Provides a unique identifier for 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.
REQUEST_TYPE_ID	I/O	NUMBER	Identifies the request type. This is normally left blank and is derived from REQUEST_TYPE_NAME.

Table A-7. KCRT_REQ_HEADER_DETAILS_INT interface table [continued]

Column	Usage	Data Type	Description
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 (200)	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 (80) VARCHAR2 (30)	Identifies the lookup type for each PARAMETER as well as 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-8. 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 sequence. This value should be the same as the parent's GROUP_ID in KCRT_ REQUEST_INT.
TRANSACTION_ID	I	NUMBER	Provides a unique identifier for each transaction.
PARENT_TRANSACTION_ID	I	NUMBER	Provides the transaction ID (from KCRT_REQUESTS_INT) of the parent table being imported.
PARENT_FIELD_TOKEN	I	VARCHAR2 (30)	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.

Table A-8. KCRT_TABLE_ENTRIES_INT interface table [continued]

Column	Usage	Data Type	Description
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 (200)	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.
LOOKUP_TYPE1 through LOOKUP_TYPE50	I	VARCHAR2 (80)	Identifies the lookup type for each PARAMETER. This is required only if user data is defined.
VALIDATION_TYPE_CODE1 through VALIDATION_TYPE_ CODE50	I	VARCHAR2 (30)	Identifies the validation type code for each PARAMETER. This is required only if user data is defined.

KCRT_FG_DEMAND_SCHEDULE_INT

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

Table A-9. 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	Provides a unique identifier for 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_MASTER_PROJ_REF_INT

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

Table A-10. 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. 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	Provides a unique identifier for 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 (200)	Creates a reference to the specified master project ID.
REF_MASTER_PROJECT_ NAME	I	VARCHAR2 (200)	Creates a reference to the specified master project name.

KCRT_FG_PROG_ISSUE_INT

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

Table A-11. 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	Provides a unique identifier for 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 (200)	Specifies the code for the escalation level.
ESCALATION_LEVEL_ MEANING	I	VARCHAR2 (200)	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 field group Program Reference on Request.

Table A-12. 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	Provides a unique identifier for 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_PROGRAM_ID	I	VARCHAR2 (200)	Creates a reference to the specified program ID.
REF_PROGRAM_NAME	I	VARCHAR2 (200)	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 field group PMO Program Resource Request.

Table A-13. 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	Provides a unique identifier for 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.
ROLE_DESCRIPTION_ CODE	I	VARCHAR2 (1800)	Provides a description of the resource's role.

KCRT_FG_PROG_RISK_INT

The KCRT_FG_PROG_RISK_INT interface table stores validation information, for each request, that is related to the field group PMO Program Risk.

Table A-14. KCRT_FG_PROG_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	Provides a unique identifier for 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 (200)	Specifies the probability code of the program risk.
PROBABILITY_MEANING	I	VARCHAR2 (200)	Specifies the description of the program risk.
RISK_IMPACT_LEVEL_ CODE	I	VARCHAR2 (200)	Specifies the code for the impact level of the program's risk.
RISK_IMPACT_LEVEL_ MEANING	I	VARCHAR2 (200)	Specifies the description of the impact level of the program's risk.

KCRT_FG_PROG_SCOPE_CHANGE_INT

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

.

Table A-15. KCRT_FG_PROG_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	Provides a unique identifier for 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 (200)	Specifies the severity code for the impact of the scope change.
IMPACT_SEVERITY_ MEANING	I	VARCHAR2 (200)	Specifies the description of the severity impact of the scope change.
CR_LEVEL_CODE	I	VARCHAR2 (200)	Speficies the code for the change request importance level of the scope change.
CR_LEVEL_MEANING	I	VARCHAR2 (200)	Specifies the description of the change request importance level of the scope change.

KCRT_FG_SLA_INT

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

Table A-16. 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	Provides a unique identifier for 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 (30)	Specifies the code for the service level agreement level.
SLA_LEVEL	I	VARCHAR2 (100)	Specifies the description for the service level agreement 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 tablestores validation information, for each request, that ais related to the field group Work Item Fields.

Table A-17. 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	Provides a unique identifier for 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.
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-17. KCRT_FG_WORK_ITEMS_INT interface table [continued]

Column	Usage	Data Type	Description
WORKLOAD_FLAG	I	VARCHAR2 (1)	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 (200)	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 (30)	Specifies the code for the category if the workload represented by this request falls under a category.
WORKLOAD_CATEGORY_ MEANING	I	VARCHAR2 (200)	Specifies the description for the category if the workload represented by this request falls under a category.
ALLOW_EXTERNAL_ UPDATE_FLAG	I	VARCHAR2 (1)	Indicates whether or not the actuals can be updated by an external system (such as Mercury Time Management™ time sheets). Valid values are: • Y • N The default value is N.
USR_SCHEDULED_START_ DATE	I	DATE	Specifies the date when the work item is scheduled 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.

Table A-17. KCRT_FG_WORK_ITEMS_INT interface table [continued]

Column	Usage	Data Type	Description
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.
ACTUAL_EFFORT	I	NUMBER	Specifies the effort (in hours). Usually equal to (duration) x (hours/day).

Table A-17. KCRT_FG_WORK_ITEMS_INT interface table [continued]

Column	Usage	Data Type	Description
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 (200)	Provides the skill name booked on this request. This must match a SKILL_NAME in KRSC_SKILLS.

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-18. KDLV_PACKAGES_INT interface table

Column	Usage	Data Type	Description
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 sequence.
PROCESS_PHASE	O	NUMBER	Indicates the current stage of the record as it is being processed. See Appendix C: *Process State Information* on page 197 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See Appendix C: *Process State Information* on page 197 for details.

Table A-18. KDLV_PACKAGES_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 (30)	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 (30)	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_BY	I	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.

Table A-18. KDLV_PACKAGES_INT interface table [continued]

Column	Usage	Data Type	Description
REQUESTED_BY_ USERNAME	I	VARCHAR2 (30)	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 (40)	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 (30)	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 (30)	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 (240)	Specifies a user-visible description of the package.
PACKAGE_TYPE_CODE	I	VARCHAR2 (30)	Provides a user-defined categorization of the package. Must be a valid LOOKUP_CODE from KNTA_LOOKUPS where `LOOKUP_TYPE = 'PACKAGE_TYPE'`.

Table A-18. KDLV_PACKAGES_INT interface table [continued]

Column	Usage	Data Type	Description
PRIORITY_CODE	I	VARCHAR2 (30)	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 (30)	Indicates the status of the package.
PROJECT_CODE	I	VARCHAR2 (30)	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 (80)	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 (1)	Indicates whether or not the interface program will release the package once it imports in into the standard Mercury Change 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.

Table A-18. KDLV_PACKAGES_INT interface table [continued]

Column	Usage	Data Type	Description
USER_DATA1 VISIBLE_USER_DATA1 through USER_DATA20 VISIBLE_USER_DATA20	I	VARCHAR2 (200)	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	I	NUMBER	Identifies the original package for this distribution package.
DISTPKG_STATUS_ MEANING	I	VARCHAR2 (80)	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 (1)	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 executed 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, as well as parameter information for the specified object type and user data for the package line.

Table A-19. 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.

Table A-19. KDLV_PACKAGE_LINES_INT interface table [continued]

Column	Usage	Data Type	Description
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 (40)	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 Appendix C: *Process State Information* on page 197 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See Appendix C: *Process State Information* on page 197 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 (30)	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.

Table A-19. KDLV_PACKAGE_LINES_INT interface table [continued]

Column	Usage	Data Type	Description
SOURCE_CODE	I	VARCHAR2 (30)	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 (80)	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 (300)	Specifies the name of the object to be processed. This value is not validated.
APP_CODE	I/O	VARCHAR2 (30)	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.

Table A-19. KDLV_PACKAGE_LINES_INT interface table [continued]

Column	Usage	Data Type	Description
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 (200)	Specifies the user-defined fields attached to the user screen. This is required only if user data is defined.
RELEASE_FLAG	I/O	VARCHAR2 (1)	Indicates whether or not the interface program will release the package once it imports in into the standard Mercury Change 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 VISIBLE_USER_DATA20	I	VARCHAR2 (200)	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	I	VARCHAR2 (300)	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 (1)	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-20. 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_ID	I	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 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.

Table A-20. KDLV_PACKAGE_NOTES_INT interface table [continued]

Column	Usage	Data Type	Description
PACKAGE_NUMBER	I	VARCHAR2 (40)	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 Appendix C: *Process State Information* on page 197 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See Appendix C: *Process State Information* on page 197 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 (30)	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 (30)	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 (1)	No longer used.

KWFL_TRANSACTIONS_INT

The KWFL_TRANSACTIONS_INT interface table is used to 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-21. KWFL_TRANSACTIONS_INT interface table

Column	Usage	Data Type	Description
TRANSACTION_ID	I	NUMBER	Provides a unique identifier for each transaction.
CREATION_DATE	I/O	DATE	Indicates the transaction date. If left blank, the current date is used.
CREATED_USERNAME	I	VARCHAR2 (30)	Identifies the USERNAME (from KNTA_ USERS) for the user performing the transaction. Supply either this or CREATED_BY.
CREATED_BY	I	NUMBER	Identifies the USER_ID (from KNTA_ USERS) for the user performing the transaction. Supply either this or 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 (30)	Identifies the USERNAME (from KNTA_ USERS) for the user performing the transaction. Supply either 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 either this or LAST_UPDATED_ USERNAME. If both are left blank, the value is derived from CREATED_USERNAME.
EVENT	I	VARCHAR2 (40)	Specifies the type of workflow transaction.

Table A-21. KWFL_TRANSACTIONS_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.
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 Appendix C: *Process State Information* on page 197 for details.
PROCESS_STATUS	O	NUMBER	Indicates the current disposition of the record. See Appendix C: *Process State Information* on page 197 for details.
SOURCE_TYPE_CODE	I	VARCHAR2 (30)	Specifies the type of external update. This should be a left blank or have a value of INTERFACE_WF.
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.
INSTANCE_SOURCE_ TYPE_CODE	I	VARCHAR2 (30)	Indicates whether or not the transaction is for a package line ('CR') or a request ('IR').
INSTANCE_SOURCE_SET_ NUMBER	I	VARCHAR2 (40)	Specifies the package number (PACKAGE_NUMBER from KDLV_ PACKAGES) or request number (REQUEST_NUMBER from KCRT_ REQUESTS). Supply either 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 either this or INSTANCE_ SOURCE_SET_NUMBER.

Table A-21. KWFL_TRANSACTIONS_INT interface table [continued]

Column	Usage	Data Type	Description
INSTANCE_SOURCE_LINE_ SEQ	I	NUMBER	Specifies the package line sequence number (SEQ from KDLV_PACKAGE_ LINES). Supply either 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 either this or INSTANCE_ SOURE_LINE_SEQ (for package lines) or INSTANCE_SOURCE_SET_ NUMBER (for requests).
WORKFLOW_STEP_NAME	I	VARCHAR2 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_ WORKFLOW_STEPS). Supply either this or WORKFLOW_ STEP_ID.
WORKFLOW_STEP_SEQ	I	VARCHAR2 (30)	Specifies the sequence number of the workflow step. Supply either this or WORKFLOW_ STEP_ID. In the case of subworkflows, the sequence numbers of the workflow steps could be in the form of 2.4.5 and so forth.
RESULT_VALUE	I	VARCHAR2 (200)	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 (200)	Indicates the result of the step. This is the result value that a user normally sees.
USER_COMMENTS	I	VARCHAR2 (200)	Specifies comments for the transaction. Any comments are appended to the notes for the package or request.
DELEGATED_TO_ USERNAME	I	VARCHAR2 (30)	Specifies the USERNAME (from KNTA_ USERS) for the user that the decision is being delegated to. Supply either this or DELEGATED_TO_ USER_ID.

Table A-21. KWFL_TRANSACTIONS_INT interface table [continued]

Column	Usage	Data Type	Description
DELEGATED_TO_USER_ID	I	NUMBER	Specifies the USER_ID (from KNTA_ USERS) for the user that the decision is being delegated to. Supply either 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 either 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.
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 (30)	Specifies the sequence number of the workflow step for the step that the package line or request should transition to. Supply either this, TO_WORKFLOW_ STEP_ID, or TO_WORKFLOW_STEP_ NAME.

Table A-21. KWFL_TRANSACTIONS_INT interface table [continued]

Column	Usage	Data Type	Description
TO_WORKFLOW_STEP_ NAME	I	VARCHAR2 (80)	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 either 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 either 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 and the values are established using the `INSERT_ROW` procedure in the `KWFL_TXN_INT` package.

Table A-22. KWFL_TXN_INT.INSERT_ROW parameters

Parameter	Usage	Data Type	Description
P_EVENT	I	VARCHAR2 (40)	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.
P_CREATED_USERNAME	I	VARCHAR2 (80)	Identifies the USERNAME (from KNTA_ USERS) for the user performing the transaction.
P_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.
P_REQUEST_NUMBER	I	VARCHAR2 (40)	Identifies the request.
P_PACKAGE_NUMBER	I	VARCHAR2 (40)	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 (80)	Specifies the name of the workflow step (STEP_NAME from KWFL_ WORKFLOW_STEPS).

Table A-22. KWFL_TXN_INT.INSERT_ROW parameters [continued]

Parameter	Usage	Data Type	Description
P_WORKFLOW_STEP_SEQ	I	VARCHAR2 (30)	Specifies the sequence number of the workflow step. In the case of 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 (200)	Indicates the result of the step. This is the result value that a user normally sees.
P_USER_COMMENTS	I	VARCHAR2 (200)	Specifies comments for the transaction. Any comments are appended to the notes for the package or request.
P_DELEGATED_TO_ USERNAME	I	VARCHAR2 (30)	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 (80)	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 (30)	Specifies the sequence number of the workflow step for the step that the package line or request should transition to.
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 (80)	Specifies the internal message name of the error that was returned. This is used mainly for debugging purposes.
O_MESSAGE	O	VARCHAR2 (1000)	Provides the error message.

Appendix B LDAP Authentication

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

The authentication steps involve:

1. The Mercury IT Governance 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. Mercury IT Governance Center does an anonymous authentication if a password is not supplied in `server.conf`.

   The `server.conf` file is described in the *System Administration Guide and Reference*.

2. Mercury IT Governance 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 the placeholder <*username*> represents 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 Mercury IT Governance 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, Mercury IT Governance Center tries to authenticate against all of them until it succeeds. If the referral option has been enabled, then Mercury IT Governance Center also queries the referral server for authentication if the user is not present in primary server.

For users who are running the Mercury IT Governance Server on a JDK 1.4 platform, Mercury IT Governance 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

In This Appendix:

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

Index