Mercury™ IT Governance Center

Open Interface Guide and Reference

Version 5.5.0

MERCURY
INTERACTIVE

Mercury Interactive Corporation
1325 Borregas Avenue
Sunnyvale, CA 94089 USA
Tel: (408) 822-5200
Fax: (408) 822-5300

If you have any comments or suggestions regarding this document, please send them via e-mail to documentation@merc-int.com.

Publication Number: OpenInterface-0304A

Table of Contents

Chapter 1 Introduction

In addition to the Web-based user interface for processing Requests, Packages, Workflows, Users, and Organizations, Mercury IT Governance (ITG) Center includes an open interface for performing key transactions. This API uses interface tables within the Mercury ITG Center database instance. Data added to these interface tables is validated and eventually imported into standard Mercury ITG tables. This generates entities that can be processed using Mercury ITG Center.

The Open Interface allows integration of third-party products with key Mercury ITG Center entities. Relevant information from these products can be used to generate Requests and Packages, perform Workflow transactions for Packages Lines and Requests, generate or update Users from a database or LDAP server, or import an organization model into Mercury ITG Center.

Note

Integration between the products in Mercury ITG 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 Change Management Packages from Request Management Requests.

About This Document

This document provides information and details for using the Open Interface. Each chapter covers specific topics on the Open Interface and includes the following information:

Organization Unit Open Interface Details the data model and process used to import organization information from an existing database.

User Open Interface Details the data model and process used to import user information from an existing database.

Request Open Interface Details the data model and process used to import Request information from an existing database.

Package Open Interface Details the data model and process used to import Package information from an existing database.

Workflow Transaction Open Interface Details the data model and process used to import Workflow transaction information from an existing database.

Intended Audience

The intended audience for this document include:

- Administrators responsible for importing users into Mercury ITG Center from another data source.
- Administrators responsible for importing organization models into Mercury ITG Center.
- Advanced configuration experts who wish to process transactions using the API, rather than use the standard interface.
- Advanced users who wish to import legacy information into Mercury ITG Center.

Note

You must have the proper Power license to access the screens and windows described in this document.

Document Conventions

Table 1-1 lists the types of conventions used in this document.

Table 1-1. Document conventions

Convention	Description	Example
Button, menu, tabs	Names of interface components that can be clicked (such as buttons, menus, and tabs) are shown in bold.	**Apply** button
Fields, Windows, Pages	Names of windows, fields, and pages are shown as displayed.	New Request window

`Code`	Code input and output are shown as displayed.	`CauchoConfigFile`
		`C:/`*`ITG_Center`*`/conf/resin.conf`

Link	Linked URLs, filenames, and cross references are shown as blue italicized text.	*www.merc-int.com*
Variable	Variables are shown as italicized text.	*ITG_Home*/bin directory
Note	Used to identify note boxes that contain additional information.	Note
Caution	Used to identify caution boxes that contain important information. Follow the instructions in all caution boxes; failure to do so may result in loss of data.	Caution
Example	Used to identify example boxes that contain examples of related procedure.	Example

Additional Resources

Mercury Interactive provides the following additional resources to help you successfully use the Open Interface:

- *Related Documentation*

- *Customer Support*
- *Education Services*

Related Documentation

The Library includes additional documents related to the topics discussed in this guide. Access the Library through the Mercury ITG Center online help.

Using the Workbench This document explains how to navigate through the Workbench interface.

Reports Guide and Reference This document provides details for running reports.

Configuring a Request Resolution System This document provides instructions for configuring a Request resolution system. This includes requirements gathering, modeling your processes in a Workflow, defining a Request Type to be integrated with the Workflow, and rolling out this system to your users.

Managing Your Resources (Resource Management) This document provides instructions for managing your Resources. This includes modeling your Organizations, setting up Resource Categories and Calendars, creating Staffing Profiles and Resource Pools, and visualizing Resource load and capacity across the system.

Customer Support

Customer support and downloads for the Mercury ITG Center and additional product information can be accessed from the Mercury Interactive Support Web site at *http://support.mercuryinteractive.com*.

Education Services

Mercury Interactive provides a complete training curriculum to help you achieve optimal results using the Mercury IT Governance Center. For more information, visit the Education Services Web site at *http://www.merc-training.com/main/ITG*.

Chapter 2 Organization Unit Open Interface

Mercury ITG Center includes an interface for importing organization information. This open interface can import organizational models from third party systems including LDAP databases, home grown organization modeling systems, and HR systems. Using this interface, an administrator can periodically synchronize the organizational model in Mercury ITG Center with the authoritative data source within their company.

In general, the synchronization process will involve importing attributes of the various resources. The following sections detail the data model and process used to import organization information from an existing database:

- *Data Model*
- *Running the Organization Open Interface for a Simple Import*
- *Running the Organization Unit Open Interface for an LDAP Import*
- *Post-Import Activities*

Note

When associating users with an Organization Model, the users/resources must exist in the system. Mercury Interactive recommends using one of two approaches to ensure a successful Organization Unit import:

- First import the users using the User Open Interface to ensure that all users are in the system. Then import the Organization information using the Org Unit Open Interface.

  OR

- Populate the KNTA_USERS_INT table (as if preparing for a User import) and the tables for the Org Unit open interface, then run the Org Unit Open Interface program. This will add the users to the system and link them to the appropriate Org Units.

For more information on mapping your Organization Model, see *Managing Your Resources (Resource Management)*.

Data Model

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

- KRSC_ORG_UNITS_INT:
  Used to define the Organization Unit. For more information, see *Table 2-1*.
- KRSC_ORG_UNIT_MEMBERS_INT:
  Used to specify user membership in the Organization Units. For more information, see *Table 2-2*.
- KNTA_USERS_INT:
  Used when adding additional users into the system; this is the same table used in the User Open Interface. For more information, see *Table 3-1 on page 32*.

Note

Each table includes a "Required for Import" column. This column, and other related requirements, are discussed in *"Loading Data into the Interface Tables"* on page 14.

Table 2-1. KRSC_ORG_UNITS_INT

Column	Required for Import?	DataType	Description
TRANSACTION_ID	N	NUMBER	System-generated identifier.
GROUP_ID	Y	NUMBER	A value from KNTA_INTERFACE_GROUPS_S. This is used to group together all records that have to be processed together.
EXISTS_FLAG	N	VARCHAR2 (1)	Used to indicate if the Org unit already exists in the system.
PROCESS_PHASE	N	NUMBER	Indicates the current stage of the record as it is being processed.
PROCESS_STATUS	N	NUMBER	Indicates the current disposition of the record. The value should initially be set to 1.

Table 2-1. KRSC_ORG_UNITS_INT

Column	Required for Import?	DataType	Description
CREATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from CREATED_BY_USERNAME or set to the user currently running the report.
CREATED_BY_USERNAME	N	VARCHAR2(30)	The USERNAME from the KNTA_USERS table for the user that is performing the transaction. This will only be used if the CREATED_BY is null.
CREATION_DATE	N	DATE	The date that the transaction is performed.
DEST_CREATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from CREATED_BY_USERNAME or set to the user currently running the report.
DEST_CREATION_DATE	N	DATE	The date that the transaction is performed.
DEST_LAST UPDATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that last performed the transaction.
DEST_LAST_UPDATE_ DATE	N	DATE	The date that the transaction was last updated.
DEST_ENTITY_UPD_DATE	N	DATE	The date that the transaction is performed.
SOURCE	N	VARCHAR2(10 0)	This is an optional, non-validated field that can be used to indicate the exact source of the Organization Information (the name of the Third Party Application, the text string “Conversion”, etc.)
SOURCE_TYPE_CODE	N	VARCHAR2(30)	For records that have been updated by an interface or migrator, indicates the type of external update (specific interface or migrator name, etc.)

Table 2-1. KRSC_ORG_UNITS_INT

Column	Required for Import?	DataType	Description
ORG_UNIT_ID	N	NUMBER	Internal identifier of the Org Unit. This should normally be left blank.
ORG_UNIT_NAME	Y	VARCHAR2(30)	The name of the Org Unit.
PARENT_ORG_UNIT_ID	N	NUMBER	Internal identifier of the Org Unit’s parent Org Unit. This should normally be left blank.
PARENT_ORG_UNIT_NAME	N	VARCHAR2(30)	The name of the Org Unit’s parent Org Unit. If none exists, then the Org Unit will appear as a top-level unit in the Organization Model.
MANAGER_ID	N	NUMBER	Internal identifier of the manager associated with the Org Unit. This should normally be left blank.
MANAGER_USERNAME	N	VARCHAR2(30)	The name of the manager associated with the Org Unit.
MANAGER_LOGON_IDENTIFIER	N	VARCHAR2(30)	The logon ID of the manager associated with the Org Unit. Used when Mercury ITG Center is started in LOGON_ID mode.
DEPARTMENT_CODE	N	VARCHAR2(30)	Code of the Department to which the Org Unit is associated.
DEPARTMENT_MEANING	N	VARCHAR2(80)	Text that is displayed to the user. This is the Department to which the Org Unit is associated. When populating the interface tables for an import, it is only necessary to specify the MEANING (the CODE is automatically derived).
LOCATION_CODE	N	VARCHAR2(30)	Code of the Location associated with the Org Unit.
LOCATION_MEANING	N	VARCHAR2(80)	Text that is displayed to the user. This is the Location associated with the Org Unit. When populating the interface tables for an import, it is only necessary to specify the MEANING (the CODE is automatically derived).
CATEGORY_CODE	N	VARCHAR2(30)	Code for the Category associated with an Org Unit.

Table 2-1. KRSC_ORG_UNITS_INT

Column	Required for Import?	DataType	Description
CATEGORY_MEANING	N	VARCHAR2(80)	Text that is displayed to the user. This is the Category associated with an Org Unit. When populating the interface tables for an import, it is only necessary to specify the MEANING (the CODE is automatically derived).
ENABLED_FLAG	N	VARCHAR2(1)	Specifies whether the Org Unit is enabled upon import.
USER_DATA_SET_ CONTEXT_ID	N	NUMBER	Parameter set context identifier for the User Data fields.
USER_DATA1 - 20	N	VARCHAR2(20 0)	These columns hold the user-defined fields attached to the user screen. Values should be entered only if User Data is defined. These columns will not be validated or defaulted.
VISIBLE_USER_DATA1 - 20	N	VARCHAR2(20 0)	These columns hold the user-defined fields attached to the user screen. Values should be entered only if User Data is defined. These columns will not be validated or defaulted.

Table 2-2. KRSC_ORG_UNIT_MEMBERS_INT

Column	Required for Import?	DataType	Description
TRANSACTION_ID	N	NUMBER	System-generated identifier.
GROUP_ID	Y	NUMBER	A value from KNTA_INTERFACE_GROUPS_S. This is used to group together all records that have to be processed together.
EXISTS_FLAG	N	VARCHAR2 (1)	Used to indicate if the Org Unit already exists in the system.
PROCESS_PHASE	N	NUMBER	Indicates the current stage of the record as it is being processed.
PROCESS_STATUS	N	NUMBER	Indicates the current disposition of the record. The value should initially be set to 1.

Table 2-2. KRSC_ORG_UNIT_MEMBERS_INT

Column	Required for Import?	DataType	Description
CREATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from CREATED_BY_USERNAME or set to the user currently running the report.
CREATED_BY_USERNAME	N	VARCHAR2(30)	The USERNAME from the KNTA_USERS table for the user that is performing the transaction. This will only be used if the CREATED_BY is null.
CREATION_DATE	N	DATE	The date that the transaction is performed.
DEST_CREATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from CREATED_BY_USERNAME or set to the user currently running the report.
DEST_CREATION_DATE	N	DATE	The date that the transaction is performed.
DEST_LAST UPDATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that last performed the transaction.
DEST_LAST_UPDATE_ DATE	N	DATE	The date that the transaction was last updated.
DEST_ENTITY_UPD_DATE	N	DATE	The date that the transaction is performed.
SOURCE	N	VARCHAR2(10 0)	This is an optional, non-validated field that can be used to indicate the exact source of the Organization Information (the name of the Third Party Application, the text string “Conversion”, etc.)
SOURCE_TYPE_CODE	N	VARCHAR2(30)	For records that have been updated by an interface or migrator, indicates the type of external update (specific interface or migrator name, etc.)

Table 2-2. KRSC_ORG_UNIT_MEMBERS_INT

Column	Required for Import?	DataType	Description
ORG_UNIT_MEMBER_ID	N	NUMBER	Internal identifier of the Org Unit member. This should normally be left blank.
ORG_UNIT_ID	N	NUMBER	Internal identifier of the Org Unit. This should normally be left blank.
ORG_UNIT_NAME	Y	VARCHAR2(30)	The name of the Org Unit.
USER_ID	N	NUMBER	Internal identifier of the user to be associated with the Org Unit.
USERNAME	Y	VARCHAR2(30)	The name of the user to be associated with the Org Unit. Either this column or the LOGON_IDENTIFIER are required for import.
LOGON_IDENTIFIER	Y (See Notes)	VARCHAR2(30)	The logon ID of the user. Used when Mercury ITG Center is started in LOGON_ID mode. Either this column or the USERNAME are required for import.

Running the Organization Open Interface for a Simple Import

Importing an Organization Model from an external database involves the following steps.

1. Loading the users/resources into Mercury ITG Center. This can be done by running the User Open Interface. For more information, see *"Running the User Open Interface for a Simple Import"* on page 41. It is also possible to populate the KNTA_USERS_INT table and run the Org Unit Open Interface program. For more information, see *Loading Data into the Interface Tables*.

2. *Loading Data into the Interface Tables*. Populate open interface tables with data before running the import program. This population can be done through any means supported by the Oracle database (such as SQL Loader). This process and the requirements for this population depends on the data that will be imported.

3. *The "Run ITG Organization Unit Interface" Program*. The Run ITG Organization Unit Interface program is defined as a standard Report Type and launched through the Request Management, Change Management, or Project Management Reports screen.

Loading Data into the Interface Tables

The initial step in importing Organization Units into Mercury ITG Center is populating the open interface table. This 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.

Certain columns in the KRSC_ORG_UNITS_INT and KRSC_ORG_UNIT_MEMBERS_INT tables must be populated for a simple user import. Additionally, if users will be added to the system, the required fields in the KNTA_USERS_INT table will need to be populated.

Note

When importing an Organization Unit from an LDAP server, this step can be skipped. For information on importing users from an LDAP server, see *""Run ITG Organization Unit Interface" Program for LDAP Import"* on page 27.

Table 2-3. KRSC_ORG_UNITS_INT Columns required for Organization import

Column	Required for Import?	DataType	Description
GROUP_ID	Y	NUMBER	A value from KNTA_INTERFACE_GROUPS_S. This is used to group together all records that have to be processed together.
ORG_UNIT_NAME	Y	VARCHAR2(30)	The name of the Org Unit.

Table 2-4. KRSC_ORG_UNIT_MEMBERS_INT Columns required for Organization import.

Column	Required for Import?	DataType	Description
GROUP_ID	Y	NUMBER	A value from KNTA_INTERFACE_GROUPS_S. This is used to group together all records that have to be processed together. The GROUP_ID should match the one used in the KRSC_ORG_UNITS_INT table.
ORG_UNIT_NAME	Y	VARCHAR2(30)	The name of the Org Unit.
USERNAME	Y (See Notes)	VARCHAR2(30)	The name of the user to be associated with the Org Unit. Either this column or the LOGON_IDENTIFIER are required for import.
LOGON_IDENTIFIER	Y (See Notes)	VARCHAR2(30)	The logon ID of the user. Used when Mercury ITG Center is started in LOGON_ID mode. Either this column or the USERNAME are required for import.

*Table 2-5. KNTA_USERS_INT Columns Required for Org Unit Import ** (see note below table)*

Column	Required for Import?	DataType	Description
GROUP_ID	Y	NUMBER	A value from KNTA_INTERFACE_GROUPS_S. This is used to group together all records that have to be processed together. Use only one group ID for each batch of imported users when running the User Open Interface program.
USERNAME	Y	VARCHAR2(30)	The username used to Logon.

*Table 2-5. KNTA_USERS_INT Columns Required for Org Unit Import ** (see note below table)*

Column	Required for Import?	DataType	Description
LOGON_ID	N (see Description)	NUMBER	Populate either this column or the USERNAME, depending 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. If LOGON_METHOD = LOGON_ID, the LOGON_ID column must be populated.
DEST_USERNAME	Y	VARCHAR2(30)	The username that will be used to Logon.
FIRST_NAME	Y (see Description)	VARCHAR2(30)	The user's first name. This field is required only when creating a new user. It is not required when re-importing an existing user.
LAST_NAME	Y (see Description)	VARCHAR2(30)	The user's last name. This field is required only when creating a new user. It is not required when re-importing an existing user.

Note

It is only necessary to populate the KNTA_USERS_INT table when adding new users.

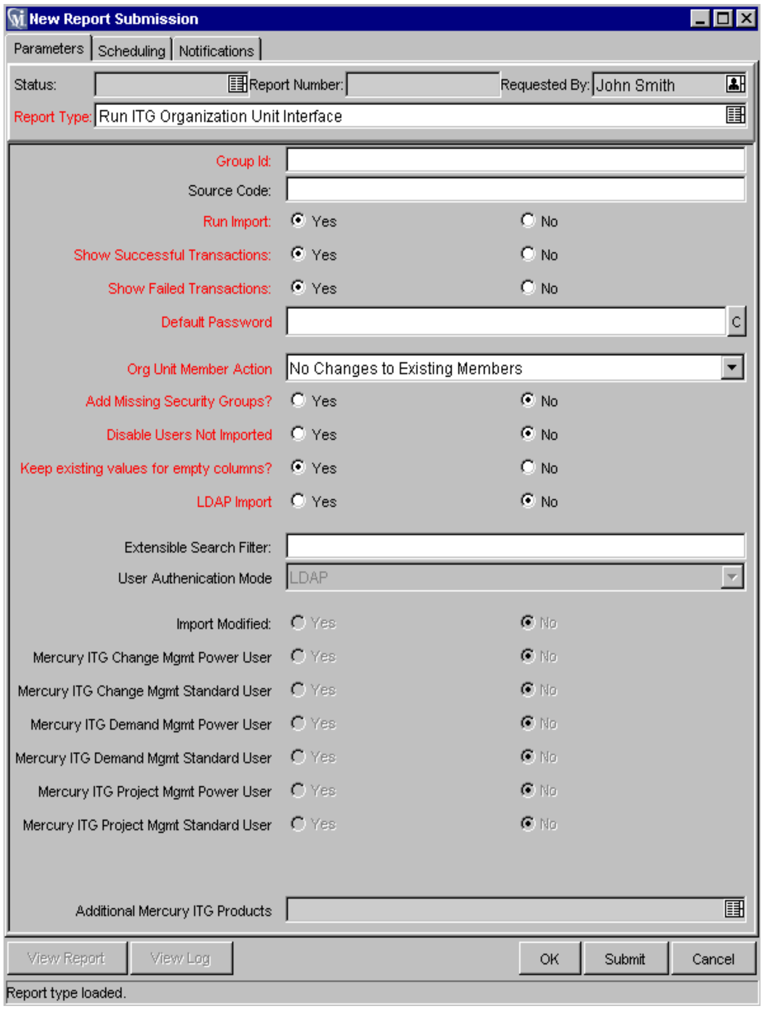
The "Run ITG Organization Unit Interface" Program

To import data from the interface tables, the "Run ITG Organization Unit Interface" program must be used. This program is defined as a standard Report Type and launched through either the Request Management, Change Management, or Project Management Reports screen.

Note

The “Run ITG Organization Unit Interface” program automatically imports relevant Organization information stored in the LDAP server to the open interface table. For more information, see *“Running the Organization Unit Open Interface for an LDAP Import”* on page 21.

For more information on using Report Types and launching Reports, see *Reports Guide and Reference* and respective product manuals.



Status:
Report Number:
Requested By: John Smith
Report Type: Run ITG Organization Unit Interface

Group Id:		
Source Code:		
Run Import:	● Yes	○ No
Show Successful Transactions:	● Yes	○ No
Show Failed Transactions:	● Yes	○ No
Default Password		
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
LDAP Import	○ Yes	● No
Extensible Search Filter:		
User Authenication Mode	LDAP	
Import Modified:	○ Yes	● No
Mercury ITG Change Mgmt Power User	○ Yes	● No
Mercury ITG Change Mgmt Standard User	○ Yes	● No
Mercury ITG Demand Mgmt Power User	○ Yes	● No
Mercury ITG Demand Mgmt Standard User	○ Yes	● No
Mercury ITG Project Mgmt Power User	○ Yes	● No
Mercury ITG Project Mgmt Standard User	○ Yes	● No
Additional Mercury ITG Products		

Report type loaded.

Figure 2-1 Report-Run ITG Organization Unit Interface

When executed, the “Run ITG Organization Unit Interface” program:

- Queries the KRSC_ORG_UNITS_INT interface table for active records matching the given selection criteria.
- Queries the KRSC_ORG_UNIT_MEMBERS_INT table.
- Queries the KNTA_USERS_INT table.
- Validates the Organization information.
- Imports validated Organization Units, Organization Unit members, and any new users into the Mercury ITG 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 they encountered.

Parameters

The “Run ITG Organization Unit Interface” program has several parameters for controlling the behavior of the program execution. These parameters are defined in *Table 2-6*.

Table 2-6. Run ITG Organization Unit Interface Parameters

Name	Type	Required	Description
Group ID	Text Field	Y	Runs the interface for a specific Group ID. The interface program will only look for records with this value in the GROUP_ID column. This is useful when importing a batch of Users.
Source Code	Text Field	N	Used to indicate the source from which the Org Units are being imported. In the case of a LDAP import, it is set to LDAP_IMPORT.
Run Import	Radio Button	Y	If set to **Yes**, the program processes the records in the interface table and tries to import them. If set to **No**, the program simply reports on the records in the interface table. This option is useful when auditing prior executions of the open interface.
Show Successful Transactions	Radio Button	Y	Shows successfully imported Org Units.
Show Failed Transactions	Radio Button	Y	Shows Org Units that were not successfully imported.
Default Password	Password Field	Y	Sets a default password for all users being associated with an organization, if they are not LDAP users.

Name	Type	Required	Description
Org Unit Member Action	Drop Down List	Y	Used to specify how the Org Unit membership is managed during the import for existing Org Units. Select one of the following options: • **No Changes to Existing Members** - The import does not add or remove any members in an existing Org Unit. • **Replace All Existing Members** - Removes all members of the Org Unit and replaces them with the members specified in the KRSC_ORG_UNIT_MEMBERS_INT table. • **Replace LDAP Imported Members** - Removes all members of the Org Unit who are associated using LDAP and replaces them with members associated with the Org Unit on the LDAP server. Other members, who have been added manually using the standard interface or by a separate open interface import, will not be altered.
Add Missing Security Groups	Radio Button	Y	Selects whether to create Security Groups. Selecting **Yes** will: • Request Management Security Groups with the same names as the Organization Units. • Add the Org Unit members to the Security Groups (if **Replace All Existing Members** is selected from the Org Unit Member Action field).
Disable Users Not Imported	Radio Button	Y	Disables all existing users currently in the Mercury ITG system (in the specified Org Unit). This happens prior to the import.
Keep existing values for empty columns	Radio Button	Y	Keep existing values on updated org units and users when interface table columns are empty.
LDAP Import	Radio Button	Y	This indicates whether the Org Units are being imported from a LDAP server. Selecting **Yes** enables multiple required fields. This field is enabled in the server.conf file. It must be set to LDAP or a comma-delimited list. The appropriate LDAP parameters should be specified. Set this to **Yes** if the authentication mode in the server.conf contains LDAP or NTLM.
Extensible Search Filter	Text Field	N	Used to import Org Units from the LDAP server which satisfy a particular LDAP search filter. If no search filter is specified, all appropriate LDAP org units will be imported. Refer to the LDAP documentation for instructions on searching for users based on specific filters. A few examples are provided following this table.

Name	Type	Required	Description
User Authentication Mode	Drop Down List	N	Specifies how users are authenticated upon their logon. For example, if LDAP is selected, the user authentication occurs on the LDAP server.
Import Modified	Radio Button	N	Select **Yes** to import only users who have been modified in the LDAP server since the last import.
Mercury ITG Change Mgmt Power User	Radio Button	N	Establishes a Change Management Power License account for the imported users.
Mercury ITG Change Mgmt User	Radio Button	N	Establishes a Change Management Standard License account for the imported users.
Mercury ITG Demand Mgmt Power User	Radio Button	N	Establishes a Demand Management Power License account for the imported users.
Mercury ITG Demand Mgmt Standard User	Radio Button	N	Establishes a Demand Management Standard License account for the imported users.
Mercury ITG Project Mgmt Power User	Radio Button	N	Establishes a Project Management Power License account for the imported users.
Mercury ITG Project Mgmt Standard User	Radio Button	N	Establishes a Project Management Standard License account for the imported users.
Additional Mercury ITG Products	Auto-complete list	N	Establish an account of the selected product (Portfolio Management, Program Management, Time Management) for the imported users.

Example

Examples of Extensible Search Filter values and their functional results:

- The following filter returns any entries in which the o (organization) attribute exactly matches ITG and any entries in which o=ITG is one of the components of the DN:

  (o:dn:=ITG)
- The following filter returns any entries in which a DN component with a syntax appropriate to the given matching rule matches ITG :

  (:dn:1.2.3.4.5.8:=ITG)

  The matching rule given by the OID 1.2.3.4.5.8 should be used.
- The following filter returns any entries in which the ou (Organization Unit) attribute exactly matches “My Org Unit” and any entries in which ou= “My Org Unit” is one of the components of the DN

  (ou:dn:=My Org Unit)

Note

All users imported and associated with the Organization Unit using the “Run ITG Organization Unit Interface” report will have the same user licenses.

Running the Organization Unit Open Interface for an LDAP Import

To import data from an LDAP server, the “Run ITG Organization Unit Interface” program must be used. This program is defined as a standard Report Type and launched through the Request Management, Change Management, or Project Management Reports screen.

The "Run ITG Organization Unit Interface" program automatically imports relevant user information stored in the LDAP server into the open interface table. It is possible to exercise more control over this process by mapping LDAP attributes to users in the LdapAttribute.conf file. For more information, see *"Mapping LDAP Attributes to Users and Organization Units"* on page 23.

If user and organization information is imported from an LDAP server, ensure that LDAP Import is set to **Yes** in the New Report Submission window. This enables other required fields, ensuring a smooth import into Mercury ITG Center.

The following sections discuss the process for importing user information using the "Run ITG Organization Unit Interface" program:

- *Prepare for an LDAP User Import*
- *"Run ITG Organization Unit Interface" Program for LDAP Import*
- *LDAP Authentication*

Prepare for an LDAP User Import

The process for importing users from an LDAP server differs slightly from the simple user import process. The following sections detail some additional preparation required for an LDAP user import.

- *Org Unit Member Action for an LDAP Import*
- *Mapping LDAP Attributes to Users and Organization Units*
- *Mercury ITG Server Configuration*

Org Unit Member Action for an LDAP Import

Similar to a simple Organization Unit import, determine which users to link to the Organization Unit. Select one of the following options from the Org Unit Member Action field:

- **No Changes to Existing Members** - The import does not add or remove any members in an existing Org Unit.
- **Replace All Existing Members** - Removes all members of the Org Unit and replaces them with the members specified in the KRSC_ORG_UNIT_MEMBERS_INT table.
- **Replace LDAP Imported Members** - Removes all members of the Org Unit who are associated using LDAP and replaces them with members in the Org Unit on the LDAP server. Other members, who have been added

manually using the standard interface or by a separate open interface import, will not be altered.

Mapping LDAP Attributes to Users and Organization Units

It is possible to map the attributes on the LDAP Server to attributes used by the Mercury ITG Server. Some of this mapping occurs by default, but it is possible to exercise greater control by mapping the attributes in the following file:

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

Table 2-7 lists the attributes and provides some contextual background for each parameter. Mercury ITG 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.

Table 2-7. LdapAttribute.conf parameters

LDAP Attribute	Description
LDAP_USER_ID	Enter the attribute which can be used to obtain the User ID on the LDAP server. LDAP_USER_ID=uid This is used to resolve the users in the Ldap server. It is mapped to unique attribute that determines a user.
LDAP_LOGON_ID	Enter the attribute which can be used to obtain the logon ID from the LDAP server. LDAP_LOGON_ID=uid Note: This needs to be different from the LDAP_USER_ID if the LOGON_METHOD = LOGON_ID in the server.conf file.
LDAP_GROUP_NAME	Enter the attribute which can be used to obtain the group names from the LDAP server. LDAP_GROUP_NAME=cn
LDAP_ORG_UNIT_NAME	Enter the attribute which can be used to obtain the Org Units from the LDAP server. LDAP_ORG_UNIT_NAME=ou
LDAP_STATIC_GROUP_ MEMBERS	Enter the attribute which identifies the Static group members on the LDAP server. LDAP_STATIC_GROUP_MEMBERS=uniquemember This parameter and the LDAP_DYNAMIC_GROUP_MEMBERS parameter are used to determine the attribute that indicates the members in the Group. In Netscape, there can be STATIC and DYNAMIC Groups; hence there are two different parameters. In Active Directory, these would map to the same attribute (member).

Table 2-7. LdapAttribute.conf parameters

LDAP Attribute	Description
LDAP_DYNAMIC_GROUP_ MEMBERS	Enter the attribute which identifies the Dynamic group members on the LDAP server. LDAP_DYNAMIC_GROUP_MEMBERS=memberurl This parameter and the LDAP_STATIC_GROUP_MEMBERS parameter are used to determine the attribute that indicates the members in the Group. In Netscape, there can be STATIC and DYNAMIC Groups; hence there are two different parameters. In Active Directory, these would map to the same attribute (member)
LDAP_OBJECTCLASS	Enter the attribute which identifies the objectclass attribute on the LDAP server. LDAP_OBJECTCLASS=objectclass This parameter determines what the objectclass is called on the LDAP server. This will always map to “objectclass”.
LDAP_USER_OBJECTCLASS	Enter the attribute which identifies the objectclass attribute for a user on the LDAP server. LDAP_USER_OBJECTCLASS=person This parameter determines the objectclass that determines if a resource is a user. It is defaulted to “person”.
LDAP_GROUP_OBJECTCLASS	Enter the attribute which identifies the objectclass attribute for a group on the LDAP server. LDAP_GROUP_OBJECTCLASS=groupofuniquenames This parameter determines the objectclass that determines if a resource is a group.
LDAP_MODIFY_TIMESTAMP	Enter the attribute which keeps track of the last modified time for an object on the LDAP server. LDAP_MODIFY_TIMESTAMP=modifyTimeStamp This parameter is mapped to attribute on the LDAP server that keeps track of the modified time for that entity.
LDAP_TIME_FORMAT	Enter the attribute which keeps track of the Time format followed by the LDAP server. Format for Netscape LDAP Servers: yyyyMMddHHmmss'Z' Format for Active Directory Servers: yyyyMMddHHmmss'.0Z' LDAP_TIME_FORMAT=yyyyMMddHHmmss'Z' This parameter gives Mercury ITG Center the information of the Time Format the LDAP server follows.

Table 2-7. LdapAttribute.conf parameters

LDAP Attribute	Description

KNTA_USERS_INT	Denotes the table name to which the import would take place. Then the format is:
	
	ColumnName= LDAP Attribute.
	
	Thus any column could be mapped to any LDAP Attribute.
	
	Map both VISIBLE_USER_DATA and USER_DATA columns always. To disable a default mapping below, either comment out or delete the mapping line.
	
	USERNAME=sAMAccountName
	FIRST_NAME=givenname
	LAST_NAME=sn
	EMAIL_ADDRESS=mail
	PHONE_NUMBER=telephonenumber
	DEPARTMENT_MEANING=departmentNumber
	LOCATION_MEANING=locality
	MANAGER_USERNAME=managerUSER_DATA1=mail**
	VISIBLE_USER_DATA1=mail**
	** These are added in as an example.

VISIBLE_USER_DATA and USER_DATA	Map both VISIBLE_USER_DATA and USER_DATA columns always.

Note

The ORG_UNIT_NAME and PARENT_ORG_UNIT_NAME columns should not be mapped in the LdapAttribute.conf file. Those attributes are specified in the KRSC_ORG_UNITS_INT table.

Mercury ITG Server Configuration

The following Mercury ITG Server parameters need to be considered when performing a user import from an LDAP server. These parameters are set in the server.conf file located in the following directory:

```
ITG_Home/server.conf
```

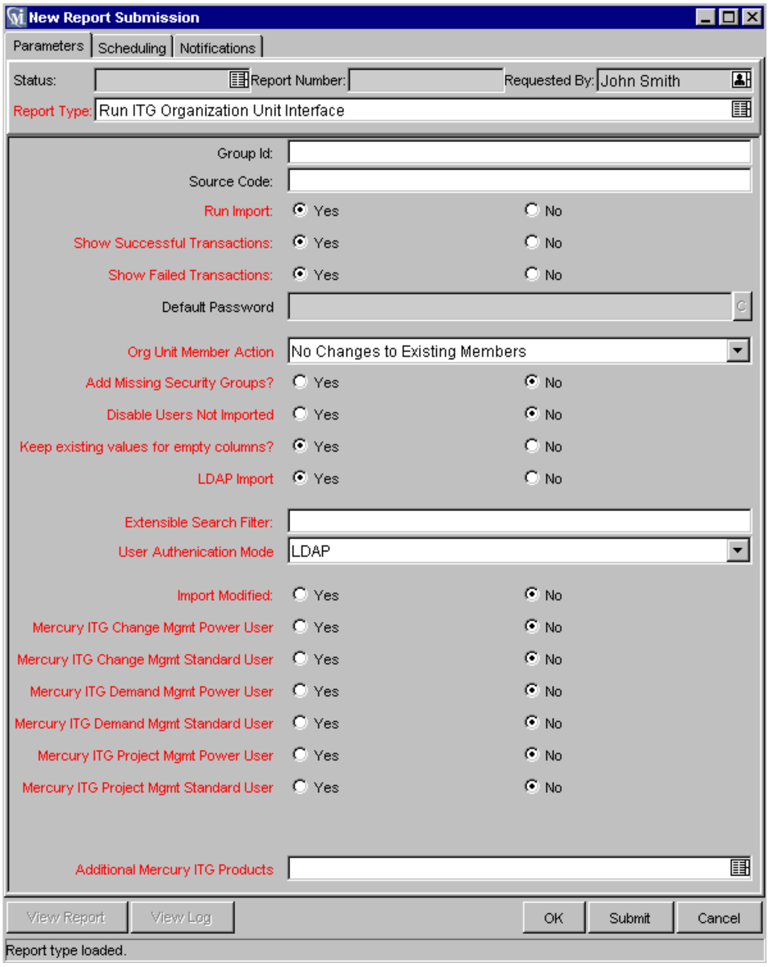
Table 2-8 lists the attributes and provides some contextual background for each parameter. After changing these parameters, stop and restart the Mercury ITG Server.

Table 2-8. server.conf Parameters

Parameter	Description
Authentication_Mode	This is required for the Mercury ITG Server to determine what method to use for authenticating users. It currently defaults to the KINTANA value. Mixed values can be separated with a comma (LDAP, KINTANA). Valid values are: KINTANA, LDAP, SITEMINDER, NTLM Example: AUTHENTICATION_MODE=KINTANA, LDAP, SITEMINDER
LDAP_Server_URL	This can be a comma-delimited list of LDAP URLs and the Mercury ITG Server will query them in the order in which they are specified. If no port number is specified, then the default port 389 will be used. Example: LDAP_URL=ldap://ldap.mydomain.com:389, ldap://ldap2.mydomain.com
LDAP_BASE_DN	This specifies the base in the LDAP server from where to start the search. If this is not specified, then the LDAP server will be queried to determine the base. Example LDAP_BASE_DN=mydomain.com
KINTANA_LDAP_ID	Mercury ITG Center account on the LDAP server. This is a required field if the LDAP server is being used at all. This will be used by the Mercury ITG Server to bind to the LDAP server.
KINTANA_LDAP_ PASSWORD	Mercury ITG Center password on the LDAP server. This is automatically encrypted by the server configuration utility. Leave this blank for an anonymous authentication. Note: To manually edit the server.conf file, surround the encrypted password with #!# delimiters. Example: KINTANA_LDAP_PASSWORD= #!#0123456789abcdefghij9876543210AB#!# Example: KINTANA_LDAP_PASSWORD=#!##!#
LDAP_SSL_PORT	SSL Port number on the LDAP server. If the LDAP server is SSL-enabled, then this should be set to the SSL port of the LDAP server (which is usually 636). This port will be used to carry out all secure transactions. If this field is not set, then all transactions will be carried out over the port specified in the parameter.

"Run ITG Organization Unit Interface" Program for LDAP Import

After determining the strategy for assigning Security Groups to the users and preparing for the import, run the import using the "Run ITG Organization Unit Interface" program. This program is launched through the Request Management, Change Management, or Project Management Reports screen.



Status: Report Number: Requested By: John Smith

Report Type: Run ITG Organization Unit Interface

Group Id:

Source Code:

Run Import: ● Yes ○ No

Show Successful Transactions: ● Yes ○ No

Show Failed Transactions: ● Yes ○ No

Default Password

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

LDAP Import ● Yes ○ No

Extensible Search Filter:

User Authenication Mode: LDAP

Import Modified: ○ Yes ● No

Mercury ITG Change Mgmt Power User ○ Yes ● No

Mercury ITG Change Mgmt Standard User ○ Yes ● No

Mercury ITG Demand Mgmt Power User ○ Yes ● No

Mercury ITG Demand Mgmt Standard User ○ Yes ● No

Mercury ITG Project Mgmt Power User ○ Yes ● No

Mercury ITG Project Mgmt Standard User ○ Yes ● No

Additional Mercury ITG Products

Report type loaded.

Figure 2-2 Report-Run ITG Organization Unit Interface for LDAP

When executed, the "Run ITG Organization Unit Interface" program:

- Populates the interface tables with records from the LDAP server.
- Validates the user information.
- Imports validated Organization Units and Organization Unit members into the Mercury ITG 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 import from the Workbench:

1. Open the Report window in either Request Management, Change Management or Project Management.

2. Select **Run ITG Organization Unit Interface** from the Report Type auto-complete list. The parameters for this program appear in the Report window.

3. Enter the parameters and select the import options. Each parameter on this Report is defined in *Table 2-6 on page 18*. For an LDAP import, pay special attention to the following fields:

   - LDAP Import: Set this field to **Yes**. This will enable the other LDAP-import related parameters.

   - Search Filter

   - User Authentication Mode

   - Import Modified

   - Standard and Power User license information

4. Click **Submit**.

The report and the log can then viewed to analyze the import results.

LDAP Authentication

Mercury ITG Center uses simple authentication to authenticate against any LDAP v.3. compliant LDAP server.

The authentication steps involves:

1. The Mercury ITG Server binds to the LDAP server using the credentials supplied in the KINTANA_LDAP_ID and KINTANA_LDAP_PASSWORD server parameters. This step is optional. Mercury ITG Center will do an anonymous authentication if a password is not supplied in server.conf.

2. Mercury ITG Center tries to obtain the distinguished name of the user by supplying a search filter to the LDAP server in the form (uid=username).

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 ITG 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_server_URL server parameter, Mercury ITG Center will try to authenticate against all of them until it succeeds. If the referral option has been enabled, then Mercury ITG Center will also query the referral server for authentication if the user is not present in primary server.

For users who are running the Mercury ITG Server on a JDK1.3 platform, Mercury ITG Center also supports LDAP authentication over SSL using passwords. To enable the SSL option, the LDAP_SSL_PORT server parameter should be set to the SSL port of the LDAP server.

Post-Import Activities

Following the user import, view the results of the import. From the Report screen, click **View Report** to open the report. This report will identify any errors with the import. After reviewing the import results, proceed with the following activities:

- *Correcting Failures*
- *Purging the Interface Tables*

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 user interface.

For users that fail to import, corrective actions are required. The first step involves 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 failures may occur due to a mapping problem between the source data and existing data in Mercury ITG Center.

Other failures may be due to missing information that cannot be defaulted. For example, users require a Username. If the Username columns were left blank for records in the User 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.

Note

During the initial implementation of the Open Interface, the mapping between the non-Mercury ITG Center source and Mercury ITG 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 ITG Center.

Purging the Interface Tables

All interface tables are automatically purged by the Purge service. The purging process depends on two parameters in the server.conf file:

- ENABLE_INTERFACE_CLEANUP - enables or disables the Purge process.
- DAYS_TO_KEEP_INTERFACE_ROWS - determines the number of days that records are retained in the interface tables.

Chapter 3 User Open Interface

For enterprises with a large number of users, Mercury ITG Center includes an open interface for user creation. This API uses interface tables within the database instance. Data added to these interface tables is validated and eventually imported into standard tables, generating users that can be processed normally within Mercury ITG Center.

Mercury ITG Center also supports importing users from an LDAP server. The specifics surrounding user import from an LDAP server are also discussed in this chapter. The following sections detail the data model and process used to import users from an existing database:

- *Data Model*
- *Running the User Open Interface for a Simple Import*
- *Running the User Open Interface for an LDAP Import*
- *Post-Import Activities*

Data Model

The interface tables used by the user open interface closely relate to the standard tables used to store user and security information. *Table 3-1* lists and defines the KNTA_USERS_INT table columns. *Table 3-2* lists and defines the KNTA_USER_SECURITY_INT table columns.

Note

Each table includes a "Required for Import" column. This column, and other related requirements, are discussed in *"Loading Data into the Interface Tables"* on page 42.

Table 3-1. KNTA_USERS_INT

Column	Required for Import?	DataType	Description
TRANSACTION_ID	N	NUMBER	System-generated identifier.
GROUP_ID	Y	NUMBER	A value from KNTA_INTERFACE_GROUPS_S. This is used to group together all records that have to be processed together.
EXISTS_FLAG	N	VARCHAR2 (1)	Used to indicate if the user already exists in the system.
PROCESS_PHASE	N	NUMBER	Indicates the current stage of the record as it is being processed.
PROCESS_STATUS	N	NUMBER	Indicates the current disposition of the record. The value should initially be set to 1.
CREATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from CREATED_BY_USERNAME or set to the user currently running the report.
CREATED_BY_USERNAME	N	VARCHAR2(30)	The USERNAME from the KNTA_USERS table for the user that is performing the transaction. This will only be used if the CREATED_BY is null.
CREATION_DATE	N	DATE	The date that the transaction is performed.
DEST_CREATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from CREATED_BY_USERNAME or set to the user currently running the report.
DEST_CREATION_DATE	N	DATE	The date that the transaction is performed.
DEST_LAST UPDATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that last performed the transaction.

Table 3-1. KNTA_USERS_INT

Column	Required for Import?	DataType	Description
DEST_LAST_UPDATE_ DATE	N	DATE	The date that the transaction was last updated.
DEST_ENTITY_UPD_DATE	N	DATE	The date that the transaction is performed.
USER_ID	N	NUMBER	Internal identifier for a user. This should normally be left blank and is defaulted from KNTA_USERS_S. If a value is entered, it should be derived from the KNTA_USERS_S sequence.
DEST_USER_ID	N	NUMBER	Internal identifier for a user. This should normally be left blank and will be defaulted from KNTA_USERS_S. If a value is entered, it should be derived from the KNTA_USERS_S sequence.
USERNAME	Y	VARCHAR2(30)	The username used to logon.
LOGON_ID	Y (SEE NOTES)	VARCHAR2(30)	The system needs either this or the USERNAME.
DEST_USERNAME	Y	VARCHAR2(30)	The username that will be used to logon.
PASSWORD	N	VARCHAR2(40)	The password for the user (defaulted from the report).
PASSWORD_EXPIRATION_ DAYS	N (BUT YOU CAN)	NUMBER	Number of days in which the current password expires.
PASSWORD_EXPIRATION_ DATE	N (BUT YOU CAN)	DATE	The effective date when the password for user id expires.
EMAIL_ADDRESS	N (BUT YOU CAN)	VARCHAR2(80)	The email address of the user.
FIRST_NAME	Y (IF NOT IN SYSTEM - SEE DESCRIPT ION)	VARCHAR2(30)	The user's first name.

Table 3-1. KNTA_USERS_INT

Column	Required for Import?	DataType	Description
LAST_NAME	Y (IF NOT IN SYSTEM -- SEE DESCRIPTION)	VARCHAR2(30)	The user’s last name.
START_DATE	N (BUT YOU CAN)	DATE	The start date for the user.
END_DATE	N (BUT YOU CAN)	DATE	The end date for the user.
DEFAULT_ACCELERATOR_ID	N	NUMBER	Parameter set context identifier for the User Data fields. Can be left blank.
SOURCE_TYPE_CODE	N	VARCHAR2(30)	For records that have been updated by an interface or migrator, indicates the type of external update (specific interface or migrator name, etc.)
SOURCE	N	VARCHAR2 (100)	This is an optional, non-validated field that can be used to indicate the exact source of the user information (the name of the third party application, the text string “Conversion”, etc.)
USER_DATA_SET_ CONTEXT_ID	N	NUMBER	Parameter set context identifier for the User Data fields.
USER_DATA1.USER_ DATA20	N	VARCHAR2 (200)	These columns hold the user-defined fields attached to the user screen. Values should be entered only if User Data is defined. These columns will not be validated or defaulted.
VISIBLE_USER_DATA1. VISIBLE_USER_DATA20	N	VARCHAR2 (200)	These columns hold the user-defined fields attached to the user screen. Values should be entered only if User Data is defined. These columns will not be validated or defaulted.
USER_DATA_SET_ CONTEXT_ID	N	NUMBER	Parameter set context identifier for the User Data fields.
AUTHENTICATION_MODE	N	VARCHAR2(30)	Authentication Mode of the user. If the user is being imported from a LDAP server, then this is automatically set to LDAP, otherwise it is set to KINTANA.

Table 3-1. KNTA_USERS_INT

Column	Required for Import?	DataType	Description
SCREEN_ID	N	NUMBER	This column is optional and is given a default value. It determines which screen is shown first after logon.
SHORTCUT_BAR_FLAG	N	VARCHAR2(1)	This column is optional and is given a default value. It determines whether or not the shortcut bar is shown in the screen manager.
SHORTCUT_BAR_LOC_ CODE	N	VARCHAR2(4)	This column is optional and is given a default value. It determines the position where the shortcut bar is displayed.
SAVE_WINDOW_BOUNDS_ FLAG	N	VARCHAR2(1)	This column is optional and is given a default value. It determines whether 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.
WINDOW_HEIGHT	N	NUMBER	This column is optional and is given a default value. It determines the default height of the screen manager window.
WINDOW_WIDTH	N	NUMBER	This column is optional and is given a default value. It determines the default width of the screen manager window.
WINDOW_X_LOCATION	N	NUMBER	This column is optional and is given a default value. It determines the horizontal position of the screen manager window.
WINDOW_Y_LOCATION	N	NUMBER	This column is optional and is given a default value. It determines the vertical position of the screen manager window.
REUSE_INTERNAL_ FRAME_FLAG	N	VARCHAR2(1)	This column is optional and is given a default value. It determines whether multiple internal frames can be opened within each screen.
SHOW_ALL_WORKFLOW_ STEPS_FLAG	N	VARCHAR2(1)	This column is optional and is given a default value. It determines whether all Workflow steps are shown within Workflow status panels.

Table 3-1. KNTA_USERS_INT

Column	Required for Import?	DataType	Description
SHOW_TRAVERSED_ STEPS_FLAG	N	VARCHAR2(1)	This column is optional and is given a default value. It determines whether steps that have been traversed and are no longer active are shown within Workflow status panels.
NUM_BRANCH_STEPS_TO _SHOW	N	NUMBER	This column is optional and is given a default value. It determines how many steps of each branch are shown within Workflow status panels, if a currently active Workflow Step leads to several branches.
NUM_KNOWN_REACH_ STEPS_TO_SHOW	N	NUMBER	This column is optional and is given a default value. It determines how many steps of a non-branching path are shown within Workflow status panels.
HIDE_IMMEDIATE_STEPS_ FLAG	N	VARCHAR2(1)	This column is optional and is given a default value. It determines whether or not Workflow steps based upon immediate executions and conditions are shown within Workflow status panels.
SHOW_CHANGE_ WARNINGS_FLAG	N	VARCHAR2(1)	This column is optional and is given a default value. It determines whether 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).
HIDE_CANCELLED_CRL_ FLAG	N	VARCHAR2(1)	This column is optional and is given a default value. It determines whether cancelled Package Lines are displayed in the Packages screen.
DEFAULT_BROWSER	N	VARCHAR2 (300)	Default browser associated with the user.
DEST_USER_PROFILE_ID	N	NUMBER	Should be left blank. Will be derived from KNTA_USER_PROFILE_S.
PHONE_NUMBER	N	VARCHAR2 (300)	This column specifies the users phone number on the Resource page.
COST_RATE	N	NUMBER	This column specifies the user’s cost rate on the Resource page.

Table 3-1. KNTA_USERS_INT

Column	Required for Import?	DataType	Description
WORKLOAD_CAPACITY	N	NUMBER	This column specifies the user's workload capacity (in percentage) on the Resource page.
MAX_ROWS_PORTLETS	N	NUMBER	Maximum number of results to be displayed on the maximized Portlet.
DEPARTMENT_CODE	N	VARCHAR2 (300)	Code of the Department to which the Resource is associated.
DEPARTMENT_MEANING	N	VARCHAR2(80)	Text that is displayed to the user. This is the Department to which the Resource is associated. When populating the interface tables for an import, it is only necessary to specify the MEANING (the CODE is automatically derived).
LOCATION_CODE	N	VARCHAR2(30)	Code of the Location associated with the user on the Resource page.
LOCATION_MEANING	N	VARCHAR2(80)	Text that is displayed to the user. This is the Location associated with the user on the Resource page. When populating the interface tables for an import, it is only necessary to specify the MEANING (the CODE is automatically derived).
MANAGER_USER_ID	N	NUMBER	This column specifies the user ID of the manager specified on the Resource page.
MANAGER_USERNAME	N	VARCHAR2(80)	This column specifies the username of the manager specified on the Resource page.
MANAGER_LOGON_ IDENTIFIER	N	VARCHAR2 (300)	The logon ID of the user's manager. Used when Mercury ITG Center is started in LOGON_ID mode.
RESOURCE_CATEGORY_ CODE	N	VARCHAR2(30)	Code for the Category associated with a user on his Resource page.

Table 3-1. KNTA_USERS_INT

Column	Required for Import?	DataType	Description
RESOURCE_CATEGORY_ MEANING	N	VARCHAR2(80)	Text that is displayed to the user. This is the Category associated with a user on his Resource page. When populating the interface tables for an import, it is only necessary to specify the MEANING (the CODE is automatically derived).
RESOURCE_TITLE_CODE	N	VARCHAR2(30)	Code for the the resource's title that appears on the Resource page.
RESOURCE_TITLE_ MEANING	N	VARCHAR2(80)	Text that is displayed to the user. This is the Title associated with a user on his Resource page. When populating the interface tables for an import, it is only necessary to specify the MEANING (the CODE is automatically derived).
DELIVER_POWER_ ENABLED	N	VARCHAR2(1)	Used to indicate if the user should have a Change Manager Power license.
DELIVER_STANDARD_ ENABLED	N	VARCHAR2(1)	Used to indicate if the user should have a Change Manager Power license.
CREATE_POWER_ ENABLED	N	VARCHAR2(1)	Used to indicate if the user should have a Request Manager Power license.
CREATE_STANDARD_ ENABLED	N	VARCHAR2(1)	Used to indicate if the user should have a Request Manager Standard license.
DRIVE_POWER_ENABLED	N	VARCHAR2(1)	Used to indicate if the user should have a Project Manager Power license.
DRIVE_STANDARD_ ENABLED	N	VARCHAR2(1)	Used to indicate if the user should have a Project Manager Standard license.
ADMINISTRATOR_ ENABLED	N	VARCHAR2(1)	Used to indicate if the user should have an Administrator license.
SOLUTIONS	N	VARCHAR2 (2000)	A ";" delimited list of Mercury ITG Center Solutions for which the user is licensed.

Table 3-2. KNTA_USER_SECURITY_INT

Column	Required for Import?	DataType	Description
TRANSACTION_ID	N	NUMBER	System-generated identifier.
PARENT_ TRANSACTION_ID	N	NUMBER	The transaction ID of the parent table being imported.
PARENT_TABLE_ NAME	N	VARCHAR2(30)	Which table was this entity imported with. (The parent_table should be KNTA_USERS_INT.)
GROUP_ID	Y	NUMBER	A value from KNTA_INTERFACE_GROUPS_S. Used to group together all records to be processed together. Only use one group ID for each batch of imported users when running the User Open Interface program.
EXISTS_FLAG	N	VARCHAR2(1)	Used to indicate if the Security Group exists for the users.
PROCESS_PHASE	N	NUMBER	Indicates the current stage of the record as it is being processed.
PROCESS_STATUS	N	NUMBER	Indicates the current disposition of the record. The value should initially be set to 1.
CREATED_BY	N	NUMBER	Identifier for the user that created the record in the source.
CREATED_BY_ USERNAME	N	VARCHAR2(30)	The USERNAME from the KNTA_USERS table for the user that is performing the transaction. This will be used only if the CREATED_BY is null.
CREATION_DATE	N	DATE	Date record was created.
DEST_CREATED_BY	N	NUMBER	Identifier for the user that created the record in the destination.
DEST_CREATION_ DATE	N	DATE	The date the record is created in the destination.

Table 3-2. KNTA_USER_SECURITY_INT

Column	Required for Import?	DataType	Description
DEST_LAST_UPDATED_BY	N	NUMBER	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from LAST_UPDATED_USERNAME.
DEST_LAST_UPDATE_DATE	N	DATE	The date that the transaction is performed.
DEST_ENTITY_UPD_DATE	N	DATE	The date that the transaction is performed.
USER_SECURITY_ID	N	NUMBER	Internal identifier for a user security. This should normally be left blank and will be defaulted from KNTA_USER_SECURITY_S. If a value is entered, it should be derived from KNTA_USER_SECURITY_S sequence.
DEST_USER_SECURITY_ID	N	NUMBER	Internal identifier for a user security. This should normally be left blank and will be defaulted from KNTA_USER_SECURITY_S. If a value is entered, it should be derived from KNTA_USER_SECURITY_S sequence.
USER_ID	N	NUMBER	This is used to tie a record to a parent record in KNTA_USERS_INT. For existing users, this should refer to the USER_ID column in KNTA_USERS.
DEST_USER_ID	N	NUMBER	This is used to tie a record to a parent record in KNTA_USERS_INT. For existing users, this should refer to the USER_ID column in KNTA_USERS.
SECURITY_GROUP_ID	N	NUMBER	Identifier of the Security Group the user is linked to.
SOURCE_TYPE_CODE	N	VARCHAR2(30)	For records that have been updated by an interface or migrator, indicates the type of external update (specific interface or migrator name, etc.)

Table 3-2. KNTA_USER_SECURITY_INT

Column	Required for Import?	DataType	Description
SOURCE	N	VARCHAR2(100)	For records that have been updated by an interface or migrator, provides additional information about the source of the external update.
USERNAME	Y	VARCHAR2(30)	USERNAME in KNTA_USERS table.
SECURITY_GROUP_NAME	Y	VARCHAR2(40)	SECURITY_GROUP_NAME in KNTA_SECURITY_GROUPS table.
USER_SECURITY_ACTION	Y	VARCHAR2(30)	Action for user security. Possible values: ADD, DROP.
LOGON_IDENTIFIER	Y	VARCHAR2(30)	LOGON_IDENTIFIER in KNTA_USERS table.

Running the User Open Interface for a Simple Import

Importing users from an external database consists of the following steps:

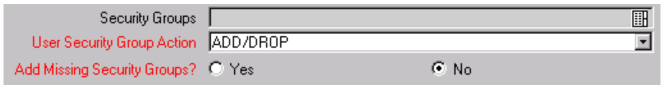
- *Determining User Security*. Decide which users should have which Security Groups linked to them. Mercury ITG Center provides a method for adding, removing, and overwriting the Security Groups associated with a batch of users.

- *Loading Data into the Interface Tables*. Populate the open interface tables with data before running the import program. This population can be done through any means supported by the Oracle database (such as SQL*Loader). This process and the requirements for this population depends on the data that will be imported (specifically regarding Security Groups).

- *The Run ITG User Interface Program*. This program is defined as a standard Report Type and is launched through either the Request Management, Change Management, or Project Management Reports screen.

Determining User Security

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: Security Groups, User Security Group Action, and Add Missing Security Groups.



Security Groups

User Security Group Action ADD/DROP

Add Missing Security Groups? ○ Yes ● No

Using the Workbench, 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. For more information, see *"Loading Data into the Interface Tables"* on page 42.
- Overwrite the Security Group specification to include only the specified Security Groups. This will delete all references to the user's Security Groups and replace them with the selected ones.

It is also possible to add missing Security Groups. This will create a new Security Group, but will not link the user to that Security Group. For information on Security Group import options, see *"The Run ITG User Interface Program"* on page 46.

Loading Data into the Interface Tables

The initial step in importing users is populating the open interface table. This 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.

The following sections discuss any requirements associated with specific import scenarios:

- *Required Data for Simple User Import*
- *Using the ADD/DROP Security Group Feature*

Note

When importing from an LDAP server, this step can be skipped. For instructions on importing users from an LDAP server, see *"The Run ITG User Interface Program for LDAP Import"* on page 58.

Required Data for Simple User Import

Specific columns in the KNTA_USERS_INT table must be populated for a simple user import. *Table 3-3* defines the columns in the KNTA_USERS_INT table that need to be populated for the import. For a complete table description, see *Table 3-1 on page 32*.

Table 3-3. KNTA_USERS_INT Columns Required for User Import

Column	Required for Import?	DataType	Description
GROUP_ID	Y	NUMBER	A value from KNTA_INTERFACE_GROUPS_S. This is used to group together all records that have to be processed together. Use only one group ID for each batch of imported users when running the User Open Interface program.
USERNAME	Y	VARCHAR2(30)	The username used to Logon.
LOGON_ID	N (see Description)	NUMBER	Populate either this column or the USERNAME, depending 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. If LOGON_METHOD = LOGON_ID, the LOGON_ID column must be populated.
DEST_USERNAME	Y	VARCHAR2(30)	The username that will be used to Logon.

Table 3-3. KNTA_USERS_INT Columns Required for User Import

Column	Required for Import?	DataType	Description
PASSWORD_ EXPIRATION_DAYS	N (see Description)	NUMBER	Number of days in which the current password expires. This column is not required, but can be populated for a group of users for convenience.
PASSWORD_ EXPIRATION_DATE	N (see Description)	DATE	The effective date when the password for user id expires. This column is not required, but can be populated for a group of users for convenience.
EMAIL_ADDRESS	N (BUT YOU CAN)	VARCHAR2(80)	The email address of the user. This column is not required, but can be populated for a group of users for convenience.
FIRST_NAME	Y (see Description)	VARCHAR2(30)	The user's first name. This field is required only when creating a new user. It is not required when re-importing an existing user.
LAST_NAME	Y (IF NOT IN SYSTEM -- SEE DESCRIPTION)	VARCHAR2(30)	The user's last name. This field is required only when creating a new user. It is not required when re-importing an existing user.
START_DATE	N (BUT YOU CAN)	DATE	The start date for the user. This column is not required, but can be populated for a group of users for convenience.
END_DATE	N (BUT YOU CAN)	DATE	The end date for the user. This column is not required, but can be populated for a group of users for convenience.

Note

Additional columns in the KNTA_USER_SECURITY_INT table must be populated when using the ADD/DROP Security Group action. For more information, see *"Using the ADD/DROP Security Group Feature"* on page 45.

Using the ADD/DROP Security Group Feature

To use the ADD/DROP Security Group feature in the Run ITG User Interface program, first populate the KNTA_USER_SECURITY_INT table. This table needs to include a record for each desired Security Group action for each user.

Example

User A and User B exist as Mercury ITG Center users 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, 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

Proceed with the import using the Run ITG User Interface report. Select the **ADD/DROP** from the User Security Group Action drop down list and run the report. The user definitions in the destination instance will be updated with the Security Group information above.

Table 3-4 defines the columns in the KNTA_USER_SECURITY_INT table that need to be populated for the import. For a complete table description, see *Table 3-2 on page 39*.

Table 3-4. KNTA_USER_SECURITY_INT Columns Required for User Import

Column	Required for Import?	DataType	Description
GROUP_ID	Y	NUMBER	A value from KNTA_INTERFACE_GROUPS_S. Used to group together all records to be processed together. Use only one group ID for each batch of imported users when running the User Open Interface program. This should be the same GROUP_ID that is specified in the KNTA_USERS_INT table.
USERNAME	Y	VARCHAR2(30)	USERNAME in KNTA_USERS table.
LOGON_IDENTIFIER	Y	VARCHAR2(30)	LOGON_IDENTIFIER in KNTA_USERS table. Populate either this column or the USERNAME, depending 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. If LOGON_METHOD = LOGON_ID, the LOGON_ID column must be populated.
SECURITY_GROUP_NAME	Y	VARCHAR2(40)	SECURITY_GROUP_NAME in KNTA_SECURITY_GROUPS table.
USER_SECURITY_ACTION	Y	VARCHAR2(30)	Action for user security. Possible values: ADD, DROP.

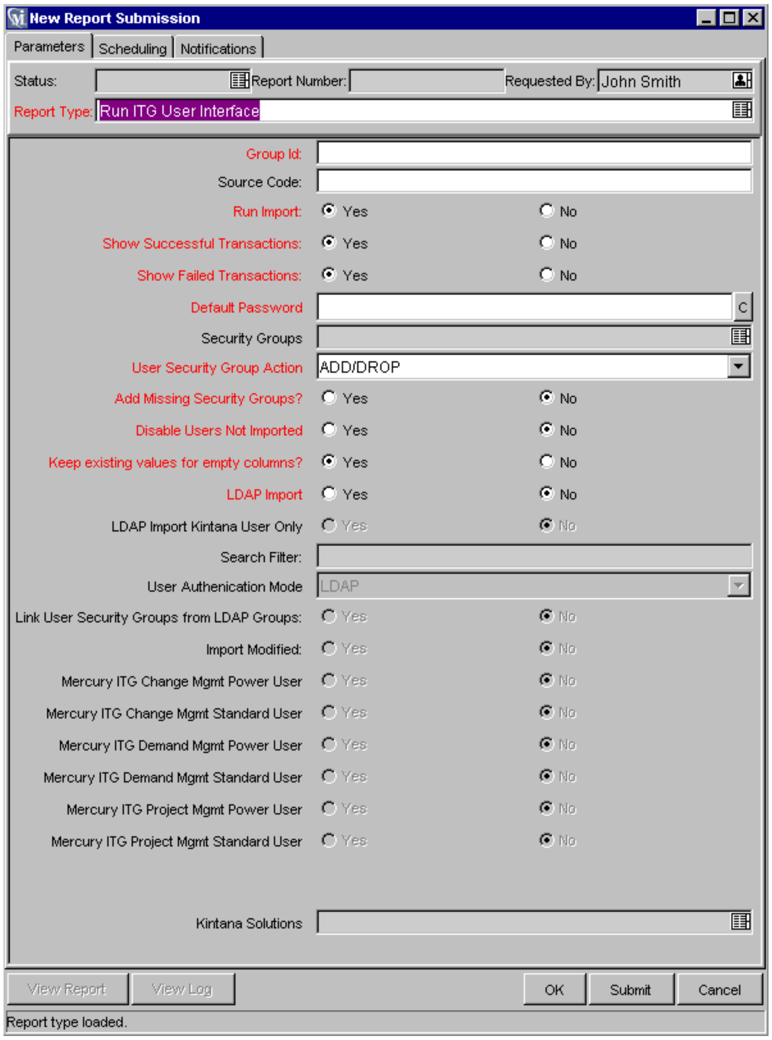
The Run ITG User Interface Program

To import data from the interface tables, use the Run ITG User Interface program. This program is defined as a standard Report Type and launched through the Request Management, Change Management, or Project Management Reports screen.

Note

The Run ITG User Interface program automatically imports relevant user information stored in the LDAP server into the open interface table. For more information, see *"Running the User Open Interface for an LDAP Import"* on page 52.

For more information on using Report Types and launching Reports, see *Reports Guide and Reference* and respective product manuals.



Status:		
Report Number:		
Requested By:	John Smith	
Report Type:	Run ITG User Interface	
Group Id:		
Source Code:		
Run Import:	● Yes	○ No
Show Successful Transactions:	● Yes	○ No
Show Failed Transactions:	● Yes	○ No
Default Password		
Security Groups		
User Security Group Action	ADD/DROP	
Add Missing Security Groups?	○ Yes	● No
Disable Users Not Imported	○ Yes	● No
Keep existing values for empty columns?	● Yes	○ No
LDAP Import	○ Yes	● No
LDAP Import Kintana User Only	○ Yes	● No
Search Filter:		
User Authenication Mode	LDAP	
Link User Security Groups from LDAP Groups:	○ Yes	● No
Import Modified:	○ Yes	● No
Mercury ITG Change Mgmt Power User	○ Yes	● No
Mercury ITG Change Mgmt Standard User	○ Yes	● No
Mercury ITG Demand Mgmt Power User	○ Yes	● No
Mercury ITG Demand Mgmt Standard User	○ Yes	● No
Mercury ITG Project Mgmt Power User	○ Yes	● No
Mercury ITG Project Mgmt Standard User	○ Yes	● No
Kintana Solutions		

Report type loaded.

Figure 3-1 Report-Run ITG User Interface

When executed, the Run ITG User Interface program:

- 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 the Mercury ITG Center tables. Partial imports are not allowed. Users with one or more failed fields will not be imported.
- Reports on the results of the execution, listing the specified Users that failed validation and the specific validation errors they encountered.

Parameters

The Run ITG User Interface program has several parameters for controlling the behavior of the program execution. These parameters are defined in *Table 3-5*.

Table 3-5. Run ITG User Interface Parameters

Name	Type	Required	Description
Group ID	Text Field	Y	Runs the interface for a specific Group ID. The interface program will only look for records with this value in the GROUP_ID column. This is useful when importing a batch of users.
Source Code	Text Field	N	Used to indicate the source from which the users are being imported. In the case of a LDAP import, it is set to LDAP_IMPORT.
Table Name	Text Field	N	This parameter is used for custom LDAP imports. To load LDAP data into some other interface table (such as KCRT_CONTACTS_INT), specify the table name by enabling this field in the Report Type.
Run Import	Radio Button	Y	If set to **Yes**, the program processes the records in the interface table and tries to import them. If set to **No**, the program simply reports on the records in the interface table. This option is useful when auditing prior executions of the open interface.
Show Successful Transactions	Radio Button	Y	Shows successfully imported users.
Show Failed Transactions	Radio Button	Y	Shows users that were not successfully imported.

Name	Type	Required	Description
Default Password	Password Field	Y	Sets a default password for all users being imported, if they are not LDAP users.
Security Groups	Auto-complete list	N	Links available Security Groups to the imported users.
User Security Group Action	Drop Down List	Y	Used to specify how the Security Groups are managed during the import. Select one of the following options: • ADD/DROP - Use this option to add certain Security Groups while removing others from user definitions. This option needs to be used in conjunction with the population of the KNTA_USER_SECURITY_INT table. See *"Using the ADD/DROP Security Group Feature"* on page 45 for details. • ADD - Adds the Security Groups listed in the Security Groups field to the users. Also adds any records in the KNTA_USER_SECURITY_INT table with an "ADD" user_security_action. • DROP - Removes the Security Groups listed in the Security Groups field from the users. Also drops any records in the KNTA_USER_SECURITY_INT table with an "DROP" user_security_action. • OVERWRITE - Removes any relationships between an existing user and their Security Groups and replaces them with the Security Groups listed in the Security Groups field.
Add Missing Security Groups	Radio Button	Y	Selects whether to create Security Groups. To use this feature, the SECURITY_GROUP_NAME and the corresponding USER_SECURITY_ACTION must be populated in the KNTA_USER_SECURITY_INT table.
Disable Users Not Imported	Radio Button	Y	Disables all existing users currently in the system. This happens prior to the import. Note that if the import fails, users will still be disabled.
Keep existing values for empty columns?	Radio Button	Y	Keep existing values on updated users when interface table columns are empty.
LDAP Import	Radio Button	Y	This indicates whether users are being imported from a LDAP server. Selecting **Yes** enables multiple required fields. Set the AUTHENTICATION_MODE parameter in the server.conf to LDAP or a comma-delimited list (such as LDAP). The appropriate LDAP parameters should be specified. Set this to **Yes** if the authentication mode in the server.conf contains LDAP or NTLM.

Name	Type	Required	Description
LDAP Import Kintana User Only	Radio Button	N	If set to **Yes**, only imports those LDAP users with the “KNTAUser” attribute. See *“Adding the KNTAUser Attribute to Users on an LDAP Server”* on page 53 for detailed instruction.
Search Filter	Text Field	N	Used to import users from the LDAP server which satisfy a particular LDAP search filter. If no search filter is specified, all appropriate LDAP users will be imported. For instructions on searching for users based on specific filters, see the LDAP documentation. A few examples are provided following this table.
User Authentication Mode	Drop Down List	N	Specifies how the user is authenticated upon their logon. For example, if LDAP is selected, the user authentication occurs on the LDAP server.
Link Security Groups from LDAP Groups	Radio Button	N	Select to link the imported users to Security Groups corresponding to the LDAP Groups on the LDAP server. If a corresponding Security Group does not exist, Add Missing Security Groups must be set to **Yes**. This option is only available when importing groups from an LDAP server. ***************************** EXAMPLE: LDAP Server: Group F1 = User A and User B --Sub-group F2 = User C and User D <-- Imported into Mercury ITG Center --> Mercury ITG User A -> Security Group F1 Mercury ITG User B -> Security Group F1 Mercury ITG User C -> Security Group F1 and F2 Mercury ITG User D -> Security Group F1 and F2 ******************************
Import Modified	Radio Button	N	Select **Yes** to import only users who have been modified in the LDAP server since the last import.
Mercury ITG Change Mgmt Power User	Radio Button	Only for LDAP	Establishes a Change Management Power License account for the imported users.
Mercury ITG Change Mgmt User	Radio Button	Only for LDAP	Establishes a Change Management Standard License account for the imported users.
Mercury ITG Demand Mgmt Power User	Radio Button	Only for LDAP	Establishes a Demand Management Power License account for the imported users.

Name	Type	Required	Description
Mercury ITG Demand Mgmt Standard User	Radio Button	Only for LDAP	Establishes a Demand Management Standard License account for the imported users.
Mercury ITG Project Mgmt Power User	Radio Button	Only for LDAP	Establishes a Project Management Power License account for the imported users.
Mercury ITG Project Mgmt Standard User	Radio Button	Only for LDAP	Establishes a Project Management Standard License account for the imported users.
Kintana Solutions	Auto-complete list	Only for LDAP	Establish an account of the selected product (Portfolio Management, Program Management, Time Management) for the imported users.

Example

Examples of Search Filter values and their functional results:

- `(cn=Babs Jensen)`: Fetch the object who has cn = Babs Jensen
- `(!(cn=Tim Howes))`: Fetch the objects who don’t have cn = Tim Howes
- `(&(objectClass=Person)(|(sn=Jensen)(cn=Babs J*)))`: Fetch all objects belonging to objectclass Person and who are either sn = Jensen or cn is like Babs J (which means cn could be Babs J or cn could be Babs Jen)
- `(o=univ*of*mich*)`: Fetch objects with o like univ%of%mich% (using sql concepts)
- `(&(ou=Development)(|(uid=test1)(uid=test2)(uid=test3)))`: Fetch all objects with ou= Development and who have uid=test1 or uid=test2 or uid=test3

Note

All users imported using the Run ITG User Interface report will 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.

Note

User Data for users is not validated as part of an import.

Running the User Open Interface for an LDAP Import

To import data from an LDAP server, the Run ITG User Interface program must be used. This program is defined as a standard Report Type and launched through the Request Management, Change Management, or Project Management Reports screen.

The Run ITG User Interface program automatically imports relevant user information stored in the LDAP server into the open interface table. It is possible to exercise more control over this process by mapping LDAP attributes to Mercury ITG Center users in the LdapAttribute.conf file. For more information, see *"Mapping LDAP Attributes to Mercury ITG Center Users"* on page 54.

If user information is imported from an LDAP server, ensure that LDAP Import is set to **Yes** in the New Report Submission window. Selecting **Yes** enables other required fields, ensuring a smooth import into Mercury ITG Center. A special command line utility can also be run to add the attribute "KNTA User" to various users on the LDAP server. For more information, see *"Adding the KNTAUser Attribute to Users on an LDAP Server"* on page 53.

Prepare for an LDAP User Import

The process for importing users from an LDAP server differs slightly from the simple user import process. The following sections detail some additional preparation required for an LDAP user import.

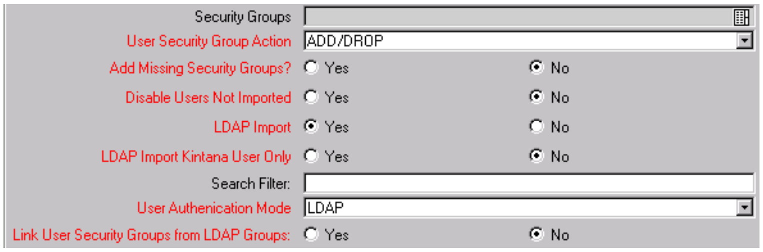
- *User Security for an LDAP Import*
- *Adding the KNTAUser Attribute to Users on an LDAP Server*
- *Mapping LDAP Attributes to Mercury ITG Center Users*
- *Mercury ITG Server Configuration*

User Security for an LDAP Import

Similar to a simple user import, the Security Groups to which the imported users will be linked need to be determined. To set security for users, either:

- Use the Security Group Action field in combination with the Security Groups field to specify which Security Groups to add to the user definitions.

- Use the Link Security Groups from LDAP Groups option. Select **Yes** to link the imported users to Security Groups corresponding to the LDAP Groups on the LDAP server. If a corresponding Security Group does not exist, Add Missing Security Groups must be set to **Yes**.



Security Groups		
User Security Group Action	ADD/DROP	
Add Missing Security Groups?	○ Yes	● No
Disable Users Not Imported	○ Yes	● No
LDAP Import	● Yes	○ No
LDAP Import Kintana User Only	○ Yes	● No
Search Filter:		
User Authenication Mode	LDAP	
Link User Security Groups from LDAP Groups:	○ Yes	● No

Note

The DROP, ADD/DROP, or OVERWRITE Security Group Action features cannot be used for LDAP user security definitions. To remove a Security Group from a user definition, either perform the update manually in the Workbench or re-run a simple User Open Interface. For more information, see *"Running the User Open Interface for a Simple Import"* on page 41.

Adding the KNTAUser Attribute to Users on an LDAP Server

Adding the KNTAUser attribute to users on an LDAP server can be a convenient way to mark users for importing, when LDAP Import Kintana User Only is set to **Yes**. However, it is not a required step. LDAP Import Kintana User Only can be set to **No**, and the Search Filter field used to query for the attribute of your choice.

If LDAP Import Kintana User Only is set to **Yes** on the Run ITG User Interface, only the LDAP users with the KNTAUser attribute will be 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 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.

- Type `kLdap.sh -s`

The LDAP parameters will be read from the server.conf file and no additional information will be requested.

Note

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

Mapping LDAP Attributes to Mercury ITG Center Users

It is possible to map the attributes on the LDAP server to attributes used by the Mercury ITG Server. Some of this mapping occurs by default, but it is possible to exercise greater control by mapping the attributes in the following file:

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

Table 3-6 lists the attributes and provides some contextual background for each parameter. Mercury ITG 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.

Table 3-6. LdapAttribute.conf parameters

LDAP Attribute	Description
LDAP_USER_ID	Enter the attribute which can be used to obtain the User ID on the LDAP server. LDAP_USER_ID=uid This is used to resolve the users in the Ldap server. It is mapped to unique attribute that determines a user.
LDAP_LOGON_ID	Enter the attribute which can be used to obtain the logon ID from the LDAP server. LDAP_LOGON_ID=uid Note: This needs to be different from the LDAP_USER_ID if the LOGON_METHOD = LOGON_ID in the server.conf file.
LDAP_GROUP_NAME	Enter the attribute which can be used to obtain the group names from the LDAP server. LDAP_GROUP_NAME=cn
LDAP_ORG_UNIT_NAME	Enter the attribute which can be used to obtain the org units from the LDAP server. LDAP_ORG_UNIT_NAME=ou

Table 3-6. LdapAttribute.conf parameters

LDAP Attribute	Description
LDAP_STATIC_GROUP_ MEMBERS	Enter the attribute which identifies the Static group members on the LDAP server. LDAP_STATIC_GROUP_MEMBERS=uniquemember This parameter and the LDAP_DYNAMIC_GROUP_MEMBERS parameter are used to determine the attribute that indicates the members in the Group. In Netscape, there can be STATIC and DYNAMIC Groups; hence there are two different parameters. In Active Directory, these would map to the same attribute (member).
LDAP_DYNAMIC_GROUP_ MEMBERS	Enter the attribute which identifies the Dynamic group members on the LDAP server. LDAP_DYNAMIC_GROUP_MEMBERS=memberurl This parameter and the LDAP_STATIC_GROUP_MEMBERS parameter are used to determine the attribute that indicates the members in the Group. In Netscape, there can be STATIC and DYNAMIC Groups; hence there are two different parameters. In Active Directory, these would map to the same attribute (member).
LDAP_OBJECTCLASS	Enter the attribute which identifies the objectclass attribute on the LDAP server. LDAP_OBJECTCLASS=objectclass This parameter determines what the objectclass is called on the LDAP Server. This will usually map to “objectclass”.
LDAP_USER_OBJECTCLASS	Enter the attribute which identifies the objectclass attribute for a user on the LDAP server. LDAP_USER_OBJECTCLASS=person This parameter determines the objectclass that determines if a resource is a user. It is defaulted to “person”.
LDAP_GROUP_OBJECTCLASS	Enter the attribute which identifies the objectclass attribute for a group on the LDAP server. LDAP_GROUP_OBJECTCLASS=groupofuniquenames This parameter determines the objectclass that determines if a resource is a group.
LDAP_MODIFY_TIMESTAMP	Enter the attribute which keeps track of the last modified time for an object on the LDAP server. LDAP_MODIFY_TIMESTAMP=modifyTimeStamp This parameter is mapped to attribute on the LDAP Server that keeps track of the modified time for that entity.

Table 3-6. LdapAttribute.conf parameters

LDAP Attribute	Description
LDAP_TIME_FORMAT	Enter the attribute which keeps track of the Time format followed by the LDAP server. Format for Netscape LDAP Servers: yyyyMMddHHmmss'Z' Format for Active Directory Servers: yyyyMMddHHmmss'.0Z' LDAP_TIME_FORMAT=yyyyMMddHHmmss'Z' This parameter gives Mercury ITG Center the information of the Time Format the LDAP server follows.

KNTA_USERS_INT	Denotes the table name to which the import would take place. The format is
	ColumnName= LDAP Attribute.
	Thus any column could be mapped to any LDAP Attribute.
	Map both VISIBLE_USER_DATA and USER_DATA columns always. To disable a default mapping below, either comment out or delete the mapping line.
	USERNAME=sAMAccountName
	FIRST_NAME=givenname
	LAST_NAME=sn
	EMAIL_ADDRESS=mail
	PHONE_NUMBER=telephonenumber
	DEPARTMENT_MEANING=departmentNumber
	LOCATION_MEANING=locality
	MANAGER_USERNAME=managerUSER_DATA1=mail**
	VISIBLE_USER_DATA1=mail**
	** These are added in for an example.

VISIBLE_USER_DATA and USER_DATA	Map both VISIBLE_USER_DATA and USER_DATA columns always.

Note

The ORG_UNIT_NAME and PARENT_ORG_UNIT_NAME columns should not be mapped in the LdapAttribute.conf file. Those attributes are specified in the KRSC_ORG_UNITS_INT table. For more information, see *"Organization Unit Open Interface"* on page 7.

Mercury ITG Server Configuration

The Mercury ITG Server parameters listed in *Table 3-7* need to be considered when performing a user import from an LDAP server. These parameters are set in the server.conf file located in the following directory:

```
ITG_Home/server.conf
```

Table 3-7 lists the attributes and provides some contextual background for each parameter. After changing these parameters, stop and restart the Mercury ITG Server.

Table 3-7. Server.conf Parameters

Parameter	Description
Authentication_Mode	This is required for the Mercury ITG Server to determine what method to use for authenticating users. It currently defaults to the KINTANA value. MIXED is no longer a valid value; mixed values can be separated with a comma. Valid values include: KINTANA, LDAP, SITEMINDER, NTLM com.kintana.core.server.AUTHENTICATION_MODE=KINTANA, LDAP, SITEMINDER
LDAP_Server_URL	This can be a comma delimited list of LDAP URLs; the Mercury ITG Server will query them in the order in which they are specified. If no port number is specified, then the default port 389 will be used. e.g. ldap://ldap.mydomain.com:389, ldap://ldap2.mydomain.com com.kintana.core.server.LDAP_URL=
LDAP_BASE_DN	This specifies the base in the LDAP server from where to start the search. If this is not specified, then the LDAP server will be queried to determine the base. e.g. o=mydomain.com com.kintana.core.server.LDAP_BASE_DN=
KINTANA_LDAP_ID	Mercury ITG Center account on the LDAP server. This is a required field if the LDAP server is being used at all. This will be used by the Mercury ITG Server to bind to the LDAP server. This is the complete distinguished name of the user. (Example: uid=admin, ou=Dev, o=mydomain.com) com.kintana.core.server.KINTANA_LDAP_ID=

Table 3-7. Server.conf Parameters

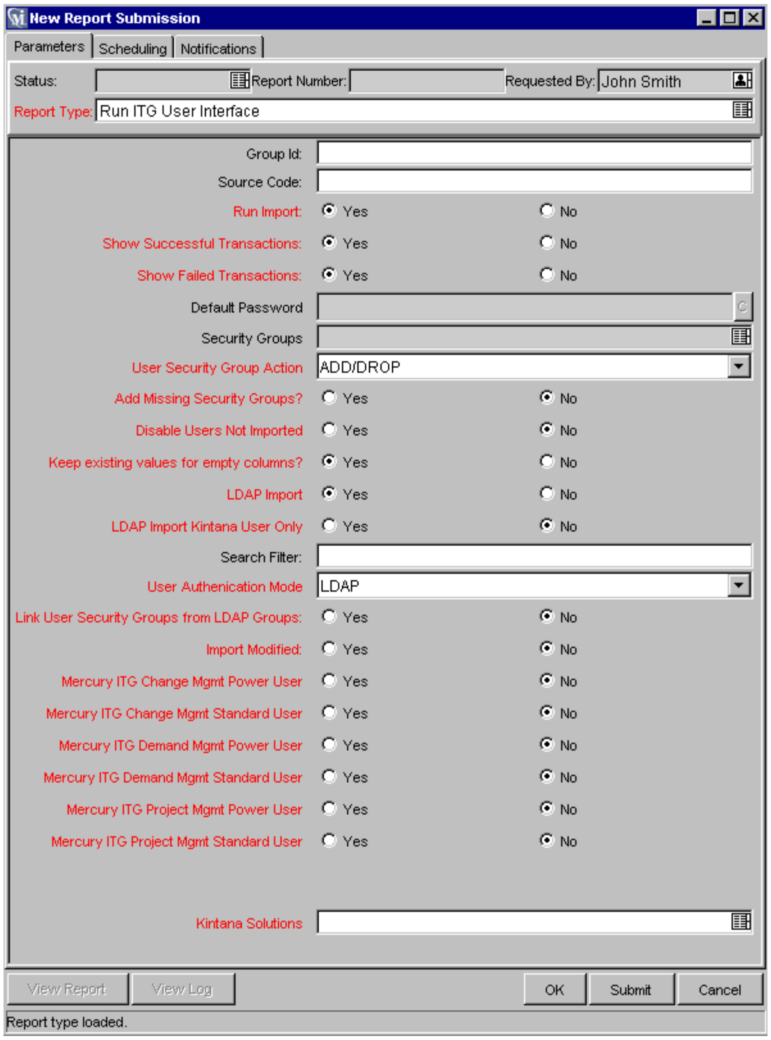
Parameter	Description

KINTANA_LDAP_ PASSWORD	Mercury ITG Center password on the LDAP server. This is automatically encrypted by the server configuration utility. Leave this blank for an anonymous authentication.
	Note: When manually editing the server.conf file, the encrypted password should be surrounded with #!# delimiters.
	com.kintana.core.server.KINTANA_LDAP_PASSWORD=#!#0123456789abcdefghij9876543210AB#!#
	com.kintana.core.server.KINTANA_LDAP_PASSWORD=#!##!#

LDAP_SSL_PORT	SSL Port number on the LDAP server. If the LDAP server is SSL enabled then this should be set to the SSL port of the LDAP server (which is usually 636). This port will be used to carry out all secure transactions. If this field is not set, then all transactions will be carried out over the port specified in the param LDAP_URL com.kintana.core.server.LDAP_SSL_PORT=

The Run ITG User Interface Program for LDAP Import

After determining the strategy for assigning Security Groups to the users and preparing for the import, run the import using the Run ITG User Interface program. This program is launched through the Request Management, Change Management, or Project Management Reports screen.



Status:		Report Number:	
Requested By:	John Smith		
Report Type:	Run ITG User Interface		

Group Id:		
Source Code:		
Run Import:	● Yes	○ No
Show Successful Transactions:	● Yes	○ No
Show Failed Transactions:	● Yes	○ No
Default Password		
Security Groups		
User Security Group Action	ADD/DROP	
Add Missing Security Groups?	○ Yes	● No
Disable Users Not Imported	○ Yes	● No
Keep existing values for empty columns?	● Yes	○ No
LDAP Import	● Yes	○ No
LDAP Import Kintana User Only	○ Yes	● No
Search Filter:		
User Authenication Mode	LDAP	
Link User Security Groups from LDAP Groups:	○ Yes	● No
Import Modified:	○ Yes	● No
Mercury ITG Change Mgmt Power User	○ Yes	● No
Mercury ITG Change Mgmt Standard User	○ Yes	● No
Mercury ITG Demand Mgmt Power User	○ Yes	● No
Mercury ITG Demand Mgmt Standard User	○ Yes	● No
Mercury ITG Project Mgmt Power User	○ Yes	● No
Mercury ITG Project Mgmt Standard User	○ Yes	● No
Kintana Solutions		

Report type loaded.

Figure 3-2 Report-Run ITG User Interface for LDAP

When executed, the Run ITG User Interface program:

- Populates the interface tables with records from the LDAP server.
- Validates the user information.
- Imports validated users into the Mercury ITG Center tables. Partial imports are not allowed. Users with one or more failed fields will not be 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 from the Workbench:

1. Open the Report window in either Request Management, Change Management or Project Management.

2. Select **Run ITG User Interface** from the Report Type auto-complete list. The parameters for this program appear in the Report window.

3. Enter the parameters and select the import options. Each parameter on this Report is defined in *Table 3-5 on page 48*. For an LDAP import, pay special attention to the following fields:

   - LDAP Import: set this field to **Yes**. This will enable the other LDAP-import related parameters.
   - Search Filter
   - User Authentication Mode
   - Link User Security Groups from LDAP Groups
   - Import Modified
   - Standard and Power User license information

4. Click **Submit**.

The report and the log can then viewed to analyze the import results.

LDAP Authentication

Mercury ITG Center uses simple authentication to authenticate against any LDAP v.3. compliant LDAP server.

The authentication involve the following steps:

1. The Mercury ITG Server binds to the LDAP server using the credentials supplied in the server.conf parameters KINTANA_LDAP_ID and KINTANA_LDAP_PASSWORD. This step is optional. Mercury ITG Center will do an anonymous authentication if a password is not supplied in server.conf.

2. Mercury ITG Center tries to obtain the distinguished name of the user by supplying a search filter to the LDAP server in the form (uid=username). The attribute uid could vary from one LDAP server to another depending on the information supplied in the LdapAttribute.conf file.

3. If Mercury ITG 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_Server_URL server parameter, Mercury ITG Center will try to authenticate against all of them until it succeeds. If the referral option has been enabled, then Mercury ITG Center will also query the referral server for authentication if the user is not present in primary server.

For users who are running the Mercury ITG Server on a JDK1.3 platform, Mercury ITG Center also supports LDAP authentication over SSL using passwords. To enable the SSL option, the LDAP_SSL_PORT server parameter should be set to the SSL port of the LDAP server.

Post-Import Activities

Following the user import, view the results of the import. From the Report screen, click **View Report** to open the report. This report will identify any errors with the import. After reviewing the import results, proceed with the following activities:

- *Correcting Failures*
- *Purging the Interface Tables*

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 using the standard user interface.

For users that fail to import, corrective actions are required. The first step involves 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 failures may occur due to a mapping problem between the source data and existing data.

Other failures may be due to missing information that cannot be defaulted. For example, users require a Username. If the Username columns were left blank for records in the User 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.

Note

During the initial implementation of the Open Interface, the mapping between the non-Mercury ITG Center source and Mercury ITG 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 ITG Center.

Purging the Interface Tables

All interface tables are automatically purged by the Purge service. The purging process depends on two parameters in the server.conf file:

- ENABLE_INTERFACE_CLEANUP - enables or disables the Purge process.
- DAYS_TO_KEEP_INTERFACE_ROWS - determines the number of days that records are retained in the interface tables.

Chapter 4 Request Open Interface

In addition to a Web-based user interface for the entry of new Requests, the Request Management application includes an open interface for Request creation. This API uses interface tables within the Request Management database instance. Data added to these interface tables is validated and eventually imported into standard Request Management tables. This generates Requests that can be processed using Request Management.

The primary purpose of the Request Open Interface is to enable integration with non-Mercury ITG 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 Request Management during initial implementation.

This chapter includes the following information related to the Request Open Interface:

- *Data Model*
- *Running the Request Open Interface*

Data Model

The interface tables used by the Request Management Request Open Interface closely relate to the standard tables used to store Request information:

- **KCRT_REQUESTS_INT** - This 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 determined by the Request Type for each specific Request.

- **KCRT_REQUEST_DETAILS_INT** - This is an internal interface table used by the Request Open Interface program as it processes new Requests. It is used to store Validation information related to the user-defined custom fields for each Request.
- **KCRT_REQ_HEADER_DETAILS_INT** - This internal table stores data for custom fields that are defined in the Request Header. Standard Request Header Type fields (such as Request Number and Priority) are stored in KCRT_REQUESTS_INT. This table can be optionally populated by the user.
- **KCRT_FG_<NAME>_INT** - Each Field Group has an associated interface table. These tables store information related to fields defined for that Field Group. These tables can be optionally populated by the user. Field Groups are installed when a Mercury ITG Center Solution is installed. Consequently, not all Field Group interface tables may be present in the database. The following Field Group interface tables currently exist in Mercury ITG Center:
    - 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

Figure 4-1 displays the relationships between the Request Open Interface tables.

Note

Each table includes a "Required for Import" column. This column, and other related requirements, are discussed in *"Loading Data into the Request Interface Tables"* on page 81.

The KNTA_INTERFACE_ERRORS table is included in the data model for reference only and should NOT be altered.

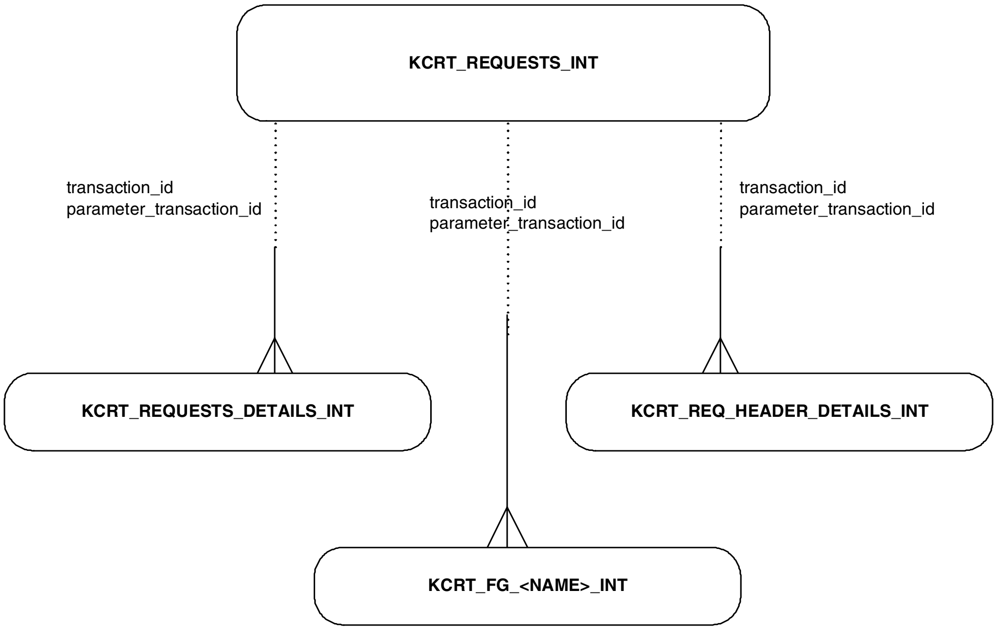


Figure 4-1 Request Interface Tables

Table 4-1. KCRT_REQUESTS_INT

Column	Required for Import?	Data Type	Description
GROUP_ID	Y	Number	Derived from the KNTA_INTERFACE_GROUPS_S sequence, this column is used to group together all the Requests to be processed (all records processed as a group should have the same value).
TRANSACTION_ID	Y (See Note)	Number	A unique identifier for the record. Enter a value from the sequence KNTA_INTERFACE_TXNS_S. If left blank, the column will be derived from this sequence. *Note: If there is a detail record, a unique value for the record must be supplied and it must match the parent_transaction_id of the detail record.*

Table 4-1. KCRT_REQUESTS_INT

Column	Required for Import?	Data Type	Description
PROCESS_PHASE	N	Number	Indicates the current stage of the record as it is being processed. Should be initially set to 1.
PROCESS_STATUS	N	Number	Indicates the current disposition of the record (whether the record has been imported or has failed validation). Should be initially set to 1.
REQUEST_ID	N	Number	The internal identifier for a Request. This should normally be left blank and will default from the KCRT_REQUESTS_S sequence. If a value is entered, it should be derived from the KCRT_REQUESTS_S sequence.
CREATION_DATE	N	Date	The creation date of the new Request. If left blank, it will default to the current date.
CREATED_USERNAME	Y	Varchar2(30)	Required column. The USERNAME from KNTA_USERS of the user generating the Request.
CREATED_BY	N	Number	The USER_ID from KNTA_USERS table for the user generating the Request. This is an internal column that should be left blank. It will be derived from CREATED_USERNAME.
LAST_UPDATE_DATE	N	Date	The last update date of the new Request. If left blank, it will default to the current date.
LAST_UPDATED_ USERNAME	Y	Varchar2(30)	Required column. The USERNAME from KNTA_USERS of the user generating the Request.
LAST_UPDATED_BY	N	Number	The USER_ID from KNTA_USERS table for the user generating the Request. This is an internal column that should be left blank. It will be derived from LAST_UPDATED_USERNAME.
ENTITY_LAST_UPDATE_ DATE	N	Date	Should be left blank. It will default to the current date.

Table 4-1. KCRT_REQUESTS_INT

Column	Required for Import?	Data Type	Description
REQUEST_NUMBER	N	Varchar2(30)	The user key for the Request. Normally, this column is left blank and is set to the REQUEST_ID which is also derived. If a value is entered, it should be unique and should match the value in the REQUEST_ID field.
REQUEST_TYPE_NAME	Y	Varchar2(80)	Required column. The REQUEST_TYPE_NAME from the KCRT_REQUEST_TYPES table that indicates the Request Type for the Request.
REQUEST_TYPE_ID	N	Number	The REQUEST_TYPE_ID from KCRT_REQUEST_TYPES that indicates the Request Type for the Request. This is an internal column that should be left blank. It will be derived from the REQUEST_TYPE_NAME.
REQUEST_SUBTYPE_ NAME	N	Varchar2(80)	The REQUEST_SUB_TYPE_NAME from the KCRT_REQUEST_SUB_TYPES table that indicates the Request Sub Type for the Request. If entered, it must be a valid Sub Type for the specified Request Type.
REQUEST_SUBTYPE_ID	N	Number	The REQUEST_SUB_TYPE_ID from KCRT_REQUEST_SUB_TYPES that indicates the Request Sub Type for the Request. This is an internal column that should be left blank. It will be derived from the REQUEST_SUBTYPE_NAME.
DESCRIPTION	N	Varchar2 (240)	A user visible description of the Request. For information purposes only.
RELEASE_DATE	N	Date	This column signifies when the Request first became active. For new Requests, leave the column blank. When converting existing Requests from a third party system, enter the initial creation date of the Request in the remote system.

Table 4-1. KCRT_REQUESTS_INT

Column	Required for Import?	Data Type	Description
STATUS_NAME	N	Varchar2(80)	The STATUS_NAME from KCRT_STATUSES indicates the current status of the Request. It must be a valid Status attached to the given Request Type and 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	N	Number	The STATUS_ID from KCRT_STATUSES. Indicates the current status of the Request. This is an internal column that should be left blank. It will be derived from the STATUS_NAME.
WORKFLOW_NAME	N	Varchar2(80)	The WORKFLOW_NAME from the KWFL_WORKFLOWS table that specifies the Request Management Workflow that the Request should follow. Should normally be left blank and will default based on the values for Request Type, Department, and Application for the Request.
WORKFLOW_ID	N	Number	The WORKFLOW_ID from the KWFL_WORKFLOWS table that specifies which Request Management Workflow the Request should follow. This is an internal column that should be left blank. It will be derived from the WORKFLOW_NAME.
DEPARTMENT_CODE	N	Varchar2(30)	A non-required column indicating the user-defined department the Request is tied to. This is an internal column that should be left blank. It will be derived from the DEPARTMENT_NAME.
DEPARTMENT_NAME	N	Varchar2(80)	A non-required column indicating the user-defined department name the Request is tied to. If entered, it must be a valid MEANING from the KNTA_LOOKUPS table where LOOKUP_TYPE = ‘DEPT’.
PRIORITY_CODE	N	Varchar2(30)	A non-required column indicating the user-defined priority of the Request. This is an internal column that should be left blank. It will be derived from the PRIORITY_NAME.

Table 4-1. KCRT_REQUESTS_INT

Column	Required for Import?	Data Type	Description
PRIORITY_NAME	N	Varchar2(80)	A non-required column indicating the user-defined priority name of the Request. If entered, it must be a valid MEANING from the KNTA_LOOKUPS table where LOOKUP_TYPE = 'REQUEST_PRIORITY'.
APPLICATION	N	Varchar2(30)	A non-required column indicating the user-defined application the Request is tied to. If entered, it must be a valid LOOKUP_CODE from the KNTA_LOOKUPS table where LOOKUP_TYPE = 'APPLICATION'.
ASSIGNED_TO_ USERNAME	N	Varchar2(30)	The USERNAME from the KNTA_USERS table that should initially be assigned the Request.
ASSIGNED_TO_USER_ID	N	Number	The USER_ID from the KNTA_USERS table for the user who should be assigned the Request. This is an internal column that should be left blank.
ASSIGNED_TO_GROUP_ NAME	N	Varchar2(30)	The SECURITY_GROUP_ID from the KNTA_SECURITY_GROUPS table for the group that should initially be assigned the Request. This column is not required.
ASSIGNED_TO_GROUP_ ID	N	Number	The SECURITY_GROUP_ID from KNTA_SECURITY_GROUPS for the group the Request should initially be assigned to. This is an internal column that should be left blank. It is derived from the ASSIGNED_TO_GROUP_NAME.
PROJECT_CODE	N	Varchar2(30)	A non-required column indicating the user-defined project the Request is tied to. If entered, it must be a valid LOOKUP_CODE from the KNTA_LOOKUPS table where LOOKUP_TYPE = 'PROJECT'.
CONTACT_FIRST_NAME	N	Varchar2(30)	This non-required column can be used to give Contact information for the Request. Should be derived from FIRST_NAME from the KCRT_CONTACTS table. If a value is entered, CONTACT_LAST_NAME should also be populated.

Table 4-1. KCRT_REQUESTS_INT

Column	Required for Import?	Data Type	Description
CONTACT_LAST_NAME	N	Varchar2(30)	This non-required column can be used to give Contact information for the Request. Should be derived from LAST_NAME from KCRT_CONTACTS table. If a value is entered, CONTACT_FIRST_NAME should also be populated.
CONTACT_ID	N	Number	This non-required column can be used to give Contact information for the Request. This is an internal column that should be left blank.
RELEASED_FLAG	N	Varchar2(1)	Required; if set to **Y**, the Request will be Released after import. Otherwise, it will remain Unreleased.
USER_DATA_SET_ CONTEXT_ID	N	Number	An internal column used when User-Defined Fields (User Data) has been specified. Should be left blank.
USER_DATA1..20 VISIBLE_USER_DATA1..20	N	Varchar2 (200)	These columns hold the User-Defined Fields attached to Request headers. Values should be entered only if Request User Data has been defined. These columns will not be validated or defaulted, so when entering values, enter both the user visible values and the internal value or id.
PARAMETER_SET_ CONTEXT_ID	N	Number	An internal column used to determine the detail fields for the Request. It is based on the Request Type attached to the Request and should be left blank.
NOTES	N	Long	This non-required column holds the free-form notes text to be attached to the Request. It will be visible in the **Notes** tab of the Request window. In the Notes field, the Carriage Returns may be represented as '{\n}' and are replaced with actual carriage returns when notes are moved into the notes table. This can be helpful when the interface table is populated through SQL*Loader.
SOURCE_TYPE_CODE	N	Varchar2(30)	Indicates how the Request was generated in Request Management. Should be left blank.

Table 4-1. KCRT_REQUESTS_INT

Column	Required for Import?	Data Type	Description
SOURCE	N	Varchar2 (100)	This is a free form column and can be used to indicate the exact source of the Request (the name of the Third Party Application, the text string 'Conversion', etc.).
WORKFLOW_STEP_ID	N	Number	An internal column that should be left blank.
COMPANY	N	Varchar2(30)	The name of the company associated with this Request; associated with any lookup.

Table 4-2. KCRT_REQUEST_DETAILS_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	Y	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_DETAIL_ID	N	Number	Refers to the REQUEST_DETAIL_ID in the KCRT_REQUEST_DETAILS table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.
PARAMETER_SET_ CONTEXT_ID	N	Number	Parameter set context identifier for the custom data fields.
BATCH_NUMBER	Y	Number	Batch number for the custom fields. This is specified in the **Storage** tab in the Field definition window on the Request Type. For more information, see *Configuring a Request Resolution System*.

Table 4-2. KCRT_REQUEST_DETAILS_INT

Column	Required for Import	Data Type	Description
PARAMETER1..40 VISIBLE_PARAMETER1..40	Y (if custom fields exist and are required)	Varchar2 (200)	Refers to Parameter1 through Parameter 40 and Visible Parameter1 through Visible Parameter 40 in the KCRT_REQUEST_DETAILS table.
PARAMETER41..50 VISIBLE_PARAMETER41..50	Y (if custom fields exist and are required)	Varchar2 (1800)	Refers to Parameter1 through Parameter 40 and Visible Parameter1 through Visible Parameter 40 in the KCRT_REQUEST_DETAILS table.
LOOKUP_TYPE1..50	N	Varchar2 (80)	Lookup Type for Parameter1.
VALIDATION_TYPE_CODE1..50	N	Varchar2 (30)	Validation Type Code for Parameter1.

Table 4-3. KCRT_REQ_HEADER_DETAILS_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQ_HEADER_DETAIL_ID	N	Number	Refers to the REQ_HEADER_DETAIL_ID in the KCRT_REQ_HEADER_DETAILS table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.
BATCH_NUMBER	Y	Number	Batch number for the custom fields. This is specified in the **Storage** tab in the Field definition window on the Request Type.

Table 4-3. KCRT_REQ_HEADER_DETAILS_INT

Column	Required for Import	Data Type	Description
PARAMETER1..40 VISIBLE_PARAMETER1..40	Y (if custom fields exist and are required)	Varchar2 (200)	Refers to Parameter1 through Parameter 40 and Visible Parameter1 through Visible Parameter 40 in the KCRT_REQ_HEADER_DETAILS table.
PARAMETER41..50 VISIBLE_PARAMETER41..50	Y (if custom fields exist and are required)	Varchar2 (1800)	Refers to Parameter1 through Parameter 40 and Visible Parameter1 through Visible Parameter 40 in the KCRT_REQ_HEADER_DETAILS table.
LOOKUP_TYPE1..50	N	Varchar2 (80)	Lookup Type for Parameter1.
VALIDATION_TYPE_ CODE1..50	N	Varchar2 (30)	Validation Type Code for Parameter1.

Table 4-4. KCRT_FG_DEMAND_SCHEDULE_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.
SCHEDULE_DATE	N	Date	Date that the Demand was scheduled.
REJECT_DATE	N	Date	Date that the Demand was rejected.
EFFORT	N	Number	Effort associated with the satisfied demand in hours.
DEMAND_SATISFIED_DATE	N	Date	Date that the demand was satisfied.

Table 4-5. KCRT_FG_MASTER_PROJ_REF_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.
REF_MASTER_PROJECT_ ID	N	Varchar2 (200)	If a value is supplied, automatically creates a reference to the specified master Project.
REF_MASTER_PROJECT_ NAME	N	Varchar2 (200)	If a value is supplied, automatically creates a reference to the specified master Project.

Table 4-6. KCRT_FG_PROG_ISSUE_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.

Table 4-6. KCRT_FG_PROG_ISSUE_INT

Column	Required for Import	Data Type	Description
ESCALATION_LEVEL_ CODE	N	Varchar2 (200)	The escalation level code.
ESCALATION_LEVEL_ MEANING	N	Varchar2 (200)	The escalation level meaning.

Table 4-7. KCRT_FG_PROG_REFERENCE_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.
REF_PROGRAM_ID	N	Varchar2 (200)	The ID of the referenced program. If a value is supplied, automatically creates a reference to the specified Program.
REF_PROGRAM_NAME	N	Varchar2 (200)	The name of the referenced Program. If a value is supplied, automatically creates a reference to the specified Program.

Table 4-8. KCRT_FG_PROG_RESOURCE_REQ_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.
ROLE_DESCRIPTION_ CODE	N	Varchar2 (1800)	Description of the resource's role.

Table 4-9. KCRT_FG_PROG_RISK_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.

Table 4-9. KCRT_FG_PROG_RISK_INT

Column	Required for Import	Data Type	Description
PROBABILITY_CODE	N	Varchar2 (200)	Probability of the program risk.
PROBABILITY_MEANING	N	Varchar2 (200)	Probability of the program risk.
RISK_IMPACT_LEVEL_ CODE	N	Varchar2 (200)	Impact level of the program's risk.
RISK_IMPACT_LEVEL_ MEANING	N	Varchar2 (200)	Impact level of the program's risk.

Table 4-10. KCRT_FG_PROG_SCOPE_CHANGE_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.
IMPACT_SEVERITY_CODE	N	Varchar2 (200)	Severity of the impact of the scope change.
IMPACT_SEVERITY_ MEANING	N	Varchar2 (200)	Severity of the impact of the scope change.
CR_LEVEL_CODE	N	Varchar2 (200)	Change request level importance of the scope change.
CR_LEVEL_MEANING	N	Varchar2 (200)	Change request level importance of the scope change.

Table 4-11. KCRT_FG_SLA_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.
SERVICE_REQUESTED_ DATE	N	Date	Date that the service was requested.
SLA_LEVEL_CODE	N	Varchar2(30)	Service level agreement level.
SLA_LEVEL	N	Varchar2 (100)	Service level agreement level.
VIOLATION_DATE	N	Date	Date that the SLA rule was violated.
SERVICE_SATISFIED_DATE	N	Date	Date that the service was satisfied.

Table 4-12. KCRT_FG_WORK_ITEMS_INT

Column	Required for Import	Data Type	Description
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
TRANSACTION_ID	N	Number	Unique identifier for a record in the group.

Table 4-12. KCRT_FG_WORK_ITEMS_INT

Column	Required for Import	Data Type	Description
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table.
REQUEST_ID	N	Number	Refers to the REQUEST_ID in the KCRT_REQUESTS table.
REQUEST_TYPE_ID	N	Number	Refers to the REQUEST_TYPE_ID in the KCRT_REQUEST_TYPES table.
PROCESS_PHASE	N	Number	Indicates the current stage of the record as it is being processed. Should be initially set to 1.
PROCESS_STATUS	N	Number	Indicates the current disposition of the record (whether the record has been imported or has failed validation). Should be initially set to 1.
WORKLOAD_FLAG	N	Varchar2 (1)	Indicates whether this Request should count as workload against Resource capacity. “Y” or “N.”
WORKLOAD_FLAG_ MEANING	N	Varchar2 (200)	Text associated with WORKLOAD_FLAG. Either “Yes” or “No.”
WORKLOAD_CATEGORY_ CODE	N	Varchar2 (30)	If the workload represented by this Request falls under a category, this is the code for that category (lookup type = RSC - Workload Category Code).
WORKLOAD_CATEGORY_ MEANING	N	Varchar2 (200)	Meaning or visible text for corresponding WORKLOAD_CATEGORY_CODE (lookup type = “RSC - Workload Category Code).
ALLOW_EXTERNAL_ UPDATE_FLAG	N	Varchar2 (1)	Indicates whether the actuals can be updated by an external system (such as Time Management Time Sheets). “Y” or “N.”
USR_SCHEDULED_START_ DATE	N	Date	Internal timestamp when work item is scheduled to start—same day as SCHEDULED_START_DATE—but the time may not be 8AM.
USR_SCHEDULED_FINISH_ DATE	N	Date	Internal timestamp when work item is scheduled to finish—same day as SCHEDULED_FINISH_DATE—but the time may not be the end of the working day.

Table 4-12. KCRT_FG_WORK_ITEMS_INT

Column	Required for Import	Data Type	Description
SCHEDULED_START_DATE	N	Date	Date that the work item is scheduled to start, at 8:00 AM on that day.
SCHEDULED_FINISH_DATE	N	Date	Date that the work item is scheduled to finish, at the end of the working day.
SCHEDULED_EFFORT	N	Number	Effort in hours. Usually equal to (Duration) x (Hours/Day)
SCHEDULED_DURATION	N	Number	(Days) Number of working days between USR_SCHEDULED_START_DATE and USR_SCHEDULED_FINISH_DATE.
SCHED_EFF_OVER_DUR	N	Number	(No units) Helper column used when calculation actuals.
USR_ACTUAL_START_DATE	N	Date	Internal timestamp when work item is scheduled to start -- same day as ACTUAL_START_DATE, but the time may not be 8AM.
USR_ACTUAL_FINISH_DATE	N	Date	Internal timestamp when work item is scheduled to finish -- same day as ACTUAL_FINISH_DATE, but the time may not be the end of the working day.
ACTUAL_START_DATE	N	Date	Date that the work item actually starts, at 8:00 AM on that day.
ACTUAL_FINISH_DATE	N	Date	Date that the work item finishes, at the end of the working day.
ACTUAL_EFFORT	N	Number	Effort in hours. Usually equal to (Duration) x (Hours/Day)
ACTUAL_DURATION	N	Number	(Days) Number of working days between USR_ACTUAL_START_DATE and USR_ACTUAL_FINISH_DATE.
ACTUAL_EFF_OVER_DUR	N	Number	(No units) Helper column used when calculation actuals.
BOOKED_SKILL_ID	N	Number	ID of booked Skill. Should match a SKILL_ID in KRSC_SKILLS table.
BOOKED_SKILL_NAME	N	Varchar2 (200)	Skill name booked on this Request. Should match a SKILL_NAME in KRSC_SKILLS table.

Running the Request Open Interface

The following sections describe the process for running Request Open Interface:

- *Loading Data into the Request Interface Tables*
- *Using the Run ITG Request Interface Program*

Loading Data into the Request Interface Tables

The initial step in importing Requests into Request Management is populating the Open Interface table. This 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 communication through database links.

The load 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).

The following section discusses the data that is required for a successful import.

Required Data for Request Import

Certain columns in the Request Open Interface tables must be populated for a Request import. *Table 4-13* defines the columns in the tables that need to be populated for the import. For a complete table description, see *"Data Model"* on page 63.

Table 4-13. KCRT_REQUESTS_INT and KCRT_REQUEST_DETAILS_INT Columns Required for Import

Column	Required for Import?	DataType	Description
KCRT_REQUESTS_INT Table			
GROUP_ID	Y	Number	Derived from the KNTA_INTERFACE_GROUPS_S sequence, this column is used to group together all the Requests to be processed (all records processed as a group should have the same value).

Table 4-13. KCRT_REQUESTS_INT and KCRT_REQUEST_DETAILS_INT Columns Required for Import

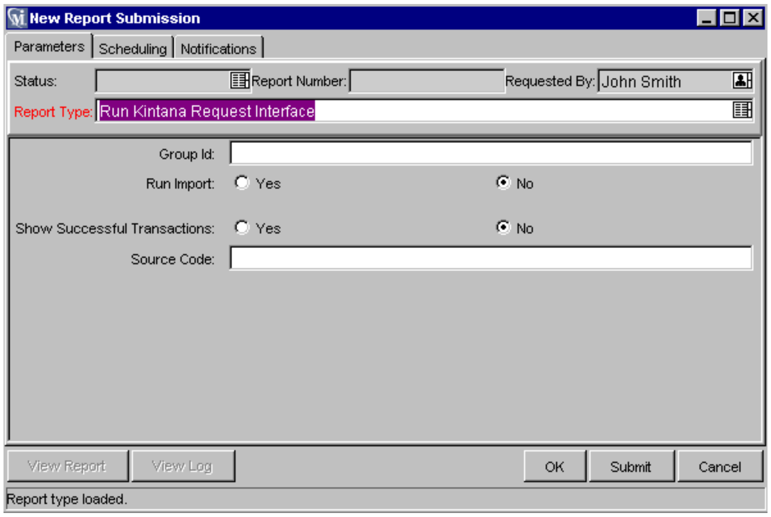
Column	Required for Import?	DataType	Description
TRANSACTION_ID	Y (See Note)	Number	A unique identifier for the record. Enter a value from the sequence KNTA_INTERFACE_TXNS_S. If left blank, the column will be derived from this sequence. *Note: If there is a detail record, a unique value for the record must be supplied and it must match the parent_transaction_id of the detail record.*
CREATED_USERNAME	Y	Varchar2(30)	Required column. The USERNAME from KNTA_USERS of the user generating the Request.
LAST_UPDATED_USERNAME	Y	Varchar2(30)	Required column. The USERNAME from KNTA_USERS of the user generating the Request.
REQUEST_TYPE_NAME	Y	Varchar2(80)	Required column. The REQUEST_TYPE_NAME from the KCRT_REQUEST_TYPES table that indicates the Request Type for the Request.
KCRT_REQUEST_DETAILS_INT Table			
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together as a group. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
PARENT_TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table. Links detailed record to a unique parent record.
BATCH_NUMBER	Y	Number	Batch number for the custom fields.
PARAMETER1..40 VISIBLE_PARAMETER1..40	Y (if custom fields exist)	Varchar2(200)	Refers to Parameter1 through Parameter 40 and Visible Parameter1 through Visible Parameter 40 in the KCRT_REQUEST_DETAILS table.

Table 4-13. KCRT_REQUESTS_INT and KCRT_REQUEST_DETAILS_INT Columns Required for Import

Column	Required for Import?	DataType	Description
PARAMETER41..50 VISIBLE_PARAMETER 41..50	Y (if custom fields exist)	Varchar2(1800)	Refers to Parameter1 through Parameter 40 and Visible Parameter1 through Visible Parameter 40 in the KCRT_REQUEST_DETAILS table.
KCRT_REQ_HEADER_DETAILS_INT Table -- Required ONLY if defined custom fields on the Request Header Types are associated with the Requests being imported.			
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together as a group. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
PARENT_ TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table. Links detailed record to a unique parent record.
BATCH_NUMBER	Y	Number	Batch number for the custom fields.
PARAMETER1..40 VISIBLE_PARAMETER 1..40	Y (if custom fields exist)	Varchar2(200)	Refers to Parameter1 through Parameter 40 and Visible Parameter1 through Visible Parameter 40 in the KCRT_REQ_HEADER_DETAILS table.
PARAMETER41..50 VISIBLE_PARAMETER 41..50	Y (if custom fields exist)	Varchar2(1800)	Refers to Parameter1 through Parameter 40 and Visible Parameter1 through Visible Parameter 40 in the KCRT_REQ_HEADER_DETAILS table.
KCRT_FG_<NAME>_INT Table -- Required ONLY if when importing field information stored in a Field Group associated with the Request.			
GROUP_ID	Y	Number	This value is used to tie records in the interface tables together as a group. Most of database processing is done based on this value. The value should be the same as the parent's GROUP_ID in KCRT_REQUEST_INT table.
PARENT_ TRANSACTION_ID	Y	Number	Identifier to link with the records in the TRANSACTION_ID in the KCRT_REQUESTS_INT table. Links detailed record to a unique parent record.

Using the Run ITG Request Interface Program

To process the data in the interface tables, use the Run ITG Request Interface program. This program is defined as a standard Request Management Report Type and launched through the Request Management Reports window. For more information on using Report Types and launching Reports, see *Reports Guide and Reference*.



Status:

Report Number:

Requested By: John Smith

Report Type: Run Kintana Request Interface

Group Id:

Run Import: ○ Yes ● No

Show Successful Transactions: ○ Yes ● No

Source Code:

Report type loaded.

Figure 4-2 The Run Request Management Request Interface Report

When executed, the Run ITG Request Interface program:

- Queries the KCRT_REQUESTS_INT interface table for active records matching the given selection criteria.
- Defaults any information that has defaulting rules in Request Management but has not been specified in the interface table records. For example, if the REQUEST_ID column is left blank, it will be 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 graphical user interface. Information in User Data fields is not validated.
- Imports validated Requests into the Request 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.

Run ITG Request Open Interface Program Parameters

The Run ITG Request Interface program has several parameters for controlling the behavior of the program execution.

Table 4-14. Run ITG Request Interface Parameters

Parameter	Field Type	Description
Group ID	Text Field	Runs the interface for a specific Group ID. 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. If this parameter is left blank, the program will only pick up records with a NULL GROUP_ID value.
Run Import	Radio Button	If set to **Yes**, the program will process the records in the interface table and try to import them. If set to **No**, the program will simply report on the records in the interface table. This option is useful when auditing prior executions of the Open Interface.
Show Successful Transactions	Radio Button	Shows Requests that were successfully imported.
Source Code	Text Field	Used to set the “source_code” column of the final Requests created with a free form text code. It is used as an indicator of how the Request was created for auditing or testing purposes.

Post-Import Activities

Following the Request import, view the results of the import. From the Report screen, click **View Report** to open the report. This report will identify any errors with the import. After reviewing the import results, proceed with the following activities:

- *Correcting Failures*

- *Purging the Interface Tables*

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 Request Management user interface.

For Requests 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 failures might occur due to a mapping problem between the source data and existing Request Management data.

Example

The source data might use a Project name that does not exist in Request Management. Corrective measures for this specific problem would include adding the specific Project in Request Management, or mapping the source Project to a Project name that already exists in Request Management.

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

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 Request 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 ITG Center source and Request 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 Request Management.

Purging the Interface Tables

All interface tables are automatically purged by the Purge service. The purging process depends on two parameters in the server.conf file:

- ENABLE_INTERFACE_CLEANUP - enables or disables the Purge process.
- DAYS_TO_KEEP_INTERFACE_ROWS - determines the number of days that records are retained in the interface tables.

Chapter 5 Package Open Interface

In addition to a Web-based user interface for the entry of new Packages and Package Lines, the Change Management application includes an open interface for Package creation and the creation of new Package Lines. This API uses interface tables within the Change Management database instance. Data added to these interface tables is validated and eventually imported into standard Change Management tables. This generates Packages and Package Lines that can be processed using Change Management.

The primary purpose of the Package Open Interface is to allow integration with non-Mercury ITG 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 Change Management during initial implementation.

This chapter includes the following information related to the Package Open Interface:

- *Data Model*
- *Running the Package Open Interface*

Note

Integration between the products in Mercury ITG 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 Change Management Packages from Request Management Requests.

Data Model

The interface tables used by the Change Management Package Open Interface closely relate to the standard tables used to store Package information.

- **KDLV_PACKAGES_INT** - This interface 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).
- **KDLV_PACKAGE_LINES_INT** - 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 App Code for the Package Line, as well as parameter information for the specified Object Type and User Data for the Package Line.
- **KDLV_PACKAGE_NOTES_INT** - This interface table is used to import 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.

Figure 5-1 displays the relationships between the **KDLV_PACKAGES_INT**, **KDLV_PACKAGE_LINES_INT**, and the **KDLV_PACKAGE_NOTES_INT** tables.

Note

Each table includes a “Required for Import” column. This column, and other related requirements, are discussed in *“Loading Data into the Package Interface Tables”* on page 101.

The KNTA_INTERFACE_ERRORS table is displayed for reference only and should NOT be altered.

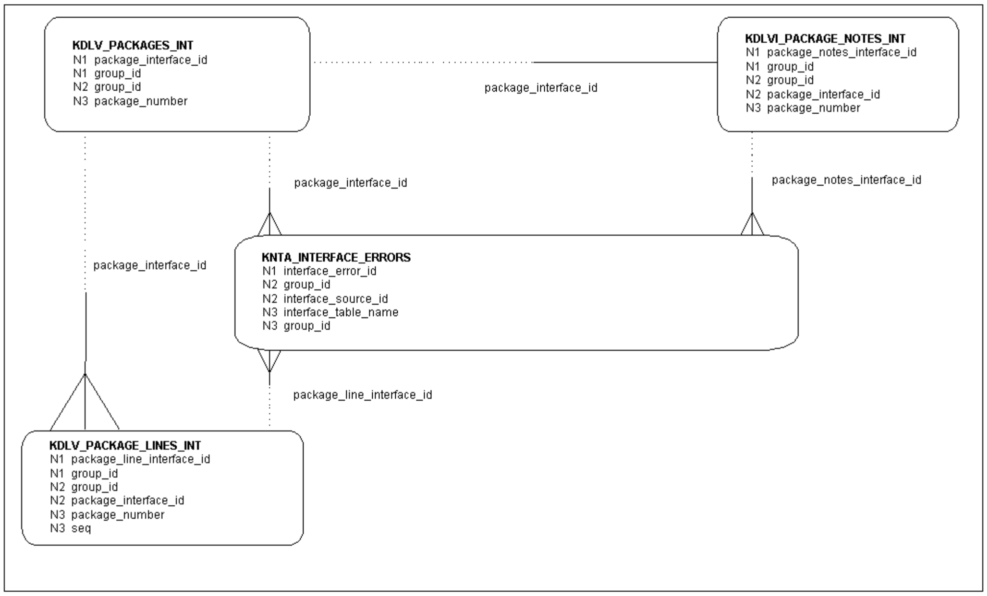


Figure 5-1 Interface and Supporting Tables

Table 5-1. KDLV_PACKAGES_INT

Column	Required for Import?	Data Type	Description
PACKAGE_INTERFACE_ID	Y (See Note)	Number	A unique identifier for the record. Enter a value from the sequence KDLV_INTERFACES_S. If left blank, the column will be derived from this sequence. *Note: This column is required if package_lines exist.*
GROUP_ID	Y	Number	Derived from KDLV_INTERFACES_S, this column is used to group together all Package Headers, Lines, and note information to be processed (all records should have the same value).
PROCESS_PHASE	N	Number	Indicates the current stage of the record as it is being processed. Should initially be set to 1.

Table 5-1. KDLV_PACKAGES_INT

Column	Required for Import?	Data Type	Description
PROCESS_STATUS	N	Number	Indicates the current disposition of the record (whether the record has been imported or has failed validation). Should initially be set to 1.
CREATED_BY	N	Number	The USER_ID from KNTA_USERS table for the user generating the Package. If left blank, it will be derived from CREATED_BY_USERNAME or will be set to the user currently running the interface.
CREATED_BY_USERNAME	Y	Varchar2(30)	The USERNAME from KNTA_USERS of the user generating the Package. Will only be used if CREATED_BY is left blank.
CREATION_DATE	N	Date	The creation date of the new Package. If left blank, it will default to the current date.
SOURCE_CODE	N	Varchar2(30)	A non-validated column used to identify the source of the record. Used for information purposes only.
PACKAGE_ID	N	Number	The internal identifier for a Package. Unlike the Request Management Open interface, this *must not be left blank*. It should be populated from KDLV_PACKAGES_S to make an association between the Package and the Package Lines.
REQUESTED_BY	Y (See Note)	Number	The USER_ID from the KNTA_USERS table for the user requesting the Package. If left blank, it will be derived from REQUESTED_BY_USERNAME or will be set to the user currently running the interface. *Note: Either this column or the REQUESTED_BY_USERNAME column must be populated.*

Table 5-1. KDLV_PACKAGES_INT

Column	Required for Import?	Data Type	Description
REQUESTED_BY_ USERNAME	Y (See Note)	Varchar2(30)	The USERNAME from the KNTA_USERS table of the user requesting the Package. Will only be used if REQUESTED_BY is left blank. *Note: Either this column or the REQUESTED_BY column must be populated.*
PACKAGE_NUMBER	N	Varchar2(40)	The user key for the Package. Unlike the Request Management Open interface, this *must not be left blank*. It should be populated with the same value as PACKAGE_ID or a unique string value.
ASSIGNED_TO_USER_ID	N	Number	The USER_ID from the KNTA_USERS table for the user who should be assigned the Package. If left blank, it will be derived from ASSIGNED_TO_USERNAME. If both columns are left blank, the Package will not have an initial Assigned To user.
ASSIGNED_TO_ USERNAME	Y	Varchar2(30)	The USERNAME from the KNTA_USERS table that should initially be assigned the Package. It will only be used if ASSIGNED_TO_USER_ID is left blank.
ASSIGNED_TO_GROUP_ID	N	Number	The SECURITY_GROUP_ID from KNTA_SECURITY_GROUPS for the group the Package should initially be assigned to. If it is left blank, it will be derived from ASSIGNED_TO_GROUP_NAME. If both columns are left blank, the Package will not have an initial Assigned To group.
ASSIGNED_TO_GROUP_ NAME	Y	Varchar2(30)	The SECURITY_GROUP_ID from the KNTA_SECURITY_GROUPS table for the group that should initially be assigned the Package. This is only used if ASSIGNED_TO_GROUP_ID is left blank.
DESCRIPTION	N	Varchar2 (240)	A user visible description of the Package. For information purposes only.

Table 5-1. KDLV_PACKAGES_INT

Column	Required for Import?	Data Type	Description
PACKAGE_TYPE_CODE	N	Varchar2(30)	A non-required user-defined categorization of the Package. If entered, it must be a valid LOOKUP_CODE from the KNTA_LOOKUPS table where LOOKUP_TYPE = ‘PACKAGE_TYPE’.
PRIORITY_CODE	N	Varchar2(30)	A non-required user-defined prioritization of the Package. If entered, it must be a valid LOOKUP_CODE from the KNTA_LOOKUPS table where LOOKUP_TYPE = ‘PACKAGE_PRIORITY’.
STATUS_CODE	N	Varchar2(30)	An internal indicator for the Package status. Should be left blank and will be defaulted to **NEW**.
PROJECT_CODE	N	Varchar2(30)	A non-required field indicating the user-defined project the Package is tied to. If entered, it must be a valid LOOKUP_CODE from the KNTA_LOOKUPS table where LOOKUP_TYPE = ‘PROJECT’.
WORKFLOW_ID	Y (See Note)	Number	The WORKFLOW_ID from the KWFL_WORKFLOWS table that specifies which Change Management Workflow the Package should follow. If left blank, this will be derived from WORFKLOW_NAME. Either WORKFLOW_ID or WORKFLOW_NAME must be entered. *Note: Either this column or the WORKFLOW_NAME column must be populated.*

Table 5-1. KDLV_PACKAGES_INT

Column	Required for Import?	Data Type	Description
WORKFLOW_NAME	Y (See Note)	Varchar2(80)	The WORKFLOW_NAME from the KWFL_WORKFLOWS table that specifies the Change Management Workflow that the Package should follow. Will only be used if WORKFLOW_ID is left blank. Either WORKFLOW_ID or WORKFLOW_NAME must be entered. *Note: Either this column or the WORKFLOW_ID column must be populated.*
PRIORITY_SEQ	N	Number	A sequence number used to determine the relative priority of Packages that are scheduled to process at the same time. If left blank, it will default to 10.
RELEASE_FLAG	N (See Description for impact)	Varchar2(1)	If set to **Y**, the Interface program will Release the Package once it imports in into the standard Change Management tables. If set to **N** or left blank, the Package will not be released.
USER_DATA_SET_ CONTEXT_ID	N	Number	An internal column used when User-Defined Fields (User Data) have been specified. Can be left blank or set to 1202.
USER_DATA1..20 VISIBLE_USER_DATA1..20	N	Varchar2 (200)	These columns hold the User-Defined Fields attached to Package headers. Values should be entered only if Package User Data has been defined. These columns will not be validated or defaulted, so if entering values, enter both the user visible values and the internal value or ID.
SOURCE_PACKAGE_ID	N	Number	Used with Release Management functionality. Identifier of the original Package for this Distribution Package.
DISTPKG_STATUS_ MEANING	N	Varchar2(80)	Used with Release Management functionality. Displayed status of this Distribution Package.

Table 5-1. KDLV_PACKAGES_INT

Column	Required for Import?	Data Type	Description
RUN_GROUP	N	Number	Used with Release Management functionality. Run Group number of a specific Distribution Package.
DISTRIBUTION_ID	N	Number	Used with Release Management functionality. Identifier of the Distribution that the Package belongs to.
ENABLED_FLAG	N	Varchar2(1)	Used with Release Management functionality. Denote whether this Distribution Package is enabled or disabled.
DIST_STEP_ TRANSACTION_ID	N	Number	Used with Release Management functionality. Path of the Distribution Workflow Step that was executed in the transaction with DIST_STEP_TRANSACTION_ID.

Table 5-2. KDLV_PACKAGE_LINES_INT

Column	Required for Import?	Data Type	Description
PACKAGE_LINE_ INTERFACE_ID	N	Number	A unique identifier for the record. Enter a value from the sequence KDLV_INTERFACES_S. If left blank, the column will be derived from this sequence.
GROUP_ID	Y	Number	Also derived from KDLV_INTERFACES_S, this column is used to group together all Package Header, Line, and note information to be processed (all records should have the same value).
PACKAGE_INTERFACE_ID	Y	Number	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. The PACKAGE_NUMBER and PACKAGE_ID columns can be used for this tie as well. For new Lines to be imported into existing Packages, this column should be left blank.

Table 5-2. KDLV_PACKAGE_LINES_INT

Column	Required for Import?	Data Type	Description
PACKAGE_ID	N	Number	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. The PACKAGE_INTERFACE_ID and PACKAGE_NUMBER columns can be used for this tie as well. For new Lines to be imported into existing Packages, this column should refer to the PACKAGE_ID of the existing Package.
PACKAGE_NUMBER	N	Varchar2(40)	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. The PACKAGE_INTERFACE_ID and PACKAGE_ID columns can be used for this tie as well. For new Lines to be imported into existing Packages, this column should refer to the PACKAGE_NUMBER of the existing Package.
PROCESS_PHASE	N	Number	Indicates the current stage of the record as it is being processed. Should be initially set to 1.
PROCESS_STATUS	N	Number	Indicates the current disposition of the record (whether the record has been imported or has failed validation). Should be initially set to 1.
CREATED_BY	Y	Number	The USER_ID from the KNTA_USERS table for the user generating the Package. If left blank, it will be derived from CREATED_BY_USERNAME or will be set to the user currently running the interface.
CREATED_BY_USERNAME	Y	Varchar2(30)	The USERNAME from the KNTA_USERS table of the user generating the Package. Will only be used if CREATED_BY is left blank.
CREATION_DATE	N	Date	The creation date of the new Package. If left blank, it will default to the current date.

Table 5-2. KDLV_PACKAGE_LINES_INT

Column	Required for Import?	Data Type	Description
SOURCE_CODE	N	Varchar2(30)	A non-validated column used to identify the source of the record. Used for information purposes only.
SEQ	Y	Number	A user visible sequence number for the Package Line. Must be a 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	N	Number	The internal identifier for a Package Line. Normally, this should be left blank and will be defaulted from the KDLV_PACKAGE_LINES_S sequence. If a value is entered, it should be derived from the KDLV_PACKAGE_LINES_S sequence.
OBJECT_TYPE_ID	N	Number	The OBJECT_TYPE_ID from the KDLV_OBJECT_TYPES table for the Object Type attached to the Package Line. If left blank, it will be derived from OBJECT_TYPE_NAME. Either OBJECT_TYPE_ID or OBJECT_TYPE_NAME must be entered.
OBJECT_TYPE_NAME	Y	Varchar2(80)	The OBJECT_TYPE_NAME from the KDLV_OBJECT_TYPES table for the Object Type attached to the Package Line. It will only be used if OBJECT_TYPE_ID is left blank. Either OBJECT_TYPE_ID or OBJECT_TYPE_NAME must be entered.
OBJECT_NAME	Y	Varchar2 (300)	A non-validated, required value indicating the name of the object to be processed.

Table 5-2. KDLV_PACKAGE_LINES_INT

Column	Required for Import?	Data Type	Description
APP_CODE	N	Varchar2(30)	The APP_CODE from the KDLV_ENVIRONMENT_APPS table indicating the application category for the Package Line. 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. This field can also be left blank.
PARAMETER_SET_ CONTEXT_ID	N	Number	An internal column derived from the OBJECT_TYPE_ID. This should be left blank.
PARAMETER1..30 VISIBLE_PARAMETER1..30	N	Varchar2 (200)	These columns hold the Parameters attached to the Package Line. The specific parameters are based on the Object Type attached to the Line. These columns will not be validated or defaulted, so if entering values, enter both the user visible values and the internal value or id.
RELEASE_FLAG	N	Varchar2(1)	An internal column used by the Interface program. It should be left blank.
USER_DATA_SET_ CONTEXT_ID	N	Number	An internal column used when user-defined fields (User Data) has been specified. It can be left blank or set to 1203.
USER_DATA1..20 VISIBLE_USER_DATA1..20	N	Varchar2 (200)	These columns hold the user-defined fields attached to Package headers. Values should be entered only if Package User Data has been defined. These columns will not be validated or defaulted, so if entering values, enter both the user visible values and the internal value or id.
OBJECT_REVISION	N	Varchar2 (300)	The denormalized object_revision of the object entered on this Line.
SOURCE_PACKAGE_LINE_ ID	N	Number	Used with Release Management functionality. Identifier of the original Package Line for this Distribution Package Line.

Table 5-2. KDLV_PACKAGE_LINES_INT

Column	Required for Import?	Data Type	Description
ENABLED_FLAG	N	Varchar2(1)	Used with Release Management functionality. Denotes whether this Distribution Package Line is enabled or disabled.

Table 5-3. KDLV_PACKAGE_NOTES_INT

Column	Required for Import?	Data Type	Description
PACKAGE_NOTE_ INTERFACE_ID	N	Number	A unique identifier for the record. Enter a value from the sequence KDLV_INTERFACES_S. If it is left blank, the column will be derived from this sequence.
GROUP_ID	Y	Number	Also derived from KDLV_INTERFACES_S, this column is used to group together all Package Header, Line, and note information to be processed. All records should have the same value.
PACKAGE_INTERFACE_ID	Y	Number	This column can be used to tie the Note 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	N	Number	This column can be used to tie the Note record to the parent record in KDLV_PACKAGES_INT. The PACKAGE_INTERFACE_ID and PACKAGE_NUMBER columns can be used for this tie as well.
PACKAGE_NUMBER	N	Varchar2 (40)	This column can be used to tie the Note record to the parent record in KDLV_PACKAGES_INT. The PACKAGE_INTERFACE_ID and PACKAGE_ID columns can be used for this tie as well.
PROCESS_PHASE	N	Number	Indicates the current stage of the record as it is being processed. Should be initially set to 1.

Table 5-3. KDLV_PACKAGE_NOTES_INT

Column	Required for Import?	Data Type	Description
PROCESS_STATUS	N	Number	Indicates the current disposition of the record (whether the record has been imported or has failed validation). Should be initially set to 1.
CREATED_BY	Y	Number	The USER_ID from the KNTA_USERS table for the user generating the Package. If left blank, it will be derived from CREATED_BY_USERNAME or will be set to the user currently running the interface.
CREATED_BY_USERNAME	N	Varchar2 (30)	The USERNAME from the KNTA_USERS table of the user generating the Package. It will only be used if CREATED_BY is left blank.
CREATION_DATE	N	Date	The creation date of the new Package. If it is left blank, it will default to the current date.
SOURCE_CODE	N	Varchar2 (30)	A non-validated column used to identify the source of the record. Used for information purposes only.
NOTE	Y	Long	The full text of the Note.
REPLACE_NOTE_FLAG	N	Varchar2(1)	This column is not used by the current Interface program.

Running the Package Open Interface

The following sections describe the process for running Package Open Interface:

- *Loading Data into the Package Interface Tables*
- *Using the Run ITG Package Interface Program*

Loading Data into the Package Interface Tables

The initial step in importing Packages and Package Lines into Change Management is populating the Open Interface tables. This 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 communication through database links.

This load process can also involve the manipulation of the table records once they have been brought into the interface tables. 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 Package (such as the third-party application or the ASCII file).

The following section discusses the data that is required for a successful import.

Required Data for Package Import

Certain columns in the KDLV_PACKAGES_INT, KDLV_PACKAGE_LINES_INT, and KDLV_PACKAGE_NOTES_INT tables must be populated for a Package import. *Table 5-4* defines the columns in the tables that need to be populated for the import. For a complete table description, see *Table 5-1 on page 91*, *Table 5-2 on page 96*, and *Table 5-3 on page 100*.

Table 5-4. Columns Required for Package Import

Column	Required for Import?	Data Type	Description
KDLV_PACKAGES_INT Table			
PACKAGE_INTERFACE_ID	Y (See Note)	Number	A unique identifier for the record. Enter a value from the sequence KDLV_INTERFACES_S. If left blank, the column will be derived from this sequence. *Note: This column is required if package_lines exist.*
GROUP_ID	Y	Number	Derived from KDLV_INTERFACES_S, this column is used to group together all Package Headers, Lines, and note information to be processed (all records should have the same value).
CREATED_BY_ USERNAME	Y	Varchar2(30)	The USERNAME from KNTA_USERS of the user generating the Package. Will only be used if CREATED_BY is left blank.

Table 5-4. Columns Required for Package Import

Column	Required for Import?	Data Type	Description
REQUESTED_BY	Y (OR NEXT)	Number	The USER_ID from the KNTA_USERS table for the user requesting the Package. If left blank, it will be derived from REQUESTED_BY_USERNAME or will be set to the user currently running the interface.
REQUESTED_BY_ USERNAME	Y (OR PREVIOUS)	Varchar2(30)	The USERNAME from the KNTA_USERS table of the user requesting the Package. Will only be used if REQUESTED_BY is left blank.
ASSIGNED_TO_ USERNAME	Y	Varchar2(30)	The USERNAME from the KNTA_USERS table that should initially be assigned the Package. It will only be used if ASSIGNED_TO_USER_ID is left blank.
ASSIGNED_TO_GROUP_ NAME	Y	Varchar2(30)	The SECURITY_GROUP_ID from the KNTA_SECURITY_GROUPS table for the group that should initially be assigned the Package. This is only used if ASSIGNED_TO_GROUP_ID is left blank.
WORKFLOW_ID	Y (WORKFLOW _NAME)	Number	The WORKFLOW_ID from the KWFL_WORKFLOWS table that specifies which Change Management Workflow the Package should follow. If left blank, this will be derived from WORFKLOW_NAME. Either WORKFLOW_ID or WORKFLOW_NAME must be entered.
WORKFLOW_NAME	Y (OR WORKFLOW_ ID)	Varchar2(80)	The WORKFLOW_NAME from the KWFL_WORKFLOWS table that specifies the Change Management Workflow that the Package should follow. Will only be used if WORKFLOW_ID is left blank. Either WORKFLOW_ID or WORKFLOW_NAME must be entered.

Table 5-4. Columns Required for Package Import

Column	Required for Import?	Data Type	Description
RELEASE_FLAG	N, BUT HAS IMPACTS	Varchar2(1)	If set to 'Y', the Interface program will Release the Package once it imports in into the standard Change Management tables. If set to 'N' or left blank, the Package will not be released.
KDLV_PACKAGE_LINES_INT Table			
GROUP_ID	Y	Number	Also derived from KDLV_INTERFACES_S, this column is used to group together all Package Header, Line, and note information to be processed (all records should have the same value).
PACKAGE_INTERFACE_ID	Y	Number	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. The PACKAGE_NUMBER and PACKAGE_ID columns can be used for this tie as well. For new Lines to be imported into existing Packages, this column should be left blank.
CREATED_BY	Y	Number	The USER_ID from the KNTA_USERS table for the user generating the Package. If left blank, it will be derived from CREATED_BY_USERNAME or will be set to the user currently running the interface.
CREATED_BY_ USERNAME	Y	Varchar2(30)	The USERNAME from the KNTA_USERS table of the user generating the Package. Will only be used if CREATED_BY is left blank.
SEQ	Y	Number	A user visible sequence number for the Package Line. Must be a positive integer and not conflict with other Package Lines in the interface table or existing Lines if importing Lines to an existing Packages.

Table 5-4. Columns Required for Package Import

Column	Required for Import?	Data Type	Description
OBJECT_TYPE_NAME	Y	Varchar2(80)	The OBJECT_TYPE_NAME from the KDLV_OBJECT_TYPES table for the Object Type attached to the Package Line. It will only be used if OBJECT_TYPE_ID is left blank. Either OBJECT_TYPE_ID or OBJECT_TYPE_NAME must be entered.
OBJECT_NAME	Y	Varchar2 (300)	A non-validated, required value indicating the name of the object to be processed.
PARAMETER1..30 VISIBLE_PARAMETER 1..30	N, BUT IF YOU HAVE DATA USE IT	Varchar2 (200)	These columns hold the Parameters attached to the Package Line. The specific parameters are based on the Object Type attached to the Line. These columns will not be validated or defaulted, so if entering values, enter both the user visible values and the internal value or ID.
KDLV_PACKAGE_NOTES_INT Table			
GROUP_ID	Y	Number	Also derived from KDLV_INTERFACES_S, this column is used to group together all Package Header, Line, and note information to be processed. All records should have the same value.
PACKAGE_INTERFACE_ID	Y	Number	This column can be used to tie the Note record to the parent record in KDLV_PACKAGES_INT. The PACKAGE_NUMBER and PACKAGE_ID columns can be used for this tie as well.
CREATED_BY	Y	Number	The USER_ID from the KNTA_USERS table for the user generating the Package. If left blank, it will be derived from CREATED_BY_USERNAME or will be set to the user currently running the interface.
NOTE	Y	Long	The full text of the Note.

Using the Run ITG Package Interface Program

To process the data in the interface tables, use the Run ITG Package Interface program. This program is defined as a standard Change Management Report Type and launched through the Change Management Reports window. For more information on using Report Types and launching Reports, see *Reports Guide and Reference*.

To generate an executable version of the program using the Report Types Workbench:

1. Click the **Configuration** shortcut group and then the **Report Types** icon. The Report Types workbench opens.

2. Search for the Run ITG Package Interface program.

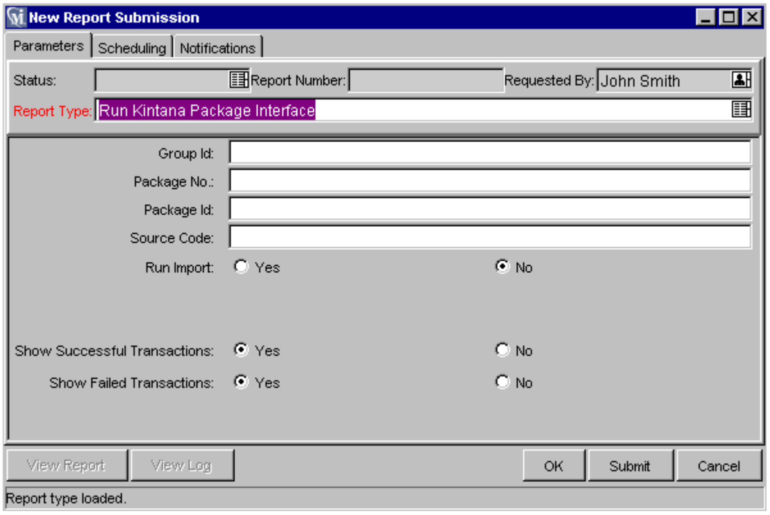
3. To copy the reference Report Type, click **Copy**.

4. Rename the copy Run ITG Package Interface.

5. Click **Yes** to the “Would you like to edit” prompt.

   The Report Type window opens.

6. Select **Yes** next to Enabled to make this Report Type available in Change Management.



Status:
Report Number:
Requested By: John Smith
Report Type: Run Kintana Package Interface

Group Id:
Package No.:
Package Id:
Source Code:
Run Import: ○ Yes ● No
Show Successful Transactions: ● Yes ○ No
Show Failed Transactions: ● Yes ○ No

Report type loaded.

Figure 5-2 The Run ITG Package Interface Report

When executed, the Run ITG Package Interface program:

- Queries the interface tables for active records matching the given selection criteria.
- Defaults any information that has defaulting rules in 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 graphical user 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.

Run ITG Package Interface Program Parameters

The Run ITG Package Interface program has several parameters for controlling the behavior of the program execution.

Table 5-5. Run ITG Package Interface Parameters

Parameter	Field Type	Description
Group Id	Text Field	Runs the interface for a specific Group ID. 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. If this parameter is left blank, the program will only pick up records with a NULL GROUP_ID value.
Package No.	Text Field	Runs the interface for a specific Package Number. The interface program will only look for records with this value in the PACKAGE_NUMBER column. This is useful when importing a specific Package.

Table 5-5. Run ITG Package Interface Parameters

Parameter	Field Type	Description
Package Id	Text Field	Runs the interface for a specific Package ID. 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	Text Field	Runs the interface for records from a specific source. The interface program will only look for records with this value in the SOURCE_CODE column.This is useful for running the open interface at different frequencies depending on the driver of the Package creation (for example, to have Packages initiated from the Project Management system imported once a day, while having Packages generated to solve problems detected by the performance monitor imported within an hour).
Run Import	Radio Button	If set to **Yes**, the program will process the records in the interface table and try to import them. If set to **No**, the program will simply report on the records in the interface table. This option is useful when auditing prior executions of the Open Interface.
Show Successful Transactions	Radio Button	Shows Packages and Package Lines that were successfully imported.
Show Failed Transactions	Radio Button	Shows Packages and Package Lines that were not successfully imported.

Post-Import Activities

Following the Request import, view the results of the import. From the Report screen, click **View Report** to open the report. This report will identify any errors with the import. After reviewing the import results, proceed with the following activities:

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 the Change Management user interface.

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 Change Management data.

Example

The source data might use a Project name that does not exist in Change Management. Corrective measures for this specific problem would include adding the specific Project in Change Management, or mapping the source Project to a Project name that already exists in Change Management.

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

Example

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 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 non-Mercury ITG Center source and 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 Change Management.

Chapter 6 Workflow Transaction Open Interface

In addition to a Web-based user interface for performing executions and approving Workflow steps, Mercury IT 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. Note that this feature is not supported through the standard user interface.

The Workflow Transactions Open Interface is a set of tables within the Mercury ITG Center database instance. 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 Open Interface is to allow integration with non-Mercury ITG 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 ITG Center during initial implementation.

This chapter includes the following information related to the Package Open Interface:

- *Data Model*
- *Running the Workflow Open Interface*

Note

Currently, the Workflow 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 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. e.g. 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.

Data Model

The interface table used by the Workflow Transactions Open Interface closely relates to the standard table used to store Workflow Transactions:

- **KWFL_TRANSACTIONS_INT** - This interface stores 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.

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

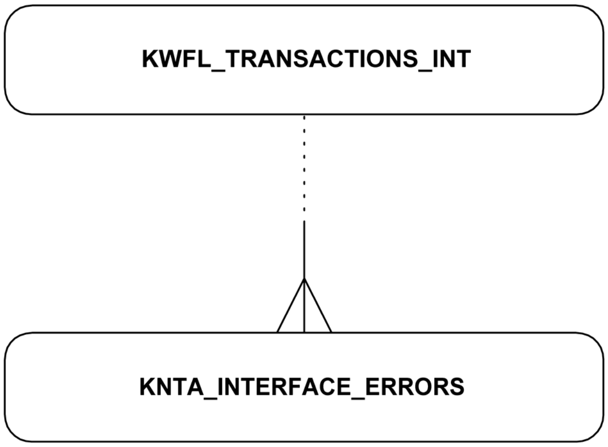


Figure 6-1 Interface and Supporting Tables

Note

Each table includes a "Required for Import" column. This column, and other related requirements, are discussed in *"Loading Data into the Workflow Interface Tables"* on page 119.

The KNTA_INTERFACE_ERRORS table is displayed for reference only and should *not* be altered by the user.

Table 6-1. KWFL_TRANSACTIONS_INT

Column	Required for Import?	Data Type	Description
TRANSACTION_ID	N	Number	Required column. A unique identifier for the record. A value for this column is derived from the sequence KNTA_INTERFACE_TXNS_S.
CREATION_DATE	N	Date	The date that the transaction is performed.
CREATED_USERNAME	Y (See Note)	Varchar2(30)	Required column. The USERNAME from the KNTA_USERS table for the user that is performing the transaction. This will be used only if the CREATED_BY is null. *Note: Either this column or the CREATED_BY column must be populated.*
CREATED_BY	Y (See Note)	Number	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from CREATED_USERNAME. *Note: Either this column or the CREATED_USERNAME column must be populated.*
LAST_UPDATE_DATE	N	Date	The date that the transaction is performed.
LAST_UPDATED_ USERNAME	Y (See Note)	Varchar2(30)	The USERNAME from the KNTA_USERS table for the user that is performing the transaction. This will be used only if the LAST_UPDATED_BY is null. *Note: Either this or the LAST_UPDATED_BY column must be populated.*
LAST_UPDATED_BY	Y (See Note)	Number	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from LAST_UPDATED_USERNAME. *Note: Either this or the LAST_UPDATED_USERNAME column must be populated.*
EVENT	N	Varchar2(40)	Required column. The type of Workflow transaction.

Table 6-1. KWFL_TRANSACTIONS_INT

Column	Required for Import?	Data Type	Description
GROUP_ID	Y	Number	A unique identifier for the group of records that are being processed. A value for this column is derived from the sequence KNTA_INTERFACE_GROUPS_S.
WORKFLOW_ENGINE_ BATCH_ID	N	Number	Derived from other information in the record. This value should always be inserted as null.
PROCESS_PHASE	N	Number	Required column. Indicates the current stage of the record as it is being processed. A record goes through the following stages as it is processed. The value should initially be set to 1. • 1 - Pending • 2 - Derivation • 3 - Validation • 6 - Final Validation • 7 - Batch Processing • 5 - Completed
PROCESS_STATUS	N	Number	Required column. Indicates the current disposition of the record. The value should initially be set to 1. • 1 - Pending • 2 - In Process • 3 - Error • 7 - Completed
SOURCE_TYPE_CODE	N	Varchar2(30)	This column should either be null or have the value 'INTERFACE_WF'.
SOURCE	N	Varchar2 (100)	This is an optional, non-validated column that can be used to indicate the exact source of the Workflow transaction (such as the name of the third party application or the text string 'Conversion').
INSTANCE_SOURCE_ TYPE_CODE	Y	Varchar2(30)	Indicates whether the transaction is for a Package Line (‘CR’) or a Request (‘IR').

Table 6-1. KWFL_TRANSACTIONS_INT

Column	Required for Import?	Data Type	Description
INSTANCE_SOURCE_SET _NUMBER	N	Varchar2(40)	The Package Number (PACKAGE_NUMBER from KDLV_PACKAGES) or Request Number (REQUEST_NUMBER from KCRT_REQUESTS). This will be used only if the INSTANCE_SOURCE_SET_ID is null.
INSTANCE_SOURCE_SET _ID	N	Number	The Package ID (PACKAGE_ID from KDLV_PACKAGES) or Request ID (REQUEST_ID from KCRT_REQUESTS) for the Workflow transaction. If null, the column is derived from INSTANCE_SOURCE_SET_NUMBER.
INSTANCE_SOURCE_ LINE_SEQ	N	Number	The Package Line Sequence Number. This will be used only if the INSTANCE_SOURCE_ID is null.
INSTANCE_SOURCE_ID	N	Number	The Package Line ID (PACKAGE_LINE_ID from KDLV_PACKAGE_LINES) or Request ID (REQUEST_ID from KCRT_REQUESTS). If null, the column is derived from INSTANCE_SOURE_LINE_SEQ (for Package Lines) or INSTANCE_SOURCE_SET_NUMBER (for Requests).
WORKFLOW_STEP_ NAME	N	Varchar2(80)	The name of the Workflow Step (STEP_NAME from KWFL_WORKFLOW_STEPS). This will be used only if the WORKFLOW_STEP_ID is null.
WORKFLOW_STEP_SEQ	N	Varchar2(30)	The sequence number of the Workflow Step. This will be used only if the WORKFLOW_STEP_ID is null. In the case of Subworkflows, the sequence numbers of the Workflow Steps could be in the form of 2.4.5, etc.
RESULT_VALUE	N	Varchar2 (200)	The result of the step. This is normally not displayed to the user, so it may be an ID or internal code.
VISIBLE_RESULT_VALUE	N	Varchar2 (200)	The displayed result of the step. This is the result value that a user normally sees.

Table 6-1. KWFL_TRANSACTIONS_INT

Column	Required for Import?	Data Type	Description
USER_COMMENTS	N	Varchar2 (200)	Comments for the transaction. Any comments are appended to the notes for the Package or Request.
DELEGATED_TO_USERNAME	N	Varchar2(30)	The USERNAME from the KNTA_USERS table for the user that the decision is being delegated to. This will be used only if the DELEGATED_TO_USER_ID is null.
DELEGATED_TO_USER_ID	N	Number	The USER_ID from the KNTA_USERS table for the user that the decision is being delegated to. If null, the column is derived from DELEGATED_TO_USERNAME.
SCHEDULE_DATE	N	Date	For scheduled executions, the date when the execution is to be performed.
WORKFLOW_ID	N	Number	Derived from other information in the record. This value should always be inserted as null.
WORKFLOW_INSTANCE_ID	N	Number	Derived from other information in the record. This value should always be inserted as null.
WORKFLOW_STEP_ID	N	Number	The Workflow Step ID (WORKFLOW_STEP_ID from KWFL_WORKFLOW_STEPS). If null, the column is derived from WORKFLOW_STEP_NAME or WORKFLOW_STEP_SEQ.
WORKFLOW_INSTANCE_STEP_ID	N	Number	Derived from other information in the record. This value should always be inserted as null.
CURRENT_STEP_TRANSACTION_ID	N	Number	Derived from other information in the record. This value should always be inserted as null.
APPROVALS_REQUIRED_CODE	N	Number	Derived from other information in the record. This value should always be inserted as null.
EVENT_GROUP_ID	N	Number	Derived from other information in the record. This value should always be inserted as null.

Table 6-1. KWFL_TRANSACTIONS_INT

Column	Required for Import?	Data Type	Description
CMD_EXECUTION_ SCHD_TASK_ID	N	Number	Derived from other information in the record. This value should always be inserted as null.
TO_WORKFLOW_STEP_ SEQ	N	Varchar2(30)	The sequence number of the Workflow Step for the step that the Package Line, Request, or Project should transition to. This will be used only if the TO_WORKFLOW_STEP_ID is null.
TO_WORKFLOW_STEP_ NAME	N	Varchar2(80)	The name of the Workflow Step (STEP_NAME from KWFL_WORKFLOW_STEPS) for the step that the Package Line, Request, or Project should transition to. This will be used only if the TO_WORKFLOW_STEP_ID is null.
TO_WORKFLOW_STEP_ ID	N	Number	The Workflow Step ID (WORKFLOW_STEP_ID from KWFL_WORKFLOW_STEPS) for the step that the Package Line or Request should transition to. If null, the column is derived from TO_WORKFLOW_STEP_NAME or TO_WORKFLOW_STEP_SEQ.

The following columns are used for internal processing and are always derived from other information in the record. These columns should always be inserted as null.

- WORKFLOW_ID
- WORKFLOW_INSTANCE_ID
- WORKFLOW_INSTANCE_STEP_ID
- CURRENT_STEP_TRANSACTION_ID
- APPROVALS_REQUIRED_CODE
- WORKFLOW_ENGINE_BATCH_ID
- EVENT_GROUP_ID
- CMD_EXECUTION_SCHD_TASK_ID

Running the Workflow Open Interface

The following sections discuss the Workflow Open Interface in more detail:

- *Loading Data into the Workflow Interface Tables*
- *Using the Run Workflow Transaction Interface Program*

Loading Data into the Workflow Interface Tables

The first step in using the Workflow Transactions Open Interface is placing information into the interface table. Instead of directly inserting records into the table, a PL/SQL procedure has been provided to hide some of the overhead in generating records in the table. For example, the PL/SQL procedure automatically initializes the Process Phase and Process Status columns.

Required Data for Workflow Transactions

Certain columns in the KWFL_TRANSACTIONS_INT tables must be populated for a Workflow Transaction. Additionally, different types of Workflow transactions require different data to first be populated. The following sections discuss any required data as well as other data population instructions:

- *Columns Inserted as Null*
- *Columns Required*
- *Parameters Required Based on Transaction Type*

Columns Inserted as Null

The following columns are used for internal processing and are always derived from other information in the record. These columns should always be inserted as null.

- WORKFLOW_ID
- WORKFLOW_INSTANCE_ID
- WORKFLOW_INSTANCE_STEP_ID
- CURRENT_STEP_TRANSACTION_ID
- APPROVALS_REQUIRED_CODE
- WORKFLOW_ENGINE_BATCH_ID
- EVENT_GROUP_ID

- CMD_EXECUTION_SCHD_TASK_ID

Columns Required

The following table lists the columns that must be populated. Refer to *Table 6-1* for a complete list of the KWKFL_TXN_INT columns.

Table 6-2. KWFL_TRANSACTIONS_INT

Column	Required for Import?	Data Type	Description
CREATED_USERNAME	Y (See Note)	Varchar2(30)	Required column. The USERNAME from the KNTA_USERS table for the user that is performing the transaction. This will be used only if the CREATED_BY is null. *Note: Either this column or the CREATED_BY column must be populated.*
CREATED_BY	Y (See Note)	Number	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from CREATED_USERNAME. *Note: Either this column or the CREATED_USERNAME column must be populated.*
LAST_UPDATED_USERNAME	Y (See Note)	Varchar2(30)	The USERNAME from the KNTA_USERS table for the user that is performing the transaction. This will be used only if the LAST_UPDATED_BY is null. *Note: Either this column or the LAST_UPDATED_BY column must be populated.*
LAST_UPDATED_BY	Y (See Note)	Number	The USER_ID from the KNTA_USERS table for the user that is performing the transaction. If null, the column is derived from LAST_UPDATED_USERNAME. *Note: Either this column or the LAST_UPDATED_USERNAME column must be populated.*
GROUP_ID	Y	Number	A unique identifier for the group of records that are being processed. A value for this column is derived from the sequence KNTA_INTERFACE_GROUPS_S.

Table 6-2. KWFL_TRANSACTIONS_INT

Column	Required for Import?	Data Type	Description
INSTANCE_SOURCE_ TYPE_CODE	Y	Varchar2(30)	Indicates whether the transaction is for a Package Line (‘CR’) or a Request (‘IR').

Parameters Required Based on Transaction Type

The Workflow Transaction Interface can be used for different types of transactions. Different parameters are required or optional depending upon the type of transaction. *Table 6-3* describes all of the parameters for the procedure INSERT_ROW in the Package KWFL_TXN_INT. The list of parameters used for each type of transaction or event is provided in *Table 6-4* through *Table 6-13*:

Table 6-3. KWFL_TXN_INT.INSERT_ROW Parameters and Descriptions

Parameter	Data Type	Description
P_EVENT	Varchar2(40)	The type of Workflow transaction. An event must always be passed to the procedure.
P_GROUP_ID	Number	An identifier for a group of records to be processed. If this is not specified, the Workflow Transaction Interface program will generate a new Group ID.
P_CREATED_USERNAME	Varchar2(80)	The USERNAME from the KNTA_USERS table for the user that is performing the transaction. A Created By Username must always be passed to the procedure.
P_SOURCE	Varchar2(100)	This is an optional, non-validated field that can be used to indicate the exact source of the Workflow Transaction (the name of the Third Party Application, the text string 'Conversion', etc.). The Source is optional.
P_REQUEST_NUMBER	Varchar2(40)	The Request Number for the Request that is being transacted.
P_PACKAGE_NUMBER	Varchar2(40)	The Package Number for the Package that is being transacted.
P_PACKAGE_LINE_SEQ	Number	The sequence number for the Package Line that is being transacted.

Table 6-3. KWFL_TXN_INT.INSERT_ROW Parameters and Descriptions

Parameter	Data Type	Description
P_WORKFLOW_STEP_NAME	Varchar2(80)	The name of the Workflow Step that is being transacted.
P_WORKFLOW_STEP_SEQ	Varchar2(30)	The sequence number of the Workflow Step that is being transacted.
P_VISIBLE_RESULT_VALUE	Varchar2 (200)	The result of the Workflow transaction. This could be a value such as ‘Approved’ or ‘Not Approved’.
P_USER_COMMENTS	Varchar2 (200)	Comments for the transaction. Any comments are appended to the notes for the Package or Request. The User Comments are optional.
P_DELEGATED_TO_USERNAME	Varchar2(30)	The username for the user to whom the Decision step is being delegated.
P_SCHEDULE_DATE	Date	For Execution steps, the date for which the execution has been scheduled to run.
P_TO_WORKFLOW_STEP_NAME	Varchar2(80)	The name of the Workflow Step that is being transitioned to.
P_TO_WORKFLOW_STEP_SEQ	Varchar2(30)	The sequence number of the Workflow Step that is being transitioned to.
O_MESSAGE_TYPE	Number	An output parameter that indicates what type of error occurred. This parameter can have one of the following values, constants for these values are defined in the Package KNTA_Constant: • SUCCESS - No error occurred. • USER_ERR - User error • INTERNAL_ERR - An internal error occurred. • WARNING - A non-fatal warning is returned.
O_MESSAGE_NAME	Varchar2(80)	The internal message name of the error that was returned. This is used mainly for debugging purposes.
O_MESSAGE	Varchar2 (1000)	The error message.

The following tables describe the parameters used for specific events:

- Table 6-4, “Parameters Used For All Events,” on page 123

- Table 6-5, “Parameters for Package or Request Submission,” on page 123
- Table 6-6, “Parameters for Decision Step Results,” on page 124
- Table 6-7, “Parameters for Decision Step Delegation,” on page 124
- Table 6-8, “Parameters for Execution Step,” on page 124
- Table 6-9, “Parameters for Execution Step Schedule,” on page 125
- Table 6-10, “Parameters for Execution Step Bypass,” on page 125
- Table 6-11, “Parameters for Changing Step Result,” on page 125
- Table 6-12, “Parameters for Forced Workflow Step Transition,” on page 126
- Table 6-13, “Parameters for Package Line or Request Cancellation,” on page 126

Table 6-4. Parameters Used For All Events

Parameter	Input/Output	Required?
P_EVENT	Input	Yes
P_GROUP_ID	Input	No
P_CREATED_USERNAME	Input	Yes
P_SOURCE	Input	No
P_REQUEST_NUMBER	Input	No (but either the Request Number or Package Number is required)
P_PACKAGE_NUMBER	Input	No (but either the Package Number or Request Number is required)
P_USER_COMMENTS	Input	No
O_MESSAGE_TYPE	Output	N/A
O_MESSAGE_NAME	Output	N/A
O_MESSAGE	Output	N/A

Table 6-5. Parameters for Package or Request Submission

Parameter	Input/Output	Required?
P_EVENT = 'INSTANCE_SET_CREATE'	Input	Use this event to submit a Package or a Request - no further parameters are necessary

Table 6-6. Parameters for Decision Step Results

Parameter	Input/Output	Required?
P_EVENT = 'APPROVAL_VOTE'	Input	For Decision steps, use this event to provide a result for a decision, such as 'Approved' or 'Not Approved'
P_PACKAGE_LINE_SEQ	Input	Yes (if the transaction is for a Package Line)
P_WORKFLOW_STEP_NAME	Input	No (but either the Step Name or Step Sequence is required)
P_WORKFLOW_STEP_SEQ	Input	No (but either the Step Sequence or Step Name is required)
P_VISIBLE_RESULT_VALUE	Input	Yes

Table 6-7. Parameters for Decision Step Delegation

Parameter	Input/Output	Required?
P_EVENT = 'APPROVAL_DELEGATE'	Input	For Decision steps, use this event to delegate a decision to another user
P_PACKAGE_LINE_SEQ	Input	Yes (if the transaction is for a Package Line) - not required for Projects
P_WORKFLOW_STEP_NAME	Input	No (but either the Step Name or Step Sequence is required)
P_WORKFLOW_STEP_SEQ	Input	No (but either the Step Sequence or Step Name is required) - not required for Projects
P_DELEGATED_TO_USERNAME	Input	Yes

Table 6-8. Parameters for Execution Step

Parameter	Input/Output	Required?
P_EVENT = 'EXECUTION_EXECUTE'	Input	For Execution steps, use this event to perform an execution at a Workflow Step
P_PACKAGE_LINE_SEQ	Input	Yes (if the transaction is for a Package Line)
P_WORKFLOW_STEP_NAME	Input	No (but either the Step Name or Step Sequence is required)
P_WORKFLOW_STEP_SEQ	Input	No (but either the Step Sequence or Step Name is required) - not required for Projects

Table 6-9. Parameters for Execution Step Schedule

Parameter	Input/Output	Required?
P_EVENT = 'EXECUTION_SCHEDULE'	Input	For Execution steps, use this event to schedule an execution for a future date
P_PACKAGE_LINE_SEQ	Input	Yes (if the transaction is for a Package Line)
P_WORKFLOW_STEP_NAME	Input	No (but either the Step Name or Step Sequence is required)
P_WORKFLOW_STEP_SEQ	Input	No (but either the Step Sequence or Step Name is required) - not required for Projects
P_SCHEDULE_DATE	Input	Yes

Table 6-10. Parameters for Execution Step Bypass

Parameter	Input/Output	Required?
P_EVENT = 'BYPASS_EXECUTION'	Input	For Execution steps, use this event to bypass an eligible execution
P_PACKAGE_LINE_SEQ	Input	Yes (if the transaction is for a Package Line)
P_WORKFLOW_STEP_NAME	Input	No (but either the Step Name or Step Sequence is required)
P_WORKFLOW_STEP_SEQ	Input	No (but either the Step Sequence or Step Name is required)
P_VISIBLE_RESULT_VALUE	Input	Yes

Table 6-11. Parameters for Changing Step Result

Parameter	Input/Output	Required?
P_EVENT = 'RESULT_OVERRIDE'	Input	For Decision, Execution, and Subworkflow steps, use this event to change the result of an active step that has not transitioned to another step
P_PACKAGE_LINE_SEQ	Input	Yes (if the transaction is for a Package Line)
P_WORKFLOW_STEP_NAME	Input	No (but either the Step Name or Step Sequence is required)
P_WORKFLOW_STEP_SEQ	Input	No (but either the Step Sequence or Step Name is required)
P_VISIBLE_RESULT_VALUE	Input	Yes

Table 6-12. Parameters for Forced Workflow Step Transition

Parameter	Input/Output	Required?
P_EVENT = 'FORCE_TRANSITION'	Input	Use this event to transition from a Workflow Step to any other step in the Workflow, regardless of whether a transition is defined; force transition only works for transitions within the same level of the Workflow (i.e. does not work between a Subworkflow and its parent Workflow)
P_PACKAGE_LINE_SEQ	Input	Yes (if the transaction is for a Package Line)
P_WORKFLOW_STEP_NAME	Input	No (but either the Step Name or Step Sequence is required)
P_WORKFLOW_STEP_SEQ	Input	No (but either the Step Sequence or Step Name is required)
P_VISIBLE_RESULT_VALUE	Input	Yes (but for this event, the Result Value is not validated against the Workflow Step)
P_TO_WORKFLOW_STEP_NAME	Input	No (but either the To Step Name or To Step Sequence is required)
P_TO_WORKFLOW_STEP_SEQ	Input	No (but either the To Step Sequence or To Step Name is required)

Table 6-13. Parameters for Package Line or Request Cancellation

Parameter	Input/Output	Required?
P_EVENT = 'INSTANCE_SET_CANCEL'	Input	Use this event to cancel a Package Line or a Request
P_PACKAGE_LINE_SEQ	Input	Yes (if the transaction is for a Package Line)

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

Example

The following 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
  clwf_txn_int.insert_row
    (p_event => 'APPROVAL_VOTE',
     p_group_id => NULL,
     p_created_username => '&p_created_username',
     p_source => NULL,
     p_request_number => '&p_request_number',
     p_package_number => NULL,
     p_package_line_seq => NULL,
     p_workflow_step_name => NULL,
     p_workflow_step_seq => '&p_workflow_step_seq',
     p_visible_result_value => '&p_visible_result_value',
     p_user_comments => NULL,
     p_delegated_to_username => NULL,
     p_schedule_date => NULL,
     p_to_workflow_step_name => NULL,
     p_to_workflow_step_seq => NULL,
     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 can be used at the operating system to run the code:

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

Using the Run Workflow Transaction Interface Program

To process the data in the interface tables, use the ‘Run Workflow Transaction Interface’ program. This program is defined as a standard Report Type and is launched through the Change Management Reports window or the Request Management Reports window.

To generate an executable version of the program using the Report Types workbench:

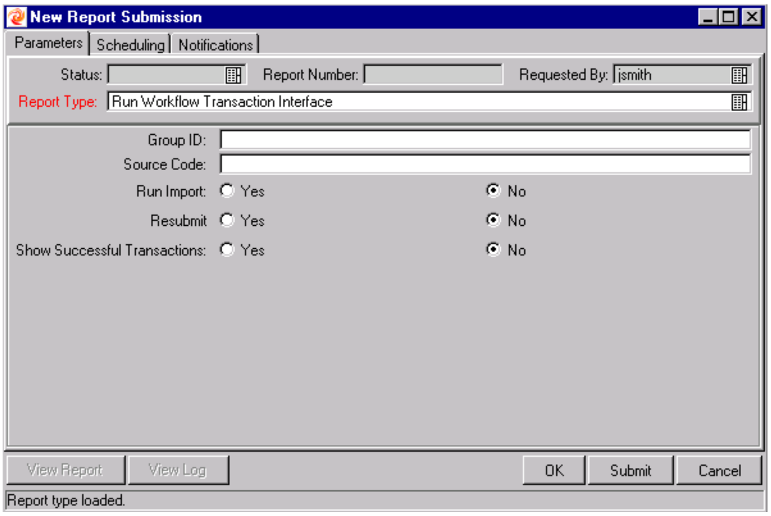
1. Click the **Configuration** shortcut group and click the **Report Types** icon. The Report Types workbench opens.

2. Search for the ‘Run Workflow Transaction Interface’ program.

3. Click **Copy** to copy the reference Report Type.

4. Rename the copy ‘Run Workflow Transaction Interface.’

5. Click **Yes** to the Would you like to edit prompt to open the Report Type window. Select **Yes** for the Enabled field to make this Report Type available in Mercury ITG Center.



Status:		Report Number:		Requested By:	jsmith
Report Type:	Run Workflow Transaction Interface				

Group ID:

Source Code:

Run Import: ○ Yes ● No

Resubmit ○ Yes ● No

Show Successful Transactions: ○ Yes ● No

Report type loaded.

Figure 6-2 The Run Workflow Transaction Interface Report

When executed, the “Run Workflow Transaction Interface” program:

- 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 graphical user interface. For example, the CREATED_BY_USERNAME must exist in Mercury ITG 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 Interface program. If desired, successful transactions can be eliminated from the report, so that only errors are displayed.

Run Workflow Transaction Interface Program Parameters

The Run Workflow Transaction Interface program has several parameters for controlling the behavior of the program execution.

Table 6-14. Run Workflow Transaction Interface Parameters

Parameter	Field Type	Description
Group ID	Text Field	Runs the interface for a specific Group ID. The interface program will only look for records with this value in the GROUP_ID column. This is useful when processing a specific batch of transactions. If this parameter is left blank, the program will only pick up records with a blank GROUP_ID value.

Table 6-14. Run Workflow Transaction Interface Parameters

Parameter	Field Type	Description
Source Code	Text Field	Runs the interface for records from a specific source. The interface program will only look for records with this value in the SOURCE_CODE column. This is useful for running the open interface at different frequencies depending on the driver of the Workflow transactions. For example, transactions initiated from the Project Management system could be imported once a day while transactions from the performance monitor could be processed within an hour.
Run Import	Radio Button	If set to **Yes**, the program will process the records in the interface table and try to import them. If set to **No**, the program will simply report on the records in the interface table. This option is useful when auditing prior executions of the Open Interface.
Resubmit	Radio Button	If set to **Yes**, 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	Radio Button	Shows Workflow Transactions that were successfully processed.

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 ITG Center data.

Example

The source data might use a result value that does not exist in Mercury ITG 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.

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 ITG 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 ITG Center source and Mercury ITG 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 ITG Center.

Index