Mercury™ IT Governance Center

Configuring a Request Resolution System (Demand Management)

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: ConfigureRequests-0304A

Table of Contents

Appendix A
Advanced Workflow Topics....251

Appendix B
Validations....283

Appendix C

Appendix D

Chapter 1 Introduction

Demand Management allows an organization to model its processes for managing technology initiatives from inception to implementation using a graphical workflow business modeler. Complex business rules can be modeled using approval methods and prioritization features that allow issues to efficiently advance through their specific workflow, routing them to relevant departments, groups or individuals. Demand Management is designed to capture data by prompting users for information specific to their Request, ensuring that required information is collected and validated at the appropriate time in the process.

About This Document

This guide provides instructions for configuring a Request resolution system using Mercury Demand Management. 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. Each chapter covers a particular topic:

Key Concepts
Defines the key concepts and definitions used when creating a Request resolution system.

Configuring a Request Resolution System - Process Overview
Provides an overview of the process used to configure a Request resolution system. It summarizes each phase of configuration.

Gathering Process Requirements and Specifications
Discusses the information that you need to collect before configuring your Request resolution system.

Mapping your Process into a Workflow Provides instructions for setting up a Workflow skeleton, including all Workflow steps, transitions and Validations included in your process.

Constructing the Request Type Provides instructions on configuring the Request Types that will be used to process Requests through your Workflow. This includes configuring Request Type fields and logic.

Integrating Participants into Your Request Resolution System Provides instructions for integrating users into your deployment process.

Setting Up Communication Paths Provides an overview for different modes of communication that you can use in your deployment system. This includes configuring Email Notifications, the Dashboard, and reports.

Rolling Out a Request Resolution System Discusses a number of topics to consider when rolling out your Request resolution process.

Advanced Workflow Topics Provides instructions for creating and using Validations in your system.

Tokens Summarizes how Tokens are used in Demand Management and provides a link to additional details.

User Data Creation and Processing Provides instructions on creating and using User Data fields in your process.

Configuration Worksheets Provides worksheets that can be printed out and used to capture data required for configuring a Request resolution system.

Intended Audience

The intended audience for this document include:

- Business or technical users who configure and maintain a Request resolution system using Demand Management
- Users responsible for Workflow configuration

- Managers responsible for reporting

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/conf/resin.conf`
Link	Linked URLs, filenames, and cross references are shown as blue italicized text.	*www.mercury.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 configure Mercury Demand Management:

- *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 Dashboard	This document provides details for defining and configuring the Dashboard and custom Portlets.
Processing Requests (Demand Management)	This document explains how to process Requests using Demand Management.
Processing Packages (Change Management)	This document explains how to process Packages using Change Management.
Managing Your Projects (Project Management)	This document explains how to work with Projects using Project Management.
Using the Workbench	This document explains how to navigate through the Workbench.
Migrators Guide and Reference	This document provides details for configuring and using the Migrator Object Types to migrate or archive data from Mercury ITG instances.
Commands and Tokens Guide and Reference	This document provides information on using Commands and Tokens.
Security Model Guide and Reference	This document presents an overview of the data security model and provides instructions for controlling access to different Mercury ITG entities.
Configuring a Deployment System (Change Management)	This document provides instructions for configuring a deployment system using Mercury Change Management. This includes requirements gathering, modeling your processes in a Workflow, defining commands used by Mercury ITG execution engine, and rolling out this system to your users.

Configuring the Dashboard	This document provides instructions for configuring custom Portlets, maintaining standard and custom Portlets, and setting a Default Dashboard for all users.
Reports Guide and Reference	This document provides details for running Reports.
Open Interface Guide and Reference	This document provides details for integrating third party products with Mercury ITG entities.
Customizing the Standard Interface	This document provides details for customizing the standard interface, including the directory structure and methods of customization.
Configuring Demand Tracking and Management	This document provides instructions for configuring the product to enable the successful management of demand in your company.
Managing Demand (Demand Management)	This document provides instructions for using Demand Management to enable the successful management of demand in your company.

Customer Support

Customer support and downloads for 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 Mercury IT Governance Center. For more information, visit the Education Services Web site at *http://www.merc-training.com/main/ITG*.

Chapter 2 Key Concepts

This chapter defines the key concepts and definitions related to configuring a Request resolution system using Mercury Demand Management.

This chapter covers the following topics:

- *Request Resolution*
- *Demand*
- *Request*
- *Request Type*
- *Request Header Type*
- *Field Logic*
- *Workflow*
- *Request/Workflow Interaction*
- *Request Type Commands*
- *Validation*
- *Token*
- *Security Groups*
- *Integration with Other IT Governance Products*
- *Reports*
- *Custom Help on Requests and Request Fields*

Request Resolution

Request resolution refers to the creation, processing and closing of Requests on a business unit. A Request can be anything from a simple question to a detailed report of a software bug.

Use Mercury Demand Management to model and enforce best practice Request resolution processes. Each type of Request leverages an optimized Workflow tailored for specific business rules to collect required data, gain appropriate approvals and perform specific actions.

As a Request progresses along its Workflow, pre-configured steps can trigger:

- Email notifications to be sent to the proper participants.
- Automated command-line executions to be performed.
- Field defaulting and logical updates, ensuring that the correct information to resolve a Request is available.
- Deployments to be created and initiated or project tasks to be updated.

When the Request has been taken to the end of its Workflow, it is considered resolved.

Demand

Demand is an item of work requested from a business unit. Mercury Demand Management provides a single application and repository to capture all demand placed on IT. Demand Management consolidates information from the many different sources so you can both view aggregate demand in real time and report against it. In addition, Demand Management streamlines the end-to-end process (from demand through deployment) of fulfilling demand.

In Mercury Demand Management, each item of demand is captured and processed on the Request level. Demand is captured using the Request resolution features of Demand Management. It can then be analyzed, consolidated, and processed (on a macro-level) using special demand-processing features in Demand Management. For additional details, see *Managing Demand (Demand Management)* and *Configuring Demand Tracking and Management*.

Request

A Request is the fundamental work unit of the Request resolution piece of Mercury Demand Management. End-users create Requests and then submit them along a resolution process (defined in the Workflow). The Request page contains all information typically required to complete a specific business process. Requests with similar or related functions can be grouped into Request Categories, making them easier to locate and use.

Each Request has an associated Request Type that determines which fields are included in the Request page. As the Request goes through its steps, the user will be prompted for any information necessary to bring the Request to closure. Once the basic Request information has been entered, the corresponding Workflow is automatically selected based on the Request Type.

Definition

A Request:

- Is the fundamental “work unit” within Mercury Demand Management.
- Is the repository for all of the information necessary to take a series of actions and move through a standard business process.
- Is a specific execution of a business process. Each Request is identified by a unique Request Number.

Request Type

A Request Type is a general category that defines the structure of the Request. Mercury Demand Management includes such pre-defined system Request Types as the Bug Request Type and Enhancement Request Type. The fields that are used when a Request is created are customizable based on the Request Type. Request Type definitions control much of the Request-specific logic in the resolution process. This includes such things as:

- Defaulting a specific Workflow to use when processing this type of Request
- Custom field definition and behavior
- Layout

- Data access and security (who can view or edit the Request)
- Configuration security (who can alter the Request Type)
- Notifications

Definition

A Request Type:

- Is the framework that defines the behavior of a Request as it moves through a business process.
- Determines the logic behind the storage and manipulation of data within a Request.
- Represents a process within a business. The Request Type can be defined to capture different kinds of data, as well as follow different business and resolution processes.

Request Header Type

Request Header Types define the collection of fields that appear in the header region of the Requests. Request Header Types typically include more general information that will be tracked between multiple types of Request. This can include such information as who logged the Request, its priority, and a description of the issue.

Request Header Types contain a set of standard pre-defined fields that can be enabled or disabled. Request Header Types can also contain custom fields.

Each Request Type must include a Request Header Type. A single Request Header Type can be used for multiple Request Types.

Field Logic

As a Request proceeds through a Workflow, certain business logic can be applied to the Request fields. For example, if the Request Priority is set to **Critical**, then another field (Assigned To) can be automatically updated to the resource responsible for dealing with critical issues.

Field behavior can be configured in the following ways:

- A field value can be updated
- A field can be automatically populated with a specific value
- A field can become invisible or non-editable
- A field can be set to required or need to be reconfirmed

This field behavior can be triggered by a number of events that occur when processing the Request. Field behavior can be triggered when:

- Another field in the Request changes
- The Request status changes (based on its Workflow)
- Request Type commands are executed

The rules that govern automated Request field changes can be configured at the Request Type level or the field level within the Request Type or Request Header Type. For additional details on configuring field logic, see *"Constructing the Request Type"* on page 111.

Workflow

A Workflow is a logical series of steps that define the process that Requests follow. The Workflow can be configured to handle virtually any business practice. This allows a department to create Workflows to automate existing processes, rather than forcing users to adopt a fixed set of processes to perform their work.

Workflow steps can range in usage from functional approvals to actual system-level executions. For example, it is possible to create an execution step to automatically connect to another application and import data into the Mercury Demand Management database.

A sample Workflow for an Application Enhancement is shown below:

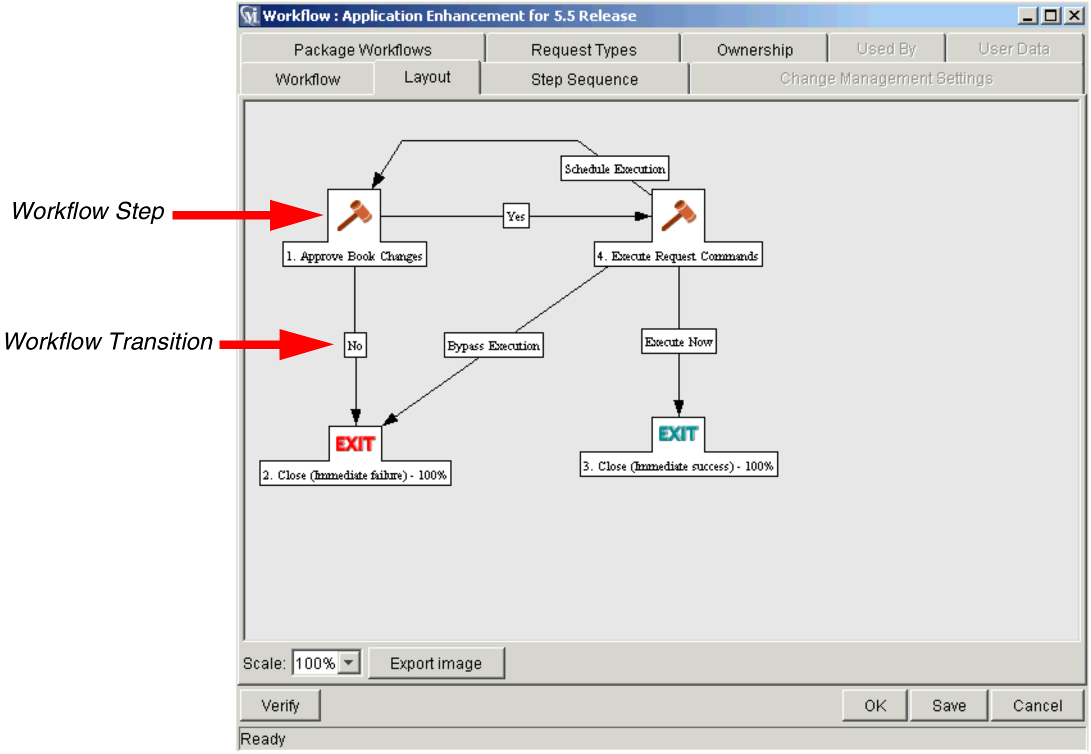


Figure 2-1 Sample Workflow

Request/Workflow Interaction

Request Types are tightly integrated with the Workflow engine. Each Request Type has a number of possible Statuses (such as Assigned, On Hold, or New). Each Status can be linked to a particular Workflow Step and drive field-level behavior. Additionally, since Request Statuses can be linked to field behavior through Status Dependencies, field properties (such as whether the field can be edited or is required) can also be altered as the Request precedes along a Workflow.

It is also possible to configure the Workflow to execute the commands contained in the Request Type at specific points in the process (Workflow step). The Request Type commands are executed at Execution Workflow steps.

Example

A Request reaches a Workflow Step that updates the Request Status to **Assigned**. At this point, the Assigned User field in the Request becomes required. This field must be filled in before the Request can continue processing.

Request Type Commands

Commands are instructions interpreted by the Execution Engine and translated into operating system commands to be dynamically executed. Commands are typically a blend between shell scripts and Mercury Demand Management system Special Commands. Commands allow the automation of an entire sequence of commands that would previously have been run manually. For example, these command sequences can automate extracting information from another enterprise application or running a report.

Request Type commands define the execution layer within Mercury Demand Management. While most of the resolution process for a Request is analytically based, cases may arise for specific Request Types where system changes are required. In these cases, Request Type commands can be used to automatically perform these changes.

Special Commands

In order to simplify programming commands, Demand Management provides a predefined set of Special Commands. These commands perform a variety of common functions, such as establishing connections to other machines for remote command execution. There are two types of Special Commands:

- System Special Commands - These commands are shipped with the product. System Special Commands are read-only and have the naming convention `ksc_command_name`. System Special Commands always begin with `ksc_`.
- User Defined Special Commands - These commands are user-defined and have the naming convention `sc_command_name`. User-defined Special Commands must begin with `sc_`. User defined Special Commands can contain one or more of the system Special Commands.

For more detailed information on Special Commands, see *Commands and Tokens Guide and Reference*.

Validation

Validations determine the acceptable input values for user-defined custom fields. Validations maintain data integrity by ensuring that the correct information is entered in a field before it is saved to the database. For example, Validations can be used to ensure that no textual information is entered into a numeric field, or that dates are entered into a field in the proper format. More complex Validations can be used to verify that only appropriate users are assigned to a Request. The values in selection Validations (drop down lists and auto-complete lists) can be configured by either listing the values or performing a SQL query.

Validations are used in the following locations:

- Every custom field generated for an Object Type, Report Type, Request Type, or User Data has a Validation.
- Every decision and execution Workflow Step has a Validation.
- Every drop down list and auto-complete list in all Demand Management windows are based upon a Validation. However, it is not always possible to change the Validations associated with predefined fields.

Token

While configuring certain features in Demand Management, it is often necessary to reference information in variables that is undefined until the product is actually used a particular context. Instead of generating objects that are valid only in those specific contexts, these variables can be used to facilitate the creation of general objects that can be applied to a variety of contexts. These variables are called **Tokens**.

There are two types of Tokens used within Demand Management: standard Tokens and custom Tokens. Standard Tokens are provided with the product. Custom Tokens are generated to represent a specific entity configurations (such as Object Type fields, Request Type fields, or Workflow parameters). Each field of the following Mercury IT Governance entities can be referenced as a Token:

- Object Types
- Request Types
- Report Types

- Project Templates
- User Data
- Workflow Parameters

Tokens can be used in the following entity windows:

- Object Type commands
- Request Type commands
- Validation commands and SQL statements
- Report Type commands
- Executions and Notifications for a Workflow
- Workflow Step commands
- Notifications in a Report Submission
- Special Command commands
- Notifications for Tasks
- Field security
- Notifications for Request Types

Security Groups

Security Groups are constructed to provide a set of users with access to specific product screens and functions. Each Security Group is configured with a set of Access Grants that enable specific access. Users are then associated with one or more Security Groups.

A user's Security Group memberships determine which windows he can view or edit, which Workflows he can use, and which Workflow Steps he has authority to act on. Each user can be a member of multiple Security Groups. The collection of Security Groups to which a user belongs defines that user's role and access within Demand Management.

Tip

Since users can be members of as many Security Groups as necessary, it is recommended that specific Security Groups are generated, each with a smaller range of responsibilities. Users can then be added to many different Security Groups to grant them their full range of access.

Security Groups control product access on the following levels:

- **Screen Security:**
  Each Security Group contains a list of Access Grants that determine a user's screen security. Access Grants are used to grant access to edit, view, manage or submit items within a specific screen. By controlling the set of Access Grants for each user, specific functional roles for the user community can be defined.
- **Workflow Step Security:**
  Each Workflow Step can be linked to a unique set of Security Groups. By adding or removing specific Security Groups from a Workflow Step in the Workflow window, it is possible to control which users can act on that step. This security level provides an extremely detailed level of control over each user's actions.
- **Request Level Security:**
  Each Request Type can be configured to control which users can create, view, edit, cancel or delete those types of Requests. It is possible to specify which Security Groups or individual users can participate with these Requests. This is set in the **User Access** tab on the Request Type window.
- **Request Field Security:**
  Security can also be controlled at the Request field level. It is possible to specify which Security Groups can view or edit specific fields on the Request. This is set in the Field window used to define the field.

Note

Only users with the Administrator licence can create, modify or delete Security Groups. This license is typically only given to a few individuals per company. This provides a centralized control over licenses and user roles within Mercury IT Governance Center.

When configuring a Request resolution process, work with the Mercury IT Governance Administrator to define the Users and Security Groups needed for your processes.

For details on configuring security and user access around your deployment system, see *Security Model Guide and Reference.*

Integration with Other IT Governance Products

This document focuses on configuring the Request resolution functions within Demand Management. With additional Mercury IT Governance products and licenses, you can address the full set of challenges related to governing your IT department.

This section provides an overview of a few key integration points within Mercury IT Governance Center. For additional information on the complete Mercury IT Governance solution, see the Products section of the Mercury Interactive Web site at *http://www.mercuryinteractive.com/products/*.

This section covers the following topics:

- *Mercury IT Governance Dashboard*
- *Mercury Project Management*
- *Mercury Change Management*
- *Mercury Change Management Extensions*

Mercury IT Governance Dashboard

Intended for large and complex environments, Mercury IT Governance Dashboard provides 360° visibility and control over technology-based initiatives and IT operational tasks. Configurable, role-based visual displays called Portlets provide relevant summary information and highlight exception conditions in initiatives. Users can then drill down to any desired level of detail.

For example, a CIO may want to see the status of the major initiatives undertaken by the IT department. Instead of relying on weekly reports patched together from different sources and often compiled from out-of-date or incomplete information, the CIO can go directly to Dashboard. The Dashboard displays the true status of the initiatives—based on current data captured automatically as part of actually performing the work. The Dashboard clearly identifies any initiative that is behind schedule, or in any other exception state, and displays the causes for the delay.

The Dashboard is beneficial to all participants throughout the Technology Chain. For example, developers can use the Dashboard to view their own action items, and end-users can consult their own Dashboards to see the status of all the Requests they have submitted.

Mercury Project Management

Mercury Project Management adds a critical dimension, automated execution, to complex project management in large IT organizations. Unlike static project management tools that simply schedule the tasks, dates, and resources, Mercury Project Management's automation proactively pushes project tasks to the assigned resource, links with Mercury Demand Management and Mercury Change Management to automatically perform issue resolution and deployment tasks, and automatically updates and reports project status as task are completed. Project managers guide projects from concept to completion from Mercury Project Management centralized environment.

Requests (used in the Request resolution process) can be added to the Mercury Project Management project plan. Dependencies can be set between Requests and Tasks on the project. This ensures that the technical aspects of the deployment process is respected by other resources on the project plan.

Mercury Change Management

Mercury Change Management allows companies to automate and manage the deployment of packaged applications, custom applications, legacy systems, and Web content. Change Management enforces deployment processes and performs all tasks required to install software changes correctly across the Development, Test, Stage, and Production system landscape.

Developers can concentrate on developing and adapting enterprise applications rather than non–value–added tasks such as code migrations. Packages group software changes they can be deployed completely, thereby eliminating errors. In the event of a failure, a complete audit trail helps developers pinpoint the cause of the problem and rollback changes if necessary.

Requests (in Mercury Demand Management) can be functionally linked to Packages (in Mercury Change Management). This allows the utilization of Change Management deployment features within the Request resolution system. For example, Requests can be configured to automatically create a Package to deploy software and application changes related to the Request. When the Package is finished deploying the changes, it can communicate with and update information on the original Request.

Mercury Change Management Extensions

Mercury Change Management Extensions simplify the complex activities required to maintain large enterprise applications like Oracle, PeopleSoft, SAP, Siebel, and Web applications built using Java and Oracle. These applications are constantly changing as new modules are added, customizations developed, configurations modified, and patches applied. The changes must be done precisely across the Development, Test, Stage and Production system landscape, usually by highly paid and hard-to-find specialists. Mercury Change Management Extensions automate these precise tasks using best practice processes designed specifically for each application.

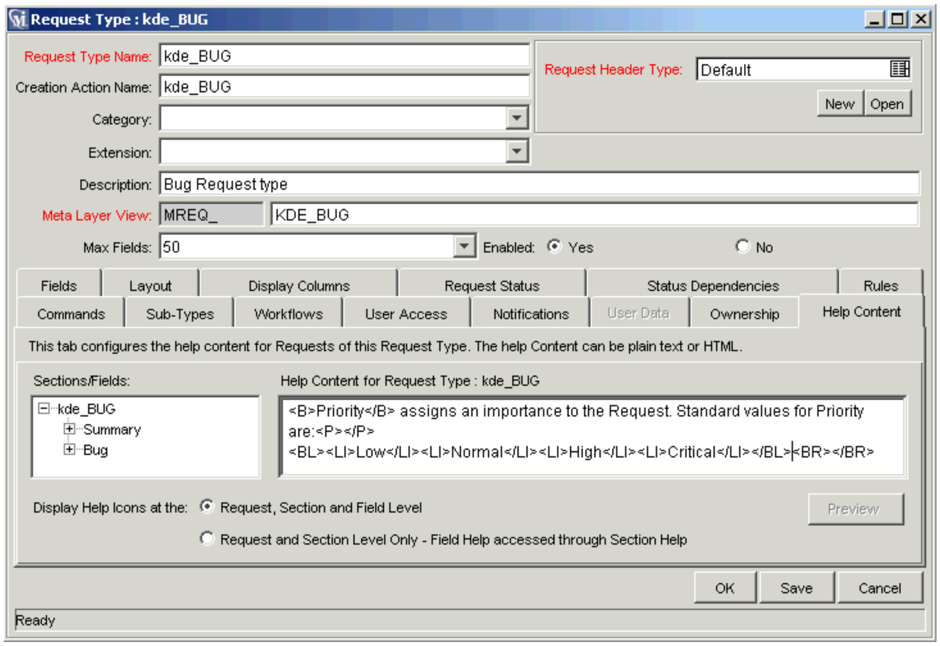
Reports

Change Management reports output text that provides information on specific entities or configurations. For a complete list of the reports used in commonly used in Request resolution systems, see *Reports Guide and Reference*.

Custom Help on Requests and Request Fields

It is possible to define custom Help content for each Request in the system. This provides the end users with accessible information related to a Request, a section on the Request, or a specific field on the Request. For example, one of your Request's includes a Priority field to specify the urgency of a particular Request. The values in the Priority field are Critical, High, Medium, and Low. You can configure Help content for that field to instruct the user when to mark a Request as Critical rather than High.

Custom Help is configured in the Help Content tab in the Request Type window. This window is shown in *Figure 2-2*.



Request Type : kde_BUG

Request Type Name: kde_BUG

Creation Action Name: kde_BUG

Category:

Extension:

Request Header Type: Default

Description: Bug Request type

Meta Layer View: MREQ_ KDE_BUG

Max Fields: 50

Enabled: ● Yes ○ No

This tab configures the help content for Requests of this Request Type. The help Content can be plain text or HTML.

Sections/Fields:

- kde_BUG
  - Summary
  - Bug

Help Content for Request Type : kde_BUG

<B>Priority</B> assigns an importance to the Request. Standard values for Priority are:<P></P>
<BL><LI>Low</LI><LI>Normal</LI><LI>High</LI><LI>Critical</LI></BL><BR></BR>

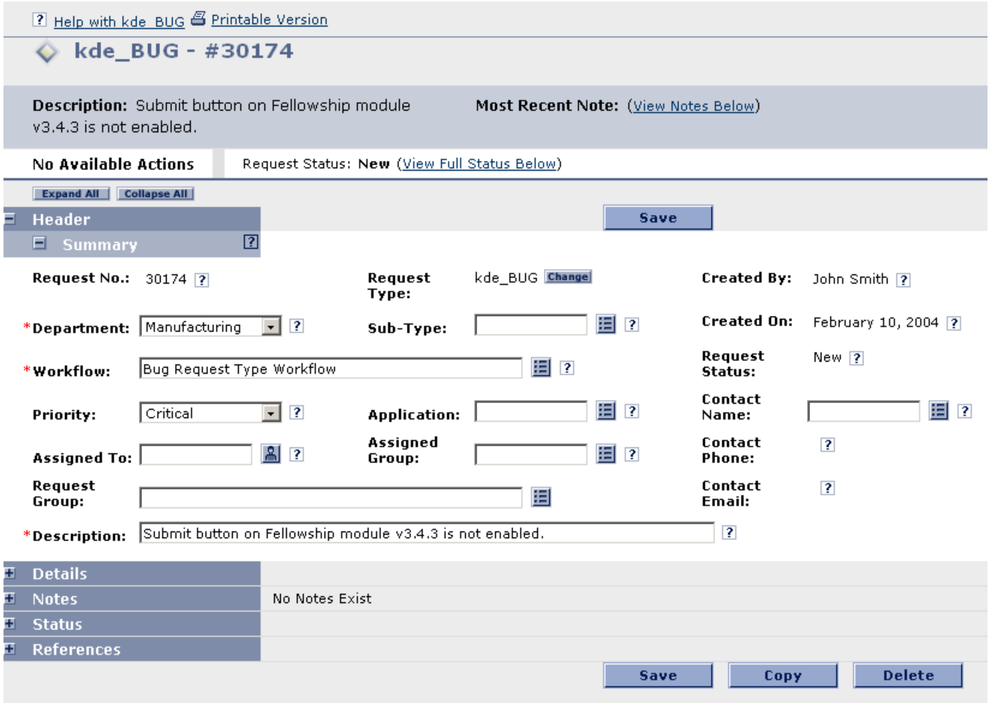
Display Help Icons at the: ● Request, Section and Field Level

○ Request and Section Level Only - Field Help accessed through Section Help

Ready

Figure 2-2 Request Type Window -- Help Content Tab

When Help is configured for a specific Request, section, or field, an icon (?) is displayed. The user can click on the icon to open a page that contains details on that field. *Figure 2-3* shows a Request with Help defined for a section and specific fields.



kde_BUG - #30174

Description: Submit button on Fellowship module v3.4.3 is not enabled.

Most Recent Note: (View Notes Below)

No Available Actions

Request Status: **New** (View Full Status Below)

Request No.:	30174	**Request Type:**	kde_BUG	**Created By:**	John Smith
***Department:**	Manufacturing	**Sub-Type:**		**Created On:**	February 10, 2004
***Workflow:**	Bug Request Type Workflow			**Request Status:**	New
Priority:	Critical	**Application:**		**Contact Name:**	
Assigned To:		**Assigned Group:**		**Contact Phone:**	
Request Group:				**Contact Email:**	
***Description:**	Submit button on Fellowship module v3.4.3 is not enabled.				

No Notes Exist

Figure 2-3 Request with Help on Sections and Fields

Chapter 3

Developing Your Configurations (Using Migrators)

This chapter introduces the concept of using multiple instances, such as Development, Testing, and Production, when configuring the Mercury IT Governance Center. It then provides and overview on how to use Mercury IT Governance Migrators to manage the deployment of configurations between these instances. Finally, this chapter discusses how to use the Migrators to archive configuration data.

Note

A Mercury Change Management Power license is required to use the Migrators.

This chapter represents a change management implementation recommendation. Using the concepts and procedures listed in this chapter can reduce the risk of down time when rolling-out Mercury ITG processes.

For detailed instructions on using the Migrators, see the *"Migrators Guide and Reference"*.

This chapter discusses the following topics:

- *Using Multiple Instances - Introduction*
- *Implementing Multiple Instances - Overview*
- *Migrating Configuration Data*
- *Archiving Configuration Data*

Using Multiple Instances - Introduction

Mercury Change Management controls the deployment of objects to mission critical applications. Before rolling-out new or modified functionality in a deployment system, thoroughly test the changes in a Development or Testing instance. For example, before rolling out a new Web Update process to manage deployments to your company's web site, test the Workflow and Object Types used to perform the deployments.

Migrators are used to capture and move configuration data (such as Workflow or Object Type definitions). This enables the sharing of configuration data between multiple Mercury ITG instances. It is then possible to test configurations in a TEST instance, and then migrate the configurations to the PRODUCTION instance.

Note

This chapter represents a change management implementation recommendation. Using the concepts and procedures listed in this chapter can reduce the risk of down time when rolling-out processes.

For additional details, see the following documents:

- *"System Administration Guide"* for instructions on setting up multiple Mercury ITG instances.
- *"Migrators Guide and Reference"* for detailed instructions on using Migrators to move configuration data.
- *"Installation Guide"* for instructions on installing new instances.

Implementing Multiple Instances - Overview

It is recommended multiple instances be used when configuring the Mercury IT Governance Center. The following sections discuss the simplest multi-instance configuration, consisting of two instances: DEV (development) and PROD (production) located on different machines. These basic migration principles can then be extended to support the number of instances used at your site.

There are two implementation scenarios for using multiple instances. The process for implementing these instances differs depending on the following scenarios:

- *Single Production Instance is Currently in Use*
  Requires you to clone the PRODUCTION instance (file system and database) to create the DEV instance.
- *New Implementation*
  Requires you to create multiple Mercury IT Governance instances by running the installation multiple times.

Single Production Instance is Currently in Use

To implement multiple instances when a single Production (PROD) instance is currently in use, clone the PROD instance. Each Mercury IT Governance instance consists of a file system and an Oracle database. These can exist on Unix or Windows machines. Contact the System Administrator for details about your site's configuration.

To move from a single active instance to multiple instances:

1. Clone the PROD instance.

   This includes the file system, database, and license information. Details for this procedure are included in the System Administration Guide. Work with the System Administrator to implement this configuration.

2. Configure any changes to Mercury product in the DEV instance.

   This includes creating or modifying Workflows, Object Types, Request Types, Validations, Security Groups, Environments, etc.

3. Configure a Package Workflow to migrate the configuration data from DEV to PROD.

   This process should be configured in the PROD instance.

4. Migrate data from the DEV instance into the PROD instance.

   Again, this activity is performed from the PROD instance. Therefore, it may be helpful to think of migrating the data as an "import" process.

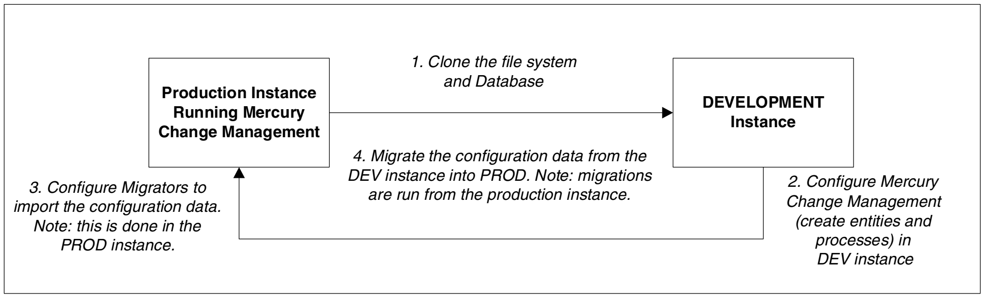


Figure 3-1 Cloning instance and configuring Migrators

New Implementation

When implementing the Mercury IT Governance for the first time, it is possible to immediately set up multiple instances. Configure one instance as the DEV instance, and the other as the PROD instance. By creating two blank instances up front, it is unnecessary to clone existing data from one instance into another. When this strategy is used, follow the instructions included in the *"Migrators Guide and Reference"* document.

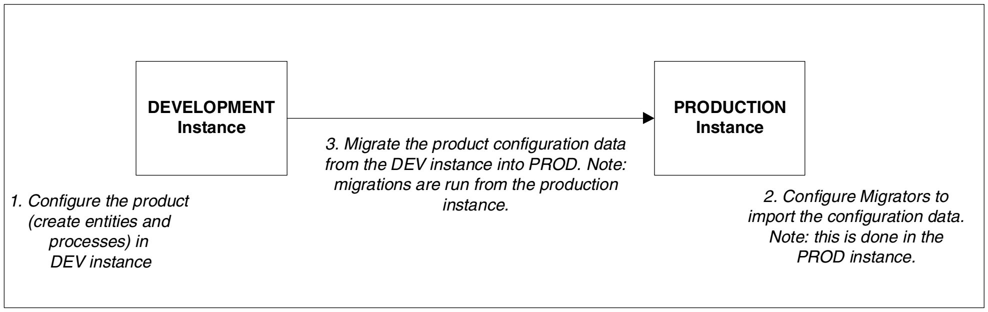

DEVELOPMENT
Instance
PRODUCTION
Instance
3. Migrate the product configuration data
from the DEV instance into PROD. Note:
migrations are run from the production
instance.
1. Configure the product
(create entities and
processes) in
DEV instance
2. Configure Migrators to
import the configuration data.
Note: this is done in the
PROD instance.

Figure 3-2 Migrating Kintana data between DEV and PROD

Migrating Configuration Data

This section provides an overview of the requirements and processes for using Migrators. This information is provided to help you communicate with the Mercury IT Governance Administrator (who maintains the Mercury ITG instances and license information) and the System Administrator (who maintains the Mercury IT Governance Server) when configuring the product.

This section covers the following topics:

- *How Migrators Work*
- *Using the Migrators - Overview*
- *Instance Requirements for Using Migrators*

How Migrators Work

Migrators are provided as Object Types in Mercury Change Management. These Migrator Object Types are run through a Workflow. Each supported entity type has its own Object Type. For example, to migrate a Workflow from one instance to another, use the Kintana Workflow Migrator Object Type.

Packages are used to process and audit the migration of configuration changes. When the Package (containing a Migrator object) enters the appropriate execution step in a Workflow, the Migrator's commands are executed. The commands extract the data that defines the entity into text (XML) files. These text files are then imported into the target instance. *Figure 3-3* represents this process.

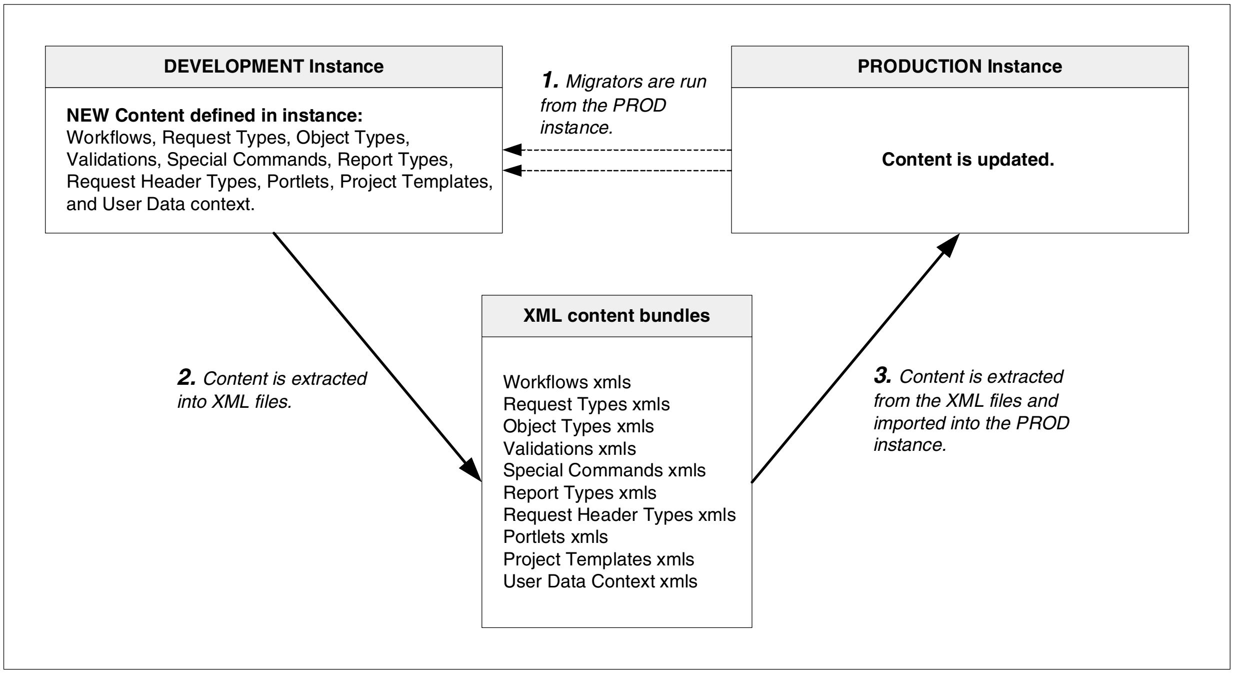

DEVELOPMENT Instance
NEW Content defined in instance:
Workflows, Request Types, Object Types, Validations, Special Commands, Report Types, Request Header Types, Portlets, Project Templates, and User Data context.
1. Migrators are run from the PROD instance.
PRODUCTION Instance
Content is updated.
XML content bundles
Workflows xmls
Request Types xmls
Object Types xmls
Validations xmls
Special Commands xmls
Report Types xmls
Request Header Types xmls
Portlets xmls
Project Templates xmls
User Data Context xmls
2. Content is extracted into XML files.
3. Content is extracted from the XML files and imported into the PROD instance.

Figure 3-3 Migrator content extraction and import overview

Using the Migrators - Overview

To use the Migrators to migrate configuration information between instances:

1. Configure the appropriate Environments (DEV and PROD) to represent instances involved in the migration.

2. Define a Package Workflow to model the desired migration process.

3. Build a Package using this Workflow, using the appropriate Migrator Object Types to create the Package Lines.

4. Submit the Package and migrate the Package Lines. Use the execution log generated to evaluate the results of the migration.

Detailed instructions for using the Migrators are included in the *"Migrators Guide and Reference"* document.

Instance Requirements for Using Migrators

To use the Migrators:

- At least two working Mercury Change Management instances (DEV and PROD) must be available

- All instances must be accessible over a network

Requirements for PROD Instance

The following items must be configured in the PROD instance. This is the destination instance that will receive the configuration data from the DEV instance.

Requirements for a successful migration include:

- Environments (PROD and DEV) defined in the Environment screen in the Workbench.

- At least one Workflow (Workflow Scope = **Packages**) must be available to run the migration. This Workflow must contain at least one execution step with the source and destination Environments configured to DEV and PROD, respectively.

- Migrator Object Types must be enabled. There is a different Object Type for each of the following entities that can be migrated: Validation, User Data, Special Command, Workflow, Report Type, Object Type, Request Type, Request Header Type, Project Template, Portlet, User Data Contexts.
- The user creating, submitting, and processing the migrations must have a Change Management power license and proper screen access. For details, see the *"Security Model"* guide.

Archiving Configuration Data

Use Migrators to document processes by exporting configuration information to text files. These text files conform to the XML (eXtended Markup Language) specification and are suitable for storage in many archiving systems including source control systems. Source control check-in can be integrated into the Migrator Object Types, allowing organizations to maintain a detailed record of the specific changes made to the production configurations.

To archive configuration data:

1. Perform an Extract only using the appropriate Migrator.

   The content is extracted into .xml files and grouped (by entity) into .zip files.

2. Check the extracts (.zip files) into your source control.

   Each extract contains a file (Source_Descriptor.xml) that describes the Mercury ITG product version and date of extraction.

3. These files can then be imported back into a Mercury IT Governance instance (of the same version) at any point in the future.

Chapter 4 Configuring a Request Resolution System - Process Overview

This chapter provides an overview of the process used to configure a Request resolution system in Mercury Demand Management. It summarizes each phase of configuration. Additional details for configuration, are included in the referenced chapters.

This chapter covers the following topics:

- *Configuring Your Request Resolution System*
- *Example: Configuring a Request Resolution System*

Configuring Your Request Resolution System

To configure a Request resolution system or process:

1. *Gathering Process Requirements and Specifications*

   Before configuring Mercury Demand Management to manage a Request resolution process, collect specific information related to the business process, any information needed to process the Request, users who will create and process Requests, and the communication devices surrounding the process.

2. *Mapping your Process into a Workflow*

   Using the information gathered in the *Gathering Process Requirements and Specifications* chapter, build the Workflow. This includes setting up required Workflow Step sources, creating Validations to be used by the transitions, and adding steps and transitions to the Workflow.

3. *Constructing the Request Type*

   Using the information gathered in the *Gathering Process Requirements and Specifications* chapter, build the Request Types. This includes creating and configuring Request Type fields and field logic. It also includes configuring the Request Types and Workflows to work together.

4. *Integrating Participants into Your Request Resolution System*

   After the Workflow and Request Types are constructed, construct security around the process. Specify who can create and process Requests, who can act on a particular Workflow step, and who can alter the process entities (such as Workflow and Request Types).

   Work with your Mercury IT Governance Administrator to configure User and Security Group definitions.

5. *Setting Up Communication Paths*

   Mercury Demand Management includes a number of features that provide visibility into Requests in the Request resolution process. This includes Notifications for Workflow steps, Portlets that provide additional real-time visibility, and built-in reports.

6. *Rolling Out Your Deployment Process*

   It is recommended a formal change management process be followed when configuring the Mercury IT Governance Center. This includes testing configurations, migrating them into a production instance, enabling the processes, and training the user base.

Example: Configuring a Request Resolution System

This section provides a business case example for configuring a Request resolution system. This example is used throughout this document to discuss configuration techniques.

Example: Request Resolution System for ACME Company

The financial group at ACME Company uses a proprietary software solution to manage the many aspects of ACME's finances:

- Billing

- Accounts Payable
- Accounts Receivable
- Fixed Asset Management
- Inventory
- Payroll
- Reporting
- Cash Management

This software solution is still relatively new and is used by hundreds of employees daily. Almost every day, an employee thinks of an idea for new or enhanced functionality to the financial system that they offer to their manager. Depending on the quality of the suggestion, a manager has the option to submit a request for new or enhanced functionality to the financial system. Initially, the request must contain the following information:

- Whether the request is for new or enhanced functionality
- The module to be enhanced (such as Billing, Inventory, or Payroll)
- The priority of the request
- Who originated the request
- A description of the new or enhanced functionality
- Any supporting documents (such as detailed proposals or Web sites)

If the request is approved, more information must be gathered:

- The user who approved the request
- Estimated time to completion
- The group assigned to build the new or enhanced functionality
- Any supporting documents (such as design documents or test plans)

During its lifecycle, the request must be approved or at least viewed by the following users or groups:

- The manager who originated the request
- The financial group director (budget approval)
- An IT analyst (feasibility, time estimates)
- The IT manager (approval to build)

- Developers (building the new/enhanced functionality)

Chapter 5 Gathering Process Requirements and Specifications

This chapter discusses the following information that needs to be collected before developing a Request resolution system:

- Business process:
  What are the steps in the process and which steps need to be reviewed and approved?
- Information needed to resolve the Request:
  What information needs to be gathered to resolve the Request? This information will translate to fields on the Request. For example, in order to resolve a Software Bug type of Request, you need to gather information on the bug: such as what is it, how can it be reproduced, and which machines does it occur on.
- Participants who will create and process Request:
  What level of security will be placed on this system?
- Communication devices surrounding the process:
  Determine whether to communicate using Notifications, the Dashboard, or reports.

This chapter covers the following topics:

- *Gathering Requirements for Workflow*
- *Gathering Requirements for Request Type*
- *Identifying Participants and Security*
- *Establishing Communication Points and Visibility*

Gathering Requirements for Workflow

The first step to configuring a Request resolution process is to define the process—the actual steps required to resolve a Request. This includes process information such as when to obtain reviews and approvals on the Request, who needs to approve the Steps, when to execute commands, and the path (transitions) between steps in the process.

The following sections discuss the specific information that needs to be gathered for Workflow configuration:

- *Defining the Business Flow*
- *Gather Information on Each Step in the Process*
- *Consider Using Subworkflows*

Defining the Business Flow

Map the business process. This consists of the steps (decisions, conditions, and any executions) and transitions needed to resolve Requests. It is helpful to graphically map these processes.

In this phase:

- Identify all decision points in the process
- Determine a flow between steps (transitions). Consider all possible exit values from each step (such as, Approved, Not Approved, Rework Required, or Error)
- Identify process closure points (success or failure)

The following example illustrates the process design issues that should be considered.

Example: Defining the Business Flow

ACME Company needs to configure a resolution process for processing change requests for their Financial Applications system. This system consists of over ten modules (including billing, accounts payable, accounts receivable, fixed asset management, inventory, reporting, payroll, and cash management). ACME's IT group needs to create a process that can address the complications related to evaluating and approving changes to this system.

Example: Business Process Overview

ACME first creates a high-level business process. The process begins when someone in the Financial group submits a Request for an enhancement.

1. **Submit:**
   The Request is submitted by someone in ACME's Financial group. If the priority of the Request is **High** or **Critical**, a manager will need to be informed right away.

2. **Validate:**
   The Request is reviewed. A feedback loop is built into the business process at this point, in case more information is needed from the original requestor.

3. **Approve:**
   The Request is approved or rejected.

4. **Schedule:**
   If the Request is approved, the work needed to create the enhancement must be scheduled. If ACME lacks the necessary resources at the moment, the Request should be put on hold.

5. **Develop:**
   The requested enhancement is developed by ACME's IT group.

6. **Deploy:**
   The finished enhancement is deployed to ACME's Financial group.

Additional Process Requirement:

When a Request is submitted, its Priority should be evaluated. If the Request is **Critical** or **High**, the Financial manager and members of the ACME's IT group should be informed.

To address this requirement, ACME added an Execution step that would evaluate the Request's Priority. If the Request is **Critical** or **High**, it is routed to another Execution step that sends a Notification to the appropriate users.

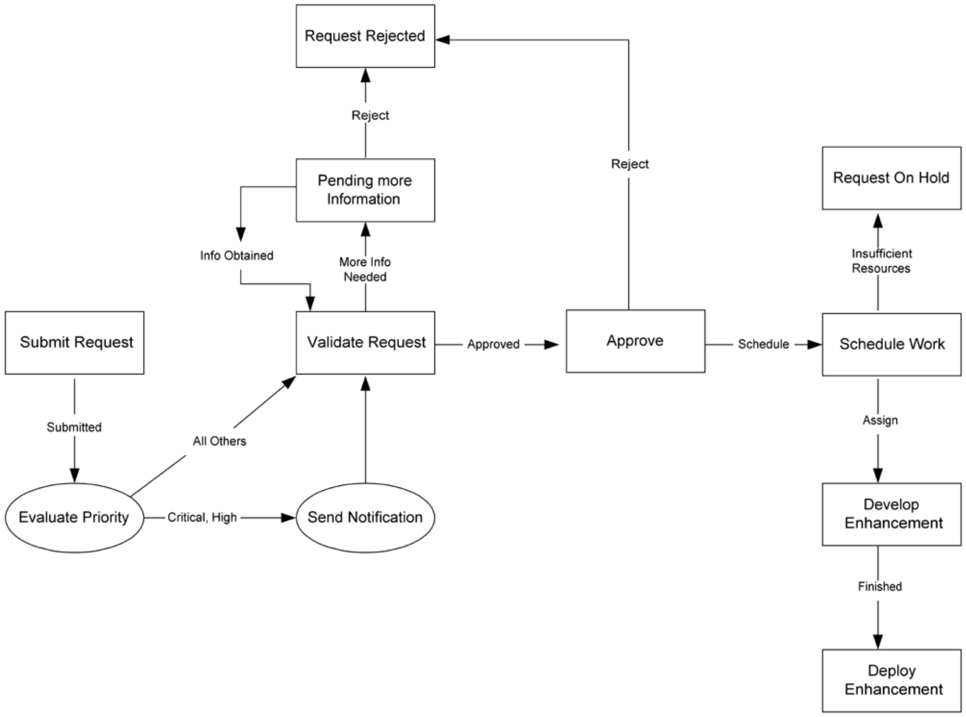


Figure 5-1 Revised Business Process

Gather Information on Each Step in the Process

After designing the business flow of a Request resolution process, gather detailed information on each step and transition in the process. This section discusses the information that needs to be collected. Use the worksheet provided in *"Configuration Worksheets"* on page 405 to collect the required information.

For each step in the process, collect the following information:

- Step name
- Description: Describe the goal of the step.
- Step Type: decision, execution, condition, or Subworkflow.
    - Decision step specific information: number of approvals required, timeouts, etc.
    - Execution step specific information:

    - The desired results of the execution. This will help you to choose the execution type and build any required commands.
    - Execution timing. Determine whether the execution will occur immediately or be processed manually.
  - Subworkflow step specific information
- Transition values and Validation:
  Transition values are the possible results for the step. Depending on the result, the process will proceed in different directions. Use one of Mercury Demand Management's system Validations or create a custom one.

Note

For each step, collect information on Participants and Notifications. This is discussed in the following sections:

- *"Identifying Participants and Security"* on page 44
- *"Establishing Communication Points and Visibility"* on page 50

Consider Using Subworkflows

A Subworkflow is any Workflow that is referenced from within another Workflow. Use Subworkflows to model complex business processes into logical, more manageable and reusable sub-processes.

Workflows can be used as Subworkflows within a parent Workflow. An entire Subworkflow is represented by a single icon in the parent Workflow window's **Layout** tab. This simplifies the potentially complex graphical layout and enables the easy reuse of common Workflow configurations.

Example: Using a Subworkflow

ACME decides to use a Subworkflow for the development portion of their process. This Subworkflow can be referenced in one part of the process. *Figure 5-2* illustrates where ACME could implement a Subworkflow.

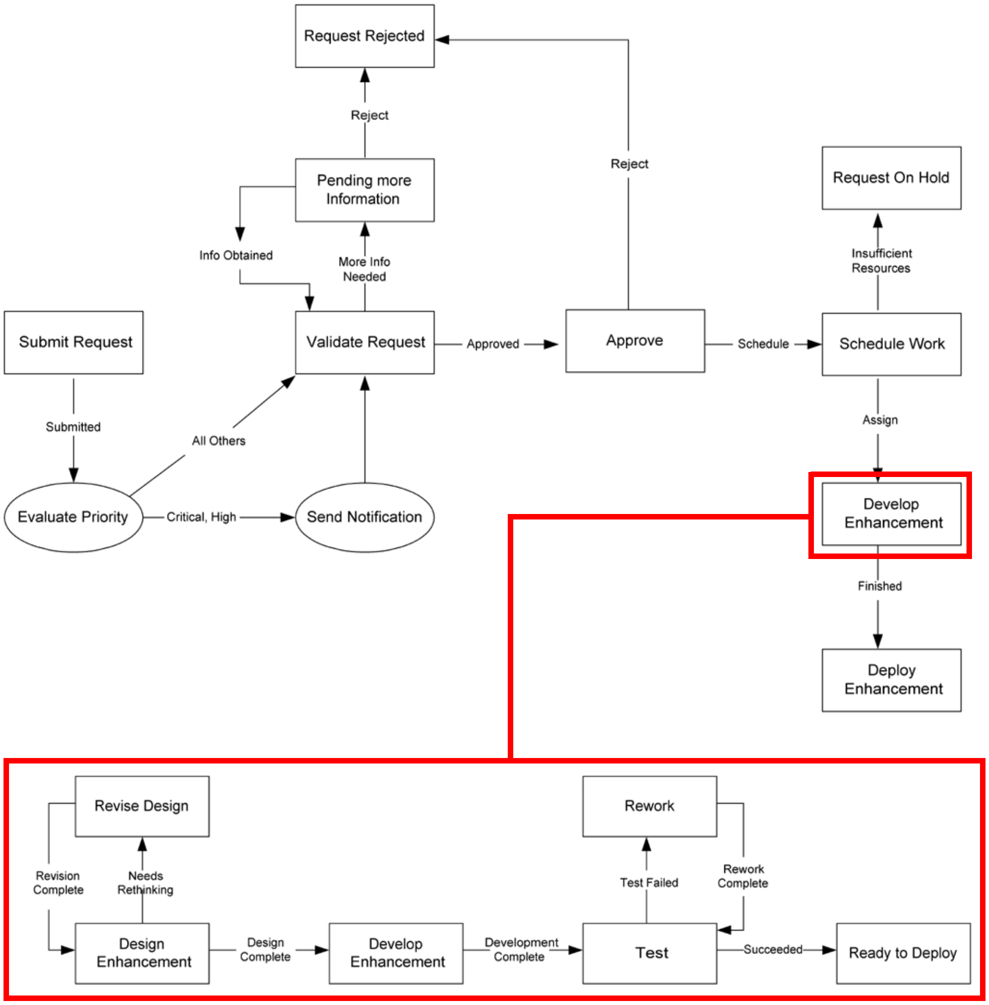


Figure 5-2 ACME Business Process with Subworkflow

Consider Request Statuses

A Request Status can be associated with a Workflow Step. The Request's Status at a particular Workflow Step will drive field logic during the Request's lifecycle. Build the Request Type before associating its Statuses with Workflow Steps. For more detailed information on Request Statuses, see *"Creating Request Statuses"* on page 157.

Gathering Requirements for Request Type

Many different types of Requests can be sent through a Workflow. As a Request moves through its resolution process, its fields and Status can change.

The following sections discuss the specific information that needs to be gathered before configuring Request Types:

- *Request Type Fields*
- *Request-Workflow Interaction*
- *Request Header Type*
- *Request Type Commands*

Request Type Fields

Each Request requires different information to process it. For example, to resolve a software bug, you need to know the software unit, product version, problem and priority. This information is captured using Request Type fields.

For each field in the Request, collect the following information:

- Field name. The field's prompt should help ensure that the correct information is captured.
- Information type. What type of information needs to be collected? Should it be a text field? Will users pick from a pre-determined list of values? The field information type is governed by its Validation, which defines the field's component type as well as what information can be entered into the field. For example, a field using a Numeric Text Field Validation will accept only numeric values. For more detailed information on Validations, see *"Request Type Field Validations"* on page 124.
- Field behavior. There are many aspects of a field that can be controlled:
    - The field can be configured to become uneditable or required depending on the value of other fields, or at a certain Workflow step.
    - The field can also be configured to automatically populate itself based on values in other fields.
    - The field can also be configured to be uneditable or invisible based on which user is looking at the Request.

More detailed information on field behavior can be found in *"Configuring Field Behavior - Overview"* on page 130.

- Field Statuses

Use the worksheets in *"Configuration Worksheets"* on page 405 to collect the Request Type field specifications.

Example: ACME collects information for the software change Request

ACME needs to know the following information in order to properly resolve a Request for a software change to the financial system:

- The name of the user creating the request
- Whether the request is for new or enhanced functionality
- The module to be enhanced (such as Billing, Inventory, or Payroll)
- The priority of the request
- A description of the new or enhanced functionality
- Any supporting documents (such as detailed proposals or Web sites)

As the request progresses along the business process, other information becomes necessary:

- Whether ACME has the budget to develop the change
- The estimated time to completion for the change
- The name of the developer assigned to build the change

To describe the Request Type, ACME decided they needed to define the following fields:

- Created By: The user who created the Request.
- New or Enhanced: Whether the change being requested is a new piece of functionality or an enhancement to an existing module.
- Priority: The priority of the Request. **Critical** Requests are acted on much faster than Requests with **Low** priority.
- Impacted Module: The software module to be changed.
- Description: A brief description of the change being requested.
- Supporting Documents: A place for the requestor to attach any supporting documents to the Request. These documents might be URLs or more detailed proposals in Rich Text Format.

See *"Constructing the Request Type"* on page 111 for additional examples on Request Type fields.

Request-Workflow Interaction

The list of possible Statuses the Request can take on as it moves through its resolution process can be configured. Each Request Status can control Request field attributes, such as whether or not the field is visible or editable. A Request Status can be tied to a Workflow Step, which means that when a Request reaches a certain Step, it acquires a Status that determines its fields' attributes. For more detailed information on Request Status Dependencies, see *"Configuring Field Behavior Using Status Dependencies"* on page 157.

In most cases, a single Request Type is associated with a single Workflow. Information contained in the Request (which is defined in the Request Type) works in conjunction with the Workflow process to ensure that the Request is correctly processed. While it is possible to use one Workflow with many different Request Types, the level of possible integration between Request Type and Workflow tends to suggest a 1:1 mapping.

It is also possible to restrict which Workflows and Request Types can be used together. Determine what, if any, restrictions are to be put in place at this level.

Example

Requesting a software change requires different information and processing than requesting a scope change to a project. Therefore, it is likely that the Workflows built around each business process will be different, and at least one field in each Request Type will be different.

Request Header Type

Request Header Types define a standard collection of fields that appear in the header or any other region of a Request using that Request Type. Each Request Type must have an associated Request Header Type. Request Header Types contain a standard set of fields that can be enabled or disabled. It is also possible to add custom fields to the Request Header Type.

Note

Request Header Types can contain Field Groups, which are collections of fields pre-configured for special product functionality like Resource Management or Program Management Office. For more detailed information on Request Header Type requirements, see *"Request Header Type"* on page 43.

When Field Groups are associated with existing Request Types (through the Request Header Type definition), database tables are updated to handle this new configuration. Due to the scope of database changes, the Database Statistics need to be re-run on the database. Instructions for this are included in *System Administration Guide*. Contact the System Administrator for help with this procedure.

Request Type Commands

Commands can be contained in a Request Type that allow it to perform command-line executions. Request Type commands often reference information stored in its fields. These commands are executed at specific points (execution steps) in the Workflow.

Collect the following information for each Request Type command that will be designed:

- The goal/purpose of the commands.
- Functional steps within the commands.
- When the commands should be run.

Use the worksheet in *"Configuration Worksheets"* on page 405 for assistance in collecting the correct data.

See *Commands and Tokens Guide and Reference* for additional information on building commands.

Identifying Participants and Security

A great deal of control can be implemented over a Request resolution system. Restrict user actions around:

- Request creation:

  - Who can create Requests.
  - Who can use a specific Workflow.
  - Who can use specific Request Types.
- Request processing:
  - Who can approve / process each step in the Workflow.
  - Who can view and edit fields in the Request.
  - Who can delete a Request.
- Configuring your Request resolution process:
  - Who can edit the Workflow.
  - Who can edit each Request Type.

Configuring this data and process security often involves a setting a number parameters: licenses, Access Grants, entity level settings and field level settings.

Tip

Use Security Groups or dynamic access (Tokens) whenever possible. Avoid specifying a list of users to control an action; for example, specifying a list of users who can act on a Workflow step. If the list of users changes (due to a departmental reorganization), that list would have to be updated manually in many places on the Workflow. By using a Security Group instead of a list of users, the Security Group can be updated once, and the changes will be propagated throughout the Workflow steps.

For the Request resolution process, collect information that will help to identify users, group them into Security Groups, and restrict access to certain functionality.

Example: ACME Determines Participants and Security

The process of approving changes to ACME's Financial system application involves many groups and individuals within the company.

- ACME Financial Group: Users of ACME's Financial system application

- Lucy Barnstorm: Director of ACME Financial Group
- Ralph Guderjahn: Business analyst
- Tyrone Chambers: Director of ACME IT group
- Hiroki Nanahara: Personnel Manager in ACME IT group
- ACME Development: Engineers for ACME IT
- Carlos Quintana: Lead Engineer
- Nora van Epstein: Manager of Release team
- Harold Lomax: Configuration Manager for ACME IT

Within this group of users, there are some logical divisions of labor. Using this division, ACME constructs the following Security Groups.

Table 5-1. ACME's Security Groups

Security Group	Members	Responsibilities
Financial Apps - Create and View Requests	ACME Financial Group Lucy Barnstorm	Responsible for creating Requests. The users can create Requests at any time and view the status of Requests they are involved in.
Financial Apps - Manage Resolution System	Harold Lomax	Responsible for Request resolution system. The user has the ability to modify the Request resolution process (Workflow, Request Types, and Security Groups). The user can also act on any step in the process.
Financial Apps - Validate and Approve Requests	Lucy Barnstorm Ralph Guderjahn Tyrone Chambers	Responsible for evaluating and approving incoming Requests. Can reject or approve Requests for development.
Financial Apps - Schedule Requests	Tyrone Chambers Hiroki Nanahara	Responsible for approving Requests for development, scheduling and assigning work, or putting Requests on hold until sufficient resources are available.
Financial Apps - Develop Requests	ACME Development Carlos Quintana Nora van Epstein	Responsible for developing enhancements specified in Requests, including functional design, implementation, and QA.
Financial Apps - Deploy Changes	Nora van Epstein	Responsible for overseeing deployments to the ACME Financial group.

Using these Security Groups and user definitions, ACME collects specific information related to their Request resolution process. This information will be considered later when defining Security Groups and Workflows. The information is presented in the following tables:

- Table 5-2, “ACME Request Creation Security,” on page 47
- Table 5-3, “ACME Request Processing Security - Financial System Change Workflow,” on page 48
- Table 5-4, “ACME - Security around managing the Financial System Change Process,” on page 49

Table 5-2. ACME Request Creation Security

Action	**Users allowed to perform action**	**Controlled by:** **(Users, Security Group, Token)**
Create a Request	ACME Financial Group; Lucy Barnstorm	Financial Apps - Create and View Requests, Financial Apps - Manage Resolution System
Use the Financial System Change Workflow	Everyone	Financial Apps - Create and View Requests, Financial Apps - Manage Resolution System, Financial Apps - Validate and Approve Requests, Financial Apps - Schedule Requests, Financial Apps - Develop Requests, Financial Apps - Deploy Changes
Use the Financial System Change Request Type	Everyone	Financial Apps - Create and View Requests, Financial Apps - Manage Resolution System, Financial Apps - Validate and Approve Requests, Financial Apps - Schedule Requests, Financial Apps - Develop Requests, Financial Apps - Deploy Changes

Notice that Harold Lomax was added to each action by adding the Financial Apps - Manage Resolution System Security Group to each step. This provides a single, relevant user with override privileges to keep the process moving.

ACME also indicates how they would like to control which users can act on each step. They select to exclusively use Security Groups and Tokens. Notice that multiple criteria can be specified to enable access to a single step: for example, you could specify two Security Groups and a TOKEN [REQ.CREATED_BY] to enable access. Users who meet any of the requirements (members of at least one Security Group or the value of the Token) can act on the step.

Table 5-3 specifies which users can act on a specific step in the Workflow. See *Figure 5-3* to see the process referenced in this table.

Table 5-3. ACME Request Processing Security - Financial System Change Workflow

Workflow Step Name	Users allowed to act on	Controlled by: (Users, Security Group, Token)
Validate Request	Lucy Barnstorm; Ralph Guderjahn; Tyrone Chambers	Financial Apps - Validate and Approve Requests Financial Apps - Manage Resolution System
Pending More Information	Financial Group member who created the Request; Lucy Barnstorm	TOKEN (REQ. CREATED_BY); Financial Apps - Create and View Requests (Security Group) Financial Apps - Manage Resolution System
Approve	Lucy Barnstorm; Ralph Guderjahn; Tyrone Chambers	Financial Apps - Validate and Approve Requests Financial Apps - Manage Resolution System
Schedule Work	Tyrone Chambers Hiroki Nanahara	Financial Apps - Schedule Requests Financial Apps - Manage Resolution System
Develop Enhancement	ACME Development Carlos Quintana Nora van Epstein	Financial Apps - Develop Requests Financial Apps - Manage Resolution System
Deploy Enhancement	Nora van Epstein	Financial Apps - Deploy Changes Financial Apps - Manage Resolution System

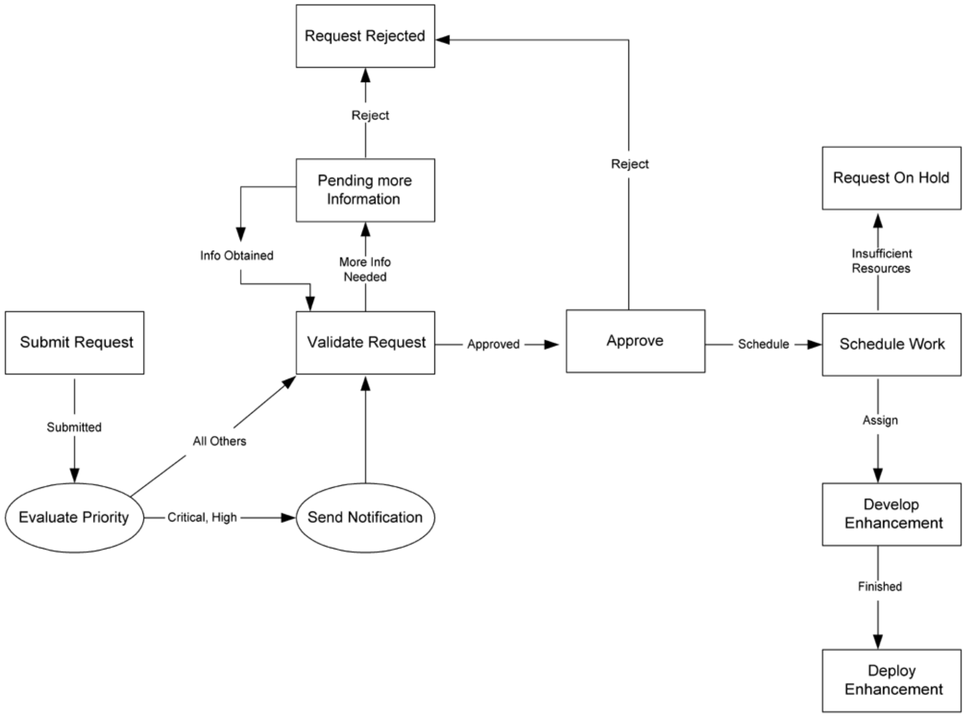


Figure 5-3 ACME Business Process

ACME must also specify who can modify the existing process. This level of security is configured using Ownership settings and Security Group Access Grants. See *"Setting Configuration Security"* on page 197 for more information on these topics.

Table 5-4. ACME - Security around managing the Financial System Change Process

Action	Users allowed to perform action	Controlled by: (Users, Security Group, Token)
Modify the Workflow	Harold Lomax	Financial Apps - Manage Resolution System
Modify the Financial System Change Request Type	Harold Lomax	Financial Apps - Manage Resolution System

Establishing Communication Points and Visibility

Determine the communication points and methods for providing visibility into the process and Request statuses. This section discusses the following features used to increase visibility:

- *Notifications on Workflow Steps*
- *Notifications on Field Changes*

Note

This section lists the information that needs to be gathered to define Notifications. For more information on defining and using Portlets and Reports, refer to the following links:

- *Using the Dashboard*
- *Reports Guide and Reference*

Notifications on Workflow Steps

It is possible to send a Notification when a Workflow step becomes eligible, has a specific outcome, or has a specific error. For each Workflow step in the process, collect the following information:

Table 5-5. Information to gather for Workflow Step Notifications

Workflow step name	Include Notification for step? (Yes / No)
Step 1 - Name	Yes
Step 2 - Name	No
Step 3 - Name	No

For each step that requires a Notification, gather the following information:

Table 5-6. Information to gather for Workflow Step Notifications

Parameter	Description
Workflow Step Name	The name of the step that requires a Workflow.

Table 5-6. Information to gather for Workflow Step Notifications

Parameter	Description
Notification Event (All, Eligible, Specific Result, Specific Error)	Specifies the event that triggers the Notification. the possible values are **All**, **Eligible, Specific Result**, or **Specific Error**.
Value (for Specific Result)	Specifies that a Notification is sent for the selected result.
Error (for Specific Error)	Specifies that a Notification is sent for the selected error.
Recipient	Determine who should receive the message. you can choose to send the Notification to users based on: **Username, Email Address, Security Group, Standard Token,** or **User Defined Token**.
Message	Determine what the message will say. Also determine if it will contain a link to the Request.

Example: ACME configures Notifications

ACME determines that a Notification needs to be added to the following steps:

Table 5-7. ACME - Workflow steps with Notifications

Workflow step name	When to send Notification	Recipients
Send Notification	[REQ.PRIORITY] = ‘High’, ‘Critical’	Financial Apps - Validate and Approve Requests Financial Apps - Schedule Requests
Pending More Information	Eligible	TOKEN [REQ.CREATED_BY]
Schedule Work	Eligible	Financial Apps - Schedule Requests
Develop Enhancement	Eligible	Financial Apps - Develop Enhancement
Deploy Enhancement	Eligible	TOKEN [REQ.CREATED_BY] Financial Apps - Deploy Enhancement

Notifications on Field Changes

Notifications can be sent when a field in a Request changes value. For more detailed information on setting up these Notifications, see *"Setting Notifications on Request Field Changes"* on page 228.

Chapter 6
Mapping your Process into a Workflow

This chapter provides an overview for how to set up a Mercury Demand Management Workflow. This includes information on configuring all Workflow steps, transitions and Validations included in a process.

This chapter discusses the following topics:

- *Building the Workflow Skeleton - Overview*
- *Create the Required Step Source*
- *Configure the Step's Transition Values (Validation)*
- *Add Steps and Transitions to the Workflow Layout*

Building the Workflow Skeleton - Overview

This section provides a high level overview for building a Workflow.

To create a Workflow:

1. Enter the general Workflow information in a new Workflow window.

   This includes entering the Name and Workflow Scope in the **Workflow** tab.

2. Create any new step sources using the Workflow Step Sources window. This includes:

   - Creating decision steps.
   - Creating execution steps.

- Creating Subworkflow steps.
- Creating any new Validations used by the above steps.

3. Add the Workflow steps to the **Layout** tab.
4. Add transitions between the Workflow steps.
5. Add Security to the Workflow.
6. Add Notifications to select Workflow steps.
7. Synchronize Workflow steps with Request Type statuses.
8. Enable the Workflow.

Required Workflow Settings for Request Resolution Process

In order to user a Workflow in a Request resolution process, the following requirements must be met:

- The Workflow Scope must be set to **Requests**
- The Workflow must be **Enabled**
- Add steps and transitions to the **Layout** of your Workflow
- If this Workflow is to integrate with a Package Workflow, specify the Package Workflow in the **Package Workflows** tab
- Enable Security for each Workflow step. This allows users to act on the step.

Note

- Workflows are created and configured using the Workbench.
- Users must have a Power License and have the proper Access Grants in order to create and edit a Workflow. See *Security Model Guide and Reference* for details.

Copying a Workflow

To copy an existing Workflow:

1. In the Workflow Workbench window, query for the Workflow to copy.

   The **Results** tab displays the results matching the search.

2. In the **Results** tab, select the Workflow.

3. Click **Copy**.

   The Copy Workflow window opens.

4. In the Workflow Name field, enter the name of the new Workflow.

5. Select the Workflow items to copy.

6. Click **OK**.

   A Question dialog opens. Click **Yes** to edit the Workflow.

7. Make any additional edits and click **Save**.

Specifying the First Step in a Process

For each Workflow, you can explicitly define the first eligible step in the Workflow.

To specify the first step:

1. Open the Workflow window and click the **Workflow** tab.

2. From the First Step field, select the name of the step to serve as the first step in the process.

   This field contains all of the fields that have been defined for the Workflow.

3. Click **Save**.

Note

Once Workflow Steps have been defined in the **Layout** tab, their sequential order can be set using the arrow buttons in the **Step Sequence** tab. The sequence relates to the graphical position of a step that a user will see when processing an entity, but it does not affect its actual processing. For example, Step A has a display sequence of 3 and Step B has a display sequence of 5. In this situation, Step A will not necessarily be eligible before Step B. Step B could, for example, be specified as the First Step in the process. Additionally, depending on the transition path of the steps, Step A may never be required.

Create the Required Step Source

Mercury Demand Management includes a number of standard Workflow step sources that can be added to a Workflow. These sources are preconfigured with standard Validations (transition values), Workflow events, and Workflow scope. These available steps specify the following common attributes, which are expected to remain consistent across all Workflows which use that step source:

- The Validation associated with the step (and thus the list of valid transition values out of the step).
- The voting requirements of the step.
- The default timeout value for the step. Each step can be configured to have a unique timeout value.
- The icon used for the step within the graphical layout.

Browse through all of the Workflow Step Sources using the Available Workflow Steps window in the Workflow Workbench. If a step source that meets the process requirements is not available, one needs to be created.

Tip

If Mercury Demand Management has a Workflow step source that meets the process requirements, simply copy and rename it. This can save configuration effort and avoid user processing errors. For example, if you need a step to route a Request based on whether it needs more analysis, you could copy and use the preconfigured Request Analysis Workflow step source.

Copy the step source so that it can be used uniquely for the processes. This allows you to control who can edit the step source, ensuring that the process will not be inadvertently altered by another user.

Create a new step source when the step requires any of the following:

- Unique Validation leaving the step
- Unique execution on the step: PL/SQL function, Token, SQL function, or Workflow Step Commands
- Different processing type: immediate vs. manual
- Specific Workflow Scope
- Unique combination of the above settings

The following sections discuss when and how to use specific settings in the Workflow Step Source:

- *Creating a Workflow Step Source - Overview*
- *Creating a Decision Type Step*
- *Create an Execution Type Step*

Creating a Workflow Step Source - Overview

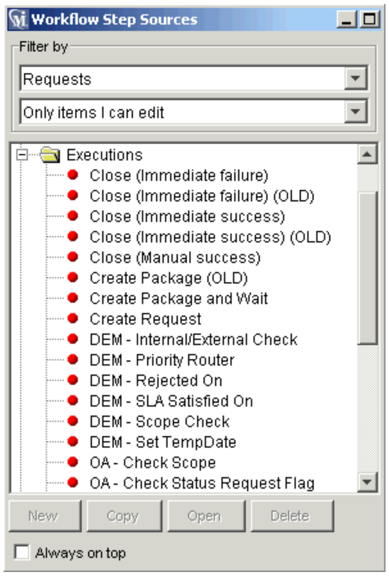
It is possible to create new Workflow step sources from the Workflow Step Sources window in the Workbench.

To create a new Workflow Step Source:

1. Click the **Configuration** screen group and click the **Workflows** icon.

   The Workflow Workbench and Workflow Step Sources windows open.

2. Select the Workflow Step Sources window.



Filter by

Requests

Only items I can edit

Executions
- Close (Immediate failure)
- Close (Immediate failure) (OLD)
- Close (Immediate success)
- Close (Immediate success) (OLD)
- Close (Manual success)
- Create Package (OLD)
- Create Package and Wait
- Create Request
- DEM - Internal/External Check
- DEM - Priority Router
- DEM - Rejected On
- DEM - SLA Satisfied On
- DEM - Scope Check
- DEM - Set TempDate
- OA - Check Scope
- OA - Check Status Request Flag

□ Always on top

3. Select to Filter by **Requests**.

4. Select the folder that corresponds to the type of Workflow step source that will be created.

   For example, to create an execution step, select the **Executions** folder.

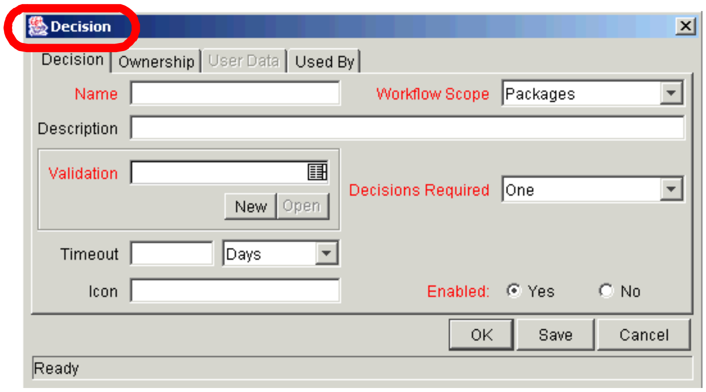
5. Click **New**.

   This opens a window that corresponds to the selected Workflow step source type. The Decision and Execution windows are shown below. For information on configuring Workflow Step Sources, see *"Creating a Decision Type Step"* on page 602 and *"Create an Execution Type Step"* on page 65.

Note

Condition steps cannot be added, deleted or modified. Demand Management supports a set number of process Condition steps. These can be added to the Workflow layout just as any other Workflow step source.

If a Condition step is selected in the Workflow Step Sources window, the **New** button will not be enabled.



Name

Workflow Scope: Packages

Description

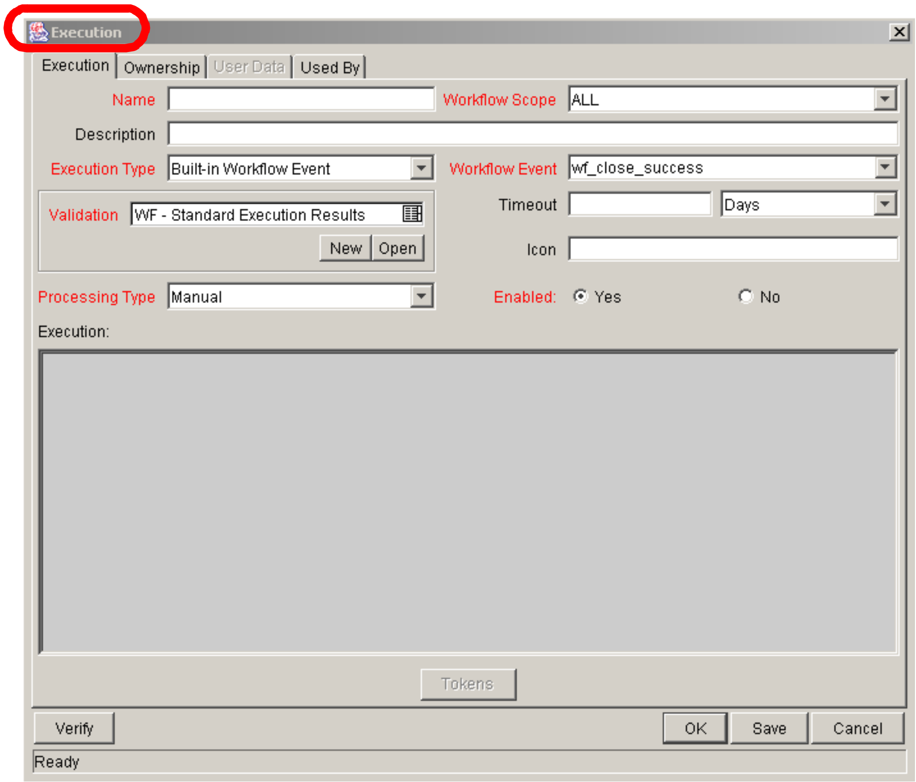
Validation

Decisions Required: One

Timeout: Days

Icon

Enabled: ● Yes ○ No



Name

Workflow Scope: ALL

Description

Execution Type: Built-in Workflow Event

Workflow Event: wf_close_success

Validation: WF - Standard Execution Results

Timeout: Days

Icon

Processing Type: Manual

Enabled: ● Yes ○ No

Execution:

6. Enter the required information and any optional information needed to define the step.

   For detailed information about setting up the steps, see *"Creating a Decision Type Step"* on page 602 and *"Create an Execution Type Step"* on page 65.

7. In the Enabled radio button, select **Yes** to be able to use this step in a Workflow.

8. Click the **Ownership** tab.

   Select which Ownership Groups will have the ability to edit this Execution or Decision.

9. To save the changes and close the Execution or Decision window, click **OK**.

The new Workflow step source is now included in the Workflow Step Sources window. It can be used in any new or existing Workflow with the corresponding Workflow Scope.

Related Topics:

- *"Creating a Decision Type Step"* on page 60
- *"Create an Execution Type Step"* on page 65

Workflow Step Source Configuration and Usage Restrictions

The following restrictions apply to Workflow step sources:

- A step source that is being used in a Workflow can not be deleted.
- A Validation for a step source that is being used can not be changed. If the Validation needs to change, copy the step source and configure a new Validation.
- The Workflow step source must be Enabled to use them on a Workflow.
- Only add step sources to a Workflow when the Workflow has a matching Workflow Scope, or the step source has a scope of **All**.
- A step from a Workflow that has been used to process a Request can not be deleted. This would compromise data integrity. Instead of deleting the step, remove all transitions to and from the step and disable it.

Creating a Decision Type Step

To create a decision step source:

1. *Enter the general information on the Decision step source* (name, scope, description)

2. *Select a Validation*

3. *Specify the voting requirements on the step*

4. *Specify the default timeout value*

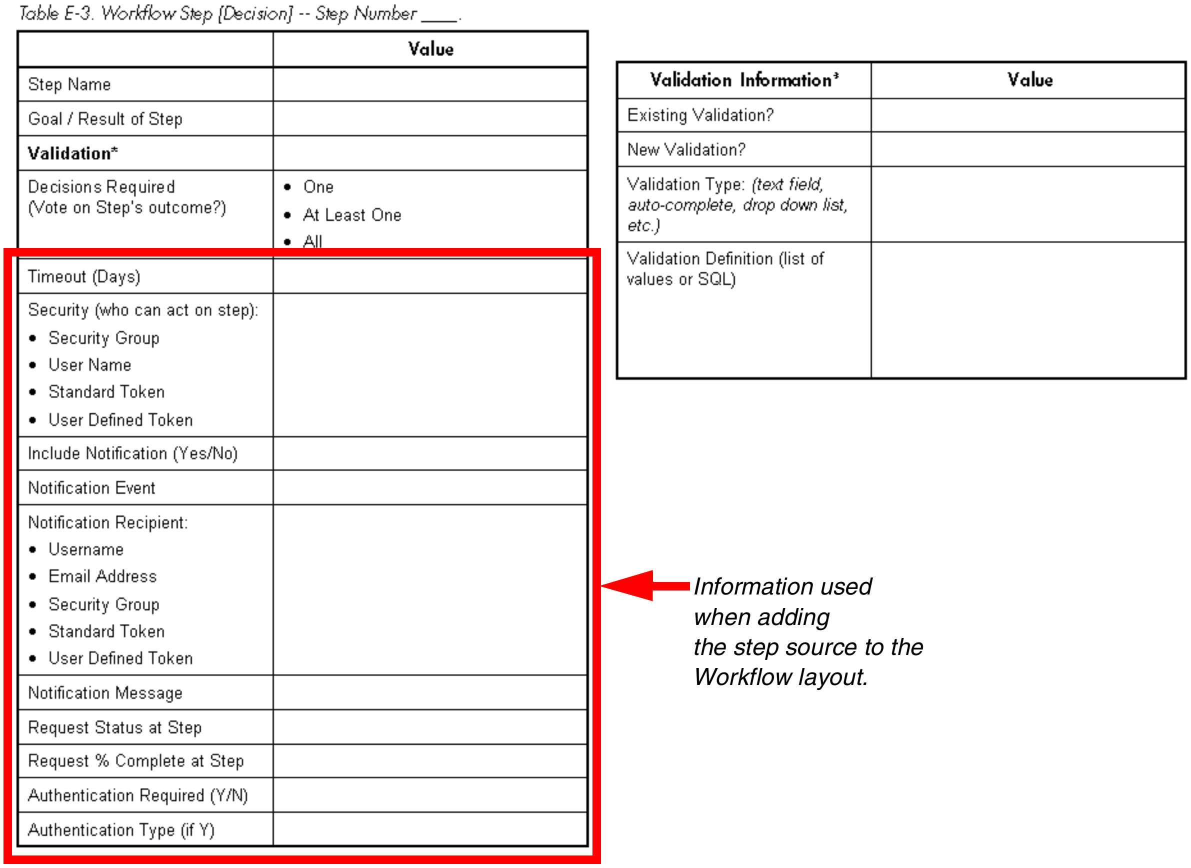


Table E-3. Workflow Step [Decision] -- Step Number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Decisions Required (Vote on Step's outcome?)	• One • At Least One • All
Timeout (Days)	
Security (who can act on step): • Security Group • User Name • Standard Token • User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: • Username • Email Address • Security Group • Standard Token • User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

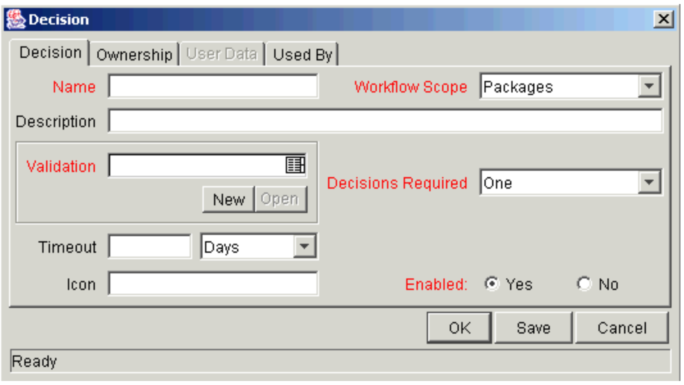
Validation Information[3]	**Value**
Existing Validation?	
New Validation?	
Validation Type: *(text field, auto-complete, drop down list, etc.)*	
Validation Definition (list of values or SQL)	

Information used when adding the step source to the Workflow layout.

Figure 6-1 Information used to create the decision step source.

Enter the general information on the Decision step source

Enter the following information in the Decision window.



Workflow Scope: Packages

Decisions Required: One

Timeout: Days

Enabled: ● Yes ○ No

Table 6-1. Decision step source worksheet to window.

Field in Decision Window	Description
Name	This is the name that describes the step source. The step can be renamed when added to the Workflow.
Workflow Scope	Describes the type of Workflow that will be using this step source. This should be set to **ALL** or **Requests** for Request resolution processes.
Description	Description of the step source.
Validation	Specifies the possible values that can exit the Workflow step. See *"Configure the Step's Transition Values (Validation)"* on page 83.
Decisions Required	This specifies the number of people who need to approve a specific step. See *"Specify the voting requirements on the step"* on page 63.
Timeout	Specifies how long (in specific units) the Step can stay eligible for before either completing with a value of **Timeout** (if Decisions Required is **All** or **One**) or a result (if Decisions Required is **At Least One**). If this Workflow Step remains eligible for the value entered in the Timeout value, the Request can be configured to send an appropriate Notification, as well as escalate to other steps in the Workflow.

Table 6-1. Decision step source worksheet to window.

Field in Decision Window	Description
Icon	A different graphic can be specified to represent steps of this source for use on the Workflow **Layout** tab. This graphic needs to exist in the `icons` subdirectory on the Mercury ITG server. All icons should be in .gif format.
Enabled	The step source must be enabled in order to add it to the Workflow layout.

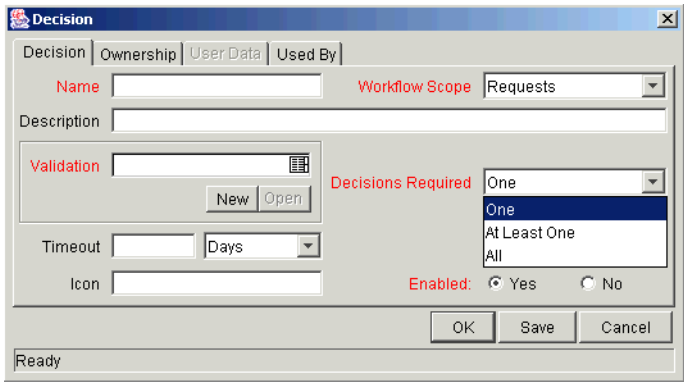
Select a Validation

Select a Validation that has the transition values required for leaving the step. If a Validation that meets your requirements is not available, create a new one from the Workflow Step Source window.

For additional details, see *"Configure the Step's Transition Values (Validation)"* on page 83.

Specify the voting requirements on the step

When a Decision step is defined, the number of decisions required for that Workflow Step can be defined. *Figure 6-2* displays the available options for the Decisions Required field.



Workflow Scope: Requests

Decisions Required: One

Timeout: Days

Enabled: ● Yes ○ No

Ready

Figure 6-2 Decision Window - Decisions Required Drop Down List

The following choices are available for Decisions Required:

- **One**

  If **One** is selected, the Workflow Step can progress if any one user who is eligible to act on this step makes a decision.

- **At Least One**

  A Timeout period must be defined to use this choice. When **At Least One** is selected for the Workflow Step, the step waits for the voters to vote on this step for a predefined amount of time, designated as the Timeout. If all voters mark their decisions before the timeout period, it takes the cumulative decision as the decision for the step and proceeds forward. If any of the voting results differ before the 'Timeout' period, the step will immediately result in a 'No consensus' outcome.

Note

> It is possible to define Specific Errors in Workflow Steps such as '**Timeout**' and '**No consensus**' as either Success or Failure in the Define Transition window. For more information, see *"Adding Transitions Between Steps"* on page 97.
>
> If all voters decide on **Approve**, the final decision is Approve. If all voters decide on **Not Approved**, the final decision is Not Approved. If some voters decide on **Approved** and one voter decides on **Not Approved**, the result is No consensus.
>
> If at the end of the Timeout, only a few voters (or only one voter) have cast their vote, the cumulative decision of the voters that voted will be used.
>
> If at the end of the Timeout no one has voted, the step will result in a Timeout.

- **All**

  The All step is also commonly used along with a specified Timeout period. Selecting **All** makes it mandatory for all voters to vote on the Workflow Step. The Workflow Step waits until the Timeout period for the voters to vote. If all voters vote, the cumulative decision is considered. If some or none of the voters voted, the step remains open or closes due to a timeout, depending on the configuration.

When using **All** or **At Least One**, all users must unanimously approve or not approve one of the Validation's selections. Otherwise, the result is **No Consensus**.

Specify the default timeout value

A timeout specifies the amount of time that a step can stay eligible for completion before completing with an error (if Decisions Required is **All** or **One**) or completing with a result (if Decisions Required is **At Least One**). Timeouts can be by minute, hour, weekday or week. Timeout parameters for Executions and Decisions are a combination of a numerical timeout value and a timeout unit (such as weekdays).

If this Workflow Step remains eligible for the value entered in the Timeout value, the Request can be configured to send an appropriate Notification and escalate to other steps in the Workflow. This field is often used in conjunction with the **At Least One** and **All** settings for Decisions Required.

Timeouts can be uniquely configured for each Workflow Step in the **Layout** tab. The timeout value specified in the step source acts as the default timeout value for the step. When adding a step to the Workflow using this step source, you can specify a different timeout value for the step.

Create an Execution Type Step

To create an execution step source:

1. *Enter the general information on the Execution step source*
2. *Define the Executions*
3. *Select a Validation*
4. *Specify the default timeout value*

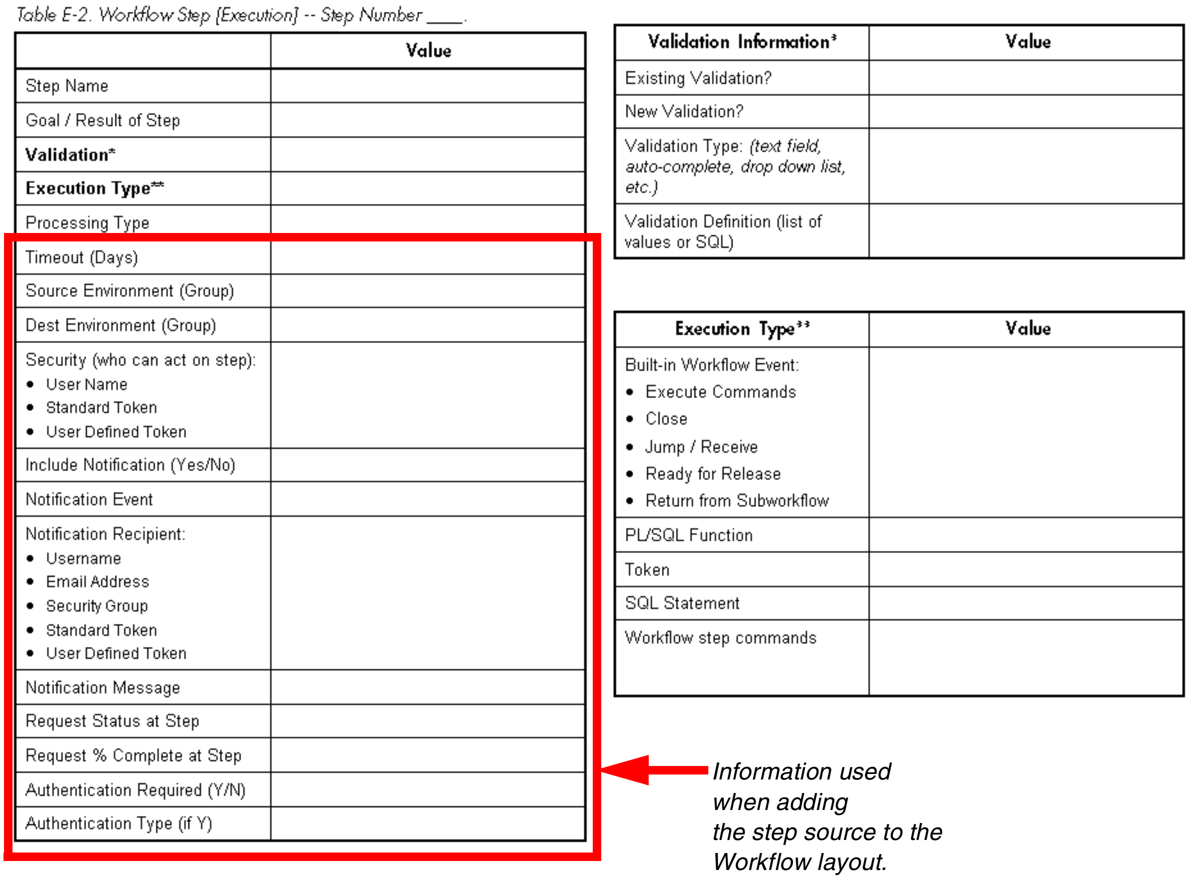


Table E-2. Workflow Step [Execution] -- Step Number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Execution Type**	
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): • User Name • Standard Token • User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: • Username • Email Address • Security Group • Standard Token • User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Validation Information*	**Value**
Existing Validation?	
New Validation?	
Validation Type: *(text field, auto-complete, drop down list, etc.)*	
Validation Definition (list of values or SQL)	

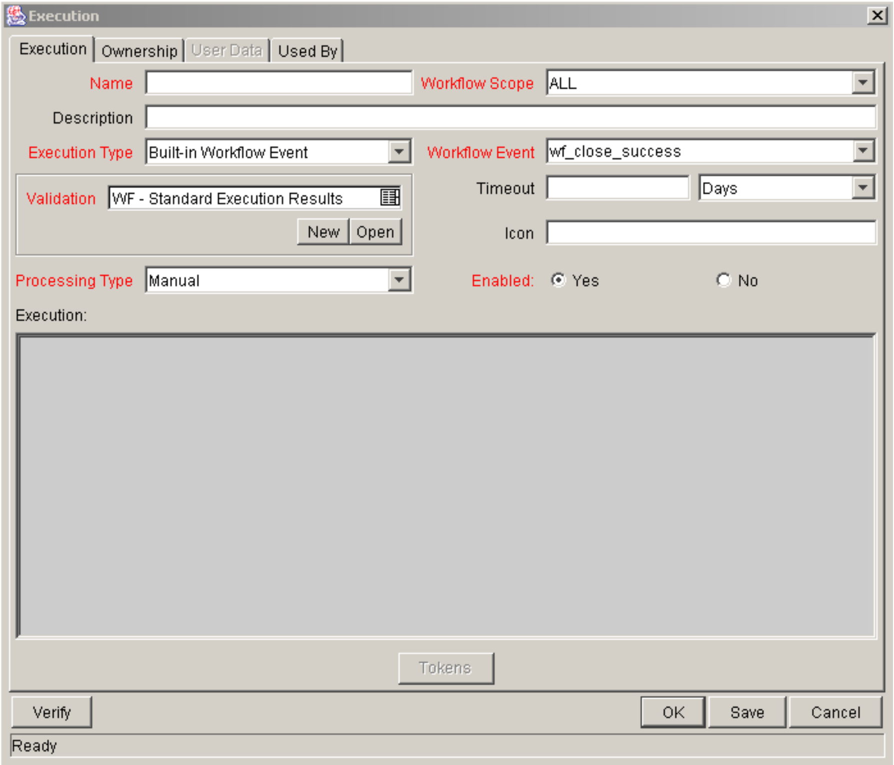
Execution Type**	**Value**
Built-in Workflow Event: • Execute Commands • Close • Jump / Receive • Ready for Release • Return from Subworkflow	
PL/SQL Function	
Token	
SQL Statement	
Workflow step commands	

Information used when adding the step source to the Workflow layout.

Figure 6-3 Information used to create the execution step source.

Enter the general information on the Execution step source

Enter the following information in the Execution window.



Name

Workflow Scope: ALL

Description

Execution Type: Built-in Workflow Event

Workflow Event: wf_close_success

Validation: WF - Standard Execution Results

Timeout: Days

Icon

Processing Type: Manual

Enabled: ● Yes ○ No

Execution:

Ready

Table 6-2. Execution step source worksheet to window.

Field in Execution Window	Description
Name	This is the name that describes the step source. The step can be renamed when added to the Workflow.
Workflow Scope	Describes the type of Workflow that will be using this step source. This should be set to **ALL** or **Requests** for Request resolution processes.
Description	Description of the step source. This should specify the execution that will occur.

Table 6-2. Execution step source worksheet to window.

Field in Execution Window	Description
Execution Type	Used to select the type of execution to be performed. The choices include: • **Built-in Workflow Event**: Executes a predefined command and returns its result as the result of the Step. • **SQL Statement**: Executes a SQL statement and returns its result as the result for the Step. • **PL/SQL Function**: Runs a PL/SQL function and returns its result as the result for the Step. • **Token**: Calculates the value of a Token and returns its value as the result for the Step. • **Workflow Step Commands**: Executes a set of commands, independent of an Object, at a Workflow Step.
Workflow Event	For Executions of type **Built-in Workflow Event**, the specific event to perform must be selected. The available choices in the drop down list depend on which Workflow Scope has been selected. The choices include: • **execute_request_commands**: Executes the Request Type commands for a Request. • **create_package**: Generates a Mercury Change Management Package. • **create_package_and_wait**: Generates a Change Management Package. The Create step that generates the Package holds it until the Package is closed. • **create_request**: Generates another Request. • **wf_close_success**: Sets the Request as closed with an end status of ‘Success.’ • **wf_close_failure**: Sets the Request as closed with an end status of ‘Failed.’ • **wf_jump:** (Mercury Change Management and Mercury Demand Management only) Instructs the Workflow to proceed to a corresponding Receive Workflow Step in another Workflow. • **wf_receive:** (Mercury Change Management and Mercury Demand Management only) Instructs the Workflow to receive a Jump Workflow Step and continue processing a Request or Package Line initiated in another Workflow. • **wf_return:** (Mercury Change Management and Mercury Demand Management only) Used to route a Subworkflow process back to its parent Workflow.

Table 6-2. Execution step source worksheet to window.

Field in Execution Window	Description
PL/SQL Function	For Executions of type **PL/SQL Function**, the actual function to run. The results of the function will determine the outcome of the step. Note: The results of the function must be a subset of the Validation values for that step.
Token	For Executions of type **Token**, the Token that will be resolved. The results of the Token resolution will determine the outcome of the step.
SQL Statement	For Executions of type **SQL Statement**, the actual query to run. The results of the query will determine the outcome of the step. Note: The results of the query must be a subset of the Validation values for that step.
Workflow step commands	For Executions of type **Workflow Step Commands**, the actual commands to run. The commands will result with a **Succeeded** or **Failed** value. Use a Validation with those values to enable transitioning out of the step based on the execution results.
Processing Type	Indicates whether the Execution is performed immediately (**Immediate**) when the Step becomes eligible or whether the Execution needs to be manually activated by a user (**Manual**).
Validation	Specifies the possible values that can exit the Workflow step. See *"Configure the Step's Transition Values (Validation)"* on page 83.
Timeout	If this Workflow Step remains eligible for the value entered in the Timeout value, the Request can be configured to send an appropriate Notification, as well as escalate to other steps in the Workflow. See *"Specify the default timeout value"* on page 83.
Icon	You can select a different graphic to represent this steps of this step source. This graphic needs to exist in the `icons` subdirectory on the Mercury ITG server. All icons should be in .gif format.
Enabled	The step source must be enabled in order to add it to the Workflow layout.

Define the Executions

Execution steps are used to perform specific actions. Mercury Demand Management provides a number of number of built in Workflow events for processing some common execution events (running Request Type commands, closing a Request, etc.). It is also possible to create custom executions based on SQL, PL/SQL, Token resolution, and custom commands.

This section discusses when to use specific types of executions and provides references for configuring these executions.

Execution steps can be created to perform the following actions:

- *Execute the Request Type Commands* and transition based on the success or failure of those commands.
- *Close the Request and mark it as a Success*
- *Close the Request and mark it as Failed*
- *Transition (jump) to a Workflow that is Processing a Package*
- *Receive control from a Workflow that is Processing a Package*
- *Return from a Subworkflow to the Parent Workflow*
- *Execute a PL/SQL function and then transition based on the result*
- *Execute a SQL statement and then transition based on the result*
- *Evaluate a Token and then transition based on the result*
- *Execute a number of system level commands and then transition based on the success or failure of those commands.*
    - o Example: Start a server
    - o Example: Stop a server

Execute the Request Type Commands

Certain process steps may require specific commands to be executed. Commands can be added to each Request Type, and the Workflow can be configured to execute Request Type commands at a specific step in the process. Each step will run its own commands, ensuring the correct execution for that Request Type.

Note

Mercury Demand Management includes a system step source that executes the Request Type commands. Use and modify (if necessary) a copy of this step source.

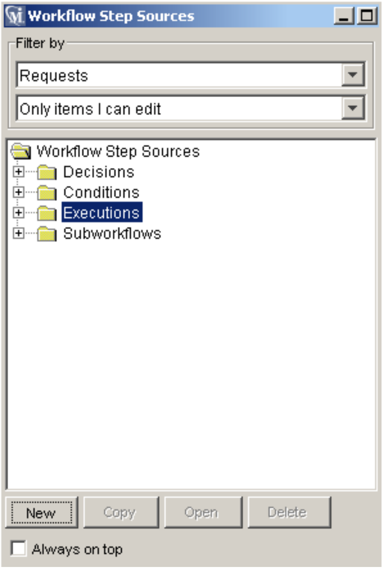
Step Source = **Execute Request Commands**

To create an execution step source that will execute the Request Type commands:

1. Open the Workflow Workbench.

2. Select the Workflow Step Sources window.

3. Select the Execution directory.

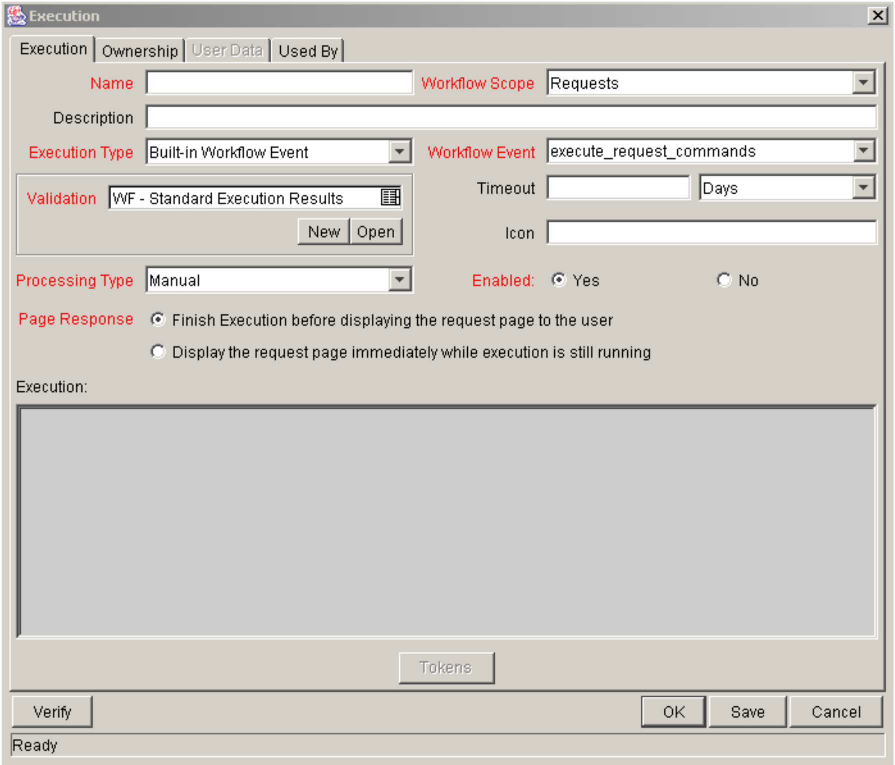


4. Click **New**.

   The Execution window opens.

5. Enter the following information:

Field in Execution Window	Value
Name	Enter a descriptive name for the step source.
Workflow Scope	**Requests**
Execution Type	**Built-in Workflow Event**
Workflow Event	**execute_request_commands**
Processing Type	**Manual** or **Immediate**
Validation	**WF - Standard Execution Results** (This is the default selection. You can select another existing or create a new Validation.)
Enabled	**Yes**
Processing Type	**Manual**
Page Response	This determines whether the step will complete the execution before reloading the Request page for the user (enabling them to make further changes), or whether the Request page will reload immediately while the execution is still in progress.



Name:

Workflow Scope: Requests

Description:

Execution Type: Built-in Workflow Event

Workflow Event: execute_request_commands

Validation: WF - Standard Execution Results

Timeout: Days

Icon:

Processing Type: Manual

Enabled: ● Yes ○ No

Page Response: ● Finish Execution before displaying the request page to the user ○ Display the request page immediately while execution is still running

Execution:

Ready

Close the Request and mark it as a Success

It is possible to create an execution step that closes a Request. Each Request Workflow should resolve with a closed Request. All the Requests that were closed successfully can then be reported on.

Tip

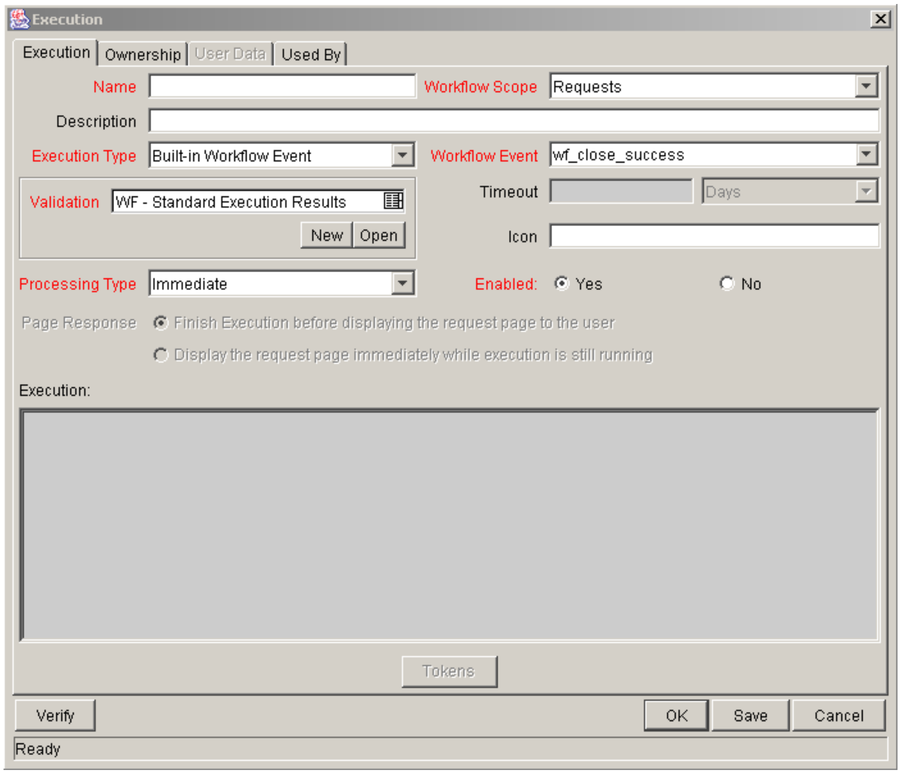
Mercury Demand Management includes a system step source that performs this task. Use this step source unless it does not meet the required specifications (such as Validation, or Processing Type).

Step Source = **Close (Immediate success)** or **Close (Manual success)**

To configure an execution step source to close a Request and mark it as a Success:

1. Create an execution step source with the following settings:

Field in Execution Window	Value
Name	Enter a descriptive name for the step source.
Workflow Scope	**Requests**
Execution Type	**Built-in Workflow Event**
Workflow Event	**wf_close_success**
Processing Type	**Manual** or **Immediate**
Validation	**WF - Standard Execution Results** (This is the default selection. You can select another existing or create a new Validation.)
Enabled	**Yes**



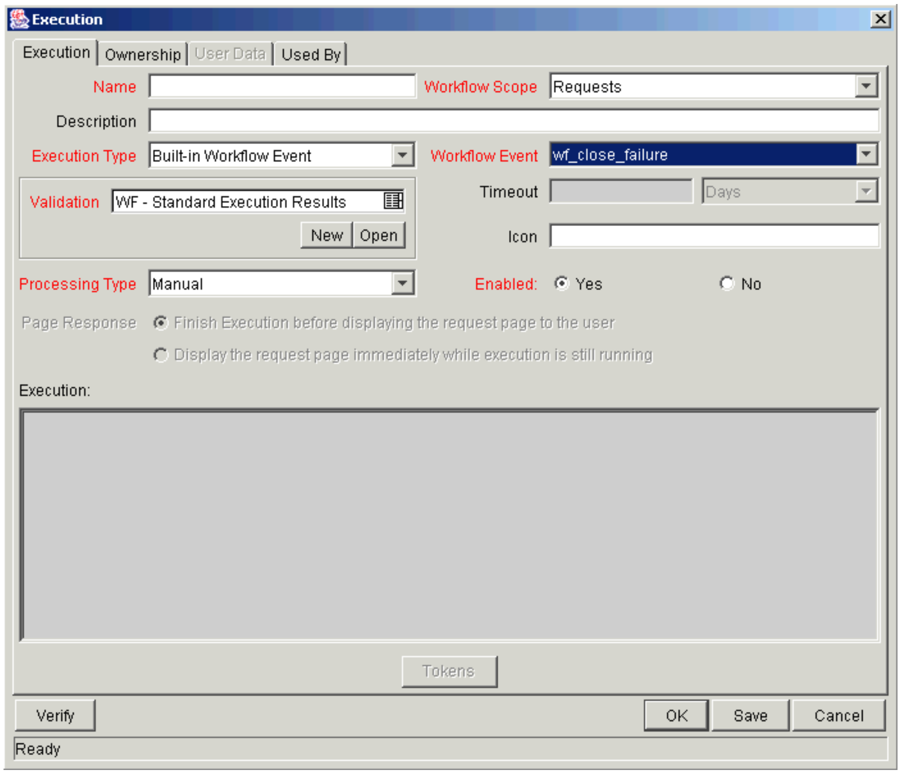
Name:
Workflow Scope: Requests
Description:
Execution Type: Built-in Workflow Event
Workflow Event: wf_close_success
Validation: WF - Standard Execution Results
Timeout: Days
Icon:
Processing Type: Immediate
Enabled: ● Yes ○ No
Page Response: ● Finish Execution before displaying the request page to the user
○ Display the request page immediately while execution is still running
Execution:

Close the Request and mark it as Failed

To configure an execution step source to close a Request and mark it as a Failed:

1. Set the following in the Execution window:

Field in Execution Window	Value
Name	Enter a descriptive name for the step source.
Workflow Scope	**Requests**
Execution Type	**Built-in Workflow Event**
Workflow Event	**wf_close_failure**
Processing Type	**Manual** or **Immediate**
Validation	**WF - Standard Execution Results** (This is the default selection. You can select another existing or create a new Validation.)
Enabled	**Yes**



Name

Workflow Scope: Requests

Description

Execution Type: Built-in Workflow Event

Workflow Event: wf_close_failure

Validation: WF - Standard Execution Results

Timeout: Days

Icon

Processing Type: Manual

Enabled: ● Yes ○ No

Page Response: ● Finish Execution before displaying the request page to the user
○ Display the request page immediately while execution is still running

Execution:

Ready

Tip

Mercury Demand Management includes a system step source that performs this task. Use this step source unless it does not meet the required specifications (such as Validation or Processing Type).

Step Source = **Close (Immediate failure)**

Transition (jump) to a Workflow that is Processing a Package

Request Workflows can communicate with Package Workflows at specific jump and receive points. To effectively utilize this functionality, properly configure both the jump and receive execution Workflow steps. See *"Advanced Workflow Topics"* on page 251 for additional details.

Receive control from a Workflow that is Processing a Package

Request Workflows can communicate with Package Workflows at specific jump and receive points. To effectively utilize this functionality, properly configure both the jump and receive execution Workflow steps. See *"Advanced Workflow Topics"* on page 251 for additional details.

Return from a Subworkflow to the Parent Workflow

Execution steps can be configured to automatically return from a Subworkflow to its parent Workflow. Create an execution step (to be used on the Subworkflow) with the following configuration:

Field in Execution Window	Value
Name	Enter a descriptive name for the step source.
Workflow Scope	**Requests**
Execution Type	**Built-in Workflow Event**
Workflow Event	**wf_return**
Processing Type	**Manual** or **Immediate**

Field in Execution Window	Value
Validation	**WF - Standard Execution Results** (This is the default selection. You can select another existing or create a new Validation.)
Enabled	**Yes**

Note

For a Request to transition back to the parent Workflow, the Subworkflow must contain a Return step. The transitions leading into the Return step must match the Validation established for the Subworkflow Step. Users must verify that the Validation defined for the Subworkflow Step is synchronized with the transitions entering the Return Step. The Subworkflow Validation is defined in the Workflow window. See *"Advanced Workflow Topics"* on page 251 for additional details.

Tip

Mercury Demand Management provides a system step source that performs this task. Use this step source unless it does not meet the required specifications (such as Validation or Processing Type).

Step Source = **Return from Subworkflow**

Execute a PL/SQL function and then transition based on the result

A PL/SQL function execution step runs a PL/SQL function and returns its results as the result of that Workflow Step. Include an execution step with the following source configuration:

Field in Execution Window	Value
Name	Enter a descriptive name for the step source.
Workflow Scope	**Requests**
Execution Type	**PL/SQL Function**
Processing Type	**Manual** or **Immediate**

Field in Execution Window	Value
Validation	Select or create a Validation that includes all of the possible values of the SQL query. Tip: you can create a Validation validated by SQL. Use the same SQL from the execution minus the WHERE clause.
Execution	Enter the PL/SQL function.
Enabled	**Yes**

Execute a SQL statement and then transition based on the result

SQL statement Execution steps are used when a Workflow needs to be routed based on the result of a query. A SQL statement execution step runs a SQL query and returns its results as the result of that Workflow Step.

Include an execution step with the following source configuration:

Field in Execution Window	Value
Name	Enter a descriptive name for the step source.
Workflow Scope	**Requests**
Execution Type	**SQL Statement**
Processing Type	**Manual** or **Immediate**
Validation	Select or create a Validation that includes all of the possible values of the SQL query. Tip: you can create a Validation validated by SQL. Use the same SQL defined for the execution minus the WHERE clause.
Execution	Enter the SQL query.
Enabled	**Yes**

Configuration notes:

- Only use select statements
- Tokens can be used within the WHERE clause
- Query must return only 1 value

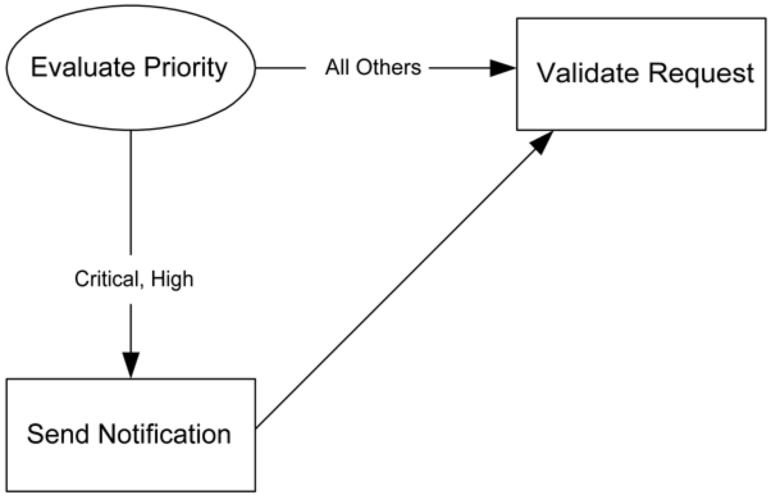
Evaluate a Token and then transition based on the result

Mercury Demand Management includes Workflow Execution steps that may be used to set up data-dependent rules for the routing of Workflow processes. Token Execution steps enable a Workflow to be routed based on the value of any field within a particular entity. A Token Execution step references the value of a given Token and uses that value as the result of the Workflow Step.

A transition may be made based on the value stored in the product by using Tokens in the Execution step. Include an execution step with the following source configuration:

Field in Execution Window	Value
Name	Enter a descriptive name for the step source.
Workflow Scope	**Requests**
Execution Type	**Token**
Processing Type	**Manual** or **Immediate**
Validation	Select or create a Validation that includes all of the possible values of the resolved Token. For example, if the Token is for the Priority field, use the Validation for the Priority field here as well.
Execution	Enter the Token for the value that the transition will be based on.
Enabled	**Yes**

For example, ACME needs to send an email Notification to the Validate and Approve Requests group if the Request's Priority is **High** or **Critical**.



ACME decides to use an Execution step to automatically evaluate the Priority of the Request and route it accordingly. If the Request's Priority is **High** or **Critical**, it gets sent to an immediate Execution step that sends a Notification to the Validate and Approve Requests group before continuing along the Workflow as normal. To accomplish this, an execution step source (Evaluate Priority) has been configured with the following parameters.

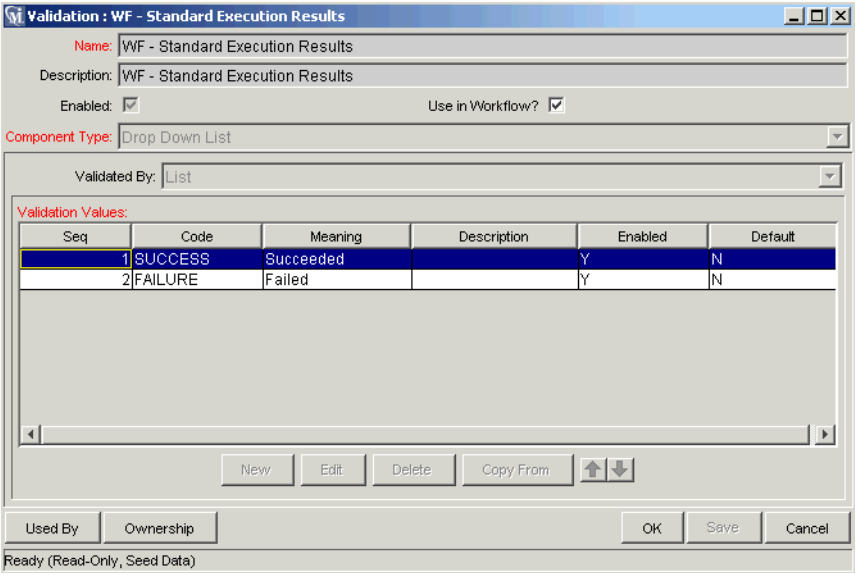
Field in Execution Window	Value
Name	Evaluate Priority
Workflow Scope	**Requests**
Execution Type	**Token**
Processing Type	**Immediate**
Validation	**CRT - Priority - Enabled**
Execution	**[REQ.PRIORITY_CODE]**
Enabled	**Yes**

Execute a number of system level commands and then transition based on the success or failure of those commands.

System level commands can be run for execution steps of the following Execution Type: **Built-in Workflow Event** (**execute_request_commands**) and

Workflow Step Commands. When either the Workflow or the Request Type commands execute at this step, the commands will either Succeed or Fail. To transition based on these results, the code for the Validation values must have the following values:

- SUCCESS
- FAILURE



Name: WF - Standard Execution Results

Description: WF - Standard Execution Results

Enabled: ☑ Use in Workflow? ☑

Component Type: Drop Down List

Validated By: List

Validation Values:

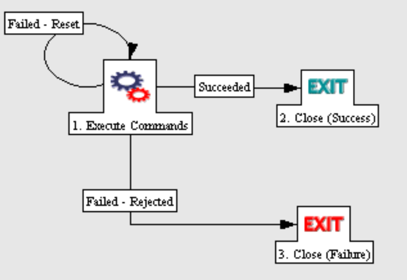
Seq	Code	Meaning	Description	Enabled	Default
1	SUCCESS	Succeeded		Y	N
2	FAILURE	Failed		Y	N

Ready (Read-Only, Seed Data)

Tip

It may be preferable to retain the option of resetting failed execution steps, rather than immediately transitioning along a “failed” path. This is often helpful when troubleshooting the execution. To configure this:

1. Create an execution step source to execute the Workflow or Request Type commands.
2. Create a Validation with the following Validation Values.
   a. SUCCEEDED
   b. FAILED
   c. FAILED - RESET
   d. FAILED - REJECTED
3. Add the step to the Workflow **Layout** tab.
4. Add transitions based on the following Specific Results:
   a. SUCCEEDED
   b. FAILED - RESET -- set the transition to return back into the same step.
   c. FAILED - REJECTED



When the commands execute successfully, they will follow the Success transition path. However, when the commands fail, they will not transition out of the step because no transition has been defined for the Failed result. The user has to manually select the Execution step and select Failed - Retry. The execution will re-run.

Select a Validation

Select a Validation that has the transition values required for leaving the step. If there is not already a Validation that meets the requirements, create a new one from the Workflow Step Source window.

For additional details, see *"Configure the Step's Transition Values (Validation)"* on page 83.

Specify the default timeout value

Timeouts in the execution steps can be set at two levels:

- Step level: the amount of time that a step is eligible before completing with an error. This is set in the Execution window.
- Command level: the amount of time that an execution is allowed to run before completing with an error. This applies to the Workflow Step Commands and Object Type Commands only. It is set in the Command window.

Timeouts can be by minute, hour, weekday or week. Timeout parameters for Executions and Decisions are a combination of a numerical timeout value and a timeout unit (such as weekdays).

If this Workflow Step remains eligible for the value entered in the Timeout value, the Request can be configured to send an appropriate Notification and escalate to other steps in the Workflow.

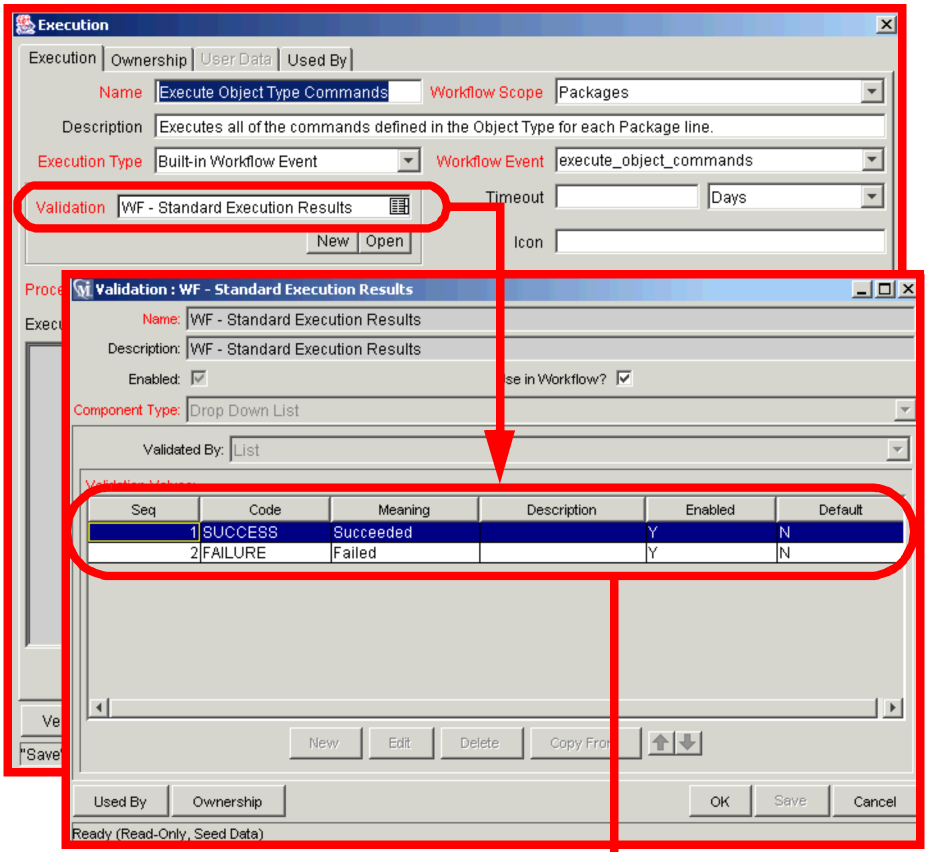
Timeouts can be uniquely configured for each Workflow Step in the **Layout** tab. The timeout value specified in the step source acts as the default timeout value for the step. When adding a step to the Workflow using this step source, specify a different timeout value for the step.

Configure the Step's Transition Values (Validation)

Workflows can be configured to transition based on values automatically returned from an execution or values selected by the user. For each Workflow step, you must define all of the possible values for the step's transition. This is set in the Validation field on the Execution window or the Decision window. The

Validation dictates the values in the Specific Result section on the Add Transition window.

1. Validation specifies all possible results for the step.



Execution

Name	Execute Object Type Commands	Workflow Scope	Packages
Description	Executes all of the commands defined in the Object Type for each Package line.		
Execution Type	Built-in Workflow Event	Workflow Event	execute_object_commands
Validation	WF - Standard Execution Results	Timeout	Days
		Icon	

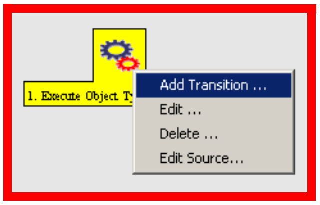
Validation : WF - Standard Execution Results

Name:	WF - Standard Execution Results
Description:	WF - Standard Execution Results
Enabled:	☑
Use in Workflow?	☑
Component Type:	Drop Down List
Validated By:	List

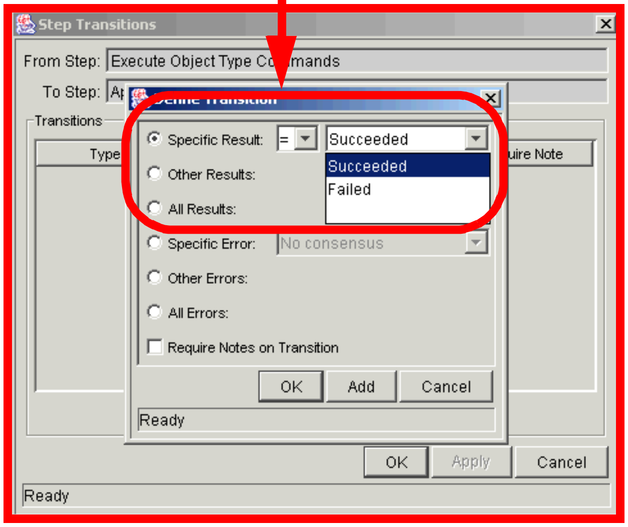
Seq	Code	Meaning	Description	Enabled	Default
1	SUCCESS	Succeeded		Y	N
2	FAILURE	Failed		Y	N

Ready (Read-Only, Seed Data)

2. Add a transition between two steps in the Workflow Layout tab.



3. Optionally base the transition on the values defined in the Validation.



Step Transitions

From Step:	Execute Object Type Commands
To Step:	Ap

● Specific Result: = Succeeded
○ Other Results:
○ All Results:
○ Specific Error: No consensus
○ Other Errors:
○ All Errors:
□ Require Notes on Transition

When specifying the Validation for the execution step source, specify all possible transition values in the Validation. When using that step source on the Workflow (add it to the Layout tab), decide whether to transition on one of the specific results, or a number of other transition options:

- Other Results
- All Results
- Specific Error
- Other Errors
- All Errors

Validations and Execution Type relationships

There is a correlation between the Validation and the Execution Type. For data-dependent transitions (Token, SQL, PL/SQL), the Validation must contain all possible values of the query or Token resolution. Otherwise, the execution step could result in a value that is not defined for the process, and the Request could become stuck in a Workflow step.

For most Built-In Workflow Events and executions that run commands, the Validation often includes the standard Workflow results (**Success** or **Failure**). If the commands or event execute without error, the result of **Success** is returned. Otherwise, **Failure** is returned.

The following table summarizes this relationship between Validations and Execution types.

Table 6-3. Relationship between Validation and Execution Type

Execution Types	Validation Notes
Built-in Workflow Event and Workflow Step Commands	Typically use a variation of the WF - Standard Execution Results Validation (Succeeded or Failed). A few of the Workflow Events have specific Validation Requirements: wf_return, wf_jump, wf_receive.
PL/SQL Function	Validation must contain all possible values returned by the function.
Token	Validation must contain all possible values for the Token.

Table 6-3. Relationship between Validation and Execution Type

Execution Types	Validation Notes
SQL Statement	Validation must contain all possible values for the SQL query. Tip: you can use the same SQL in the Validation (drop down or auto-complete list) minus the WHERE clause.

Tip

Use the information captured in the *"Configuration Worksheets"* on page 405 to construct the Validation.

Consider copying existing Validations to save time and ensure that the SQL or other Validation technique is configured properly.

Add Steps and Transitions to the Workflow Layout

Build the process graphically by dragging and dropping Workflow step sources onto the Workflow window's **Layout** tab. When a Workflow step source is included in a Workflow, it is then referred to as a "Workflow step." If a step source does not exist that meets the necessary requirements (such as decision, execution, correct transition Validation values, and processing type) one will need to be created.

When adding the step source to the **Layout** tab, supplemental information must be provided. The following sections discuss the configuration required when:

- *Adding Decision Steps*
- *Adding Execution Steps*
- *Adding a Subworkflow*
- *Adding Transitions Between Steps*

Adding Decision Steps

To add a Decision step to your Workflow:

1. Drag the Step Source onto the Workflow's **Layout** tab.

2. *Enter the general information on the Decision step*
3. *Specify the Security*
4. *Configure Notifications for the Workflow Step*
5. *Specify the Timeout value for the step*

Figure 6-4 shows the information used when creating a decision step.

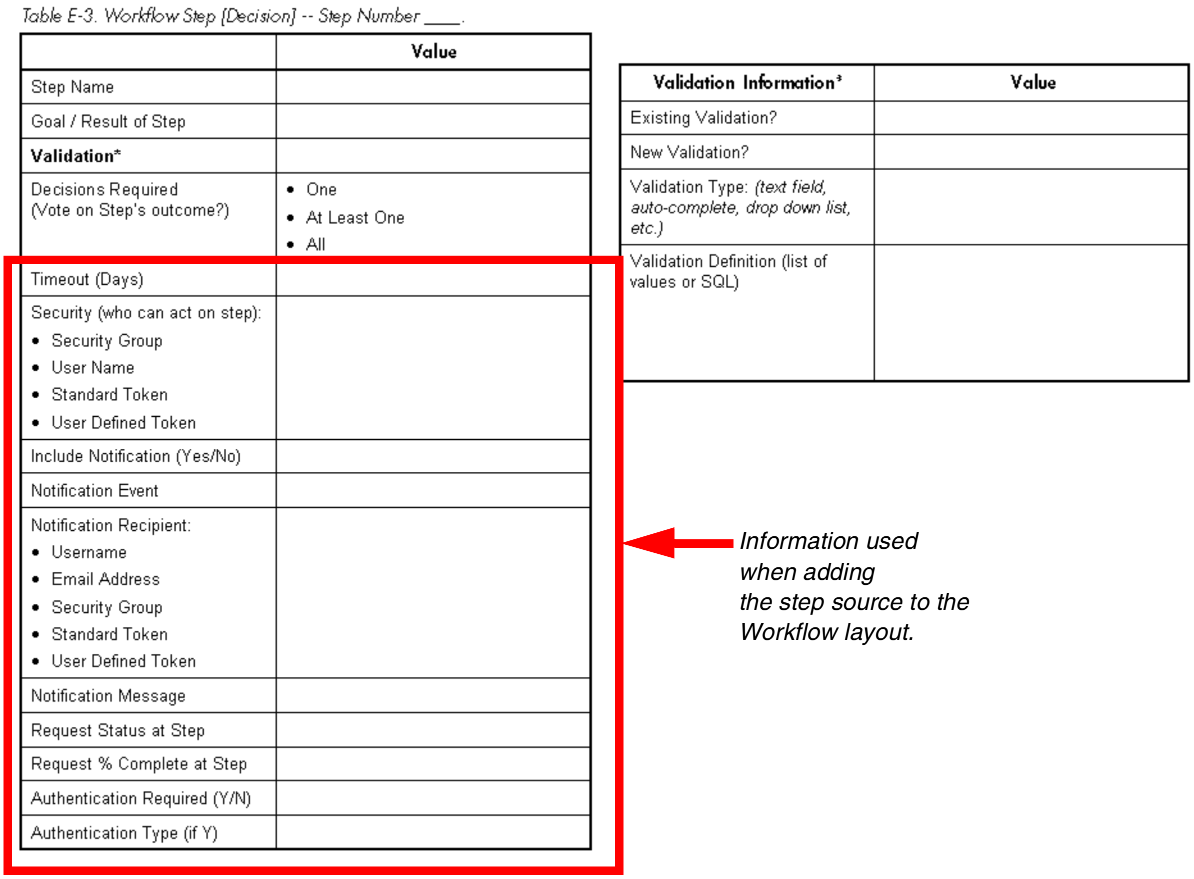


Table E-3. Workflow Step (Decision) -- Step Number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Decisions Required (Vote on Step's outcome?)	• One • At Least One • All
Timeout (Days)	
Security (who can act on step): • Security Group • User Name • Standard Token • User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: • Username • Email Address • Security Group • Standard Token • User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

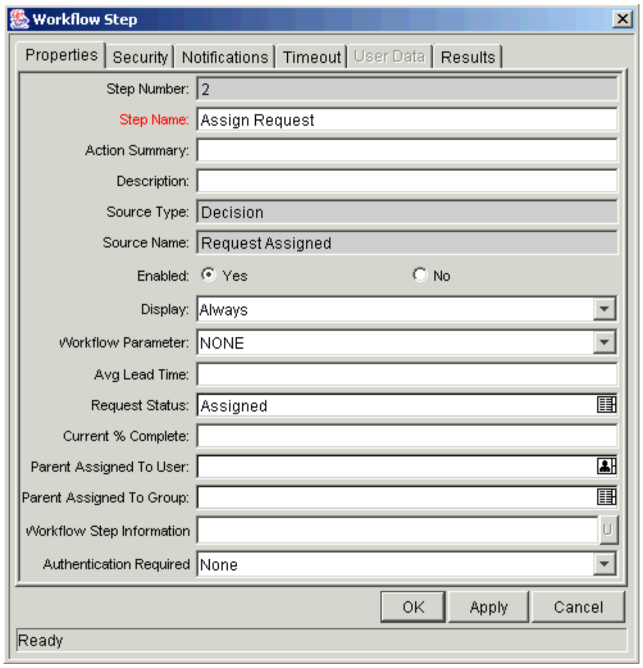
Validation Information[3]	**Value**
Existing Validation?	
New Validation?	
Validation Type: *(text field, auto-complete, drop down list, etc.)*	
Validation Definition (list of values or SQL)	

Information used when adding the step source to the Workflow layout.

Figure 6-4 Information used to create the decision step.

Enter the general information on the Decision step

Enter the following information in the Workflow Step window.



Step Number: 2
Step Name: Assign Request
Action Summary:
Description:
Source Type: Decision
Source Name: Request Assigned
Enabled: ● Yes ○ No
Display: Always
Workflow Parameter: NONE
Avg Lead Time:
Request Status: Assigned
Current % Complete:
Parent Assigned To User:
Parent Assigned To Group:
Workflow Step Information
Authentication Required None

Ready

Table 6-4. Decision Workflow Step window fields

Field/Tab on Workflow Step Window	Description
Step Name	Name of the step that appears on the Workflow window **Layout** tab.
Action Summary	The text that appears on the action button in the Request status panel. If it is left blank, the Step Name is used.
Description	A description of the step; could describe the goal of the step.
Enabled	Indicates whether the Workflow Step is currently enabled. If it is not enabled, the step will not be included in any new Requests using this Workflow. **This field cannot be changed if the step has any incoming Transitions.**
Display	Whether or not to show the step on the Request status panel.
Workflow Parameter	Used to save the results of a Workflow Step for later use in the Workflow processing.

Table 6-4. Decision Workflow Step window fields

Field/Tab on Workflow Step Window	Description
Avg Lead Time	A user-specified metric for comparing actual performance to estimated goals. It does not affect any transactional logic.
Request Status	The Status acquired by the Request when it reaches this step.
Current % Complete	The percentage of the resolution process that is complete at this step. This value rolls up into a Mercury Project Management Project when a Task has been created from a Request.
Parent Assigned to User	If this field is not empty when the step becomes Eligible, the Assigned to User of the Package automatically changes to the user specified in the field.
Parent Assigned to Group	If this field is not empty when the step becomes Eligible, the Assigned to Group of the Request automatically changes to the Security Group specified in the field.
Workflow Step Information	A text entry field in which any URL can be entered. This is where users can find documents, instructions or comments to aid them in working the Workflow Step.
Authentication Required	Determines whether the user acting on the step will need to provide authentication before acting, and if so, what kind.
Security Tab	Determines who can act on the step.
Notifications Tab	Specify who will receive an email Notification when this step becomes eligible or has a specific result or error.
Timeout Tab	Specify the Timeout value for this step. In the Timeout tab, select to use the Workflow step source timeout value or specify your own in the Specific Value section.

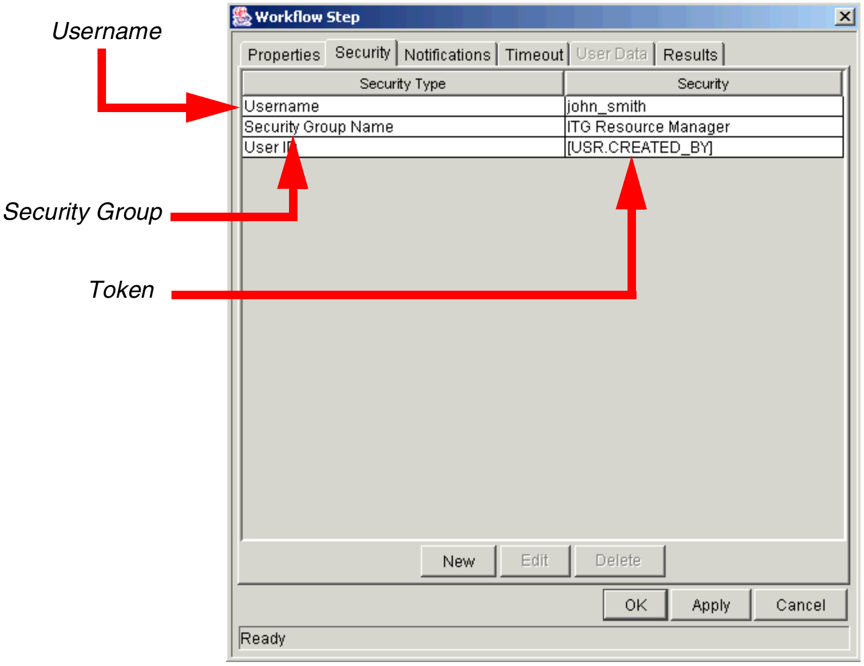
Specify the Security

"Integrating Participants into Your Request Resolution System" on page 177 provides information on setting up security for a Request resolution process. This includes such things as controlling who can create Requests and who can

act on specific steps in the process. Security related directly to processing a Workflow step is configured in the Workflow Step window.

Define who can act on the step by:

- Security Group
- Username
- Token (standard or user-defined)



Security Type	Security
Username	john_smith
Security Group Name	ITG Resource Manager
User ID	[USR.CREATED_BY]

Tip

Consider using Security Groups or dynamic access (Tokens) when defining the Workflow step security. Avoid specifying a list of users to control an action; for example, specifying a list of users who can act on a Workflow step. If the list of users changes (due to an organizational reorganization), that list would need to be updated manually in many places on the Workflow. By using a Security Group instead of a list of users, the Security Group can be updated once, and the changes will be propagated throughout the Workflow steps.

Configure Notifications for the Workflow Step

"Setting Up Communication Paths" on page 203 provides information on setting up Notifications for steps in your process. This includes such things as configuring the Notification's recipients and message.

See the following sections for more details:

- *"Establishing Communication Points and Visibility"* on page 50
- *"Setting Up Communication Paths"* on page 203

Specify the Timeout value for the step

Use the **Timeout** tab on the Workflow Step window to set a timeout for the Workflow Step. Timeout can be set according to either of the following options:

- **Use Workflow Step Source**—This is the default setting. The Workflow Step Source determines the Step's timeout.
- **Specific Value**—Enter a value for the Workflow Step's timeout according to:
    - **Constant**—Enter a numerical value and specify **Minutes**, **Hours**, **Days**, **Weeks**, or **Weekdays**.
    - **Token**—Enter a Token that resolves to a number and specify **Minutes**, **Hours**, **Days**, **Weeks**, or **Weekdays**.

Adding Execution Steps

To add an Execution step to your Workflow:

1. Drag the Step Source onto the **Layout** tab.
2. *Enter the general information on the Execution step*
3. *Specify the Security*
4. *Configure Notifications for the Workflow Step*
5. *Specify the Timeout value for the step*

Figure 6-5 shows the information used when creating an execution step.

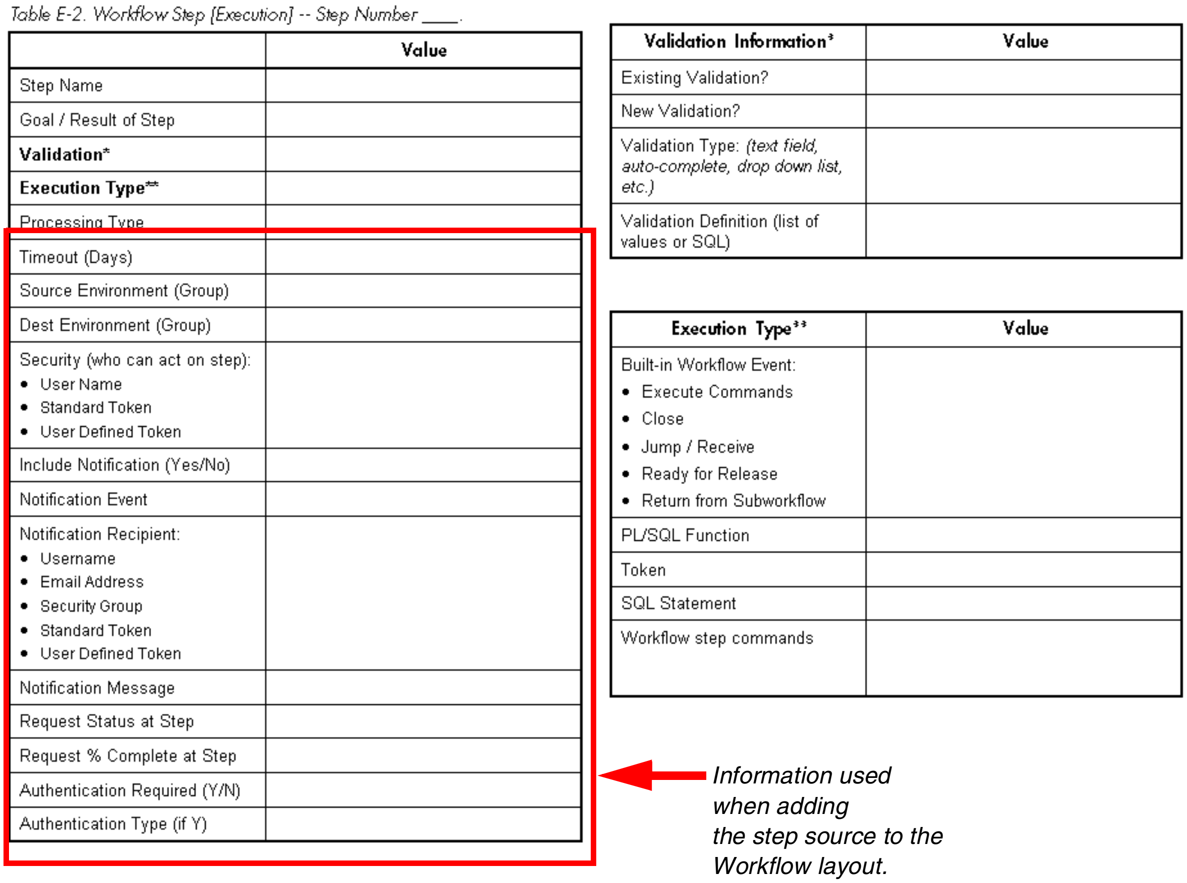


Table E-2. Workflow Step (Execution) -- Step Number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Execution Type**	
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): • User Name • Standard Token • User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: • Username • Email Address • Security Group • Standard Token • User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Validation Information*	Value
Existing Validation?	
New Validation?	
Validation Type: *(text field, auto-complete, drop down list, etc.)*	
Validation Definition (list of values or SQL)	

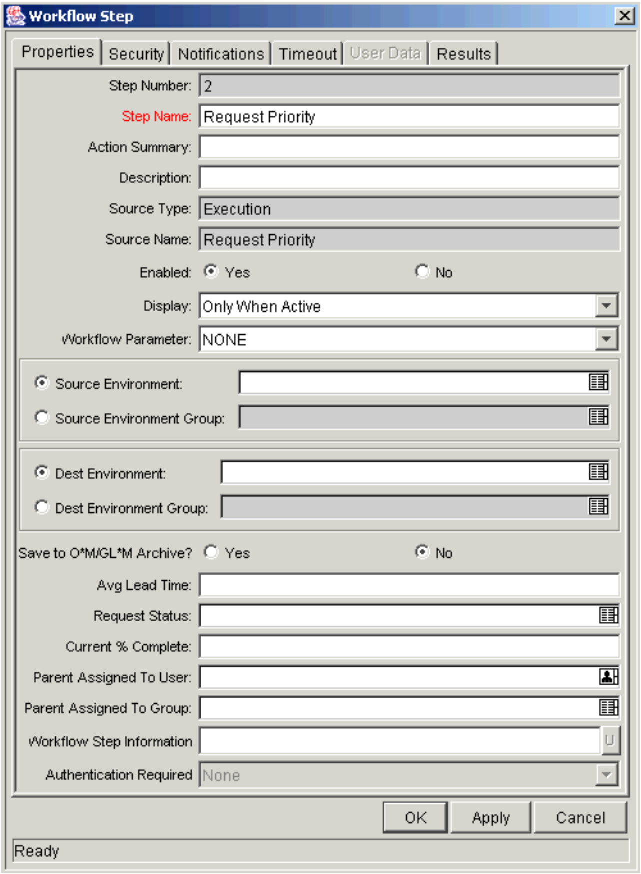
Execution Type**	Value
Built-in Workflow Event: • Execute Commands • Close • Jump / Receive • Ready for Release • Return from Subworkflow	
PL/SQL Function	
Token	
SQL Statement	
Workflow step commands	

Information used when adding the step source to the Workflow layout.

Figure 6-5 Information used to create the execution step.

Enter the general information on the Execution step

Enter the following information in the **Properties** tab in the Workflow Step window.



Step Number: 2
Step Name: Request Priority
Action Summary:
Description:
Source Type: Execution
Source Name: Request Priority
Enabled: ● Yes ○ No
Display: Only When Active
Workflow Parameter: NONE
● Source Environment:
○ Source Environment Group:
● Dest Environment:
○ Dest Environment Group:
Save to O*M/GL*M Archive? ○ Yes ● No
Avg Lead Time:
Request Status:
Current % Complete:
Parent Assigned To User:
Parent Assigned To Group:
Workflow Step Information
Authentication Required None

Table 6-5. Execution Workflow Step window fields

Field on Workflow Step Window	Description
Step Name	This is the name of the step that appears on the Layout tab.
Action Summary	The text that appears on the action button in the Request status panel.
Description	A description of the step.
Enabled	Whether or not the step is enabled.
Display	Whether or not to show the step on the Request status panel.
Workflow Parameter	Used to save the results of a Workflow Step for later use in the Workflow processing.
Source Environment	Specifies the Source Environment where the software that is to be changed is located.

Table 6-5. Execution Workflow Step window fields

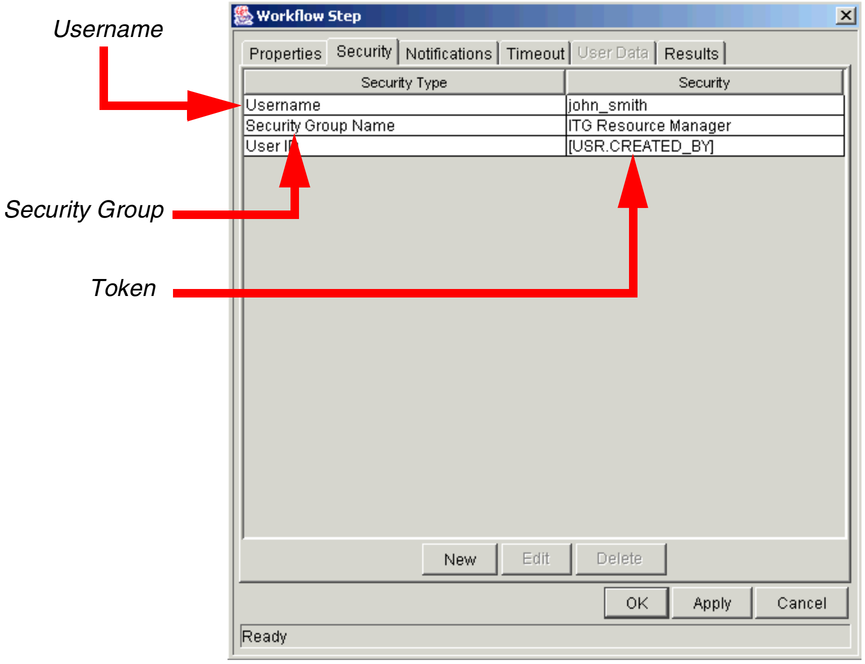
Field on Workflow Step Window	Description
Source Environment Group	Specifies the Source Environment Group which contains the Environment from which the software change is obtained. The Source Environment Group can also be used in conjunction with the Environment Application Codes to provide a dynamic Source Environment selection.
Dest Environment	Specifies the Destination Environment to which the software change is deployed.
Dest Environment Group	Specifies the destination Environment Group. The destination consists of multiple Environments to which the software change is deployed.
Avg Lead Time	A user-specified metric for comparing actual performance to estimated goals. It does not affect any transactional logic.
Request Status	The Status acquired by the Request when it reaches this step.
Current % Complete	The percentage of the resolution process that is complete at this step. This value rolls up into a Mercury Project Management Project when a Task has been created from a Request.
Parent Assigned to User	If this field is not empty when the step becomes Eligible, the Assigned to User of the Request automatically changes to the user specified in the field.
Parent Assigned to Group	If this field is not empty when the step becomes Eligible, the Assigned to Group of the Request automatically changes to the Security Group specified in the field.
Workflow Step Information	A text entry field in which any URL can be entered. This is where users can find documents, instructions or comments to aid them in working the Workflow Step.
Authentication Required	Specifies how to authenticate the user when they act on the Step. Username and password may be prompted according to this setting.

Specify the Security

"Integrating Participants into Your Request Resolution System" on page 177 provides information on setting up security for a Request resolution process. This includes such things as controlling who can create Requests and who can act on specific steps in the process. Security related directly to processing a Workflow step is configured in the Workflow Step window.

Define who can act on the step by:

- Security Group
- Username
- Token (standard or user-defined)



Security Type	Security
Username	john_smith
Security Group Name	ITG Resource Manager
User ID	[USR.CREATED_BY]

Tip

Consider using Security Groups or dynamic access (Tokens) when defining the Workflow step security. Avoid specifying a list of users to control an action; for example, specifying a list of users who can act on a Workflow step. If the list of users changes (due to an organizational reorganization), that list would need to be updated in many places on the Workflow. By using a Security Group instead of a list of users, the Security Group can be updated once, and the changes will be propagated throughout the Workflow steps.

Configure Notifications for the Workflow Step

"Setting Up Communication Paths" on page 203 provides information on setting up Notifications for steps in your process. This includes such things as configuring the Notification's recipients and message. You can configure Notifications for specific steps.

See the following sections for more details:

- *"Establishing Communication Points and Visibility"* on page 50
- *"Setting Up Communication Paths"* on page 203

Specify the Timeout value for the step

Use the **Timeout** tab on the Workflow Step window to set a timeout for the Workflow Step. Timeout can be set according to either of the following options:

- **Use Workflow Step Source** — This is the default setting. The Workflow Step Source determines the Step's timeout.
- **Specific Value** — You can enter a value for the Workflow Step's timeout according to:
  - **Constant** — Enter a numerical value and specify **Minutes**, **Hours**, **Days**, **Weeks**, or **Weekdays**.
  - **Token** — Enter a Token that resolves to a number and specify **Minutes**, **Hours**, **Days**, **Weeks**, or **Weekdays**.

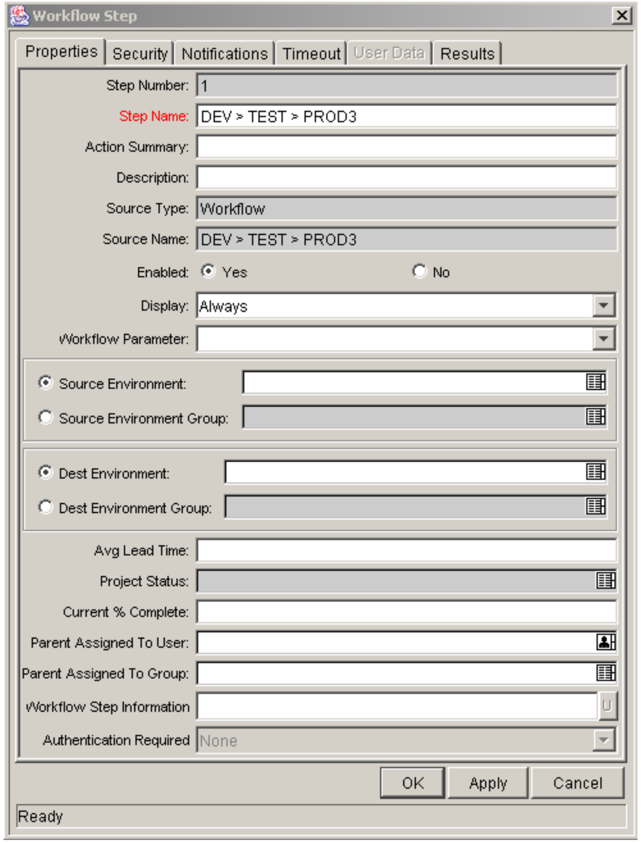
Adding a Subworkflow

A Subworkflow can be selected from the Workflow Step Sources window and dragged onto the **Layout** tab. When the Request reaches the Subworkflow Step, it follows the path defined in that Subworkflow. The Subworkflow will either close within that Workflow or return to the parent Workflow.

To add an enabled Subworkflow to another Workflow:

1. Select the desired Subworkflow and drag it to the **Layout** tab.

   The Workflow Step window opens. This window contains preconfigured information which is specific to the selected Workflow Step.



Step Number: 1
Step Name: DEV > TEST > PROD3
Action Summary:
Description:
Source Type: Workflow
Source Name: DEV > TEST > PROD3
Enabled: ● Yes ○ No
Display: Always
Workflow Parameter:
● Source Environment:
○ Source Environment Group:
● Dest Environment:
○ Dest Environment Group:
Avg Lead Time:
Project Status:
Current % Complete:
Parent Assigned To User:
Parent Assigned To Group:
Workflow Step Information
Authentication Required None

2. Configure this step as if configuring an execution or decision step.

3. Click **OK**.

See *"Advanced Workflow Topics"* on page 251 for a detailed discussion of using Subworkflows.

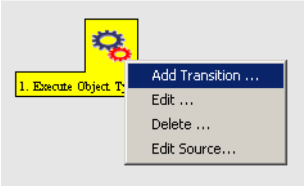
Adding Transitions Between Steps

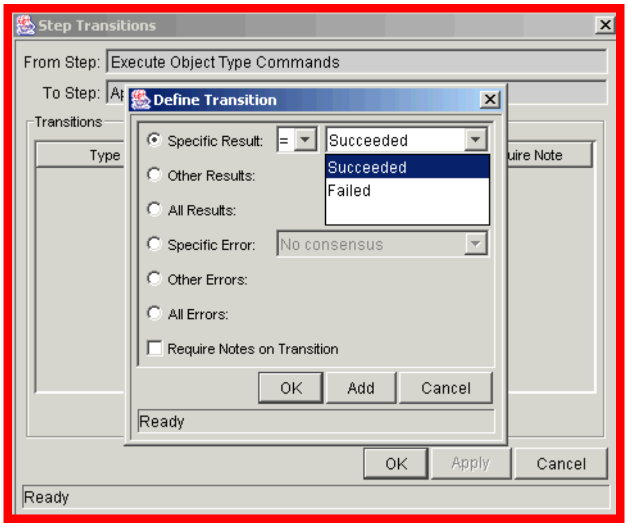
After adding the steps to the Workflow **Layout** tab, configure the transitions between them. Choose to transition between steps based on the following step results:

- **Specific Result** (based on the Validation configured in the step source)
- **Other Results**
- **All Results**
- **Specific Error**

- **Other Errors**
- **All Errors**

Select the proper transition between steps. The following section provides some example scenarios and transition configuration options:





From Step: Execute Object Type Commands

To Step: A

● Specific Result: = Succeeded

Succeeded
Failed

○ Other Results:

○ All Results:

○ Specific Error: No consensus

○ Other Errors:

○ All Errors:

□ Require Notes on Transition

Ready

Ready

Transition based on a specific result

Transitioning based on the result of a specific decision or execution allows the business process to branch based on anticipated results of a step in the Workflow.

To transition based on a specific Workflow step result:

1. Add a transition between two steps.

   a. Right-click on a step and select **Add Transition.**

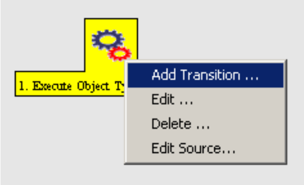
   b. Connect the arrow to the appropriate target step.

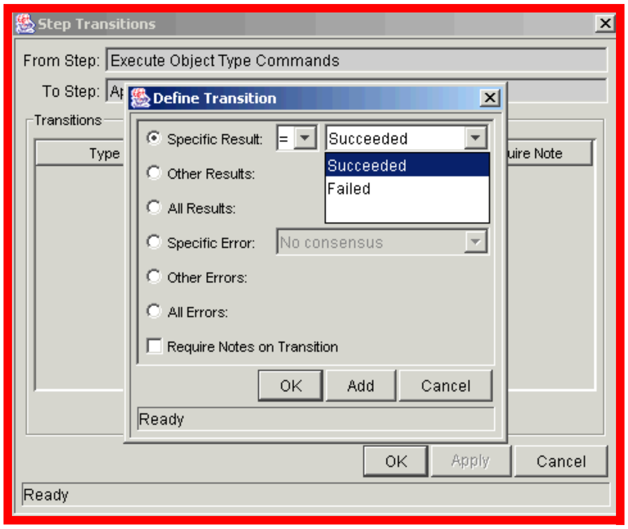
   The Define Transition window opens.

2. Select the Specific Result radio button.

3. Select the desired result from the drop down list.

   The values in this list will vary depending on the Validation set in the Workflow step source for the transitioning step.





From Step: Execute Object Type Commands

To Step: Ap

● Specific Result: = Succeeded

○ Other Results:

○ All Results:

○ Specific Error: No consensus

○ Other Errors:

○ All Errors:

□ Require Notes on Transition

Ready

Ready

Transition based on a value in a field

It is possible to transition a Request based on the value in a particular field. This can be a general field in the Request Header (such as Priority, Assigned To, or Request Group) or a custom field specified in the Request (defined on the Request Type). For example, if the Request's Priority field is set to **Critical**, then you may want the Request to follow a different, more robust process. This is done by resolving a field Token in a Workflow execution step. The Workflow engine evaluates the field's value at a specific step and then can route the Request accordingly.

To transition based on the value in a field:

1. Add an immediate execution step source to the Workflow.

   It may be necessary to create a custom Workflow step source for this operation. The step source should be configured as shown in the following table:

Field in Execution Window	Value
Workflow Scope	**Requests**
Execution Type	**Token**
Processing Type	**Immediate**
Validation	Select or create a Validation that includes all of the possible values of the resolved Token. For example, if you plan on branching based on the Priority field, use the **[REQ.PRIORITY_CODE]** Token and the **CRT - Priority - Enabled** Validation. The Validation contains all possible values of the Token.
Execution	Enter the Token for the value that you would like to transition based on.
Enabled	**Yes**

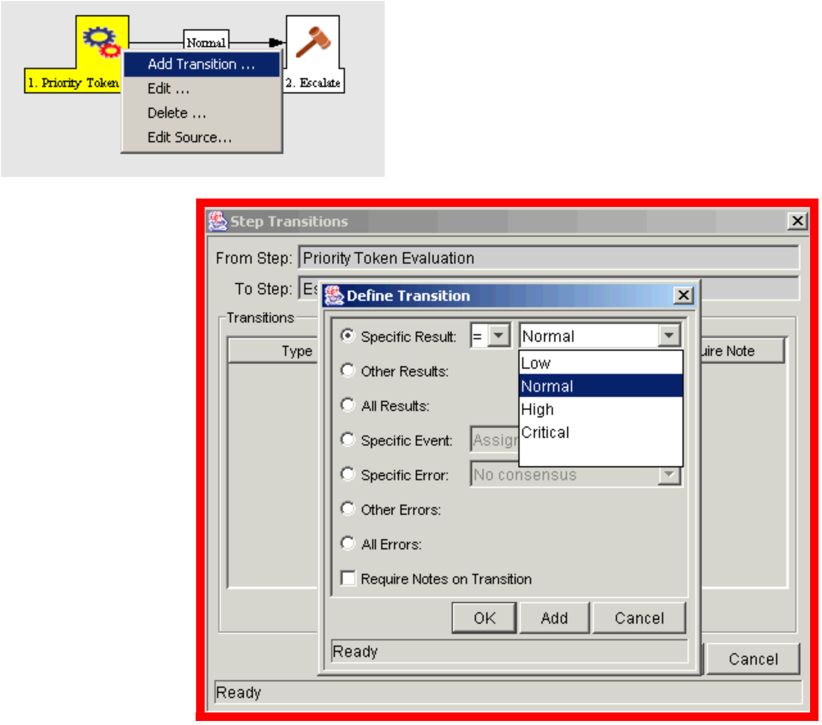
2. Add transition between two steps.

   The Define Transition window opens.

3. Select the Specific Result radio button.

4. Select the desired result from the drop down list.

   The values in this list will vary depending on the Validation set in the Workflow step source for the transitioning step. For the previous Priority example, the possible values will be the values allowed in the Request's Priority field.



From Step: Priority Token Evaluation

To Step: Es

● Specific Result: = Normal

○ Other Results:

○ All Results:

○ Specific Event: Assign

○ Specific Error: No consensus

○ Other Errors:

○ All Errors:

□ Require Notes on Transition

Ready

Ready

Figure 6-6 Example: Transitioning based on value in a field (Token)

Transition based on data in a table

It is possible to transition based on information stored in a table. To transition using this method, use a Workflow execution step with an execution type of SQL. For more information, see *"Execute a SQL statement and then transition based on the result"* on page 78.

When transitioning from a properly configured execution step (Execution Type = **SQL Statement**), transition based on a Specific Result. The possible results are defined in the Workflow step source's Validation. The values in this list are determined by a SQL query of a database table.

As with any execution step, configure this transition to be an immediate or manual step.

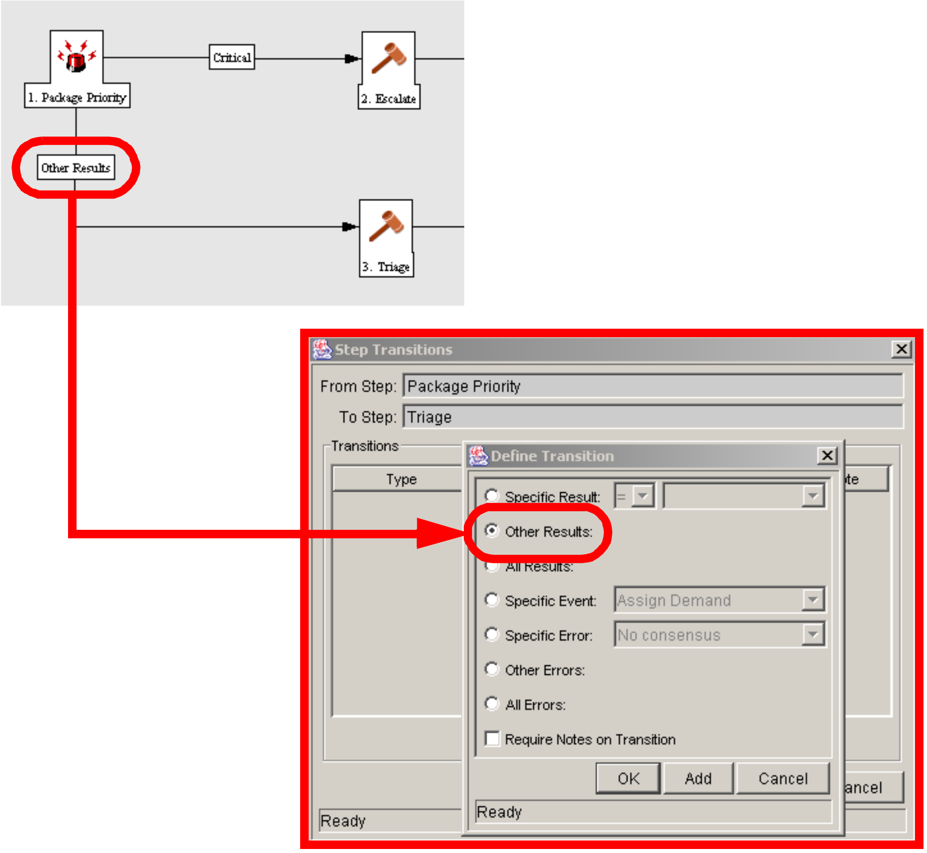
Transition based on all but one specific value

It is possible to transition based on all but one specified value. For example, you may want to transitional all "Critical" Requests one way and all other results another.

To transition based on all but one specific value:

1. Create a transition from a step based on a specific result.

2. Create another transition from the same step and specify **Other Results**.



From Step: Package Priority

To Step: Triage

○ Specific Result: =

● Other Results:

○ All Results:

○ Specific Event: Assign Demand

○ Specific Error: No consensus

○ Other Errors:

○ All Errors:

□ Require Notes on Transition

Ready

In the above example, only Requests with a "Critical" priority will follow the Escalate path. All other results are sent to Triage.

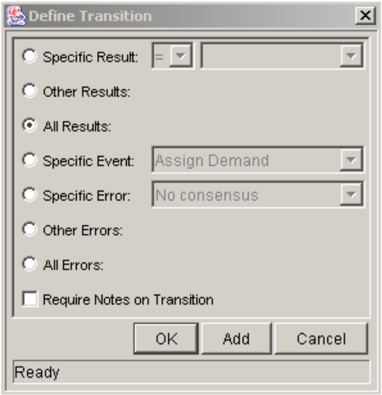
Tip

You can use Other Results when multiple transitions are exiting a single step. Other Results will act as the transition if none of the other explicit transition conditions are satisfied.

Transition based on all results

It is possible to define a Request to transition regardless of the step's actual results. For example, you may want to run a Subworkflow to perform server maintenance after the on-call server contact is identified. To do this, add a transition from the Specify Contact step to the Subworkflow. Since the next step in the process does not depend on the result of the step, it is appropriate to use the **All Results** transition.

To do this, define a transition from the step and select **All Results**. The Define Transitions window is shown below.



Define Transition

○ Specific Result: =
○ Other Results:
● All Results:
○ Specific Event: Assign Demand
○ Specific Error: No consensus
○ Other Errors:
○ All Errors:
□ Require Notes on Transition

Tip

Consider using an **All Results** transition when kicking off a sub-process. Note that you can still define transitions based on Specific Results or errors when you select **All Results**. You can bring the process together later using an AND step.

Transition based on error

It is possible to transition based on a specific error that occurs during an execution step. The business process may then be branched based on likely execution errors such as **Timeout**, **Command Execution** or **Invalid Token**.

To transition based on a specific Workflow step error:

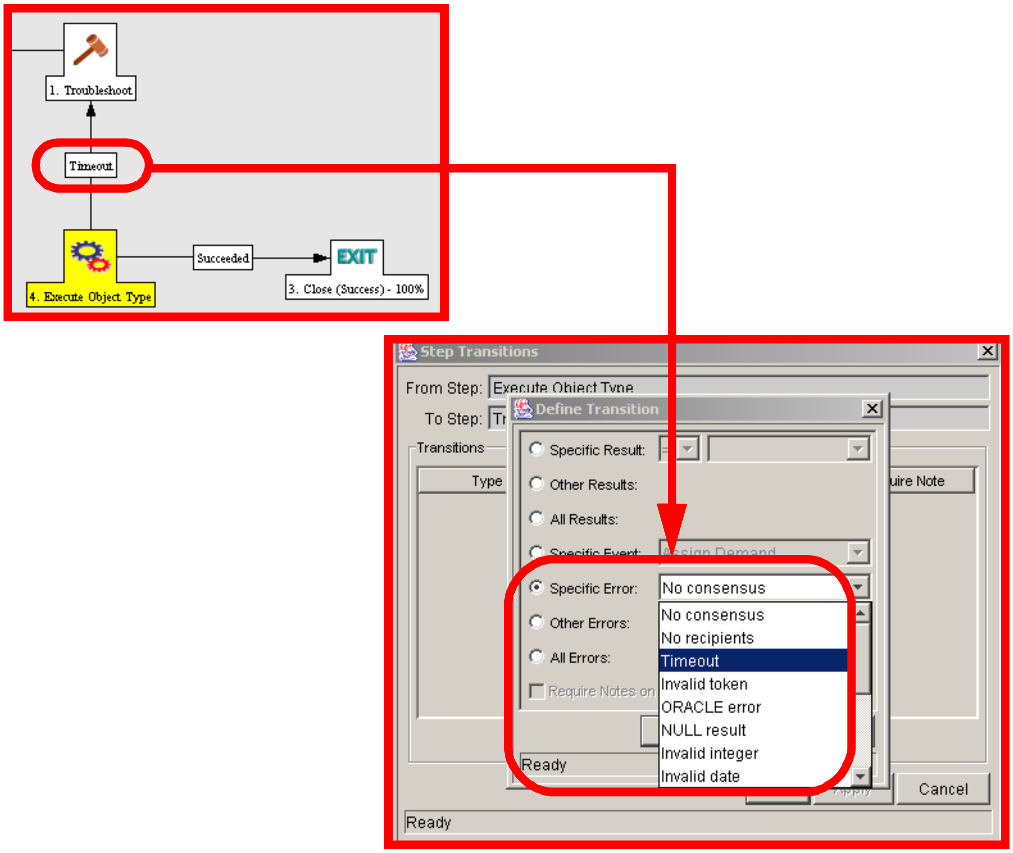
1. Add transition between two steps.

   The Define Transition window opens.

2. Select the Specific Error radio button.

3. Select the error from the drop down list.

   All values in this list are defined in *Table 6-6*.



From Step: Execute Object Type

To Step: Tr

● Specific Error: No consensus

No consensus
No recipients
Timeout
Invalid token
ORACLE error
NULL result
Invalid integer
Invalid date

Table 6-6. Workflow Transition Errors

Transition Option	Meaning
Multiple Return Results	When the Package Level Subworkflow receives multiple results from Package Lines that traversed through it.
No consensus	When all users of all Security Groups, or users linked to the Workflow Step need to vote, and there is no consensus.
No recipients	When none of the Security Groups linked to the Workflow Step has users linked to it, no user can act on the Workflow Step.
Timeout	When the Workflow Step times out. Used for Executions and Decisions.
Invalid token	Invalid Token used in the execution.
ORACLE error	Failed PL/SQL Execution.
NULL result	No result is returned from the execution.
Invalid integer	Validation includes an invalid value in the Integer field.
Invalid date	Validation includes an invalid value in the Date field.

Table 6-6. Workflow Transition Errors

Transition Option	Meaning
Command execution error	Execution engine has failed or has a problem.
Invalid Result	Execution or Subworkflow has returned a result not included in the Validation.
Parent closed	For wf_receive or wf_jump steps, a Package Line is expecting a message from a Request that is cancelled or closed.
Child closed	For wf_receive or wf_jump steps, a Request is expecting a message from a Package Line that is cancelled or closed.
No parent	For wf_receive or wf_jump steps, a Package Line is expecting a message from a Request that has been deleted.
No child	For wf_receive or wf_jump steps, a Request is expecting a message from a Package Line that has been deleted.
Multiple jump results	For wf_jump steps in a Package Line, different result values were used to transition to the step.

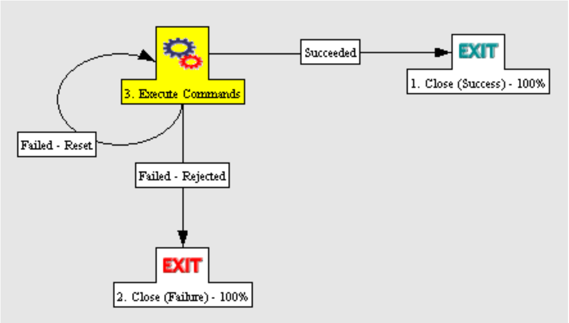
Transition back to the same step

It is possible to retain the option of resetting failed execution steps, rather than immediately transitioning along a “failed” path. This is often helpful when troubleshooting the execution.

To transition back to the same execution step:

1. Create an execution step source to execute the Workflow or Request Type commands.
2. Create a Validation with the following Validation Values.
    a. SUCCEEDED
    b. FAILED
    c. FAILED - RESET
    d. FAILED - REJECTED
3. Add the step to the Workflow **Layout** tab.

4. Add transitions based on the following Specific Results:
   a. SUCCEEDED
   b. FAILED - RESET -- set the transition to return back into the same step.
   c. FAILED - REJECTED



When the commands execute successfully, they will follow the Success transition path. However, when the commands fail, they will not transition out of the step because no transition has been defined for the **FAILED** result. The user has manually select the Workflow step and select **FAILED - RETRY**. The execution will re-run.

Note

Be careful when using an immediate execution step when the **FAILED** result is not feeding directly back into the execution step. This would result in a continual execution-failure loop.

Transition based on a previous Workflow Step result (parameters)

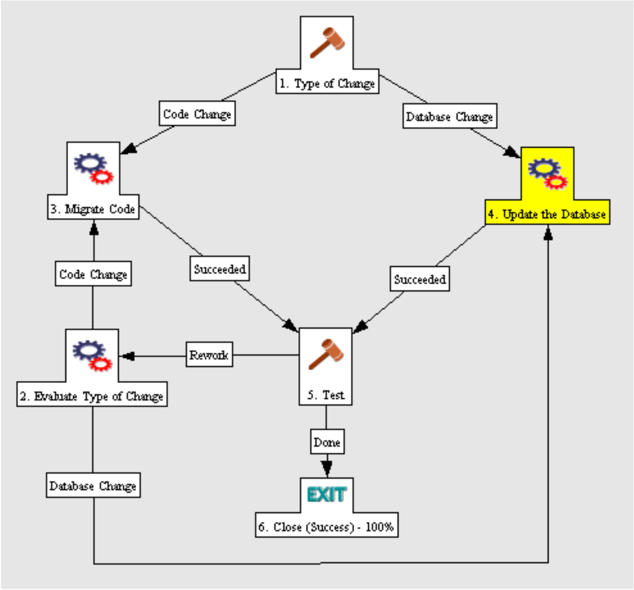
It is possible to use Workflow parameters to store the result of a Workflow Step. This value can then be used later to define a transition.

To create a transition based on a Workflow parameter:

1. Create a Workflow Parameter in the Workflow window.
2. Specify that Workflow Parameter in a Workflow step on the **Layout** tab.
3. Create a Token execution step that will resolve the value in the Workflow parameter.

Example: Using a Workflow Parameter to Transition

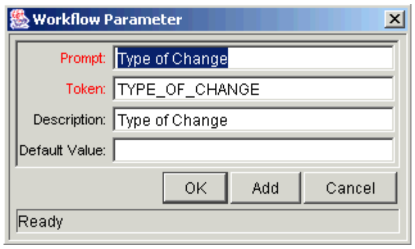
One step in this example process requires the user to route the Request based on the type of change (code or database). The decision made at this step is then considered later in the process to correctly route rework of the specific type.



To enable this process, set the following in the Workflow:

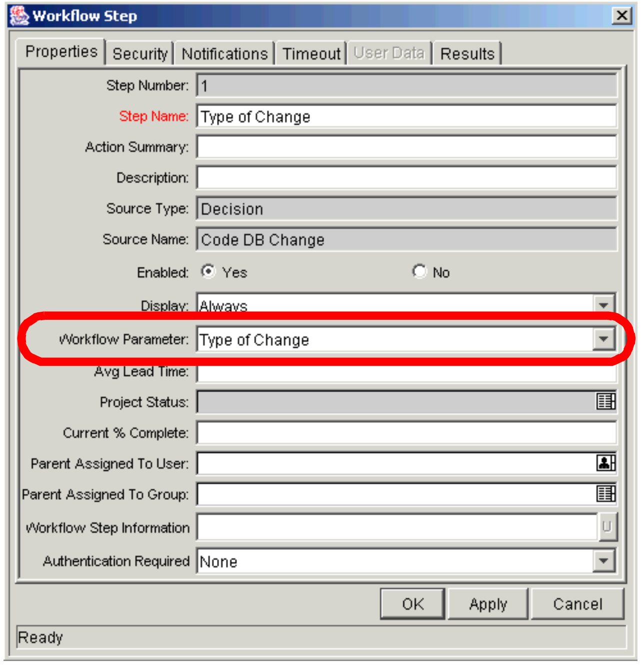
1. Create a Workflow Parameter in the Workflow window.

   This is done by clicking **Add** in the **Workflow** tab on the Workflow window.



Prompt:	Type of Change
Token:	TYPE_OF_CHANGE
Description:	Type of Change
Default Value:	

2. Select **Type of Change** for the Workflow Parameter in Type of Change Workflow step on the **Layout** tab.

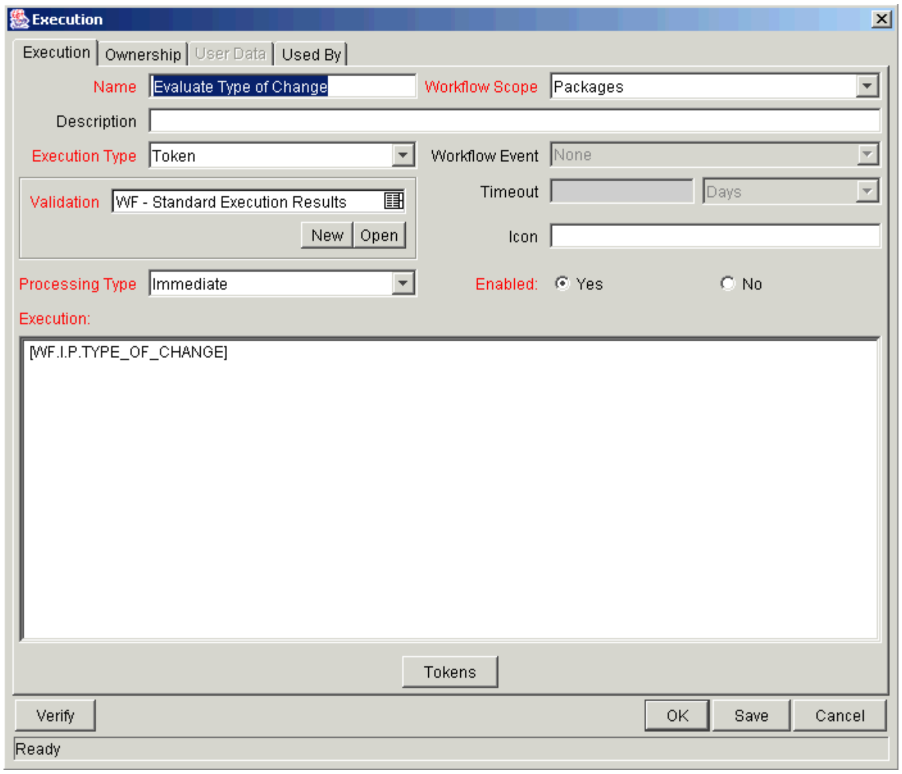


Step Number: 1
Step Name: Type of Change
Action Summary:
Description:
Source Type: Decision
Source Name: Code DB Change
Enabled: ● Yes ○ No
Display: Always
Workflow Parameter: Type of Change
Avg Lead Time:
Project Status:
Current % Complete:
Parent Assigned To User:
Parent Assigned To Group:
Workflow Step Information
Authentication Required None

Ready

3. Create a Token execution step that will resolve the value in the Workflow parameter.

   Note that the Validation used in this step should contain the same values as the Validation specified in the Type of Change decision step.



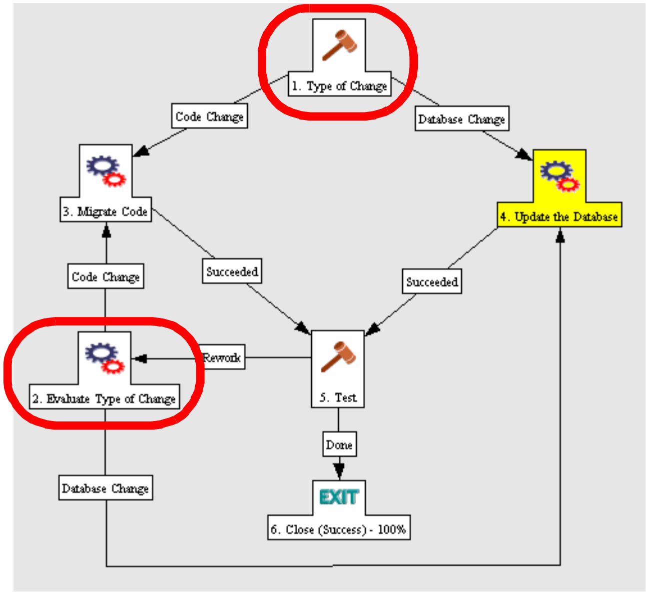
Execution

Execution | Ownership | User Data | Used By

Name: Evaluate Type of Change
Workflow Scope: Packages
Description:
Execution Type: Token
Workflow Event: None
Validation: WF - Standard Execution Results
Timeout: Days
Icon:
Processing Type: Immediate
Enabled: ● Yes ○ No
Execution:

```
[WF.I.P.TYPE_OF_CHANGE]
```

4. Add the steps and transitions as shown below.



Transition to and from Subworkflows

There are special configuration requirements when transitioning to and from Subworkflows. For detailed instructions, see *"Advanced Workflow Topics"* on page 251.

Transition to and from a Package Workflow

There are special configuration requirements when transitioning to and from a Package Workflow using Jump and Receive steps in the Workflows. For detailed instructions, see *"Advanced Workflow Topics"* on page 251.

Chapter 7 Constructing the Request Type

This chapter provides an overview for how to configure the Request Type that will be used to process Requests through a Workflow. This includes configuring Request Type and Request Header Type fields, field behavior, and Commands.

This chapter covers the following topics:

Creating a Request Type - Overview

Request Types are created and configured using the Workbench.

To create a new Request Type:

1. From the menu bar, select **Administration > Open Workbench**.

2. Click the **Request Mgmt** screen group on the Workbench and click the **Request Types** icon.

   The Request Type Workbench window opens.

3. Click **New Request Type**.

   The Request Type window opens.

4. Enter the Request Type general information.

   This includes the fields defined in the following table:

Field	Description
Creation Action Name	A description of the Request Type's function. An example of a Creation Action Name is **Log a Product Bug**. Creation Action Names display on the Create New Request page.
Category	The category containing the Request Type. Categories are created by an application administrator and are based on the business needs of the organization. Examples of categories which an organization might use are Sales and Support and General Administration. Categories display on the Create New Request window in the standard interface. [Validation = CRT - Request Type Category]

5. Select a Request Header Type to be used with this Request Type.

   Either select an existing Request Header Type from the auto-complete list, or create a new Request Header Type by clicking **New**. For more detailed information, see *"Choosing a Request Header Type"* on page 114.

6. Create fields that describe the Request.

   This includes configuring:

   - Field names.
   - Validations and component types (dictated by the Validation).
   - Field Behaviors: whether fields are displayed or have any defaulting behavior. Other aspects of field behavior should be taken into account later, and are discussed below.

See *“Request Type Field Validations”* on page 124.

7. Configure the Field layout.

   This will determine how the fields are positioned on the Request. See *“Modifying the Request Layout”* on page 169.

8. Determine which fields require more intricate defaulting behaviors and configure them in the **Rules** tab of the Request Type window.

   For example, configure a set of fields to automatically populate based on a change to another field in the Request. See *“Configuring Request Type Defaulting Behavior (Rules)”* on page 145 for more detailed information.

9. Link or create the Statuses the Request will take on as it moves through its Workflow.

   Request Statuses are linked to Workflow Steps. Set Status Dependencies on fields to define additional field behavior. See *“Creating Request Statuses”* on page 157 for more detailed information.

10. Open the Workflow to be used by the Request Type and link the proper Request Statuses to the appropriate Workflow Steps.

    See *“Assigning Request Statuses to Workflow Steps”* on page 164 for more detailed information.

11. Create the Request Type’s commands, if any have been determined necessary.

    Request Type commands can be useful for carrying out complex operations on fields. See *“Defaulting”* on page 132 for an example.

12. Set Ownership for the Request Type.

    This controls who can modify or delete the Request Type. See *Security Model Guide and Reference* for details.

13. Set up any desired Notifications for Request field changes.

    See *“Setting Notifications on Request Field Changes”* on page 228 for more details.

14. Configure Request, Request Section, or Request Field Help for the Request Type.

It is possible to provide accessible online information for users who are processing the Requests. Configure the Request type to display additional, custom information about the Request, sections or fields. See *"Creating Custom Help for Requests"* on page 238 for additional details.

Tip

It is often advantageous to use the **Copy** functionality to copy an existing Request Type and then edit the new copy. To reduce the amount of editing required, choose an existing Request Type similar to the Request Type to be generated.

Note

Only users with the appropriate security can create or edit Request Types. To edit Request Types, you must belong to a Security Group that has the Access Grant Demand Mgmt: Edit Request Types. See the *"Integrating Participants into Your Request Resolution System"* on page 177 for more information.

Choosing a Request Header Type

Request Header Types define the collection of fields that appear in the header region of the Requests. Request Header Types typically include more general information that will be tracked between multiple types of Request. This can include such information as who logged the Request, its priority, and a description of the issue.

Request Header Types contain a set of standard pre-defined fields that can be enabled or disabled. Request Header Types can also contain custom fields.

Each Request Type must include a Request Header Type. A single Request Header Type can be used for multiple Request Types.

Table 7-1 lists the Request Header Types delivered with a base installation of Mercury Demand Management.

Table 7-1. System Request Header Types

System Header Type	Description
(REFERENCE) Default	The default Request Header Type. Includes a % Complete field.

Table 7-1. System Request Header Types

System Header Type	Description
(REFERENCE) Comprehensive	Displays all information. Consistent with previous versions of Mercury Demand Management.
(REFERENCE) Simple	Displays only the most essential information.
(REFERENCE) Departmental	An example Header Type for simple cross-departmental Requests.
(REFERENCE) Application	An example Header Type for simple cross-application Requests.
(REFERENCE) Help Desk	An example Header Type for help desk Requests, including contact and assignment information.

Note

When Field Groups are associated with existing Request Types (through the Request Header Type definition), Mercury ITG database tables are updated to handle this new configuration. Because of the scope of database changes, the Database Statistics should be re-run on your database. Instructions for this are included in the System Administration Guide. Contact the System Administrator for help with this procedure.

Setting a Default Request Header Type

It is possible to select a Request Header Type to default when a user selects a specific Request Type. It is also possible specify the default Workflow that is selected for all Requests.

To set the default Request Header Type:

1. Open the Request Type Workbench window.
2. Click the **Results** tab.
3. Click **Setup Request Header**.

   The Request Header Setup Dialog window opens.
4. Select information as described in the following table.

Field	Description
Default Workflow	Select a default Workflow. This default Workflow is used for all new Requests, unless the associated Request Type has a defaulting rule for the Workflow.
Default Request Header Type	Select a default Request Header Type. This Request Header Type will be used for all new Request Types, unless a different Request Header Type is specified in the individual Request Type windows.

5. Click **OK**.

Creating a New Request Header Type

To create a new Request Header Type:

1. Click the **Demand Mgmt** screen group on the Workbench and click the **Request Header Types** icon.

   The Request Header Type Workbench window opens.

2. Click **New Request Header Type**.

   The Request Header Type window opens.

3. Fill in any general information for the Request Header Type.

4. Modify the existing fields in the Request Header Type as appropriate.

   This includes configuring field security. See *"Modifying Existing Request Header Type Fields"* on page 118 for more information.

5. (Optional) Create new Request Header Type fields.

   For more information, see *"Creating a New Request Header Type Field"* on page 121.

6. In the **Filter** tab, select the options as described in the following table.

Field	Description
This section of the Contact Name field is limited by:	• All Contacts: Limit the number of contact names seen in the Contact Name field when creating or updating a Request Header Type by selecting one of the Contact Name options available in the **Filter** tab. Selecting this option will display all users with no restrictions on the list of contact names. • The Company field of the Request: Users can limit the number of contact names they would see in the Contact Name field when creating or updating a Request Header Type by selecting one of the 'Contact Name' options available in the Filter tab. Selecting this option will restrict the list of contact names the user would see to those found in the Company field of the Request.
This section of the Assigned Group Field is limited by:	• Only Security Groups with the Request option enabled: Users can limit the number of group names they would see when creating or updating a Request Header Type by selecting one of two 'Assigned Group' options available in the Filter tab. Selecting this option will restrict the list of group names the user would see to only those Security Groups where the Request option is enabled. • Participants only: Users can limit the number of group names they would see when creating or updating a Request Header Type by selecting one of two 'Assigned Group' options available in the **Filter** tab. Selecting this option will restrict the list of group names the user would see to participants in the Request.
This section of the Assigned To field is limited by:	• Only users who are in Security Groups with the Request option enabled: Limit the number of user names seen in the Assigned To field when creating or updating a Request Header Type by selecting one of two Assigned To options available in the **Filter** tab. Selecting this option restricts the list of user names the user would see to only those Security Groups where the Request option is enabled. • Participants only: Users can limit the number of user names they would see in the Assigned To field when creating or updating a Request Header Type by selecting one of two Assigned To options available in the **Filter** tab. Selecting this option restricts the list of user names the user would see to Participants of the Request. In this instance, Participants are defined as: the Assigned User, the creator of the Request, Members of the Assigned Group, or Members of the Workflow.

7. To save the Request Header Type, click **Save**.

8. Open the Request Type to associate with the Request Header Type.

9. In the Request Header Type field, select the Request Header Type.

10. In the Request Type window, click **Save**.

Modifying Existing Request Header Type Fields

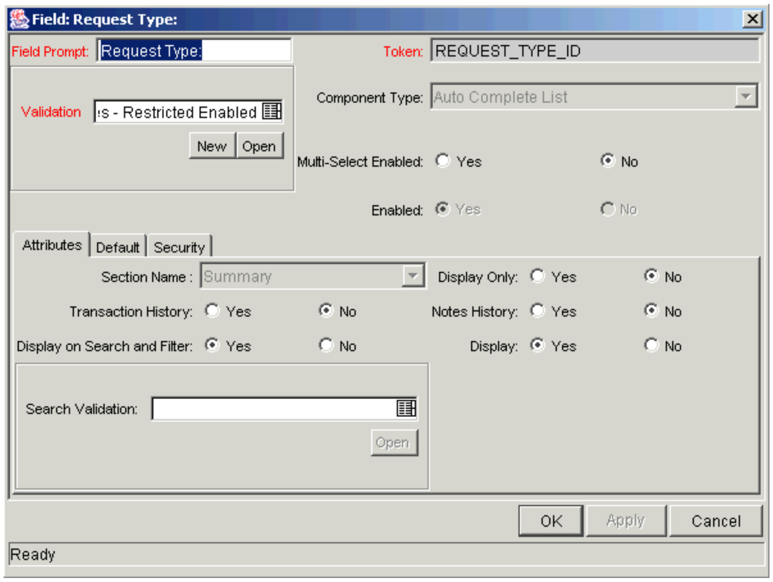
Request Header Type fields are modified similarly to the way Request Type fields are. There are, however, a few notable differences specifically when referring to the standard fields that are delivered with all Request Header Types. The differences include:

- The Validation for an existing system-provided Request Header Type field cannot be changed. A different Validation can be specified when creating a new Request Header Type field or copying a Request Header Type field.
- Request Header Type fields can be used with many Request Types simultaneously. This may necessitate specifying a different Validation to be used in Search/Filter functionality. The Search Validation can be specified from the Request Header Type Field window's **Attributes** tab.

To modify a Request Header Type field:

1. In the Request Header Type window, select a field.
2. Click **Edit**.

   The Field window opens.



Field: Request Type:

Field Prompt: Request Type:

Token: REQUEST_TYPE_ID

Validation: ·s - Restricted Enabled

Component Type: Auto Complete List

Multi-Select Enabled: ○ Yes ● No

Enabled: ● Yes ○ No

Attributes | Default | Security

Section Name : Summary

Display Only: ○ Yes ● No

Transaction History: ○ Yes ● No

Notes History: ○ Yes ● No

Display on Search and Filter: ● Yes ○ No

Display: ● Yes ○ No

Search Validation:

Ready

3. Modify the field by altering the selections in the Field window.

The Field window controls the behavior of the Request Header Type field. For discussion of each field, as well as what can and cannot be altered on existing system-provided Request Header Type fields, see *Table 7-2*.

Table 7-2. Request Header Type Field Window - General Information Region

Field	Behavior
Field Prompt	The prompt visible for the Request Header Type field on the Request.
Token	An uppercase text string used to identify this field. The Token name must be unique for the specific Request Header Type. An example of a Token name is ASSIGNED_TO_USER_ID. On existing Request Header Type fields, this field cannot be edited.
Validation	Indicates the Validation logic to determine the valid values for this field. This could be a list of user-defined values, a rule that the result has to be a number, etc. See *"Determining the Field Type (Selecting a Validation)"* on page 125 for more details.
Component Type	Defines the visual characteristics of the field (drop down list, free form text field, etc.). This is derived from the Validation chosen. This field cannot be edited.
Multiselect	Determines whether or not the field allows users to select more than one entry. Only valid for fields with an auto-complete component for the Validation.
ENABLED	Determines whether or not the field is turned on for this Request Header Type.

Table 7-3. Request Header Type Field Window - Attributes Tab

Field	Behavior
Section Name	The area of the Request on which the field is displayed.
Display Only	Determines if the field is only displayed and cannot be updated, even at initial Request entry.
Transaction History	Turns transaction auditing on or off for this field. If the field is set to **Yes**, whenever it field changes in a Request, the change is logged in a transaction history table.
Notes History	Turns Notes auditing on or off for this field. If the field is set to **Yes**, whenever it changes in a Request, the change is logged in Notes for the Request.
Display on Search and Filter	Determines whether or not the field will be displayed in Search and Filter pages in the standard interface.

Table 7-3. Request Header Type Field Window - Attributes Tab

Field	Behavior
Display	Determines whether or not the field is seen by Requests that use the given Request Header Type. If Disabled, the Request Header Type field will no longer be displayed.
SEARCH VALIDATION	Useful for fields that are used in multiple Request Header Types. Allows you to specify a Validation to be used as the Search/Filter Validation across all Request Header Types.

Table 7-4. Request Header Type Field Window - Default Tab

Field	Behavior
Default Type	Defines whether or not the field will have a default value. Either default the field with a constant value (**Default from Constant**) or default it from the value in another field (**Default from Parameter**).
Visible Value	Defines the constant value that defaults in the field. Enabled only when the Default Type of **Default from Constant** is selected.

Table 7-5. Request Header Type Field Window - Storage Tab (for new fields)

Field	Behavior
Max Length	Determines the maximum field character length. The two possible values are 200 and 1800.
Batch Number	Based on the number of maximum fields. For every 50 fields, one batch is created. 10 of these 50 fields can be more than 200 characters in length. Enabled only when there are more than 50 fields (creating more than one batch).
Parameter Col	Determines the internal database column that the field value is stored in. These values are then stored in the corresponding column in the Request Details table for each batch of the given Request Header Type. Information can be stored in up to 50 columns using Request Header Types, allowing up to 50 fields/batch. No two fields in a Request Header Type can use the same column number within the same batch.

Table 7-6. Request Header Type Field Window - Security Tab

Field	Description
Visible To	Lists all users, Security Groups, and linked Tokens for which this field will be visible.
Editable By	Lists all users, Security Groups, and linked Tokens for which this field will be editable.
Edit	Opens the Edit Field Security window, which configures the users, Security Groups, and linked Tokens that will be able to view and/or edit this field. See *"Creating a Field"* on page 133 for more detailed information.

4. Click **OK** to save your changes and close the Field window.

Creating a New Request Header Type Field

If none of the existing Request Header Type fields match your needs, custom fields can be created. New Request Header Type fields can be created and modified in much the same way as Request Type fields. See *"Request Type Field Validations"* on page 124 for more detailed instructions.

Copying a Request Header Type

To copy a Request Header Type:

1. Click the **Demand Mgmt** screen group on the Workbench and click the **Request Header Types** icon.

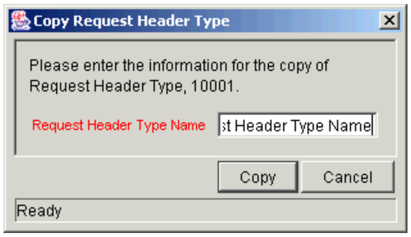
   The Request Header Type Workbench window opens.

2. Enter any necessary search criteria and click **List**.

   The **Results** tab displays the results of your search.

3. Select the desired Request Header Type and click **Copy**.

   The Copy Request Header Type window opens.


Copy Request Header Type

Please enter the information for the copy of Request Header Type, 10001.

Request Header Type Name: st Header Type Name

Ready

4. Enter a new Request Header Type Name.

5. Click **Copy**.

   A question dialog opens, prompting whether you would like to edit the new Request Header Type.

6. Click **Yes**.

   The Request Header Type window opens.

7. Fill in any general information for the Request Header Type.

8. Modify the existing fields in the Request Header Type as needed.

   For more information, see *"Modifying Existing Request Header Type Fields"* on page 118.

9. (optional) Create any additional new Request Header Type fields.

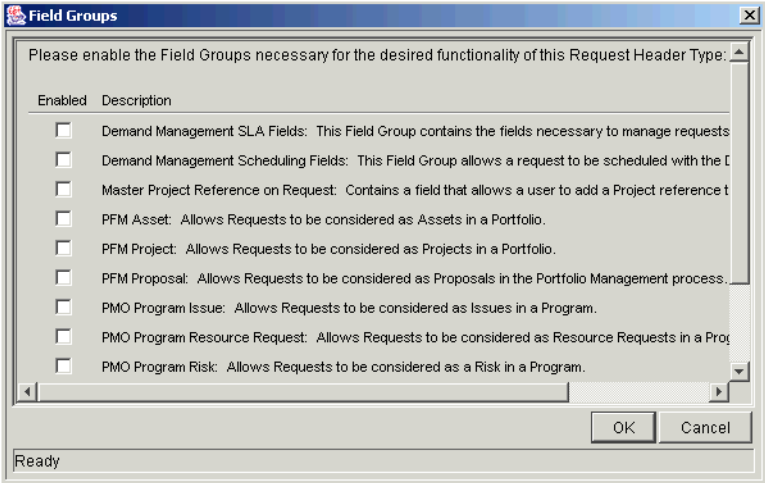
   For more information, see *"Creating a New Request Header Type Field"* on page 121.

10. Click **Save** to save the Request Header Type.

Field Groups

Field Groups are a way for Mercury ITG to distribute a collection of fields required for certain functionality. For example, Mercury Demand Management distributes a collection of fields for SLA in a SLA Field Group. Field Groups can be added to Request Header Types. The fields will behave just like normal fields, with the restrictions that the user cannot remove them except by removing the entire Field Group and the user might not be able to modify some of the field properties. *Table 7-7* lists the Field Groups that are delivered with various Mercury ITG products.

Each Field Group has a custom Token prefix that allows the user to access the data of that field by using:
REQ.P.<field_group_token_starting_with_KNTA_>



Please enable the Field Groups necessary for the desired functionality of this Request Header Type:

Enabled	Description
☐	Demand Management SLA Fields: This Field Group contains the fields necessary to manage requests
☐	Demand Management Scheduling Fields: This Field Group allows a request to be scheduled with the [
☐	Master Project Reference on Request: Contains a field that allows a user to add a Project reference t
☐	PFM Asset: Allows Requests to be considered as Assets in a Portfolio.
☐	PFM Project: Allows Requests to be considered as Projects in a Portfolio.
☐	PFM Proposal: Allows Requests to be considered as Proposals in the Portfolio Management process.
☐	PMO Program Issue: Allows Requests to be considered as Issues in a Program.
☐	PMO Program Resource Request: Allows Requests to be considered as Resource Requests in a Pro
☐	PMO Program Risk: Allows Requests to be considered as a Risk in a Program.

Figure 7-1 Field Groups Window

Table 7-7. Field Groups

Field Group	Description
Demand Management SLA	This Field Group contains the fields necessary to manage Requests with SLA.
Demand Management Scheduling	This Field Group allows a Request to be scheduled with Mercury Demand Management.
Master Project Reference on Request	Contains a field that allows a user to add a Project Reference to a Request.
PMO Program Issue	Allows Requests to be considered as issues in a Program.
PMO Program Resource Request	Allows Requests to be considered as Resource Requests in a Program.
PMO Program Risk	Allows Requests to be considered as a Risk in a Program.
PMO Program Scope Change	Allows Requests to be considered as a Scope Change in a Program.
Portfolio Management Proposal	Contains the fields necessary to create a PFM Proposal.
Portfolio Management Project	Contains the fields necessary to create a PFM Project.
Portfolio Management Asset	Contains the fields necessary to create a PFM Asset.

Table 7-7. Field Groups [continued]

Field Group	Description
Program Reference on Request.	Contains a field that allows a user to add a Program Reference to a Request.
Work Item Fields	Work item fields.

Specific Field Groups and related configuration notes are further documented in other Mercury ITG configuration guides.

Request Type Field Validations

Request Type fields define the information collected from the end users when a Request is created, and during the Request resolution process. Prompts, Tokens, and Validations can be configured for each field in a Request Type.

In addition, field properties can change based on the Status of the Request, which in turn can be set by the Workflow. As the Request moves through its lifecycle, these changes in field properties can reflect your business process.

Example

To properly prioritize and resolve a software bug, the system must know the severity of the bug's impact. A field named Impact can be created to capture that information. The value entered in this field will later be used in conjunction with the Workflow to process and resolve the Request.

To create and configure a Request Type field:

1. Open the Request Type window.
2. Click **New**.

   The Field window opens.
3. Enter the general field information: Field Prompt, Token, and Description.
4. Select a Validation for the field.

If a Validation does not exist that meets your needs, one can be created. The Validation dictates the possible values that can be entered in the field and the field type (such as text field, drop down list, or date field).

5. Configure the field’s behavior.

   This includes setting the following aspects of the field:

   - Visibility
   - Editability
   - Basic field defaulting
   - Other advanced field behavior

   Field behavior configuration consists of setting options in the Field window’s **Attributes**, **Default**, **Storage**, and **Security** tabs, as well as in the Request Type’s **Rules** and **Status Dependencies** tabs. See *“Configuring Field Behavior - Overview”* on page 130. Note that some field behavior is dependent on other Request Type fields. This step may need to be revisited after creating the other fields in your Request Type.

6. Enable the field.

Tip

To save time, you can copy fields from other Request Types and modify them to suit your needs. For more detailed information, see *“Copying a Request Type Field”* on page 141.

Determining the Field Type (Selecting a Validation)

When configuring a Request Type or Request Header Type, it is possible to specify a different Validation for each field (except existing system-provided Request Header Type fields). The Validation dictates the possible values that can be entered in the field. They also dictate the field type (for example, text field or drop down list). The following sections provide some general information related to Validations.

- *“Available Field Types”* on page 126
- *“Selecting the Validation”* on page 127
- *“Building a Validation”* on page 128

See *"Validations"* on page 283 for more detailed implementation instructions.

Available Field Types

Fields located in the Request Type can be of the following types.

Table 7-8. Field types.

Field Type	Description
Text Field, Text Area	Text fields and text areas are generic entry fields. Text fields are displayed on a single line, while text areas are displayed on multiple lines using a scroll bar if necessary. The values that are entered can be constrained. If an attempt is made to type non-conforming values into a text field or text area, the entries are ignored. For example, if the letter "A" is typed into a numeric field, the character does not appear. Text fields can also be configured to display the data according to a certain format. For example, to configure a text field to accept and format a nine digit, hyphenated social security number or a ten digit telephone number.
Drop down list	Field that allows the user to select from a predefined set of values. The values in a drop down list can be specified in two ways: • In the Validated By field, by selecting List to enter specific values. • By selecting SQL to use a SQL statement to build the contents of the list.
Auto-complete list	Field that allows the user to select from a predefined set of values. The values in an auto-complete list can be specified in the following ways. In the Validate By field, select one of the following: • **List**: used to enter specific values. • **SQL**: uses a SQL statement to build the contents of the list. • **Command With Delimited Output**: uses a system command to produce a character-delimited text string and uses the results to define the list. • **Command With Fixed Width Output**: uses a system command to produce a text file and parses the result on the basis of the width of columns, as well as the headers.
Radio Button	Radio buttons are used for fields where there is a possible Yes/No choice. Selecting on option disables the other. For example, clicking **Yes** in a Yes/No radio button pair disables the **No** option. To select a choice, click the button to the left of the appropriate choice.
Date Field	Date fields can accept a variety of formats. The current date field Validations are separated into two categories: all systems, and systems using only the English language.

Table 7-8. Field types.

Field Type	Description
Web Address (URL)	The Web Address field is a generic text entry field in which any URL can be entered. When this field is used, a U button appears next to the field. If U is clicked, a Web page is opened using the field value as the Web address.
Password	The Password Field component type creates a text field with an associated **C** button. Data is entered through a dialog that asks for the new password and a verification of the password. The text is then displayed in the field as *****.
Table Component	Used to enter multiple records into a single component on the Request. The table component can be configured to include multiple columns of varied data types. Additionally, this component supports rules for populating elements within the table and provides functionality for capturing column totals. See *"Configuring the Table Component"* on page 339 for details.
Budget	Field that can be added to the Request Type to enable access to view, edit or create Budgets associated with a Request or Project.
Staffing Profile	Field that can be added to the Request Type to enable access to view, edit or create Staffing Profiles associated with a Request or Project.
Resource Pool	Field that can be added to the Request Type to enable access to view, edit or create Resource Pools associated with a Request or Project.

Note

Fields of type Directory Chooser and File Chooser cannot be used in Request Types.

Selecting the Validation

Use the information gathered in *"Gathering Process Requirements and Specifications"* on page 35 to determine the appropriate Validation for the Request Type field. If a Validation does not exist that meets your requirements (has the appropriate values), one can be created. See *"Validations"* on page 283 for a complete list of Validations that are delivered with Mercury Demand Management.

It is also possible to select a Validation that has been configured for use at your site.

Tip

Be careful when using a Validation that has been configured for use in another process. If the owner of the other process changes the Validation, it will also be changed for the items in your process. Consider creating a new Validation by copying the existing one. To control who can alter the Validation values, set Ownership on that Validation.

Building a Validation

If a Validation does not exist that meets the requirements (has the appropriate values), one can be created. For instructions on creating the Validation, see *"Validations"* on page 283.

Table 7-9 highlights when to use certain component types.

Table 7-9. Field/Validation Examples

Component Type	Examples: When to Use
Auto-complete list Drop down list	• List of all users • List of all users in a specific Security Group • Desired actions • List of information located in another (non-Mercury ITG) system. (example: list of managers stored in PeopleSoft)
Radio buttons Check boxes	• Question specifying an action. (Example: File needs to be compiled?) • Results of an execution step (radio button). The result can be used later in a processing decision.
Text field Text area Date field	• Description of the change (text field or text area). • Release version number (numeric text field) • Specify a deadline (date field)

Auto-Complete Versus Drop Down List

Auto-completes and drop down lists are often applied in similar situations. They both present a predefined / limited list of choices to the user, but both have unique features which could be more appropriate for a given situation. Consider the following comparison chart when selecting to use a drop-down or auto-complete list.

Table 7-10. Auto-complete versus drop down comparison chart.

Action	Auto-complete	Drop-down
Can contain a static list of choices	Yes	Yes
Can contain choices derived from a database query	Yes	Yes
Can contain choices from system executions (commands)	Yes	
Can display multiple columns of information	Yes	
Allows users to select multiple values from the list	Yes	
List is determined at the time of page/screen load		Yes
List is determined when the field is selected. this is useful when making the values in the list dependent on other parameters in the screen. For example, listing only users in a specified Security Group.	Yes	
Allows for partial value returns (for example, type “A” and and view only the choices beginning with “A.”)	Yes	

Finally, consider usability. Drop down lists become less efficient when the selection list gets large. Consider using auto-completes in these situations.

Tips for Configuring Validations

Consider the following tips when creating a Validation for your Request Type:

- Be careful when creating Validations (drop down lists and auto-complete lists) that are validated by lists. Each time the set of values changes, the Validation must be updated. Consider, instead, validating using a SQL query or PL/SQL function. For example, to create an auto-complete field that lists all users in a specific department, validate the list by SQL.

```
SELECT U.USER_ID, U.USERNAME, U.FIRST_NAME, U.LAST_NAME
FROM KNTA_USERS U, KNTA_SECURITY_GROUPS SG,
   KNTA_USER_SECURITY US
WHERE SG.SECURITY_GROUP_ID = US.SECURITY_GROUP_ID AND
   US.USER_ID = U.USER_ID
AND SG.SECURITY_GROUP_NAME = 'Support Team'
```

```
and UPPER(u.username) like UPPER('?%')
and (u.username like upper(substr('?',1,1)) || '%'
  or   u.username like lower(substr('?',1,1)) || '%')
order by 2
```

In the previous example, when a new user account is created and included in the "Support Team" Security Group, that user will automatically be included in the auto-complete list.

- Reuse SQL and PL/SQL from existing Validations. Review the seeded Validations (see *"Validations"* on page 283) to see if there are other similar Validations in the system. If there are, copy the Validation and modify the Validated By specifications to meet your requirements.
- The same Validation can often be used for Workflow step source and the field Validation.

Configuring Field Behavior - Overview

This section summarizes the field behavior that can be configured for your Requests. The following attributes can be configured on each field in the Request Type:

- *Visibility*
- *Editability*
- *Defaulting*
- *Required/Reconfirm*
- *Notifications*

Visibility

Fields can be set to be visible or hidden to the user based on the following criteria:

- Field attributes—The field can be set to display or be hidden at all times. This is controlled from the Field window's **Attributes** tab. See *"Creating a Field"* on page 133 for details.
- Request Status—Based on the Status of the Request itself (linked to the Workflow Step), the field can be set to display or be hidden. See *"Configuring Field Behavior Using Status Dependencies"* on page 157 for details.

- Field security—Fields can also be configured to be invisible to particular users or Security Groups. This is controlled from the Field window's **Security** tab. See *"Creating a Field"* on page 133 for details.

Figure 7-2 illustrates how the product determines whether a field is visible to a particular user. This diagram assumes that the user has access to view the Requests, which requires the correct license, Access Grants, and settings in the Request Type window's **User Access** tab.

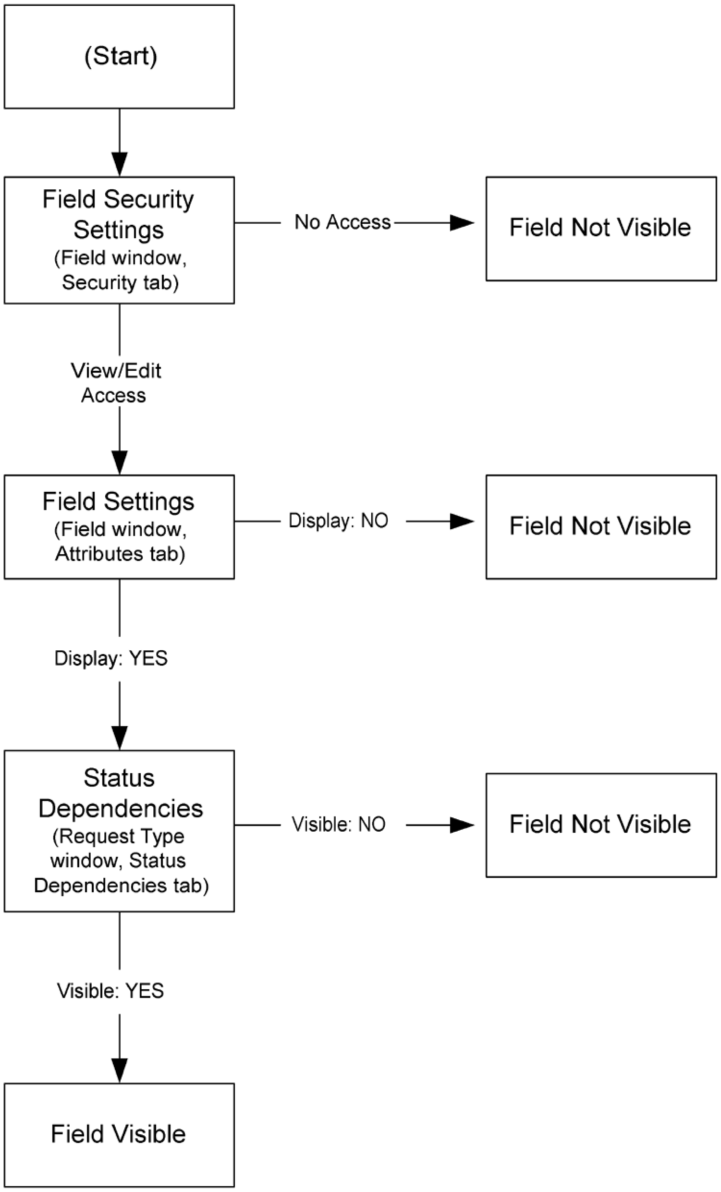

(Start)
Field Security Settings (Field window, Security tab)
No Access
Field Not Visible
View/Edit Access
Field Settings (Field window, Attributes tab)
Display: NO
Field Not Visible
Display: YES
Status Dependencies (Request Type window, Status Dependencies tab)
Visible: NO
Field Not Visible
Visible: YES
Field Visible

Figure 7-2 Field Visibility Interactions

Table 7-11 provides a quick reference guide for setting field visibility parameters.

Table 7-11. Quick Reference Guide - Field Visibility Parameters

Field Parameter	Location	Section & Page
Field Security Settings	Request Type window -> **Fields** tab -> Field window -> **Security** tab	*"Creating a Field"* on page 133
Field Settings	Request Type window -> **Fields** tab -> Field window -> **Attributes** tab	*"Request Type Field window - Attributes Tab"* on page 136
Status Dependencies	Request Type window -> **Status Dependencies** tab	*"Status Dependencies - Visible"* on page 162

Editability

Fields can be set to become display-only, so that their contents are frozen and become non-editable, based on the following criteria:

- Request Status—Based on the Status of the Request itself, the field can be set to become non-editable. For more information, see *"Configuring Field Behavior Using Status Dependencies"* on page 157.
- Field security—Fields can be configured to be visible but non-editable to particular users or Security Groups. This is controlled from the Field window's **Security** tab. For more detailed information, see *"Creating a Field"* on page 133.

Defaulting

A field can be configured to automatically update the value in that field based on the following settings:

- Field defaulting—The value of a single field can be linked to the value of other fields defined for that entity. For example, a Request Type field can default to a particular manager's username when the value in another field in that Request Type equals the text **Critical**. This is controlled from the Field window's **Default** tab. See *"Creating a Field"* on page 133 for additional details.
- Request Type Rules—A Request Type can be configured to automatically populate multiple fields based on the value of one field. For example, if a field has the value **Bug Report**, the Workflow, Contact Name, Contact Phone,

and Department fields can be automatically filled. This is controlled from the Request Type window's **Rules** tab. See *"Configuring Request Type Defaulting Behavior (Rules)"* on page 145 for additional details.

- Request Type Commands — Commands can also be used to control certain behavior of Request Type fields. At specific points (Workflow execution steps) in a resolution process, it is possible to select to run the commands stored in the Request Type. These commands can then manipulate the data inside a Request Type field. For example, you can construct a Command to consider a number of parameters and then default a field based on those parameters. This provides an advantage over the defaulting features in the Field window, which can only default based on a single parameter stored on the same Request Type.

  Controlling field values using Commands can be useful in the following situations (examples):

  - Store a value from an execution (Note: this can also be done using Workflow parameters. See *"Transition based on a previous Workflow Step result (parameters)"* on page 106 for details.)
  - Clearing a field after evaluating a number of parameters.

  See the *Commands and Tokens Guide and Reference* for more information on setting up commands to control field defaulting.

Required/Reconfirm

Fields can be configured to be required or need to be reconfirmed based on the Status of the Request. See *"Configuring Field Behavior Using Status Dependencies"* on page 157 for additional details.

Notifications

Fields can be configured to send an email Notification based on a change in value. See *"Setting Notifications on Request Field Changes"* on page 228 for additional details.

Creating a Field

New fields are created and configured using the Field window, accessed from the Request Type window's **Fields** tab.

From the Field window, it is possible to configure:

- Whether the field is displayed
- Whether a field can be edited under different circumstances
- Whether the field defaults to a certain value
- Dependencies to values in other fields in the Request Type

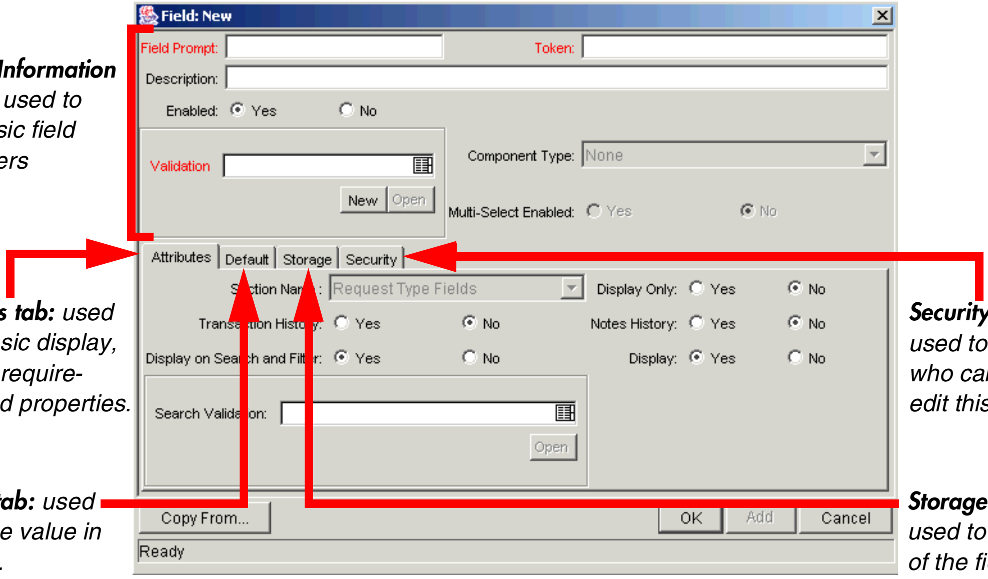
Note

> Since field behavior is often dependent on other fields in the Request Type, other Request Type fields will often have to be created before configuring a field's behavior.

General Information Region: *used to enter basic field parameters*

Attributes tab: *used to set basic display, edit and requirement field properties.*

Default tab: *used to set the value in the field.*



Field: New

Field Prompt: Token:

Description:

Enabled: ● Yes ○ No

Validation

New Open

Component Type: None

Multi-Select Enabled: ○ Yes ● No

Attributes | Default | Storage | Security

Section Name: Request Type Fields

Display Only: ○ Yes ● No

Transaction History: ○ Yes ● No

Notes History: ○ Yes ● No

Display on Search and Filter: ● Yes ○ No

Display: ● Yes ○ No

Search Validation:

Open

Copy From... OK Add Cancel

Ready

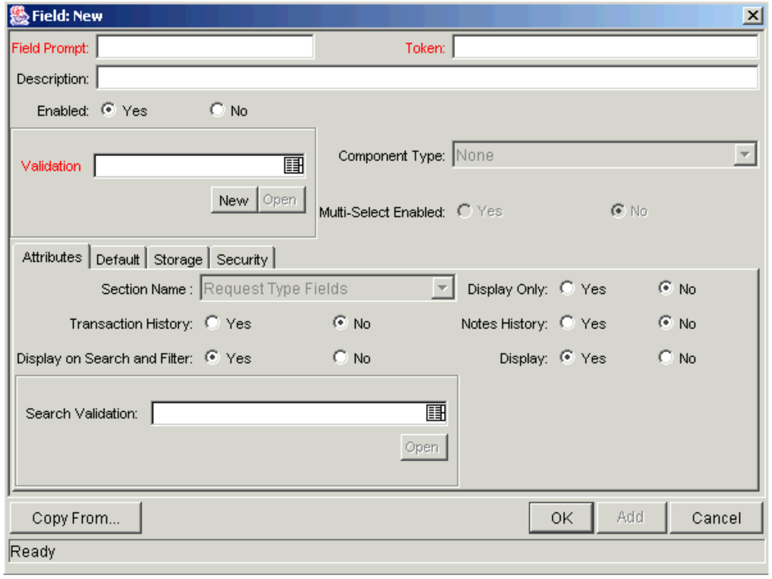
Security tab: *used to set users who can view and edit this field.*

Storage tab: *used to set properties of the field relating to its storage in the database.*

Figure 7-3 Request Type Field Window

To create a new field on the Request Type:

1. Open the Request Type window.
2. In the **Fields** tab, click **New**.



Field Prompt:
Token:
Description:
Enabled: ● Yes ○ No
Validation
Component Type: None
Multi-Select Enabled: ○ Yes ● No

Attributes | Default | Storage | Security

Section Name : Request Type Fields
Display Only: ○ Yes ● No
Transaction History: ○ Yes ● No
Notes History: ○ Yes ● No
Display on Search and Filter: ● Yes ○ No
Display: ● Yes ○ No
Search Validation:

Ready

3. Enter the general information for the Request field.

   Table 7-12 provides a definition of the fields in this area.

Table 7-12. Request Type Field window - General Information Region

Field	Description
Field Prompt	The prompt visible for the Request Type field on the Request.
Token	An uppercase text string used to identify this field. The Token name must be unique for the specific Request Type. An example of a Token name is ASSIGNED_TO_USER_ID.
DESCRIPTION	A description of the Request Type field.
ENABLED	Determines whether or not the field is turned on for this Request Type.
Validation	Indicates the Validation logic to determine the valid values for this field. This could be a list of user-defined values, a rule that the result has to be a number, etc. See *“Determining the Field Type (Selecting a Validation)”* on page 125 for more details.
Component Type	Defines the visual characteristics of the field (drop down list, free form text field, etc.). This is derived from the Validation chosen. This field cannot be edited.

Table 7-12. Request Type Field window - General Information Region

Field	Description
Multiselect	Determines whether or not the field allows users to select more than one entry. Only valid for fields with an auto-complete component for the Validation.

4. Click the **Attributes** tab to define the field's basic properties (such as Display Only, Transaction History, and Notes History).

   Table 7-13 defines the fields in this area.

Table 7-13. Request Type Field window - Attributes Tab

Field	Description
Section Name	The area of the Request on which the field is displayed.
Display Only	Determines if the field is only displayed and cannot be updated, even at initial Request entry.
Transaction History	Turns transaction auditing on or off for this field. If it is set to Yes, whenever this field changes in a Request, the change is logged in a transaction history table.
Notes History	Turns Notes auditing on or off for this field. If it is set to Yes, whenever this field changes in a Request, the change will be logged in Notes for the Request.
Display On Search and Filter	Determines whether or not the field will be displayed in Search and Filter pages in the standard interface.
Display	Determines whether or not the field is seen by Requests that use the given Request Type. If set to **No**, the Request Type field will no longer be displayed.

Note

The number of fields in a Request Type that can have the Notes History and Transaction History attributes enabled is limited. The total number of fields in a Request Type that has Notes History and Transaction History enabled separately or both attributes enabled at the same time should not exceed forty fields.

5. Click the **Defaults** tab to define the default value for that field.

Table 7-14 defines the fields in this area.

Table 7-14. Request Type Field window - Defaults Tab

Field	Description
Default Type	Defines if the field will have a default value. Either default the field with a constant value, default it from the value in another field, or default to a parameter.
Visible Value	If a Default Type of **Constant** is selected, the constant value can be entered here. This value should be what the user would normally enter in the field.
Depends On	If defaulting from another field, enter the Token name of that field. At runtime, when using this Request Type, every time a value is entered or updated in the source field, it will automatically be entered or updated in this destination field.

6. Click the **Storage** tab to view the field's location in the database.

   The **Storage** tab automatically places the field into the next available position within the database based on the current field's attributes. By opening the Field window for a specific field found within a Request, administrators can use the **Storage** tab to locate a field within the database. This is useful for reporting purposes. If necessary, the **Storage** tab can also be used to specify a field location within the database when creating a new field, but the standard method is to allow the interface to automatically position the field for the administrator.

   The **Storage** tab automatically stores the value for a Text Field of maximum length 1800 in column 41 or higher.

   Table 7-15 defines the fields in this area.

Table 7-15. Request Type Field window - Storage Tab

Field	Description
Max Length	Determines the maximum field character length. The two possible values are 200 and 1800.

Table 7-15. Request Type Field window - Storage Tab

Field	Description
Batch Number	Based on the number of maximum fields. For every 50 fields, one batch is created. 10 of these 50 fields can be more than 200 characters in length. Enabled only when there are more than 50 fields (creating more than one batch).
Parameter Col	Determines the internal database column that the field value is stored in. These values are then stored in the corresponding column in the Request Details table for each batch of the given Request Type. Information can be stored in up to 50 columns using Request Types, allowing up to 50 fields/batch. No two fields in a Request Type can use the same column number within the same batch.

7. Click the **Security** tab to view the field's security configuration.

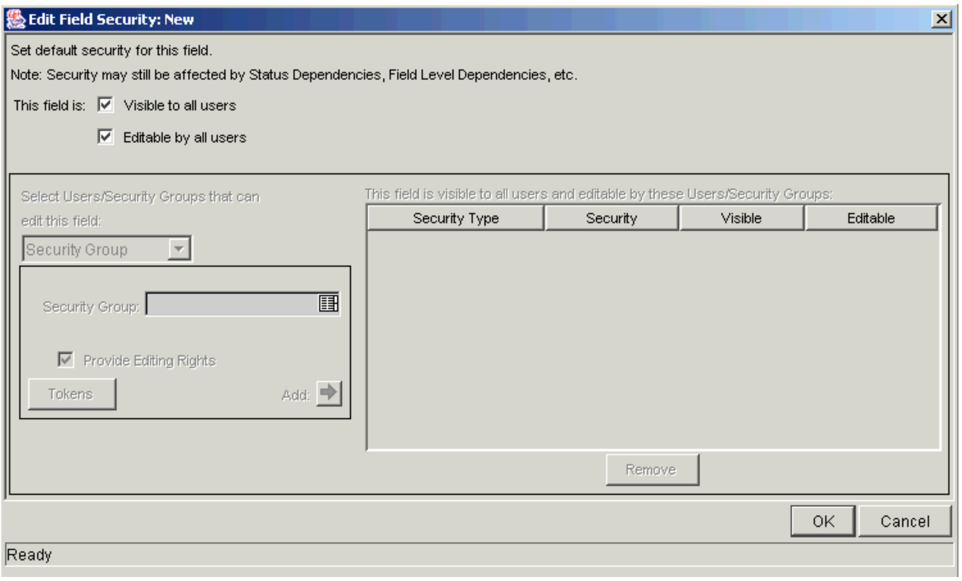
   Table 7-16 defines the fields in this area.

Table 7-16. Request Type Field window - Security Tab

Field	Description
Visible To	Lists all users, Security Groups, and linked Tokens for which this field will be visible.
Editable By	Lists all users, Security Groups, and linked Tokens for which this field will be editable.
Edit	Opens the Edit Field Security window, which configures the users, Security Groups, and linked Tokens that will be able to view and/or edit this field. See *"Creating a Field"* on page 133 for more detailed information.

   a. Click **Edit**.

      The Edit Field Security window opens.



Edit Field Security: New

Set default security for this field.

Note: Security may still be affected by Status Dependencies, Field Level Dependencies, etc.

This field is: ☑ Visible to all users

☑ Editable by all users

Select Users/Security Groups that can edit this field:

Security Group

Security Group:

☑ Provide Editing Rights

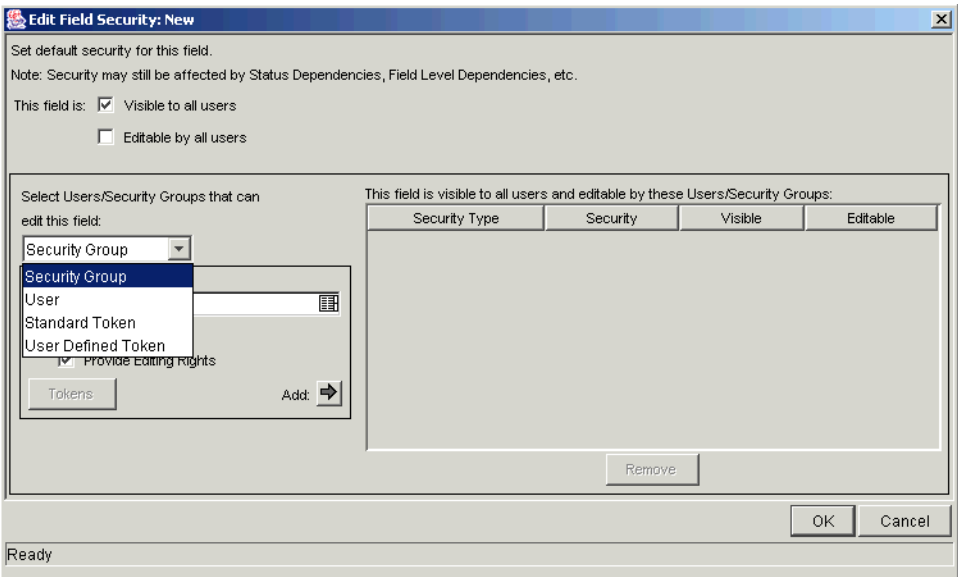
This field is visible to all users and editable by these Users/Security Groups:

Security Type	Security	Visible	Editable

Ready

b. Uncheck the Visible to all users box to begin fine-tuning field properties.

c. Make a choice from the Select Users/Security Groups that can view this field drop down list. Possible choices include **User**, **Security Group**, **Standard Token**, or **User Defined Token**.



Edit Field Security: New

Set default security for this field.

Note: Security may still be affected by Status Dependencies, Field Level Dependencies, etc.

This field is: ☑ Visible to all users

□ Editable by all users

Select Users/Security Groups that can edit this field:

Security Group

☑ Provide Editing Rights

This field is visible to all users and editable by these Users/Security Groups:

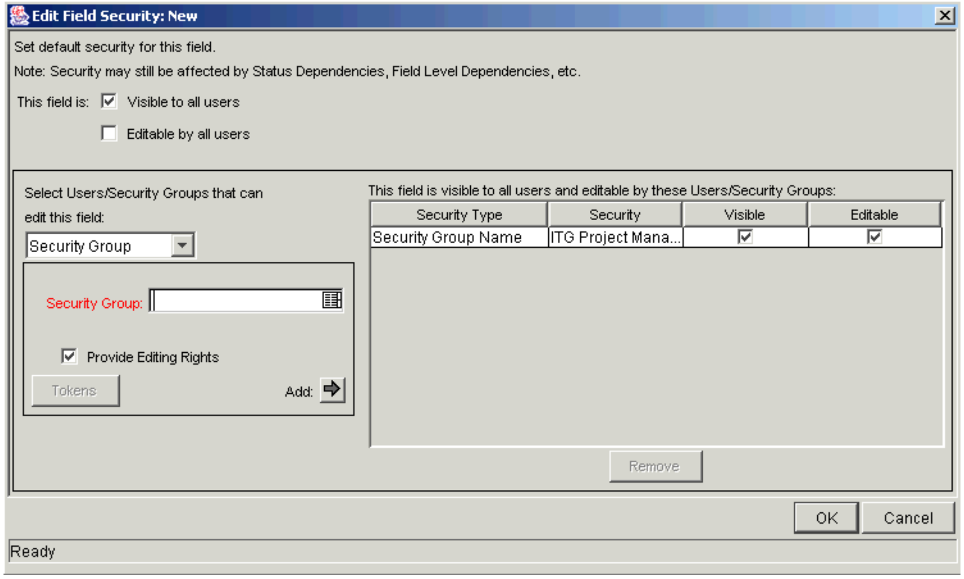
Security Type	Security	Visible	Editable

Ready

d. After selecting a choice from the drop down list, select the User, Security Group, or Token from the auto-complete list.

To assign the selected User, Security Group, or Token editing rights as well as viewing rights to the field, check the Provide Editing Rights box.

e. Click the **Add** arrow button to add the selected User, Security Group, or Token to the This field is visible to these Users/Security Groups area.



Set default security for this field.

Note: Security may still be affected by Status Dependencies, Field Level Dependencies, etc.

This field is: ☑ Visible to all users

☐ Editable by all users

Select Users/Security Groups that can edit this field:

Security Group

Security Group:

☑ Provide Editing Rights

This field is visible to all users and editable by these Users/Security Groups:

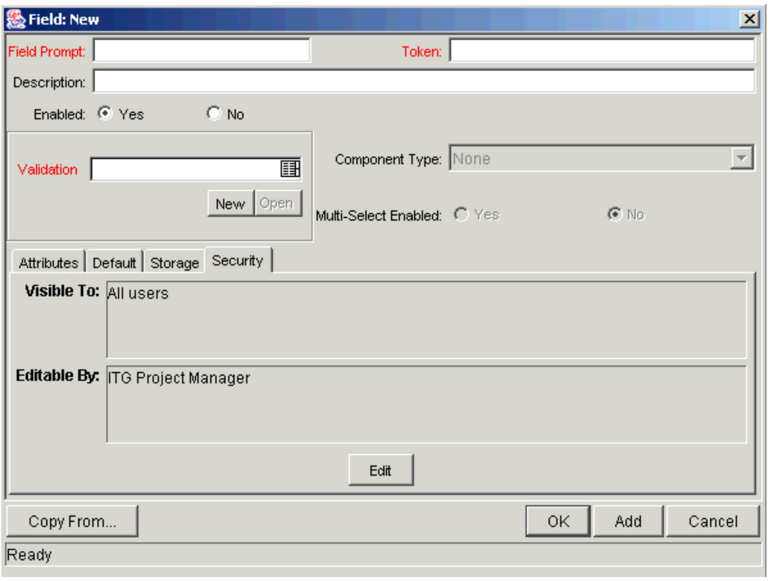
Security Type	Security	Visible	Editable
Security Group Name	ITG Project Mana...	☑	☑

Ready

To change the Visible and Editable settings for each entry directly in the Edit Field Security window, uncheck the box in the Visible or Editable column of the This field is visible to these Users/Security Groups area. To remove viewing rights entirely, select the User, Security Group, or Token and click **Remove**.

8. When you are finished adding Users, Security Groups, or Tokens to the This field is visible to these Users/Security Groups area, click **OK** to save changes and return to the **Security** tab.

   The **Security** tab is updated with the list of Users, Security Groups, or Tokens with viewing or editing rights to the field.



Enabled: ● Yes ○ No

Component Type: None

Multi-Select Enabled: ○ Yes ● No

Visible To: All users

Editable By: ITG Project Manager

Ready

Note

When adding field-level security to existing fields on a Request Type that has been used to create Requests, the Mercury IT Governance database tables are updated to handle this new configuration. Due to the scope of database changes, the Database Statistics need to be re-run on the database. Instructions for this are included in the System Administration Guide. Contact the System Administrator for help with this procedure.

Note

There can only be 500 rows per column, 3 columns per tab, and a maximum of 20 tabs for each Request Type.

When taking advantage of the Reporting Meta Layer functionality, those fields contained within the first four batches (200 fields) will be available for reporting.

Copying a Request Type Field

Use the **Copy From** functionality to streamline the process of adding fields to a Request Type by copying the definition of existing fields (from other Request Types).

To copy a Request Type field:

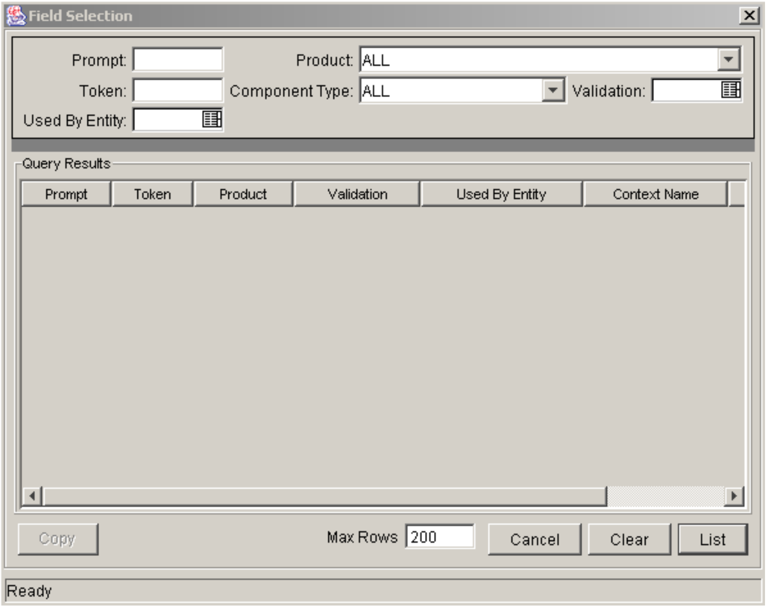
1. Open the Request Type.

2. In the **Field** tab, click **New**.

   The Field window opens.

3. Click **Copy From**.

   The Field Selection window opens.



Field Selection

Prompt:

Token:

Used By Entity:

Product: ALL

Component Type: ALL

Validation:

Query Results

Prompt	Token	Product	Validation	Used By Entity	Context Name

Max Rows 200

Ready

4. Enter search criteria into the header fields, such as the Token name or field Prompt.

   More complex queries can also be performed, such as listing all fields that reference a certain Validation or are used by a certain entity. Due to the large number of fields in the system, limit the list of fields by one or more of the query criteria.

5. Click **List**.

6. Select the desired field, and click **Copy**.

   This closes the window and copies the definition of the selected field into the New Field window.

7. Make any necessary modifications.

8. Click **OK**.

This saves the changes and closes the Field window.

Removing Request Type Fields

To remove a field permanently from a Request Type:

1. Open the Request Type.

2. Select the field in the **Fields** tab.

3. Click **Remove.**

4. Click **OK**.

This removes the field and closes the window.

Setting the Number of Maximum Fields for a Request Type

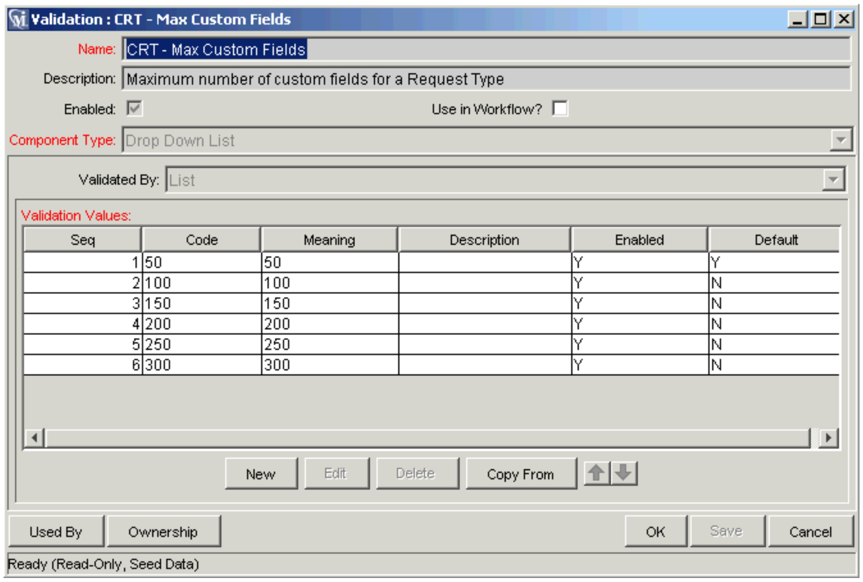
Initially, Request Types have a set group of selectable, maximum field values. The values are in increments of 50 and stop at 300.

To change the number of maximum allowable fields for all Request Types:

1. Click the **Configuration** screen group and the Validations screen.

   The Validation Workbench opens.

2. Open the CRT- Max Custom Fields Validation.



Validation : CRT - Max Custom Fields

Name: CRT - Max Custom Fields

Description: Maximum number of custom fields for a Request Type

Enabled: ☑ Use in Workflow? ☐

Component Type: Drop Down List

Validated By: List

Validation Values:

Seq	Code	Meaning	Description	Enabled	Default
1	50	50		Y	Y
2	100	100		Y	N
3	150	150		Y	N
4	200	200		Y	N
5	250	250		Y	N
6	300	300		Y	N

Ready (Read-Only, Seed Data)

3. Click **New**.

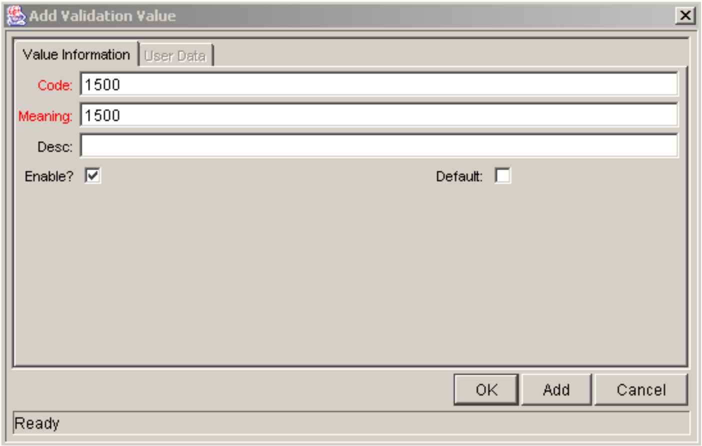
   The Add Validation Value window opens.

4. Enter the new maximum field value in both the Code and Meaning fields (the values entered must be the same).

Note

To change an existing Validation Value, select the desired line and click **Edit**. Enter the new values and then click **OK**.

All values entered must be multiples of 50 (i.e. 350, 1500, 1550, etc.)



Code: 1500

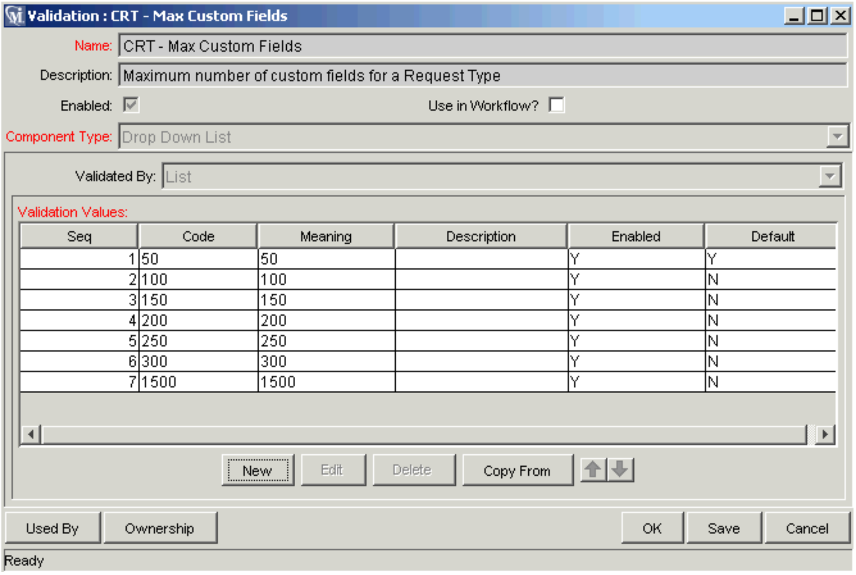
Meaning: 1500

Desc:

Enable? ☑

Default: □

5. Click **OK**.



Name: CRT - Max Custom Fields

Description: Maximum number of custom fields for a Request Type

Enabled: ☑ Use in Workflow? ☐

Component Type: Drop Down List

Validated By: List

Validation Values:

Seq	Code	Meaning	Description	Enabled	Default
1	50	50		Y	Y
2	100	100		Y	N
3	150	150		Y	N
4	200	200		Y	N
5	250	250		Y	N
6	300	300		Y	N
7	1500	1500		Y	N

The maximum custom fields value is now one of the selectable value options from the Max Fields drop down list on the Request Type window.

Configuring Request Type Defaulting Behavior (Rules)

Request Rules can be used to set up the automatic population of Request fields based on various dependencies. Request Rules are ideal for the following scenarios:

- A default Workflow, Assigned To user or Assigned Group should be specified when a Request of this Type is initially created.
- Multiple Request fields should be populated depending on the value of a single field.

There are two types of Rules:

- **Simple Default Rules** allow a default Workflow to be specified, as well as the Assigned To and Assigned Group fields, depending on the Department or Application filled in by the user. The Workflow, Assigned To and Assigned Group fields can also be specified upon Request creation.
- **Advanced Default Rules** define logic for the automatic population of fields in the Request based on user entries.

When configuring Request Rules, use the Rule Type drop down list to switch between **Simple** and **Advanced Defaults**.

Note

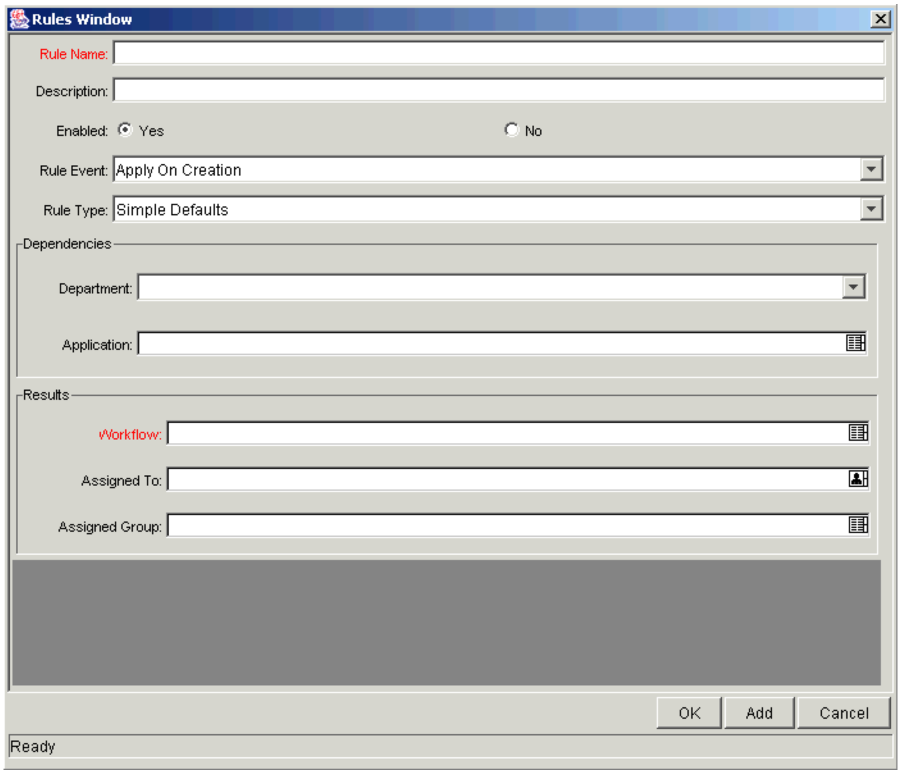
When switching between Rule Types, whatever work has been done in the first Type will be lost when the switch is made.

The following sections discuss setting up Request Rules in more detail:

- *Configuring Simple Default Rules*
- *Configuring Advanced Default Rules*

Configuring Simple Default Rules

Simple Default Rules are used to auto-fill the Workflow, Assigned To and Assigned Group fields.



Rules Window

Rule Name:

Description:

Enabled: ● Yes ○ No

Rule Event: Apply On Creation

Rule Type: Simple Defaults

Dependencies

Department:

Application:

Results

Workflow:

Assigned To:

Assigned Group:

These fields can be filled based on the Rule Event and Dependencies fields, discussed in *Table 7-17*.

Table 7-17. Simple Default Rule Control Fields

Field	Description
Rule Event	Specifies the event that triggers the Rule. **Apply On Creation** - The Rule will fire when the Request is created, filling in whichever of the Results fields have been specified. **Apply On Field Change** - The Rule will fire when one of the Dependencies fields is changed to the specified value. **Apply On Field Change And Stop Processing Rules** - The Rule will fire when one of the Dependencies fields is changed to the specified value, and all subsequent Rules in the **Rules** tab will not.
Department	Specifies the department that triggers the Rule.
Application	Specifies the application that triggers the Rule.

Using any appropriate combination of these control fields, the Workflow, Assigned To, or Assigned Group fields can be specified.

Note

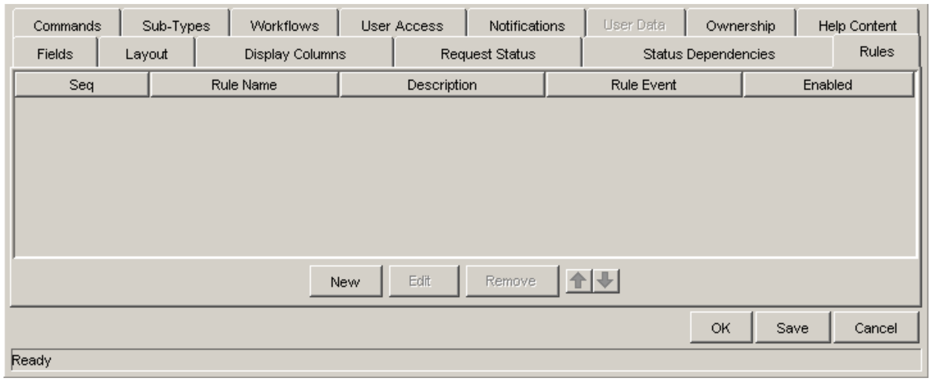
The Workflow field is the only required field for Simple Default Rules.

By setting the desired Workflow and the Rule Event to **Apply On Creation**, it is possible to set the default Workflow that will be used each time a Request of that type is used.

Creating a Simple Default Rule

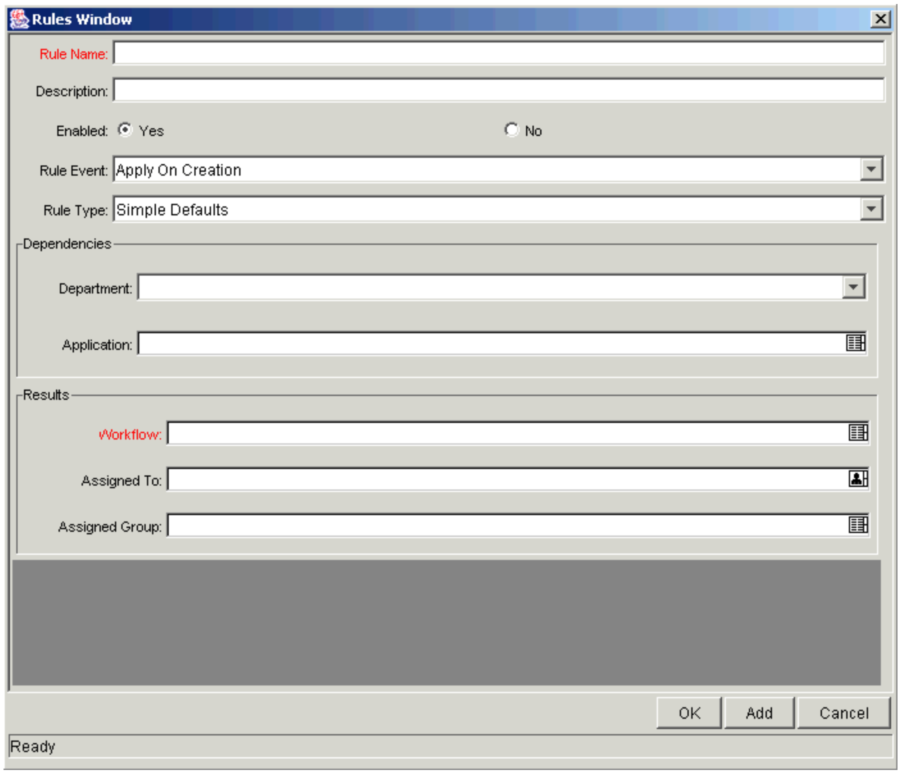
To add a new Simple Default Rule to a Request Type:

1. Open the Request Type window for the desired Request Type and click the **Rules** tab.



2. Click **New**.

   The Request Type Rules window opens in **Simple Defaults** mode.



Enabled: ● Yes ○ No

Rule Event: Apply On Creation

Rule Type: Simple Defaults

3. Enter the required Rule Name and Workflow.

4. Enter any desired Dependencies.

   These Dependencies must be met in order for the rule to be executed.

5. Enter the Results for this rule.

   A Workflow must be entered, but all other fields are optional.

6. Click **OK** to save this rule and close the window, or click **Add** to add this default rule and clear the window to add another rule.

Note

> The fields displayed in the Rules window shown are from the Request Header Type.

Once this rule is saved, any new Request matching the combination of Request Type, Department, and Application for the rule automatically updates the Workflow, Assigned To, and Assigned Group fields to the default values specified in the rule.

If more than one rule applies for a given Request, then the system uses the most specific rule. See *"Configuring Advanced Default Rules"* on page 152 for more detailed information.

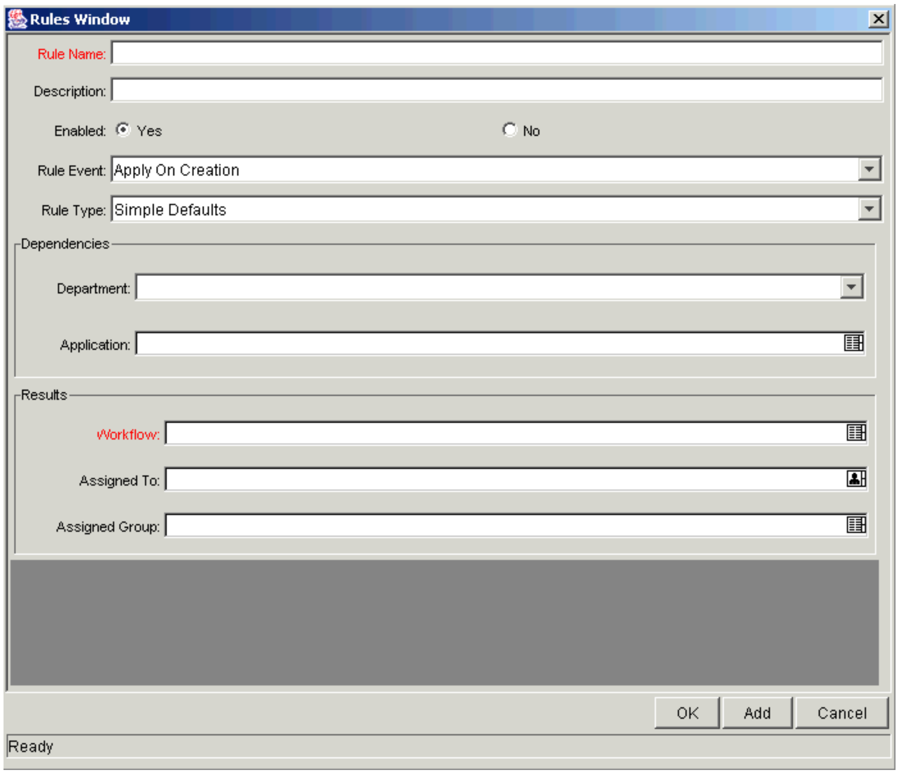
Example: ACME Defaults Software Change Workflow

ACME is creating their Financial Software Change Request Type, and have decided that it would be convenient for the Workflow field to be populated automatically with the value **Financial Software Change Workflow** whenever a Request of this type is created.

1. Harold Lomax, ACME's IT Configuration Manager, opens the Financial Software Change Request Type and clicks the **Rules** tab.

2. Lomax clicks **New**.

   The Rules window opens in **Simple Defaults** mode.



Rules Window

Rule Name:

Description:

Enabled: ● Yes ○ No

Rule Event: Apply On Creation

Rule Type: Simple Defaults

Dependencies

Department:

Application:

Results

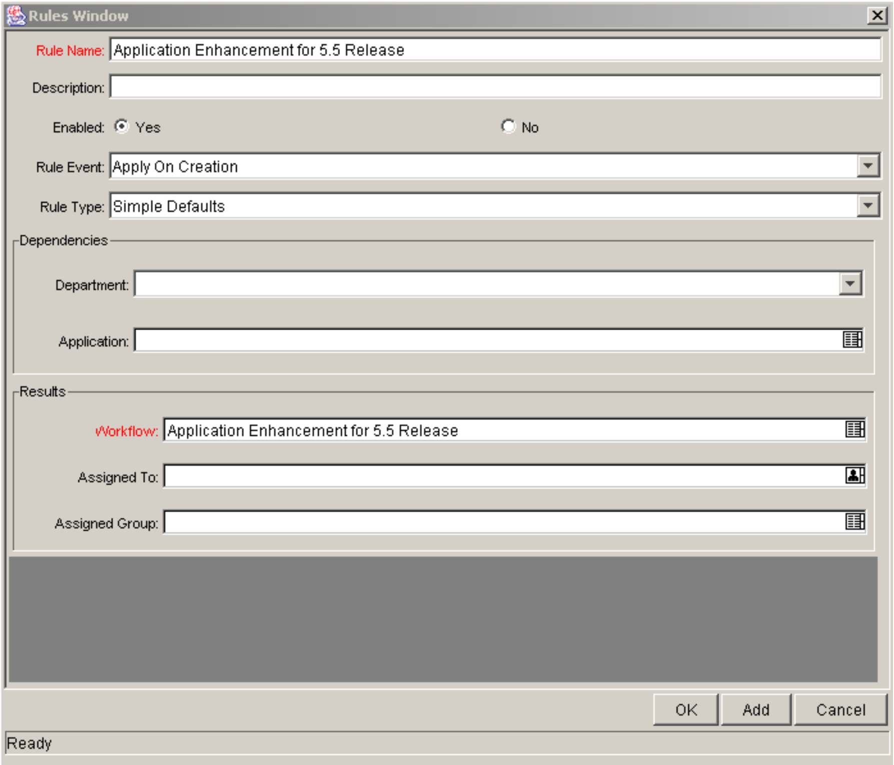
Workflow:

Assigned To:

Assigned Group:

3. In the Rule Name field, Lomax enters the name of the rule.

4. From the Workflow field, he selects **Financial Software Change Workflow**.



Rule Name: Application Enhancement for 5.5 Release

Description:

Enabled: ● Yes ○ No

Rule Event: Apply On Creation

Rule Type: Simple Defaults

Dependencies

Department:

Application:

Results

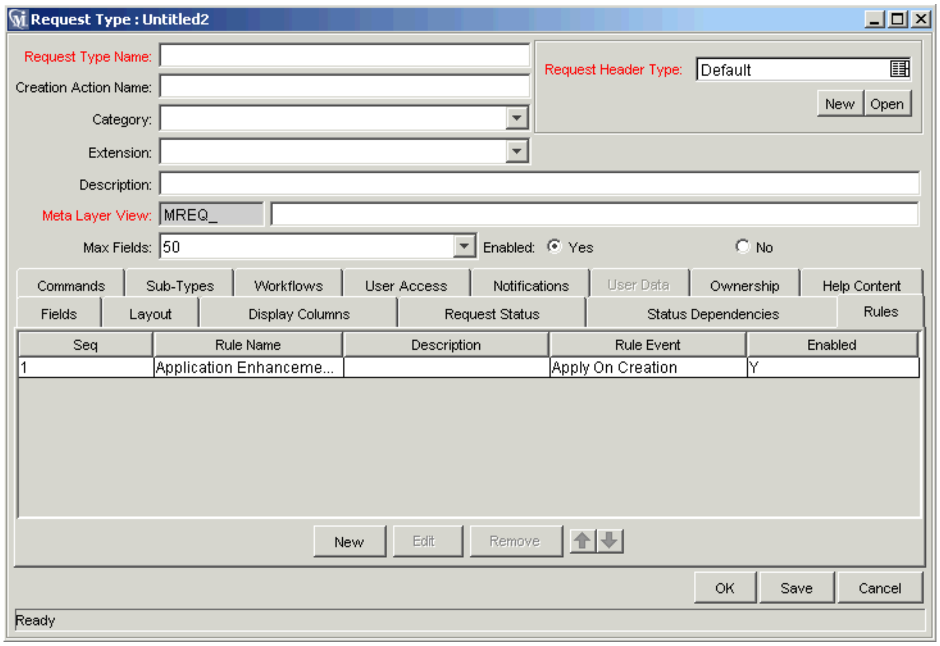
Workflow: Application Enhancement for 5.5 Release

Assigned To:

Assigned Group:

5. Lomax clicks **OK**.

   The new Rule is added to the **Rules** tab.



Request Type Name:
Creation Action Name:
Category:
Extension:
Description:
Meta Layer View: MREQ_
Max Fields: 50
Enabled: ● Yes ○ No
Request Header Type: Default
New Open

Seq	Rule Name	Description	Rule Event	Enabled
1	Application Enhanceme...		Apply On Creation	Y

Configuring Advanced Default Rules

Advanced Default Rules define logic for the automatic population of fields in the Request based on user entries. Advanced Default Rules differ from Simple Default Rules in the following ways:

- Simple Default Rules can only trigger from Request creation or changes to the Department or Application fields. Advanced Default Rules can trigger from changes to any field in the Request.
- Simple Default Rules can only populate the Workflow, Assigned To, or Assigned Group fields. Advanced Default Rules can populate any field or set of fields in the Request simultaneously, including fields in the Request or in the Request Header.

Note

Configuring Advanced Default Rules requires knowledge of SQL.

Advanced Default Rules are often used with the following values from the Rule Event field:

- **Apply On Creation**—The Rule will fire when the Request is created, filling in whichever of the Results fields have been specified.
- **Apply On Field Change**—The rules applies when values in other fields change. This functions two ways:

  a. Specific value — The rule applies when a field specified in the Dependencies area is changed to a specific user-defined value. If multiple dependency fields are defined for a rule, all of them must match in actual use for the rule to take effect.

  b. All values — The rule applies for any value of a field specified in the Dependencies area.

  When the field or fields specified in the Dependencies area are changed, any fields specified in the Results area are automatically populated according to rule order. This is useful in the event of multiple Dependency field matches.

Example

Harold Lomax specifies Rule One with Dependency Field A = **123** and Rule Two with Dependency Field B = **XYZ.**

A user fills in Field B with **XYZ**, then fills in Field A with **123**.

The system executes Rule One first, followed by Rule Two.

- **Apply On Field Change And Stop Processing Other Rules** — The rule applies when a field specified in the Dependencies area is changed to a user-defined value. When the field is changed, any fields specified in the Results area will be automatically populated according to the first rule defined. Any other rule processing will stop immediately after the last Result field is populated. This is useful for multiple Dependency field matches where one particular Rule should be evaluated without changing.

Example

Harold Lomax specifies Rule One with Dependency Field A = **123** and Field B = **XYZ**, and Rule Two with Dependency Field B = **XYZ**.

A user fills in Field A with **123**, then fills in Field B with **XYZ**.

The system executes Rule One only and stops.

Creating an Advanced Default Rule

To create an Advanced Default Rule to set up automatic population of fields for a Request Type:

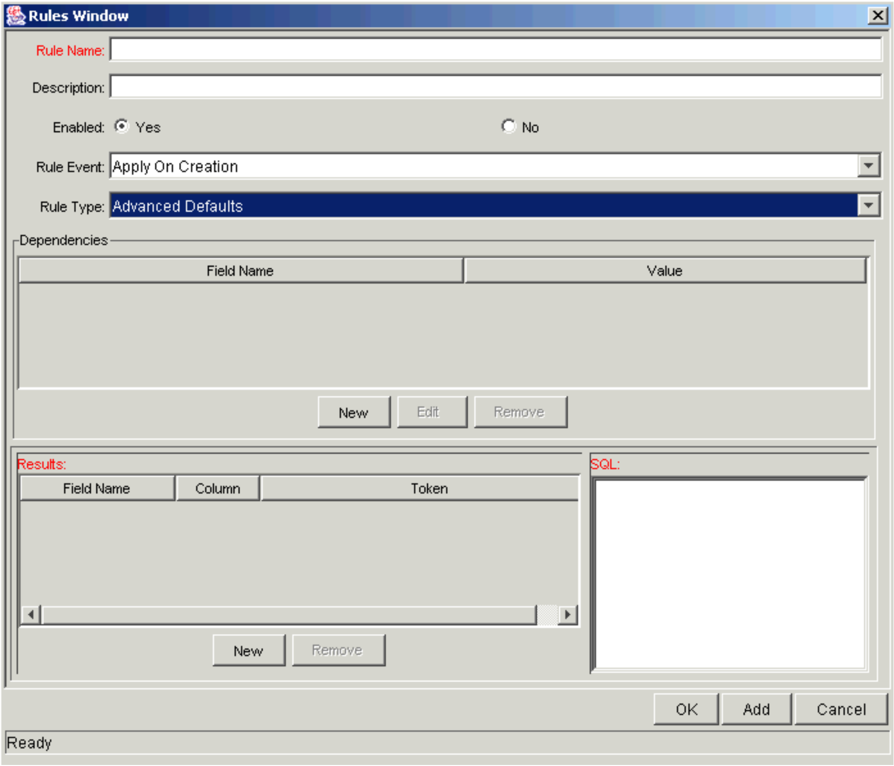
1. Open the Request Type window for the desired Request Type

2. Click the **Rules** tab.

3. Click **New**.

   The Rules window opens in **Simple Defaults** mode.

4. From the Rule Type drop down list, select **Advanced Defaults**.



Rules Window

Rule Name:

Description:

Enabled: ● Yes ○ No

Rule Event: Apply On Creation

Rule Type: Advanced Defaults

5. Enter a name for the new rule in the Rule Name field.

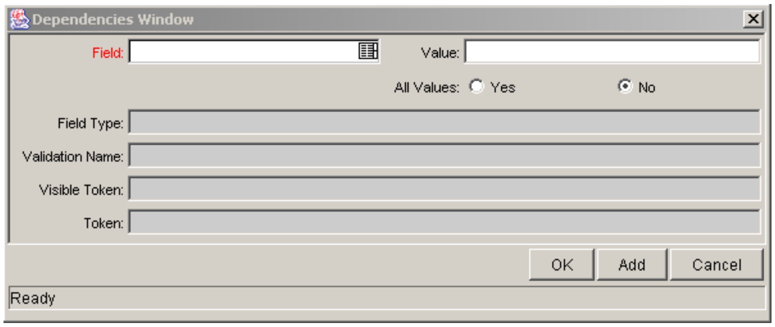
6. From the Enabled field, select **Yes**.

7. Select an event that will trigger the auto-population from the Rule Event drop down list (**Apply On Creation**, **Apply On Field Change**, **Apply On Field Change And Stop Processing Other Rules**).

8. Select a field or fields to trigger the rule.

   a. Click **New** in the Dependencies area.

      The Dependencies window opens.



Note

Request Default Rules cannot be configured to trigger from a multi-select auto-complete field. Do not choose a multi-select auto-complete for the Field.

   b. Select the field and value that will trigger the auto-population.

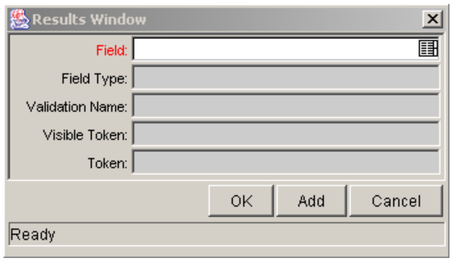
      To make all values for the field trigger the auto-population, select **Yes** next to All Values.

   c. Click **Add** to add the field to the Dependencies area and continue to add more without closing the Dependencies window. Click **OK** to add the field and close the Dependencies window.

9. Select the fields for the rule to auto-populate.

   a. Click **New** in the Results area.

      The Results window opens.



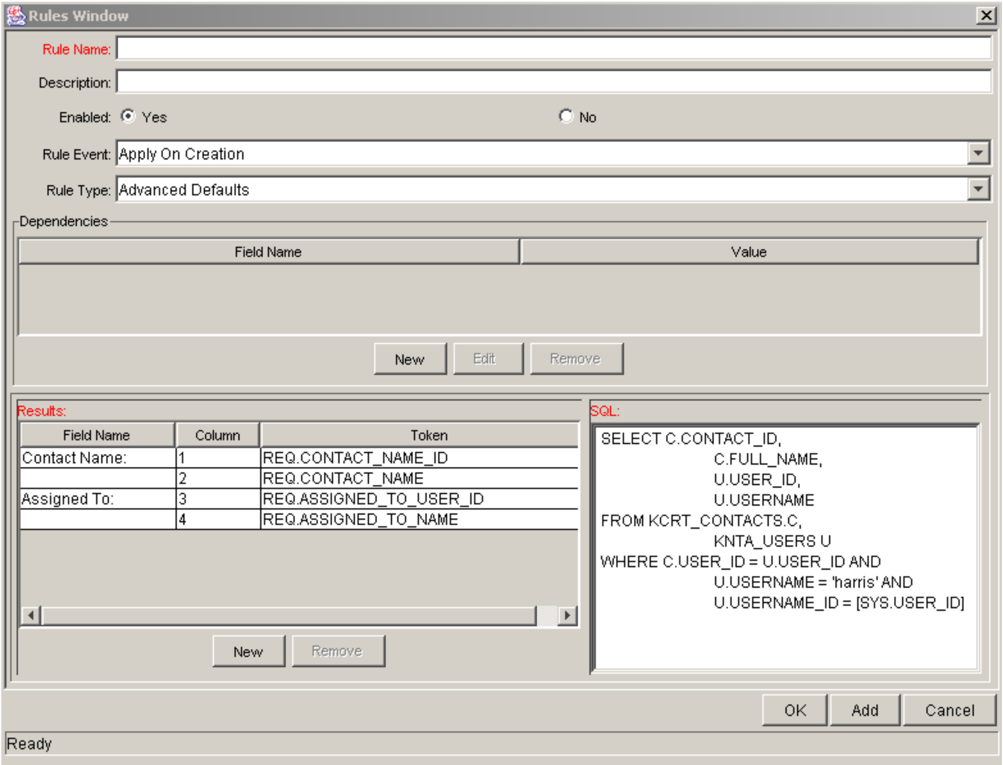
b. Specify the field you want to be populated in the Field auto-complete list.

c. Click **Add** to add the field to the Results area and continue to add more without closing the Results window. Click **OK** to add the field and close the Results window.

The Results area's table displays the Field name, its column number, and Token.

10. In the SQL area, define the SQL statement that will load values into the fields specified in the Results area.

Each "select" value will be loaded into its corresponding column in the Results table in order.



Rule Name:

Description:

Enabled: ● Yes ○ No

Rule Event: Apply On Creation

Rule Type: Advanced Defaults

Field Name	Value

Results:

Field Name	Column	Token
Contact Name:	1	REQ.CONTACT_NAME_ID
	2	REQ.CONTACT_NAME
Assigned To:	3	REQ.ASSIGNED_TO_USER_ID
	4	REQ.ASSIGNED_TO_NAME

SQL:

```
SELECT C.CONTACT_ID,
              C.FULL_NAME,
              U.USER_ID,
              U.USERNAME
FROM KCRT_CONTACTS.C,
              KNTA_USERS U
WHERE C.USER_ID = U.USER_ID AND
              U.USERNAME = 'harris' AND
              U.USERNAME_ID = [SYS.USER_ID]
```

Example

The SQL statement shown above will load C.CONTACT_ID into column 1, C.FULL_NAME into column 2, U.USER_ID into column 3, and U.USERNAME into column 4.

11. Click **Apply** to apply the rule and continue to create more, or click **OK** to apply the rule and close the Rules window.

    The system validates the SQL statement in the SQL area to ensure that it contains the correct Tokens: [SYS] Tokens, [AS] Tokens, or Tokens of fields present in the Dependencies area. If the SQL statement is invalid, an error message will be displayed. See *"Tokens"* on page 357 for more detailed information.

12. Click **Save** in the **Rules** tab to save any changes.

Configuring Field Behavior Using Status Dependencies

During a Request's resolution process, it can acquire different Statuses as it progresses along its Workflow. These Statuses can be used to drive field behavior, linking Workflow processes to specific information in the Request.

The following sections discuss setting up field behavior using Status dependencies in more detail:

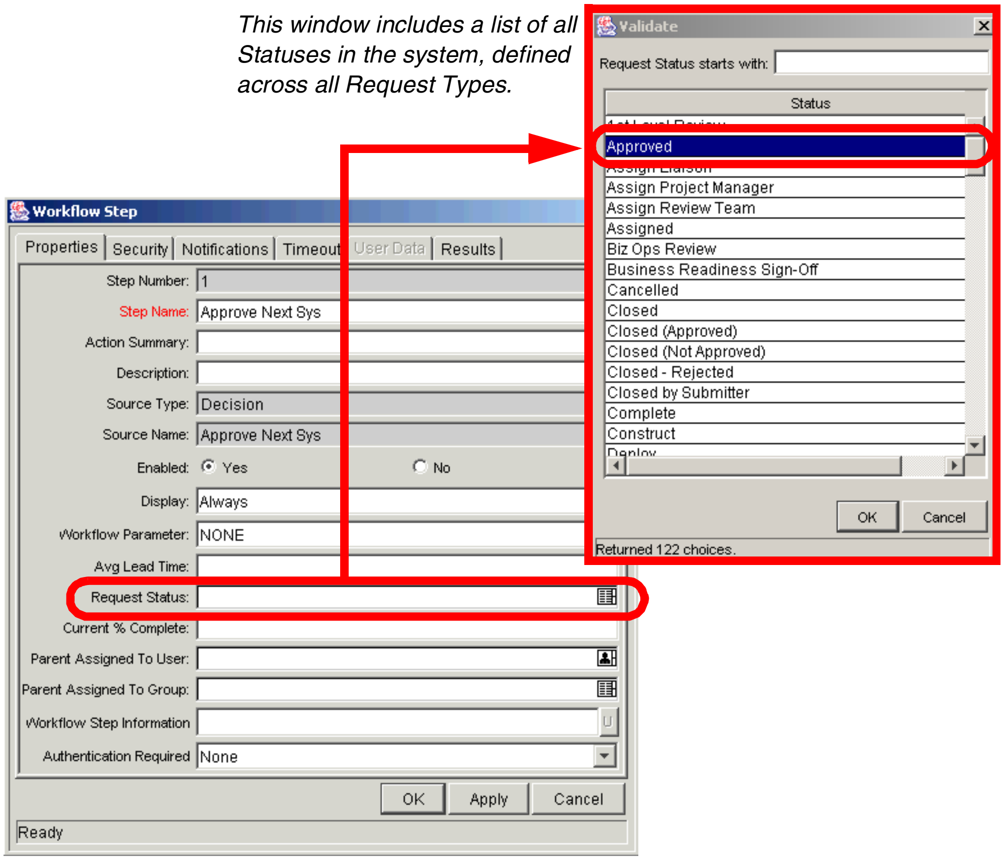
- *Creating Request Statuses*
- *Configuring Field Status Dependency Behavior*
- *Assigning Request Statuses to Workflow Steps*

Creating Request Statuses

Requests can take on different Statuses as they progress along their lifecycle. Some possible Request Statuses include:

- Submitted
- Assigned
- In Progress
- On Hold
- Complete

These Statuses are then linked to the Workflow Steps to drive Request logic. *Figure 7-4* shows how Statuses are linked to Workflow Steps.



This window includes a list of all Statuses in the system, defined across all Request Types.

Validate

Request Status starts with:

Status
1st Level Review
Approved
Assign Liaison
Assign Project Manager
Assign Review Team
Assigned
Biz Ops Review
Business Readiness Sign-Off
Cancelled
Closed
Closed (Approved)
Closed (Not Approved)
Closed - Rejected
Closed by Submitter
Complete
Construct
Deploy

Returned 122 choices.

Workflow Step

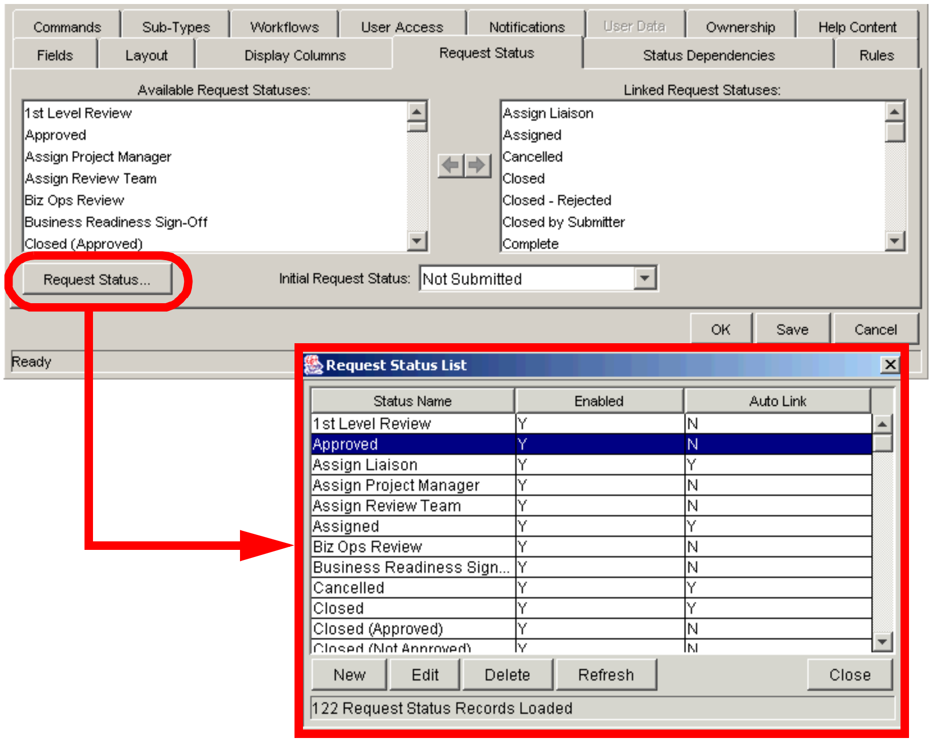
Step Number:	1
Step Name:	Approve Next Sys
Action Summary:	
Description:	
Source Type:	Decision
Source Name:	Approve Next Sys
Enabled:	● Yes ○ No
Display:	Always
Workflow Parameter:	NONE
Avg Lead Time:	
Request Status:	
Current % Complete:	
Parent Assigned To User:	
Parent Assigned To Group:	
Workflow Step Information	
Authentication Required	None

Ready

Figure 7-4 Request Status Specified in the Workflow Step

As a Request moves along this Workflow, its Status changes at particular steps. Each Status can be linked to Request field behavior through the **Status Dependencies** tab. For more information on linking Request Statuses to field behavior, see *"Configuring Field Status Dependency Behavior"* on page 161.

Before linking Request Statuses to Workflow Steps, the Request Type must first possess all desired Statuses. The list of possible Request Statuses is created in the Request Type window's **Request Status** tab. *Figure 7-5* shows the interface for creating Request Statuses.



Available Request Statuses:

1st Level Review
Approved
Assign Project Manager
Assign Review Team
Biz Ops Review
Business Readiness Sign-Off
Closed (Approved)

Linked Request Statuses:

Assign Liaison
Assigned
Cancelled
Closed
Closed - Rejected
Closed by Submitter
Complete

Initial Request Status: Not Submitted

Status Name	Enabled	Auto Link
1st Level Review	Y	N
Approved	Y	N
Assign Liaison	Y	Y
Assign Project Manager	Y	N
Assign Review Team	Y	N
Assigned	Y	Y
Biz Ops Review	Y	N
Business Readiness Sign...	Y	N
Cancelled	Y	Y
Closed	Y	Y
Closed (Approved)	Y	N
Closed (Not Approved)	Y	N

122 Request Status Records Loaded

Figure 7-5 Request Status Tab and Request Status List Window

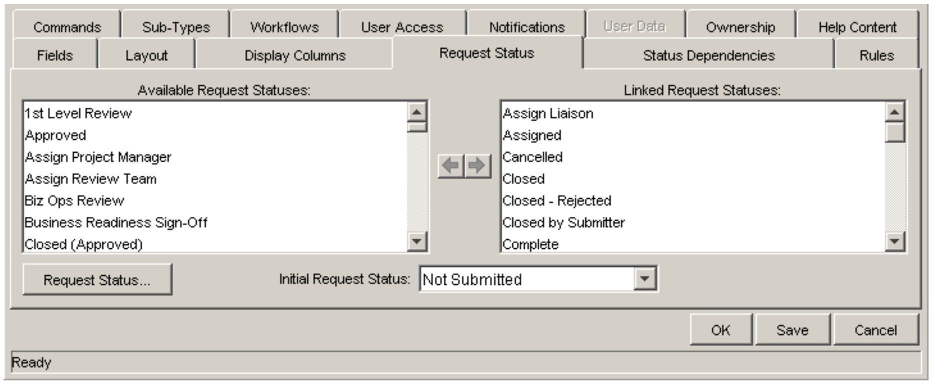
If a desired Status does not appear in the Available Request Statuses list, it can be created.

The Request's initial Status can be set using the Initial Request Status drop down list.

Adding and Linking a Request Status

To add a new Request Status to the list of available Request Statuses:

1. Open the Request Type window and click the **Request Status** tab.



Available Request Statuses:

- 1st Level Review
- Approved
- Assign Project Manager
- Assign Review Team
- Biz Ops Review
- Business Readiness Sign-Off
- Closed (Approved)

Linked Request Statuses:

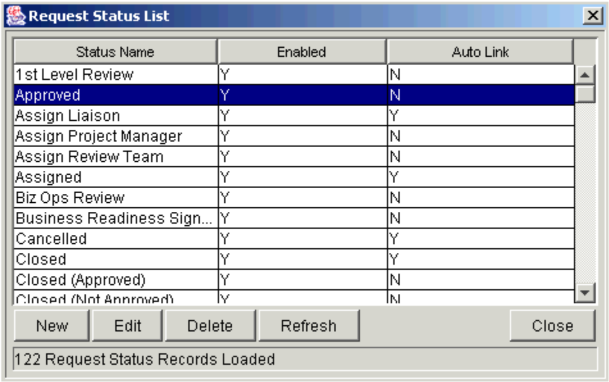
- Assign Liaison
- Assigned
- Cancelled
- Closed
- Closed - Rejected
- Closed by Submitter
- Complete

Initial Request Status: Not Submitted

Ready

2. Click **Request Status**.

   The Request Status List opens.

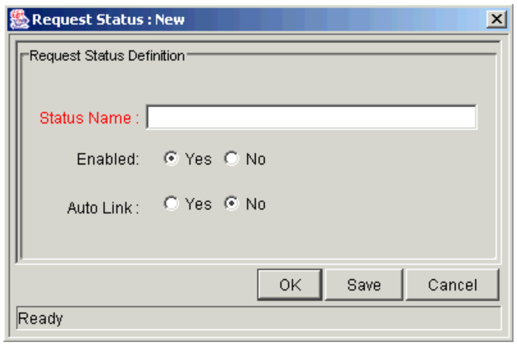


Status Name	Enabled	Auto Link
1st Level Review	Y	N
Approved	Y	N
Assign Liaison	Y	Y
Assign Project Manager	Y	N
Assign Review Team	Y	N
Assigned	Y	Y
Biz Ops Review	Y	N
Business Readiness Sign...	Y	N
Cancelled	Y	Y
Closed	Y	Y
Closed (Approved)	Y	N
Closed (Not Approved)	Y	N

122 Request Status Records Loaded

3. Click **New**.

   The Request Status: New window opens.



Status Name :

Enabled: ● Yes ○ No

Auto Link : ○ Yes ● No

Ready

4. Enter a Status Name.

5. Select Enabled = **Yes** for the status to appear in the Available Request Status column for all new Request Types.

6. Select Auto Link = **Yes** for the status to automatically link to all new Request Types.

7. Click **OK** to save the information and close the Request Status: New window, **Save** to save the information and leave the window open, or **Cancel** to exit the window without saving the changes.

Request Statuses can also be edited and deleted from the Request Status List. The Name, Enabled, and Auto Link fields can be changed when a Status from this list is edited.

Configuring Field Status Dependency Behavior

Certain Request field behavior can be linked directly to the Request Statuses. This is done in the **Status Dependencies** tab of the Request Type window.

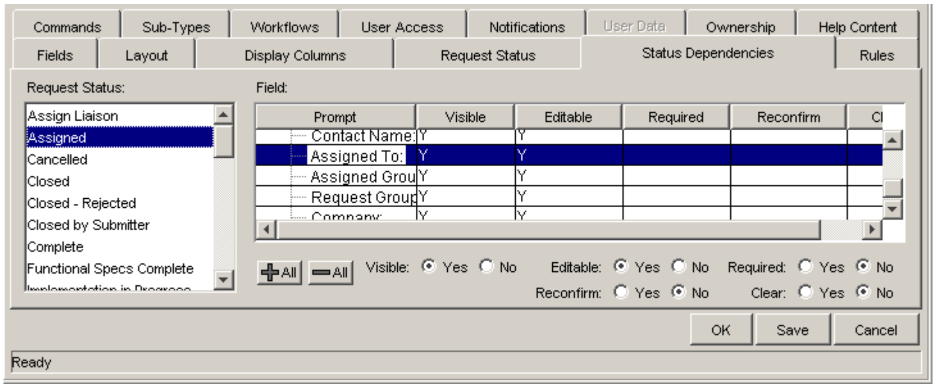
To assign field properties based on Request Status:

1. From the Request Status list, select a Request Status.

2. In the Field table, select a field.

3. Assign the field's attributes under the given Request Status.

   This is done by toggling the radio buttons at the bottom of the screen.



Request Status:

- Assign Liaison
- Assigned
- Cancelled
- Closed
- Closed - Rejected
- Closed by Submitter
- Complete
- Functional Specs Complete
- Implementation in Progress

Field:

Prompt	Visible	Editable	Required	Reconfirm	Cl
Contact Name:	Y	Y			
Assigned To:	Y	Y			
Assigned Grou	Y	Y			
Request Group	Y	Y			
Company:	Y	Y			

Visible: ● Yes ○ No   Editable: ● Yes ○ No   Required: ○ Yes ● No

Reconfirm: ○ Yes ● No   Clear: ○ Yes ● No

Ready

Multiple fields can be configured simultaneously by using the Ctrl or Shift keys to select the fields and then change the attribute values. Select a tab row, such as **Header Fields**, to configure all fields in the tab simultaneously. It is also possible to select multiple statuses and change the same fields if those states require the same attribute values for the same fields.

The following sections discuss each field attribute in more detail:

- *Status Dependencies - Visible*
- *Status Dependencies - Required Field*
- *Status Dependencies - Updateable Field*
- *Status Dependencies - Reconfirm Field*
- *Status Dependencies - Clear Field*
- *Status Dependencies Interactions*

Status Dependencies - Visible

The Visible radio button determines whether or not a field is visible for a specific Request Status. If it is set to Visible = **No**, then the field is hidden.

Status Dependencies - Required Field

When a field is required, it is necessary to enter a value for the field when changes are made to the Request that would affect the Request Status.

Example

A Request is not to be allowed to reach the “Assigned” Request Status unless the Assigned To User field has a value. Additionally, if a Request is at a status of “Assigned,” a user cannot clear the Assigned To User field.

In order to make this work, the Assigned To User field is set to the following parameters for the “Assigned” status:

- Visible = **Yes**
- Editable = **Yes**
- Required = **Yes**
- Reconfirm - **No**
- Clear = **No**

Status Dependencies - Updateable Field

If a field is set to Updateable = **No** for a specific Request Status, then it is not possible to edit the field at the given Request Status. If a field is set up as Required, Reconfirm, or Clear, it must be set to Updateable = **Yes**.

At certain stages in a Request Resolution process, it may be desirable to ensure that specific fields do not get updated. For example, when a Request of type "Vendor Bug" is at the status "Patch Applied," it may be desirable to make sure that the Patch Number field is not updated. This logic is controlled at the Request Type level. For each Request Type, it is possible to determine which Request fields are updateable and non-updateable when a Request is at each possible Request Status.

When a field of a Request is non-updateable due to this logic, the field is grayed out in the Request. The value is visible but cannot be changed.

Status Dependencies - Reconfirm Field

When a field in the Request Type is set to Reconfirm = **Yes**, it is presented to the user before the Request moves to the next step in the Workflow. The contents of these fields can then be reviewed and changed.

Status Dependencies - Clear Field

The Clear flag is used in conjunction with other dependencies to remove the contents of a field. The basic uses of the Clear flag are:

- When Clear is set to **Yes** and the Required and Reconfirmed are set to **No**, the field is not presented to the user or cleared entering this status, but the contents of that field are cleared before moving to the next step in the Workflow.
- Any fields that have the Clear, Required, and Reconfirmed enabled cause the field to show up in red, but cleared. Appropriate values must then be entered.
- All of the Clear events are logged in the Request Notes section as a status change from the old value to " "; if a new value for that field is chosen, then the new value is indicated in the Notes.

Status Dependencies Interactions

Table 7-18 illustrates the results of different combinations of the Required, Reconfirm, and Clear functions.

Table 7-18. Status Dependencies Interactions

Dependencies			Results at Given Status		
Required	**Reconfirmed**	**Clear**	**Display**	**Color**	**Data Shown**
No	No	No	No	N/A	N/A
No	No	Yes	No	N/A	N/A
No	Yes	No	Yes	Black	Current Data
No	Yes	Yes	Yes	Black	None
Yes	No	No	Yes, if NULL	Red	None
Yes	No	Yes	Yes	Red	None
Yes	Yes	No	Yes	Red	Current Data
Yes	Yes	Yes	Yes	Red	None

Note

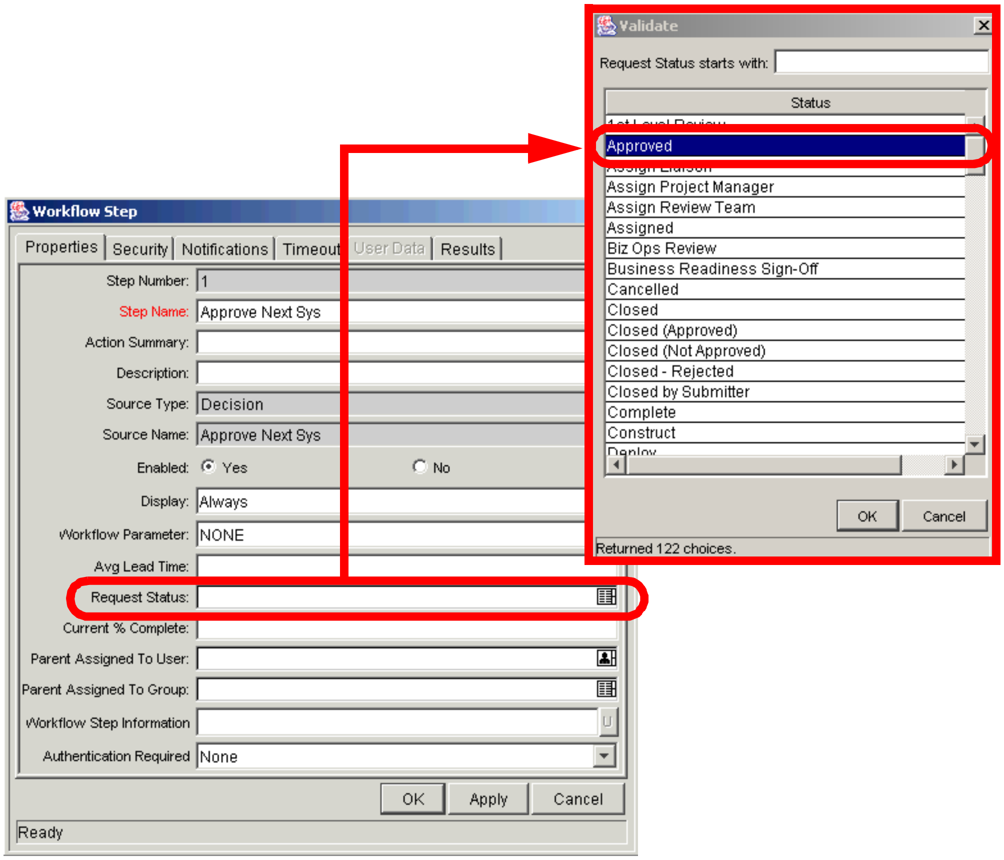
> For each Request Status within a Request Type, there can be up to a maximum of 250 fields with a Required state and 250 fields with a Reconfirm state.

Assigning Request Statuses to Workflow Steps

The final step in linking Request field logic to a Workflow is to assign the appropriate Request Statuses to their respective Workflow Steps.

To assign a Request Status to a Workflow Step:

1. Open the Workflow window.
2. Click the **Layout** tab.
3. Double-click on the desired Workflow Step.
4. Select the desired Request Status from the Request Status auto-complete list.



Step Number:	1
Step Name:	Approve Next Sys
Action Summary:	
Description:	
Source Type:	Decision
Source Name:	Approve Next Sys
Enabled:	● Yes ○ No
Display:	Always
Workflow Parameter:	NONE
Avg Lead Time:	
Request Status:	
Current % Complete:	
Parent Assigned To User:	
Parent Assigned To Group:	
Workflow Step Information	
Authentication Required	None

Ready

Request Status starts with:

1st Level Review
Approved
Assign Liaison
Assign Project Manager
Assign Review Team
Assigned
Biz Ops Review
Business Readiness Sign-Off
Cancelled
Closed
Closed (Approved)
Closed (Not Approved)
Closed - Rejected
Closed by Submitter
Complete
Construct
Deploy

Returned 122 choices.

5. Repeat as needed with all necessary Workflow Steps.

6. Save the Workflow.

Note

Not all Workflow Steps need to have a Request Status assigned. A Request retains the last-encountered Status.

As the Request progresses through this Workflow, it will take on the Status assigned in each Workflow Step.

Adding Custom Online Help to the Request Type

It is possible to provide accessible online information for users who are processing the Requests. Configure the Request type to display additional, custom information about the Request, sections or fields.

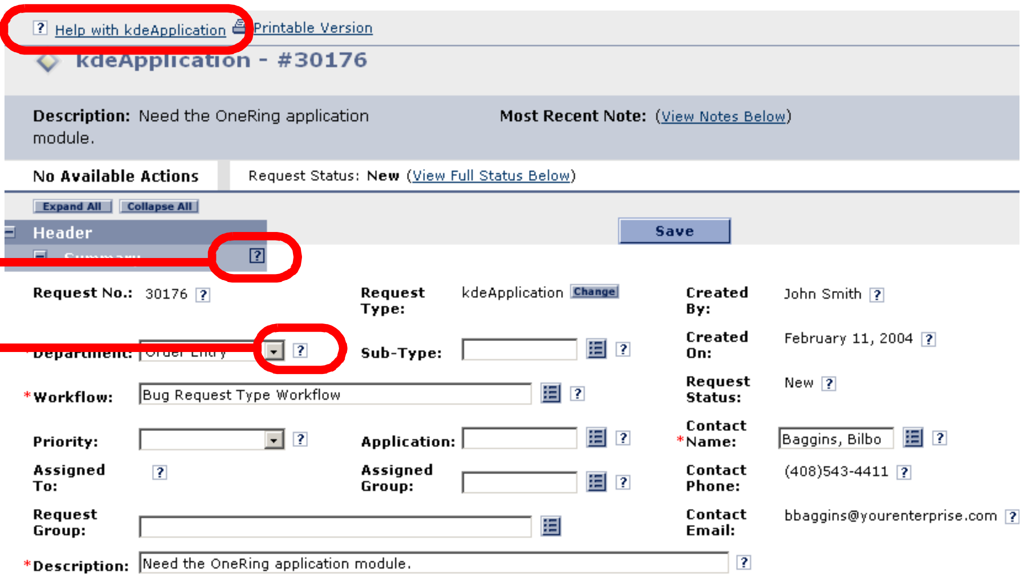
This section covers the following topics:

- *Accessing Custom Online Help*
- *Adding Help to the Request*

Accessing Custom Online Help

Mercury Demand Management configuration experts can define online help content for end users who are processing a specific Request. Online help can be configured for the following areas on a Request:

- Request Type Level—This help content can provide a general description of when a particular Request Type should be used. This online help is access from the top of the Request page, as shown in *Figure 7-6*.
- Request Section Level—This help content can provide a general description of the fields in a section of the Request. This can help end users determine if they need to fill in the fields within a particular section of the Request. This online help is accessed from the section title on the Request page, as shown in *Figure 7-6*.
- Request Field Level—This help content can provide a specific description of a field in the Request. For example, a Request might include a Priority field to specify the urgency of a particular Request. The values in the Priority field are **Critical**, **High**, **Medium**, and **Low**. It is possible to configure help content for that field to instruct the user when to mark a Request as **Critical** rather than **High**. This online help is accessed from the field on the Request, as shown in *Figure 7-6*.



Click to access Request level help.

Click to access section level help.

Click to access field level help.

? Help with kdeApplication Printable Version

kdeApplication - #30176

Description: Need the OneRing application module.

Most Recent Note: (View Notes Below)

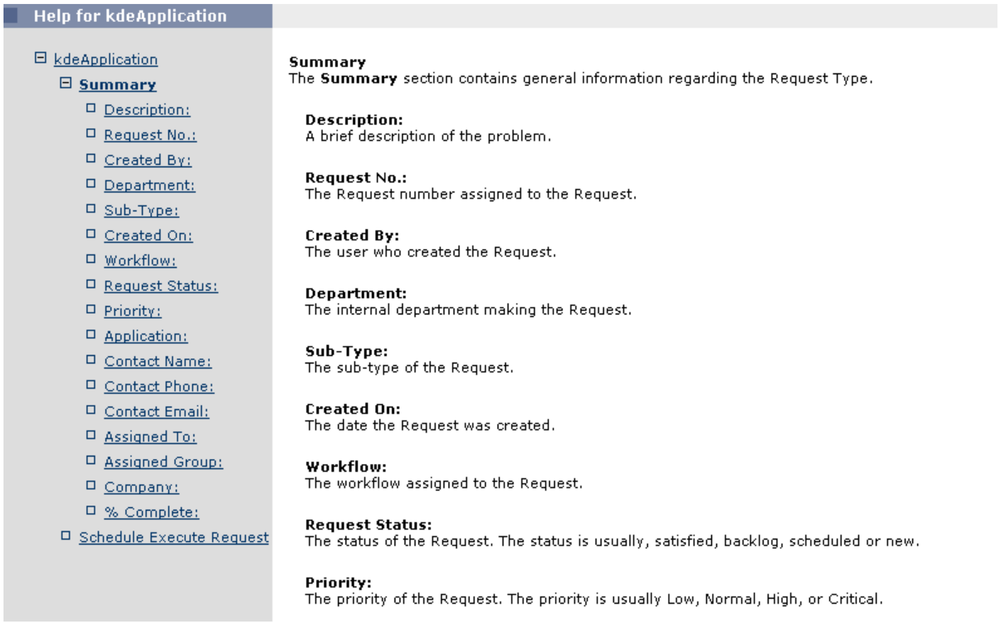
No Available Actions

Request Status: **New** (View Full Status Below)

Request No.:	30176	**Request Type:**	kdeApplication	**Created By:**	John Smith
Department:	Order Entry	**Sub-Type:**		**Created On:**	February 11, 2004
***Workflow:**	Bug Request Type Workflow			**Request Status:**	New
Priority:		**Application:**		***Contact Name:**	Baggins, Bilbo
Assigned To:		**Assigned Group:**		**Contact Phone:**	(408)543-4411
Request Group:				**Contact Email:**	bbaggins@yourenterprise.com
***Description:**	Need the OneRing application module.				

Figure 7-6 Accessing the Section and Field Level Help

Click on the help icon (question mark) to view the help defined for a particular Request Type, section, or field. The help icon is only displayed when there has been help configured for that Request, section or field. The online help interface is shown in *Figure 7-7*.



Summary
The **Summary** section contains general information regarding the Request Type.

Description:
A brief description of the problem.

Request No.:
The Request number assigned to the Request.

Created By:
The user who created the Request.

Department:
The internal department making the Request.

Sub-Type:
The sub-type of the Request.

Created On:
The date the Request was created.

Workflow:
The workflow assigned to the Request.

Request Status:
The status of the Request. The status is usually, satisfied, backlog, scheduled or new.

Priority:
The priority of the Request. The priority is usually Low, Normal, High, or Critical.

Figure 7-7 Request Help Interface

Adding Help to the Request

Custom online help can be added to a Request, Request section, and Request Fields from the Request Type Workbench.

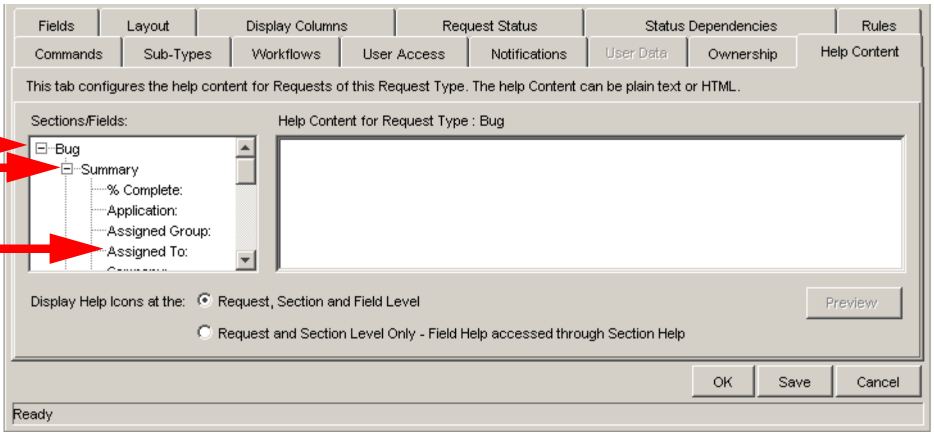
To add custom online help to the Request Type:

1. From the Request Type Workbench, open the Request Type to configure.

2. Click the **Help Content** tab.

3. In the Sections/Fields section, select the item to which content will be added.

Select the Request name, Section name, or Field name to define help for that item.



This tab configures the help content for Requests of this Request Type. The help Content can be plain text or HTML.

Sections/Fields:

- Bug
  - Summary
    - % Complete:
    - Application:
    - Assigned Group:
    - Assigned To:

Help Content for Request Type : Bug

Display Help Icons at the: ● Request, Section and Field Level

○ Request and Section Level Only - Field Help accessed through Section Help

4. In the Help Content for Request Type/Section/Field section, enter the help content for the selected item.

   Enter plain text or HTML-formatted text. To see what the text looks like in the actual help display, click **Preview**.

5. (Optional) Select other sections or fields to define help content for those items.

6. From the Display Help Icons at the: field, specify how the help icons will be shown in the standard interface.

   - **Request, Section and Field Level**—Display a help icon (question mark) beside each Request, section and field that has associated help content.

   - **Request and Section Level Only**—Does not display the help icon at the individual field level. Any help content defined for the fields can be accessed from the section level help.

7. Click **Save**.

The online help can now be accessed from the standard interface. See *"Accessing Custom Online Help"* on page 166 for additional details.

Modifying the Request Layout

Modifying the layout of a Request Type can consist of the following activities:

- *Modifying the Request Type Field Width*
- *Moving Fields in a Request Type*
- *Adding Sections to the Request Type*

Modifying the Request Type Field Width

To change the column width of a field:

1. Open the Request Type.
2. Click the **Layout** tab.
3. Select the section of the Request that contains the field.
4. Select the field.
5. In the Field Width drop down list, select a width of **1**, **2**, or **3**.

Note

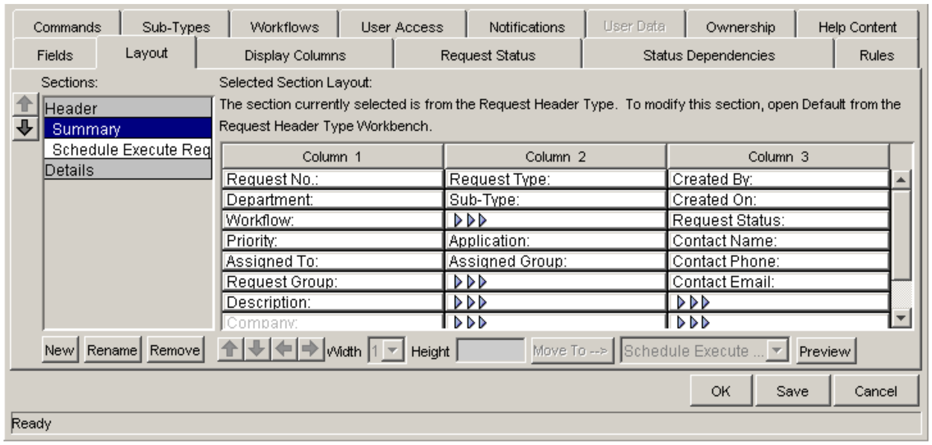
The field width has to correspond to the column location. For example, a field located in column two cannot have a width set to **3**.

Additionally, for fields of component type Text Area, it is possible to determine the number of lines the Text Area will display. Select the field and change the value in the Component Lines field. If the selected field is not of type Text Area, this attribute will be blank and non-updateable.

Moving Fields in a Request Type

To move a field or a set of fields:

1. Open the Request Type.
2. Click the **Layout** tab.



Sections:

- Header
  - Summary
  - Schedule Execute Req
- Details

Selected Section Layout:

The section currently selected is from the Request Header Type. To modify this section, open Default from the Request Header Type Workbench.

Column 1	Column 2	Column 3
Request No.:	Request Type:	Created By:
Department:	Sub-Type:	Created On:
Workflow:	▷▷▷	Request Status:
Priority:	Application:	Contact Name:
Assigned To:	Assigned Group:	Contact Phone:
Request Group:	▷▷▷	Contact Email:
Description:	▷▷▷	▷▷▷
Company:	▷▷▷	▷▷▷

Width 1 · Schedule Execute ...

Ready

3. Select the section of the Request that contains the field(s) to move.

4. Select the field(s).

   To select more than one field, press the Shift key while selecting the last field in the set. Selection is either singular or a sequential group. Only groups of adjacent fields can be selected. Blank cells can be selected as part of the group.

5. Move the fields to the desired location in the layout builder, either by clicking the arrow buttons or using the corresponding keyboard arrow keys.

If a Request Type has multiple sections in its field layout, fields can be moved from one section to another.

To move a field to a different section:

1. Select the field.

2. If there is more than one section, select the desired section from the **Move To -->** drop down list.

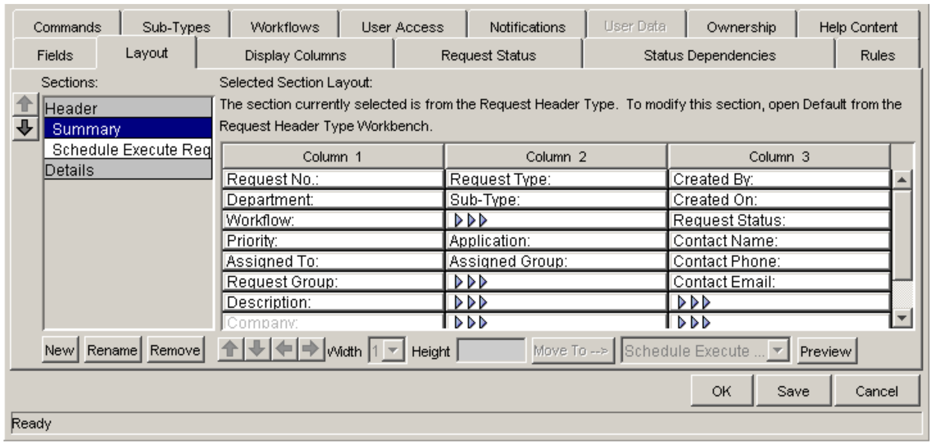
3. Click **Move To -->**.

Adding Sections to the Request Type

The layout for a new Request Type defaults to have only one section for the custom fields.

To add a new section:

1. Open the Request Type.
2. Click the **Layout** tab.



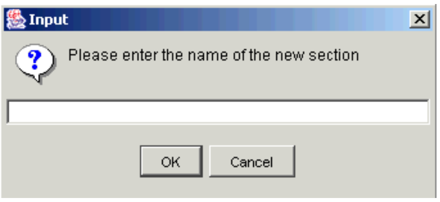
Selected Section Layout:

The section currently selected is from the Request Header Type. To modify this section, open Default from the Request Header Type Workbench.

Column 1	Column 2	Column 3
Request No.:	Request Type:	Created By:
Department:	Sub-Type:	Created On:
Workflow:	▷▷▷	Request Status:
Priority:	Application:	Contact Name:
Assigned To:	Assigned Group:	Contact Phone:
Request Group:	▷▷▷	Contact Email:
Description:	▷▷▷	▷▷▷
Company:	▷▷▷	▷▷▷

3. Click **New** in the Sections area (on the left side of the window).

   The Input window opens.



Please enter the name of the new section

4. Enter a new section name.

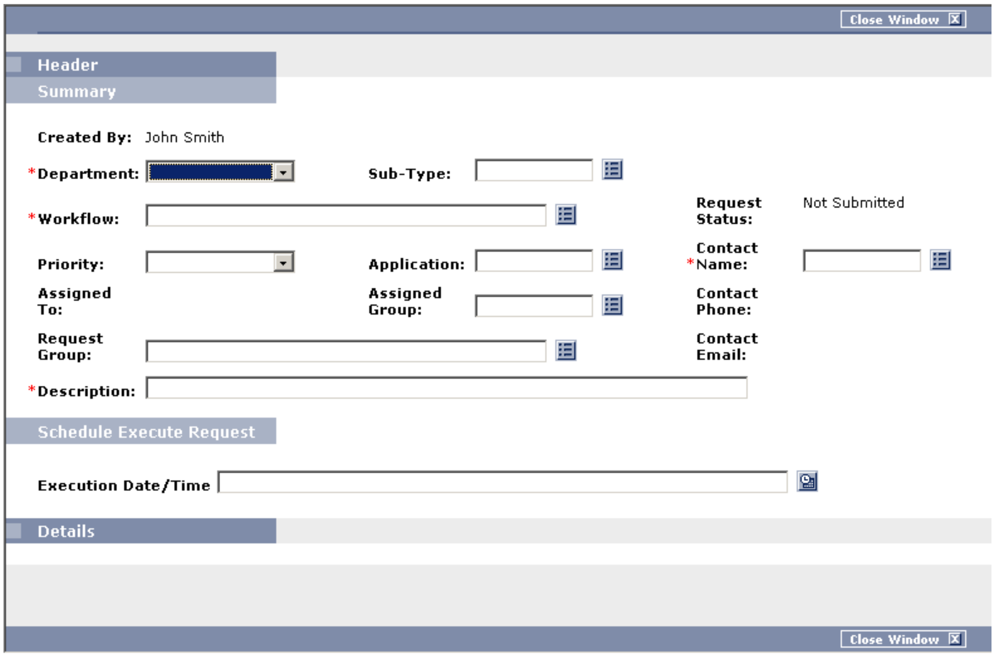
   Custom section names can be up to 30 characters in length.

   When Requests are generated for the given Request Type, the new section with the defined custom fields will be visible.

5. To view what the layout will look like to the user processing the Request, click **Preview**. This opens an HTML window that shows the fields as they will appear.

Note

- If all the fields have a width of one column and are all in the same column, all displayed columns automatically span the entire available area when a Request of the given Request Type is viewed or edited.
- Any non-displayed fields do not affect the layout. The layout engine considers them the same as a blank field.



Created By: John Smith

*Department:		Sub-Type:			
*Workflow:				Request Status:	Not Submitted
Priority:		Application:		Contact *Name:	
Assigned To:		Assigned Group:		Contact Phone:	
Request Group:				Contact Email:	
*Description:					

Execution Date/Time

Figure 7-8 Request Field Layout Preview

It is possible to change the section names later.

To change the name of a section:

1. In the Request Type **Layout** tab, select the section name to be changed.
2. Click **Rename** and enter a new value in the Input window that appears.

The change will be reflected immediately.

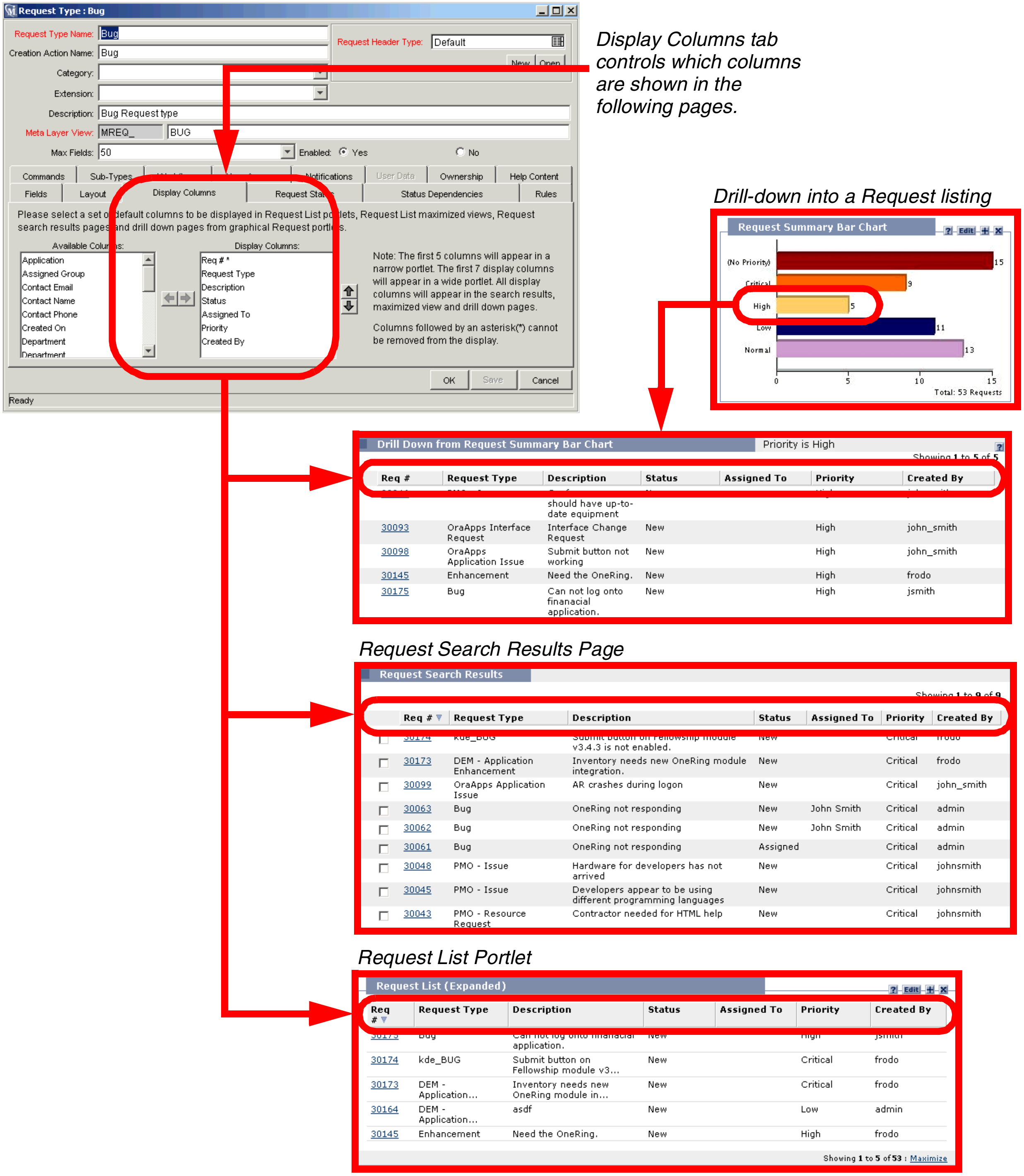
Configuring the Request Columns Displayed in Portlets and Search Result Pages

Certain information in the Request can provide a useful summary-level description of the Request. This can include information such as the Request Type, a description of the Request, and a priority. For each Request Type, it is possible to control which Request columns are displayed to the user in the following pages:

- Request list Portlets
- Request Search Results page
- Request drill-down pages accessed by clicking on graphical Request Portlets

Users can view the displayed information on these pages to decide if they need to view the details of a specific Request.

Figure 7-9 shows how the settings in the Request Type window control the columns that are displayed in these windows.



Request Type : Bug

Request Type Name:	Bug
Creation Action Name:	Bug
Category:	
Extension:	
Description:	Bug Request type
Meta Layer View:	MREQ_ BUG
Max Fields:	50
Request Header Type:	Default
Enabled:	● Yes ○ No

Commands | Sub-Types | Notifications | User Data | Ownership | Help Content | Fields | Layout | Display Columns | Request Status | Status Dependencies | Rules

Please select a set of default columns to be displayed in Request List portlets, Request List maximized views, Request search results pages and drill down pages from graphical Request portlets.

Available Columns:
Application
Assigned Group
Contact Email
Contact Name
Contact Phone
Created On
Department
Department

Display Columns:
Req # *
Request Type
Description
Status
Assigned To
Priority
Created By

Note: The first 5 columns will appear in a narrow portlet. The first 7 display columns will appear in a wide portlet. All display columns will appear in the search results, maximized view and drill down pages.

Columns followed by an asterisk(*) cannot be removed from the display.

OK | Save | Cancel

Ready

Display Columns tab controls which columns are shown in the following pages.

Drill-down into a Request listing

Drill Down from Request Summary Bar Chart Priority is High

Showing **1** to **5** of **5**

Req #	Request Type	Description	Status	Assigned To	Priority	Created By
		should have up-to-date equipment				
30093	OraApps Interface Request	Interface Change Request	New		High	john_smith
30098	OraApps Application Issue	Submit button not working	New		High	john_smith
30145	Enhancement	Need the OneRing.	New		High	frodo
30175	Bug	Can not log onto finanacial application.	New		High	jsmith

Request Search Results Page

Request Search Results

Showing **1** to **9** of **9**

	Req #	Request Type	Description	Status	Assigned To	Priority	Created By
□			v3.4.3 is not enabled.				
□	30173	DEM - Application Enhancement	Inventory needs new OneRing module integration.	New		Critical	frodo
□	30099	OraApps Application Issue	AR crashes during logon	New		Critical	john_smith
□	30063	Bug	OneRing not responding	New	John Smith	Critical	admin
□	30062	Bug	OneRing not responding	New	John Smith	Critical	admin
□	30061	Bug	OneRing not responding	Assigned		Critical	admin
□	30048	PMO - Issue	Hardware for developers has not arrived	New		Critical	johnsmith
□	30045	PMO - Issue	Developers appear to be using different programming languages	New		Critical	johnsmith
□	30043	PMO - Resource Request	Contractor needed for HTML help	New		Critical	johnsmith

Request List Portlet

Request List (Expanded)

Req #	Request Type	Description	Status	Assigned To	Priority	Created By
		application.				
30174	kde_BUG	Submit button on Fellowship module v3...	New		Critical	frodo
30173	DEM - Application...	Inventory needs new OneRing module in...	New		Critical	frodo
30164	DEM - Application...	asdf	New		Low	admin
30145	Enhancement	Need the OneRing.	New		High	frodo

Showing **1** to **5** of **53** : Maximize

Figure 7-9 Displayed columns set in the Request Type window

To configure the columns that are displayed in Portlets and search result pages:

1. In the Request Type window, click the **Display Columns** tab.
2. From the Available Columns list, select the columns to display.

   Select multiple columns by pressing the Ctrl key while clicking different items.
3. Click the right arrow.

   The selected items are moved into the Display Columns list.
4. Remove any columns that you do not want to display from the Display Columns list.
5. Click **Save**.

Note

In Request Portlets, this setting represents the default columns that are displayed in the Portlet. The user can select to display alternate columns when personalizing the Portlet.

Similarly this setting represents the default columns that are displayed when using the Advanced Search functionality in the Request List Portlet or Request Search Results page.

Chapter 8 Integrating Participants into Your Request Resolution System

This chapter provides an overview for how to integrate users into a Request resolution process. It provides information for defining Security Groups and controlling users' access to actions in Mercury Demand Management.

This chapter covers the following topics:

- *User Security and Participation - Overview*
- *Establishing Security Groups*
- *Prerequisite Settings for Users and Security Groups*
- *Setting Request Creation Security*
- *Setting Request Processing Security*
- *Setting Configuration Security*

Note

Only users with an Administrator license can create User accounts and Security Groups, which are critical when integrating participants into a Request resolution system. Work with your Administrator to configure the User accounts and Security Groups required for the process.

User Security and Participation - Overview

Mercury Demand Management enables a great deal of control over who can participate in a Request resolution process. User actions can be restricted around:

- **Request creation:**
    - Who can create Requests.
    - Who can use a specific Workflow.
    - Who can use specific Request Types.
- **Request processing:**
    - Who can act on each step in the Workflow.

      For this restriction, access can be enabled by specifying users or Security Groups. Access can also be provided dynamically by having a Token resolve to provide access.
    - Who can view or edit certain fields in a Request.

      For this restriction, enable view or edit access to Request fields by specifying users or Security Groups. Access can also be provided dynamically by having a Token resolve to provide access.
- **Managing your Request resolution process:**
    - Who can change the Workflow.
    - Who can change each Request Type.

Tip

Use Security Groups or dynamic access (Tokens) whenever possible. Avoid specifying a list of users to control an action; for example, specifying a list of users who can act on a Workflow step. If the list of users changes (due to any departmental reorganization), that list would need to be updated manually in many places on the Workflow. By using a Security Group instead of a list of users, the Security Group can be updated once, and the changes will be propagated throughout the Workflow steps.

Establishing Security Groups

Security Groups are used to control who can access certain screens and functionality in Mercury Demand Management. See *"Security Groups"* on page 15 for an overview. The following sections provide instructions on defining Security Groups:

- *Creating a Security Group by Specifying a List of Users*
- *Using Mercury's Resource Management to Control User Security*

The general process for creating a Security Group is as follows:

1. Specify Security Group membership on the **Users** tab.

   This can be accomplished by providing a list of users or by associating the group with an organization unit defined in Mercury IT Governance Center.

2. Specify the screen and feature access by linking the appropriate Access Grants.

   See *"Access Grants"* on page 271 for details.

Note

Consider creating and maintaining two types of Security Groups:

- Security Groups to control who can act on a specific Workflow step (list of users without any special Access Grants)
- Security Groups to control who can access a particular screen or function (list of users and appropriate Access Grants)

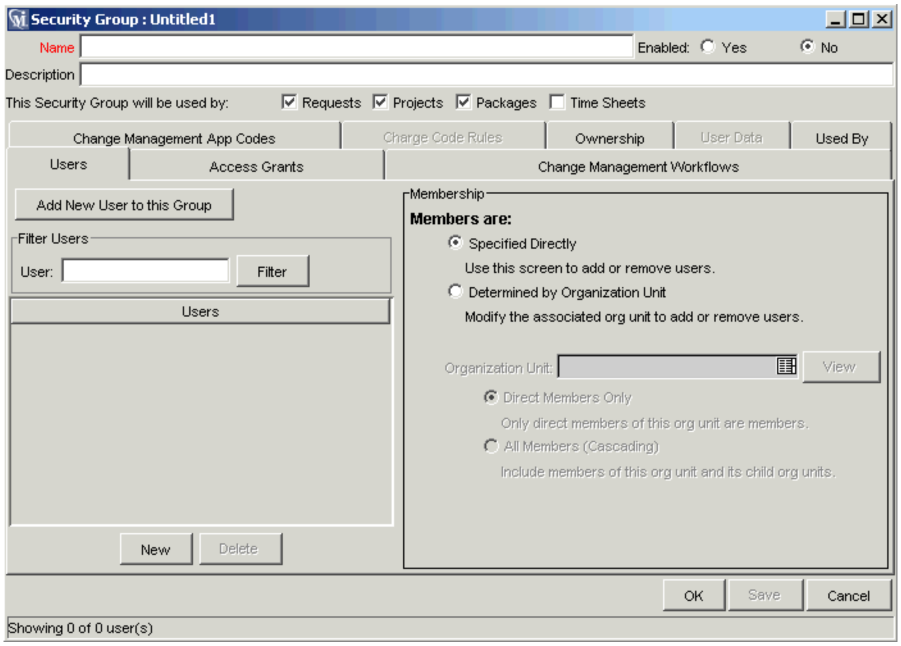
This can greatly simplify the maintenance of a security model around your processes. As new users are added to the system, they can be granted to appropriate screen and function access and associated with specific Workflows.

Creating a Security Group by Specifying a List of Users

To generate and define a new Security Group:

1. Click **New Security Group** in the Security Group Workbench, or select **File -> New -> Security Group** from the menu.

   The Security Group window opens.



Security Group : Untitled1

Name | Enabled: ○ Yes ● No

Description

This Security Group will be used by: ☑ Requests ☑ Projects ☑ Packages □ Time Sheets

Membership

Members are:

● Specified Directly

Use this screen to add or remove users.

○ Determined by Organization Unit

Modify the associated org unit to add or remove users.

Organization Unit:

● Direct Members Only

Only direct members of this org unit are members.

○ All Members (Cascading)

Include members of this org unit and its child org units.

Showing 0 of 0 user(s)

2. Enter the Name and Description.

3. Select **Yes** to enable this Security Group.

   If the Security Group is not enabled, it does not appear as a choice when generating or updating users or Workflows.

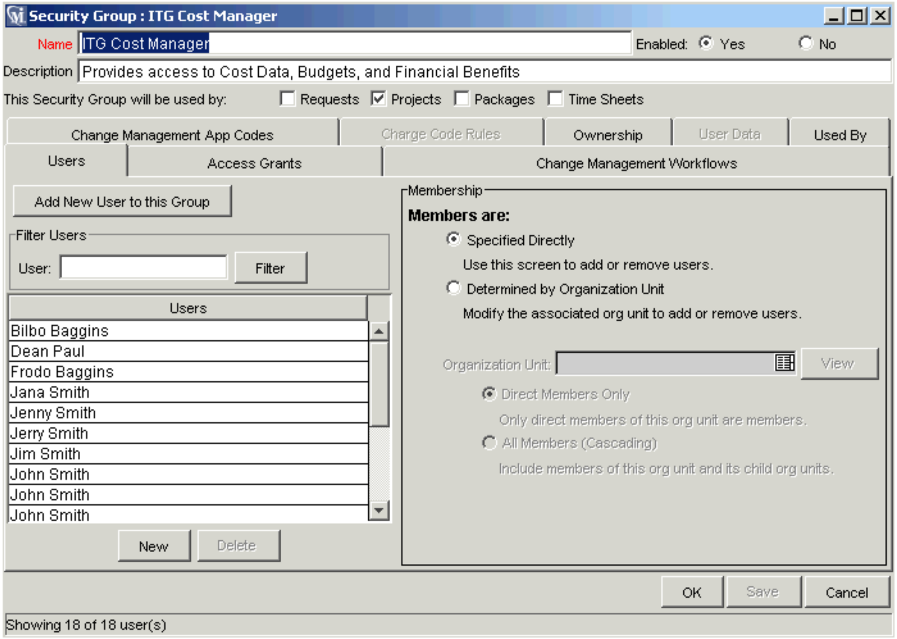
4. Select which entities (Requests, Projects or Packages) will use the Security Group by clicking their respective check boxes in the This Security Group will be used by field.

5. Link the desired Users to the Security Group.

   a. Click **Add New User to this Group** in the **Users** tab.

      The Users window opens.

   b. Select the desired usernames from the Users field and click **OK** to add your selection to the **Users** tab.



Security Group : ITG Cost Manager

Name: ITG Cost Manager

Enabled: ● Yes ○ No

Description: Provides access to Cost Data, Budgets, and Financial Benefits

This Security Group will be used by: □ Requests ☑ Projects □ Packages □ Time Sheets

Membership

Members are:

● Specified Directly

Use this screen to add or remove users.

○ Determined by Organization Unit

Modify the associated org unit to add or remove users.

Organization Unit:

● Direct Members Only

Only direct members of this org unit are members.

○ All Members (Cascading)

Include members of this org unit and its child org units.

Users
Bilbo Baggins
Dean Paul
Frodo Baggins
Jana Smith
Jenny Smith
Jerry Smith
Jim Smith
John Smith
John Smith
John Smith

Showing 18 of 18 user(s)

6. Link the desired Access Grants.

   Each Access Grant enables certain functions performed on a window or page. See *Security Model Guide and Reference* for a description of each available Access Grant.

   a. Select the desired Access Grants in the Available Access Grants list.

   b. Click the right arrow button pointing to the Linked Access Grants list.

      The selected Access Grants are moved into the column.

7. (Optional) To use the same Security Group for processing Packages in Mercury Change Management, specify which Workflows the Security Group can use in the **Change Management Workflows** tab.

   See *Security Model Guide and Reference* for more information about allowing or restricting Change Management Workflows in Security Groups.

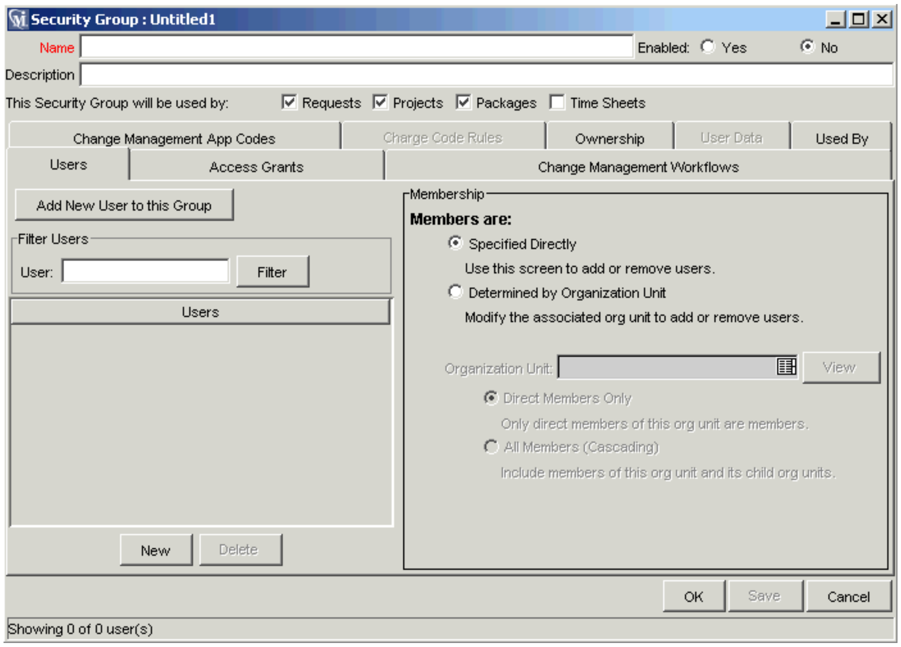
8. Click the **Ownership** tab and select the Ownership Groups that have the right to edit, copy or delete the current Security Group. See *"Setting Ownership for Security Groups"* on page 92 for more information about setting Ownership for a new or existing Security Group.

9. (Optional) Enter any necessary information in the **User Data** tab's custom fields.

10. Click **OK** to save the current Security Group and close the Security Group window.

Using Mercury's Resource Management to Control User Security

Users can also be associated to Security Groups through their inclusion in an organization model definition. Using the Mercury IT Governance Center's resource management capabilities, a user can be placed into a model that includes security and access information. See *Managing Your Resources (Resource Management)* for details.



Security Group : Untitled1

Name

Enabled: ○ Yes ● No

Description

This Security Group will be used by: ☑ Requests ☑ Projects ☑ Packages □ Time Sheets

Membership

Members are:

● Specified Directly

Use this screen to add or remove users.

○ Determined by Organization Unit

Modify the associated org unit to add or remove users.

Organization Unit:

● Direct Members Only

Only direct members of this org unit are members.

○ All Members (Cascading)

Include members of this org unit and its child org units.

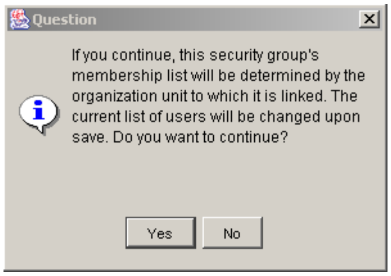
Showing 0 of 0 user(s)

To define a Security Group to use the members of an organization unit:

1. Open the Security Group window.

2. Select Determined by Organization Unit in the Membership section of the **Users** tab.

   The following question dialog opens.


Question

If you continue, this security group's membership list will be determined by the organization unit to which it is linked. The current list of users will be changed upon save. Do you want to continue?

3. Click **Yes**.

Note

When an Organization Unit has been selected to control user access to the Security Group, any users specified in the Users list will be replaced with the members of the organization unit.

4. Select the Organization Unit.

5. Select whether to include:

   - Direct Members Only:
     Only direct members of the specified organization unit.

   - All Members (cascading)
     Members of this organization unit and its child units.

6. Click **Save**.

Related Topics:

- *Managing Your Resources (Resource Management)*
- *Security Model Guide and Reference*

Prerequisite Settings for Users and Security Groups

General access to Request Types and certain functions related to processing Requests are controlled by Access Grants associated with Security Groups. Users in those Security Groups are then given all functionality enabled by those Access Grants.

This section lists the Access Grants and Security Group settings that should be set to enable general access to Request processing.

Note

Only users with an Administrator license can create or modify User and Security Group accounts. Work with the administrator to provide users with the basic settings required to process Requests. Process and data restrictions can later be implemented using settings in the Workflow and Request Type definitions.

Licenses

Users must have one of the following product licenses in order to create and process Requests:

- Demand Management: Standard License
  The Standard License provides a user with access to the standard interface, where the Request is created and processed.
- Demand Management: Power License
  The Power License provides a user with all of the access included in the Standard License. In addition, it provides the user with access to the Workbench and advanced functionality such as deleting Requests.

See *Security Model Guide and Reference* for details on what functionality is enabled with each license. Also, the following sections discuss how the functionality provided with each Access Grant differs depending on the license type that the user has.

Access Grants

Table 8-1 lists the common Access Grants used to provide general access to Request processing functionality. The table also discusses how the functionality provided with each Access Grant differs depending on the license type that the user has (Standard or Power). Users must have some of these Access Grants to access basic functionality.

Table 8-1. Access Grants Related to Request Creation and Processing

Access Grant	Description
Demand Mgmt: Edit Requests	Perform basic Request processing actions: create Requests, edit certain Requests, and delete your un-submitted Requests depending on your license (standard versus power). Standard License: • Allows the user to generate Requests. • User cannot change the Workflow when creating or editing a Request. • Allows the user to edit the Request as specified in the **User Access** tab on the Request Type window. Power License: • Has the same permissions as those listed for Standard License. • Allows the user to delete the Request as specified in the **User Access** tab on the Request Type window. • Allows the user to cancel the Request as specified in the **User Access** tab on the Request Type window.
Demand Mgmt: Manage Requests	Perform advanced Request processing actions: creating, editing, deleting, changing the Request's Workflow, and overriding references. This access enables different functions depending on your license (standard versus power). Standard License: • User always has permission to edit the Request. • Override and/or remove any References on any Request. Power License: • Has the same permissions as those listed for Standard License. • User always has permission to delete or cancel a Request. • User can change the Workflow when creating and editing a Request.
Demand Mgmt: Override Participant Restriction	This Access Grant allows the user to view a Request regardless of whether the user is its creator, the Assigned To user, a member of the Assigned Group or a member of the Workflow Steps Security Group.

Tip

Screen and function access provided through Access Grants are cumulative. If a user belongs to three different Security Groups, he will have all access provided to each of the groups. Therefore, to restrict certain screen and feature access, remove the user from any Security Group that grants that access.

Use the **Access Grants** tabs in the User window to see all Security Groups where specific Access Grants are included, then:

- Remove the user from the Security Group (using the **Security Group** tab on the User window)
- Remove the Access Grants from the Security Group (in the Security Group window). Note: only do this if no one in that Security Group needs the access provided in that Access Grant.

Note

Mercury IT Governance Center includes other Access Grants that can be used to control access to other functions in Demand Management. See *Security Model Guide and Reference* for details.

Setting Request Creation Security

It is possible to control who can create certain Requests or use specific Request Types and Workflows. This provides a great deal of control over who can process changes of a certain type to specific environments. The following sections discuss how to control security related to Request creation:

- *Enabling Users to Create Requests*
- *Restricting Users from Selecting a Specific Workflow*

Note

The following sections assume that your users have the appropriate license and Access Grants to perform basic Request creation and processing.

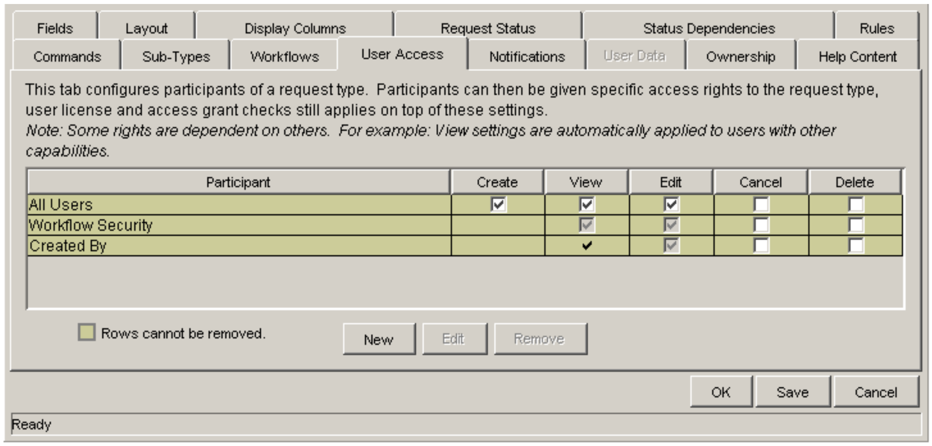
Enabling Users to Create Requests

It is possible to specify which users are allowed to create Requests of a specific Request Type in the Request Type window's **User Access** tab. You can enable all users with appropriate Access Grants to create a Request, or enable only certain users to create Requests of a certain Request Type.

The **User Access** tab can include multiple lines that grant access to create or process the Request. If a user meets any of the requirements listed in the tab, he will have access to perform that action in the Request.

To enable all users to create and submit a Request:

1. Open the Request Type window.
2. Click the **User Access** tab.



This tab configures participants of a request type. Participants can then be given specific access rights to the request type, user license and access grant checks still applies on top of these settings.
Note: Some rights are dependent on others. For example: View settings are automatically applied to users with other capabilities.

Participant	Create	View	Edit	Cancel	Delete
All Users	☑	☑	☑	□	□
Workflow Security		☑	☑	□	□
Created By		✓	☑	□	□

🟩 Rows cannot be removed.

Ready

3. In the All Users row, select the Create checkbox.
4. Click **Save**.

To enable only members of a specific Security Group to create a Request:

1. In the Request Type window's **User Access** tab, de-select the Create checkbox in the All Users row.
2. Click **New**.

   The Participant Security window opens.

3. From the Security Group field, select the Security Group that can create Requests of this Request Type.

4. Click **OK**.

   A new line listing the selected Security Group is added to the **User Access** tab.

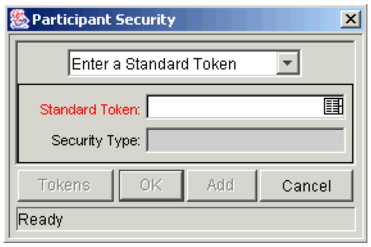
5. On the **User Access** tab, click **Save**.

To enable specific users to create a Request:

1. In the Request Type window's **User Access** tab, de-select the Create checkbox in the All Users row.

2. Click **New**.

   The Participant Security window opens.



3. From the drop down list, select one of the following items:

   - **Enter a Username**—Specify individual user names.
   - **Enter a Standard Token**—Control Request creation dynamically, depending on the value in a standard field. Select from a list of system Tokens that corresponds to a User or Security Group.
   - **Enter a User Defined Token**—Control Request creation dynamically, depending on the value in a custom field. Select from any field Token that corresponds to a User or Security Group.

   Selecting a value will automatically update the field below. For example, selecting **Enter a Username** will change the field below to Username.

4. Enter the specific value that corresponds to the recipient type selected above.

   This can be a Username or a Token.

5. Click **OK**.

   A new line listing the selected user or Token is added to the **User Access** tab.

6. On the Request Type window, click **Save**.

Restricting Users from Selecting a Specific Workflow

When creating a Request, a Workflow must be selected for the Request to proceed through. It is possible to control which Workflows can be used with each Request Type.

To restrict users from selecting a specific Workflow when creating a new Request.

1. In the Request Type window, click the **Workflow** tab.

2. De-select the All Workflows are allowed for this Request Type checkbox.

3. Click **New**.

   The Workflow: New window opens.

4. From the Workflow field, select only the Workflows that can be used with this Request Type.

5. Click **OK**.

6. The selected Workflows are added to the **Workflow** tab.

7. Click **Save**.

Only Workflows specified in the Workflow tab can be used when creating Requests of this Request Type.

Tip

Request Types can be associated with Workflows such that only certain Request Types can be processed through the Workflow. The selected Request Type must be enabled so that the user can create a Request when using that Workflow.

You can also opt to restrict all new Request Types.

The default Request Type to be used with this Workflow can also be specified.

This is set on the Workflow window - **Request Types** tab.

Setting Request Processing Security

It is possible to control who can process Requests following a Request submission. This includes specifying who can edit fields on Request, cancel a Request, and delete a Request. You can also control who can act on certain steps (decisions and executions) in a process. The following sections discuss how to control security related to Request processing:

- *Enabling Users to Edit Fields on a Request*
- *Enabling Users to Cancel or Delete a Request*
- *Enabling Users to Act on a Specific Workflow Step*

Note

The following sections assume that your users have the appropriate license and Access Grants to perform basic Request creation and processing.

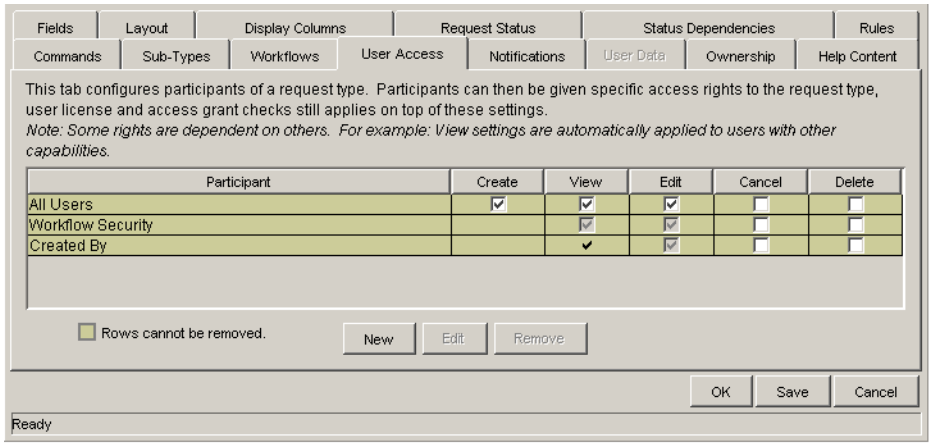
Enabling Users to Edit Fields on a Request

It is possible to control which users are allowed to edit fields on Requests of a specific Request Type.

To enable all users to edit fields on a Request:

1. Open the Request Type window.

2. Click the **User Access** tab.



This tab configures participants of a request type. Participants can then be given specific access rights to the request type, user license and access grant checks still applies on top of these settings.
Note: Some rights are dependent on others. For example: View settings are automatically applied to users with other capabilities.

Participant	Create	View	Edit	Cancel	Delete
All Users	☑	☑	☑	☐	☐
Workflow Security		☑	☑	☐	☐
Created By		✓	☑	☐	☐

🟨 Rows cannot be removed.

Ready

3. In the All Users row, select the Edit checkbox.

4. Click **Save**.

To enable only members of a specific Security Group to edit a Request:

1. In the Request Type window's **User Access** tab, de-select the Edit checkbox in the All Users row.

   By default, the Edit field remains selected in the Workflow Security row. This indicates that any user who is included in the associated Workflow's security (defined in any Workflow Step in the Workflow window) can edit Request fields.

2. Click **New**.

   The Participant Security window opens.

3. From the Security Group field, select the Security Group that can edit Requests of this Request Type.

4. Click **OK**.

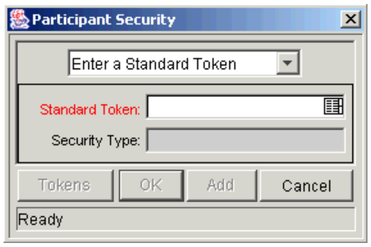
   A new line listing the selected Security Group is added to the **User Access** tab. By default, the Edit field is selected.

5. On the Request Type window, click **Save**.

To enable specific users to create a Request:

1. In the Request Type window's **User Access** tab, de-select the Edit checkbox in the All Users row.
2. Click **New**.

   The Participant Security window opens.



3. From the drop down list, select one of the following items:
   - **Enter a Username**—Specify individual user names.
   - **Enter a Standard Token**—Control Request security dynamically, depending on the value in a standard field. Select from a list of system Tokens that corresponds to a User or Security Group.
   - **Enter a User Defined Token**—Control Request security dynamically, depending on the value in a custom field. Select from any field Token that corresponds to a User or Security Group.

   Selecting a value will automatically update the field below. For example, selecting **Enter a Username** will change the field below to Username.

4. Enter the specific value that corresponds to the recipient type selected above.

   This can be a Username or a Token.

5. Click **OK**.

   A new line listing the selected user or Token is added to the **User Access** tab. By default, the Edit field is selected.

6. On the Request Type window, click **Save**.

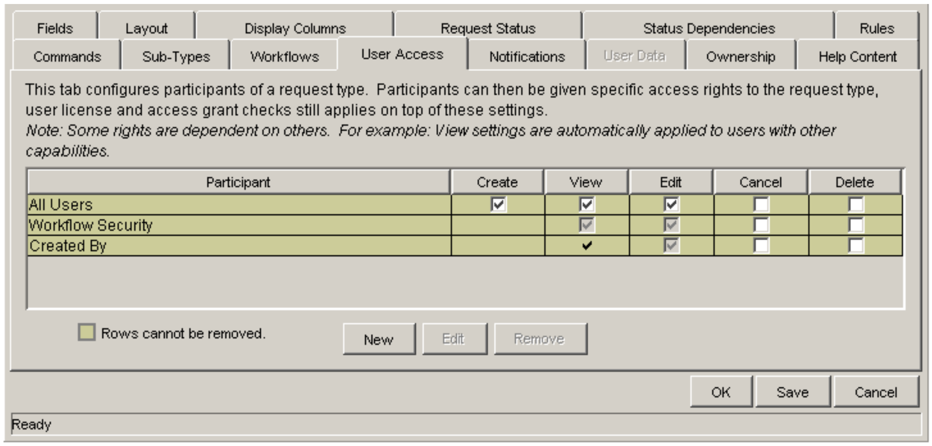
Enabling Users to Cancel or Delete a Request

It is possible to control which users are allowed to cancel or delete Requests of a specific Request Type.

To enable all users to cancel or delete a Request:

1. Open the Request Type window.

2. Click the **User Access** tab.



This tab configures participants of a request type. Participants can then be given specific access rights to the request type, user license and access grant checks still applies on top of these settings.

Note: Some rights are dependent on others. For example: View settings are automatically applied to users with other capabilities.

Participant	Create	View	Edit	Cancel	Delete
All Users	☑	☑	☑	☐	☐
Workflow Security		☑	☑	☐	☐
Created By		✓	☑	☐	☐

☐ Rows cannot be removed.

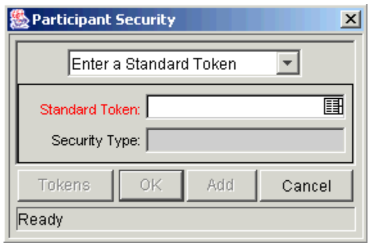
3. In the All Users row, select the Cancel and Delete checkboxes.

4. Click **Save**.

To enable only specific users or members of a specific Security Group to cancel or delete a Request:

1. In the Request Type window's **User Access** tab, click **New**.

   The Participant Security window opens.



2. From the drop down list, select one of the following items:

    - **Enter a Security Group**—Specify all users in a Security Group.
    - **Enter a Username**—Specify individual user names.
    - **Enter a Standard Token**—Control Request security dynamically, depending on the value in a standard field. Select from a list of system Tokens that corresponds to a User or Security Group.
    - **Enter a User Defined Token**—Control Request security dynamically, depending on the value in a custom field. Select from any field Token that corresponds to a User or Security Group.

    Selecting a value will automatically update the field below. For example, selecting **Enter a Username** will change the field below to Username.

3. Enter the specific value that corresponds to the recipient type selected above.

4. Click **OK**.

    A new line listing the selected user or Token is added to the **User Access** tab.

5. In the new row, select the Cancel and Delete checkboxes.

6. On the Request Type window, click **Save**.

To enable the user who logged the Request to cancel or delete that Request:

1. Open the Request Type window.

2. Click the **User Access** tab.

3. In the Created By row, select the Cancel and Delete checkboxes.

4. Click **Save**.

Enabling Users to Act on a Specific Workflow Step

It is necessary to specify who can act on each step in the Request resolution Workflow. Only users who are specified on the **Security** tab in the Workflow Step window will be able to process the Request at that step.

To specify the users who can act on a specific Workflow step:

1. Open the Workflow.

2. Click the **Layout** tab.

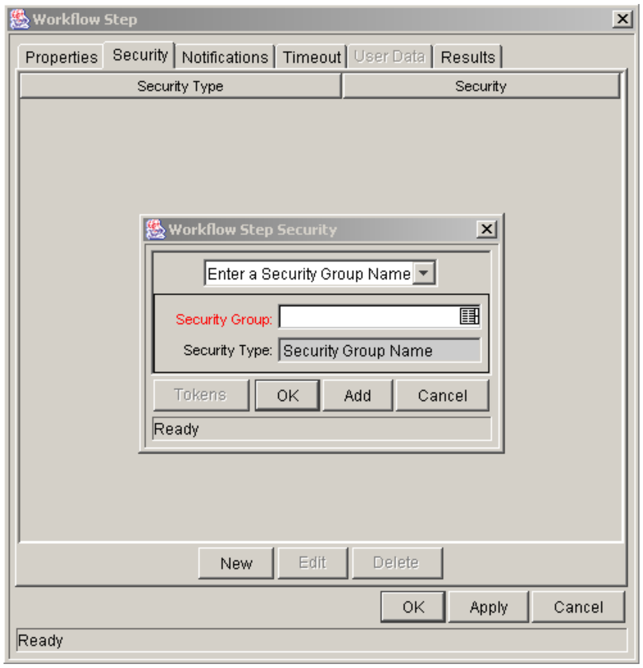
3. Double click on the step that you would like to configure.

   The Workflow Step window opens. Note: the Workflow Step window also opens when first adding a step to the **Layout** tab.

4. Click the **Security** tab.

5. Click **New**.

   The Workflow Step Security window opens.



Enter a Security Group Name

Security Group:

Security Type: Security Group Name

Ready

Ready

6. Select the method for specifying the step security from the drop down list:

   - Security Group Name
   - Username
   - Standard Token
   - User Defined Token.

   Selecting a value from this field automatically updates the other fields on this window. For example, selecting **Enter a Username** will change the Security Group field to Username.

7. Specify the Security Groups, Usernames, or Tokens that will control the access to this step.

8. Click **OK**.

   The security specification is added to the **Security** tab. Additional specifications can then be added to the step by clicking **New** and repeating the above process. The step's security can therefore be controlled using a combination of multiple Security Groups, Usernames and Tokens.

9. Click **OK** to save and close the window.

Tip

10. Consider assigning a Security Group to each decision, execution and condition step, even though many of these steps will proceed automatically. If a command fails, or a condition is not met, it may be necessary to manually override the step.

11. Also consider assigning a “Request Manager” Security Group to each step. That group could be configured with global access to act on every step in the process. Again, this could help avoid bottlenecks by providing a small group with permission to process stalled Requests.

12. Avoid specifying a single user as the only person who can act on a Workflow step. This would require a process update (re-configuration) when that user changes roles or leaves the company. Better to grant access dynamically using a Token or Security Group.

Setting Configuration Security

A critical part of ensuring successful Request resolution is ensuring that the resolution process is altered only by the correct people. Configuration security can be established around all of the Mercury Demand Management configuration entities. This includes such activities as controlling:

- Who can change the Workflow.
- Who can change each Request Type.
- Who can change the Security Group definitions.

The following sections discuss some options for securing configurations:

- *Setting Ownership for Configuration Entities*
- *Removing Access Grants*

Setting Ownership for Configuration Entities

Different groups of users have ownership and control over Demand Management entities. These groups are referred to as Ownership Groups.

Unless a global permission has been designated to all users for an entity, members of Ownership Groups are the only users who have the right to edit, delete or copy that entity. The Ownership Groups must also have the proper Access Grant for the entity in order to complete those tasks. For example, the Edit Workflows Access Grant is needed to edit Workflows and Workflow Steps.

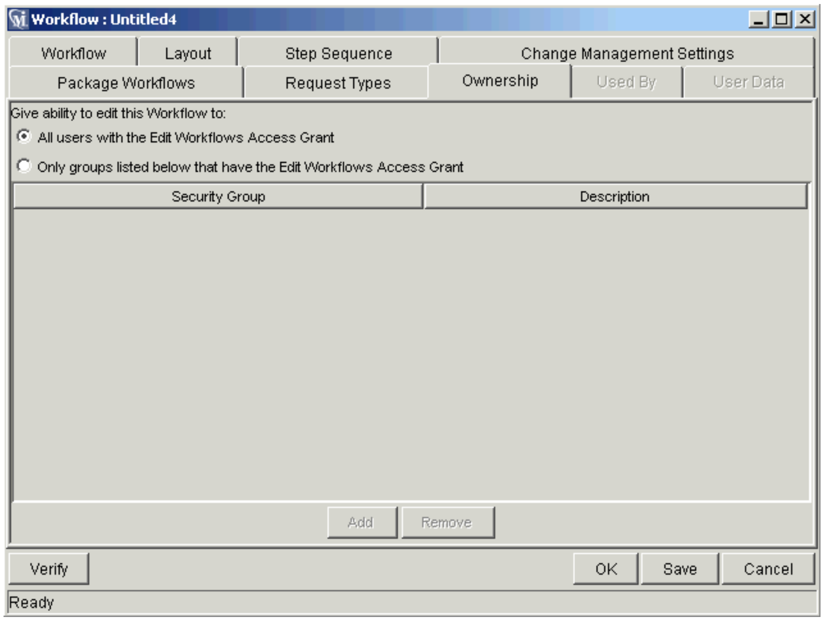
Multiple Ownership Groups can be applied to the various entities. The Ownership Groups will have sole control over the entity, providing greater security. Ownership Groups are defined in the Security Group window. Security Groups become Ownership Groups when used in the Ownership capacity.

It is possible to specify Ownership Groups for the following entities involved in your process:

- Workflows
- Workflow Steps
- Request Types
- Request Header Types
- Security Groups
- User Definitions
- Report Types
- Validations
- Special Commands

The Ownership setting is accessed through the individual entity windows in the Workbench. For example, to set the Ownership for Workflows:

1. Open the Workflow.
2. Click the **Ownership** tab.



Give ability to edit this Workflow to:

● All users with the Edit Workflows Access Grant

○ Only groups listed below that have the Edit Workflows Access Grant

3. Click the Only groups listed below that have the Edit Workflows Access Grant radio button.

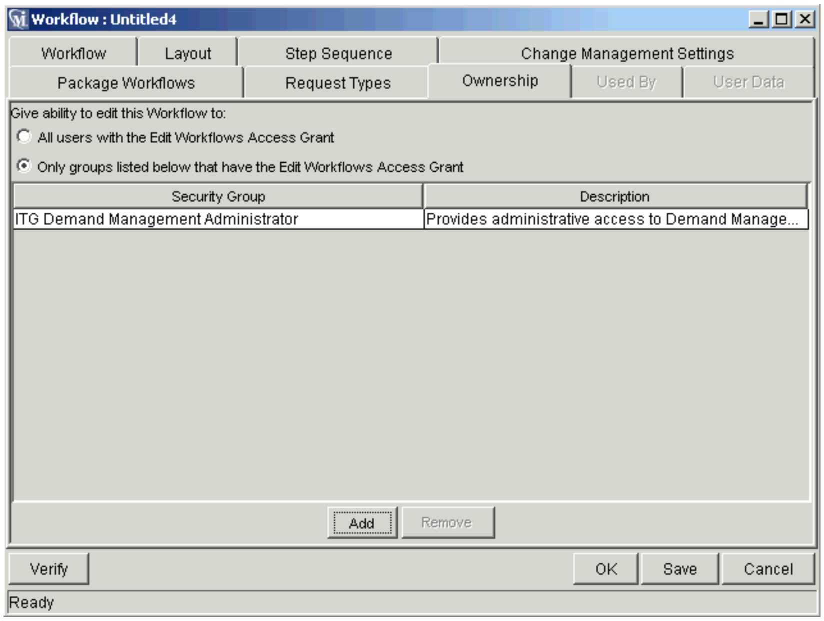
4. Click **Add**.

   The Add Security Group window opens.

5. Select the Security Group.

6. Click **Add** to add the current Security Group and continue adding more Security Groups. Click **OK** to add the current Security Group and close the Add Security Group window.

   The Security Group(s) you selected displays in the **Ownership** tab under the Security Group column.



Give ability to edit this Workflow to:

○ All users with the Edit Workflows Access Grant

● Only groups listed below that have the Edit Workflows Access Grant

Security Group	Description
ITG Demand Management Administrator	Provides administrative access to Demand Manage...

7. Click **OK** to save the selection and close the Workflow window. Click **Save** to save the selection and leave the Workflow window open.

Note

The System: Ownership Override Access Grant allows the user to access and edit configuration entities even if he is not a member of one of the entity’s Ownership Groups.

Removing Access Grants

It is also possible to restrict the ability to modify configuration entities by removing the user from any Security Group that grants that access.

Use the **Access Grants** tabs in the User window to see all Security Groups where specific Access Grants are included.

- Remove the user from the Security Group (using the **Security Group** tab on the User window)
- Remove the Access Grants from the Security Group (in the Security Group window). Note: only do this if no one in that Security Group needs the access provided in that Access Grant.

Note

Only users with an Administrator license can create User accounts and Security Groups, which are critical when integrating participants into a Request resolution system. Work with the Administrator to configure the User accounts and Security Groups required for the process.

Table 8-2 lists the Access Grants that provide edit access to different configuration entities.

Table 8-2. Access Grants for editing configuration entities

Access Grant	Description
Config: Edit Report Types	Allows the user to generate, update and delete Report Types.
Config: Edit Special Commands	Allows the user to generate, update and delete Special Commands.
Config: Edit User Data	Allows the user to generate, update and delete User Data.
Config: Edit Validation Values	Allows the user to generate, update and delete Validation Values.
Config: Edit Validations	Allows the user to generate, update and delete Validations.
Config: Edit Workflows	Allows the user to generate, update and delete Workflows.
Demand Mgmt: Edit Contacts	Allows the user to generate, update and delete Contacts.
Demand Mgmt: Edit Request Header Types	Allows the user to generate, update and delete Request Header Types.
Demand Mgmt: Edit Request Types	Allows the user to generate, update and delete Request Types.
Sys Admin: Edit Security Groups	Allows the user to generate, update and delete Security Groups.
Sys Admin: Edit Users	Users Allows the user to generate, update and delete Users.

Chapter 9 Setting Up Communication Paths

This chapter provides an overview for different modes of communication that can be used in a Request resolution system. The following features help increase visibility into Request processes and data:

- Email Notifications:
  Each Workflow step can be configured to send an email to specified users when the step becomes eligible, has a specific result, or encounters an error. Using Notifications at key points in a process ensures a speedy resolution by notifying appropriate parties of actions required by them or complications during the process.

  Email Notifications can also be sent when a field in a Request changes.
- Dashboard:
  The Dashboard provides an interface through which the current state of the Requests can be quickly assessed. Personalize your Dashboard to display status information that is most meaningful to your role. For example, Financial system users may only want to see the Requests that they submitted, whereas the Financial manager may want to have visibility into each critical Request currently in progress.
- Reports:
  Mercury Demand Management includes a number of reports that can be used to assess Request status. Demand Management also publishes a reporting meta layer that can be used to build custom reports. Reports can be scheduled to run periodically.
- Custom Help on Requests and Request Fields
  Custom Help content can be defined for each Request in the system. This enables you to provide end users with direct information related to a specific Request, a section on the Request, or a field on the Request.

This chapter illustrates how Notifications, Dashboard components, reports, and custom online help can be used to monitor and control the resolution process. This chapter discusses the following topics:

- *Adding Notifications to Workflow Steps*
- *Setting Notifications on Request Field Changes*
- *Configuring Your Dashboard*
- *Configuring Reports*
- *Creating Custom Help for Requests*

Adding Notifications to Workflow Steps

When configuring a Notification for a Workflow step, consider the following:

- When to send it.
- Who should receive it.
- What the message should say.

This section covers the following topics:

- *Adding a Notification to a Workflow step - Overview*
- *Configuring When to Send a Notification*
- *Configuring the Notification Recipients*
- *Configuring the Notification Message*
- *Using Notification Templates*

Adding a Notification to a Workflow step - Overview

To add a Notification to a Workflow step:

1. Open the Workflow.
2. Click the **Layout** tab.

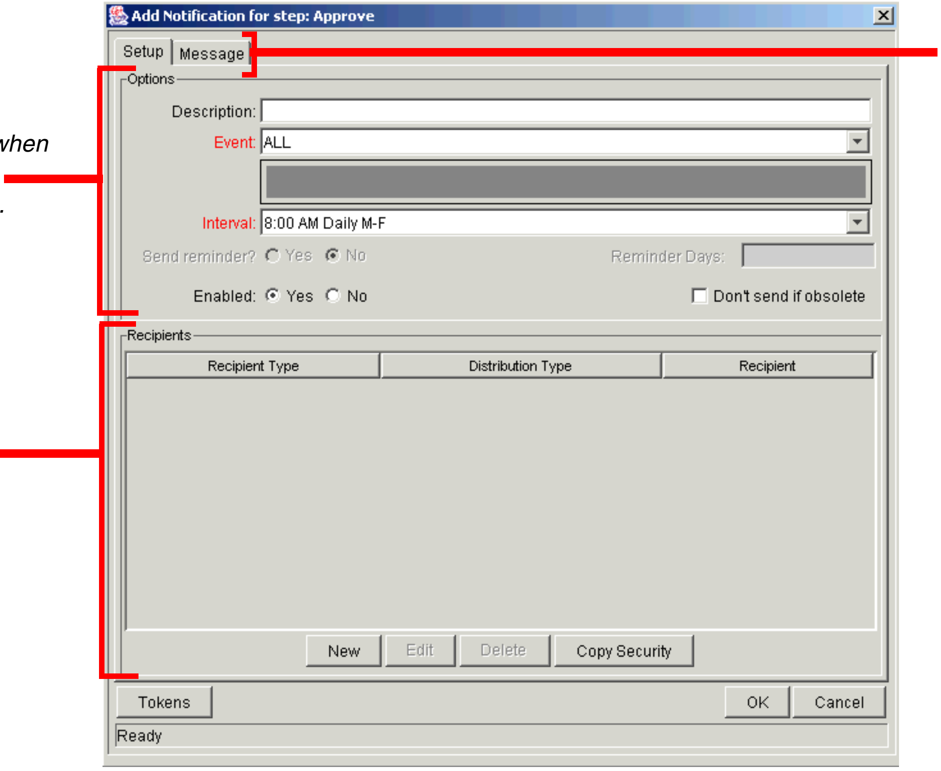
3. Double click on the step to configure.

   The Workflow Step window opens. The Workflow Step window also opens when first adding a step to the **Layout** tab.

4. Click the **Notifications** tab.

5. Click **New**.

   The Add Notification for Step window opens.



Description:

Event: ALL

Interval: 8:00 AM Daily M-F

Send reminder? ○ Yes ● No

Reminder Days:

Enabled: ● Yes ○ No

☐ Don't send if obsolete

Recipient Type	Distribution Type	Recipient

Ready

Configure the message.

Configure when to send the Notification.

Select the recipients.

6. Configure the following:

   - When the Notification is sent (Event and Interval)
   - Who receives the Notification (Recipients)
   - The body of the Notification (Message)

7. Click **OK**.

The Notification specification is added to the **Notifications** tab. Additional specifications can then be added to the step by clicking **New** and repeating the above process. You can therefore send a different Notification to different recipients for different events.

8. Click **OK** to save and close the window.

Configuring When to Send a Notification

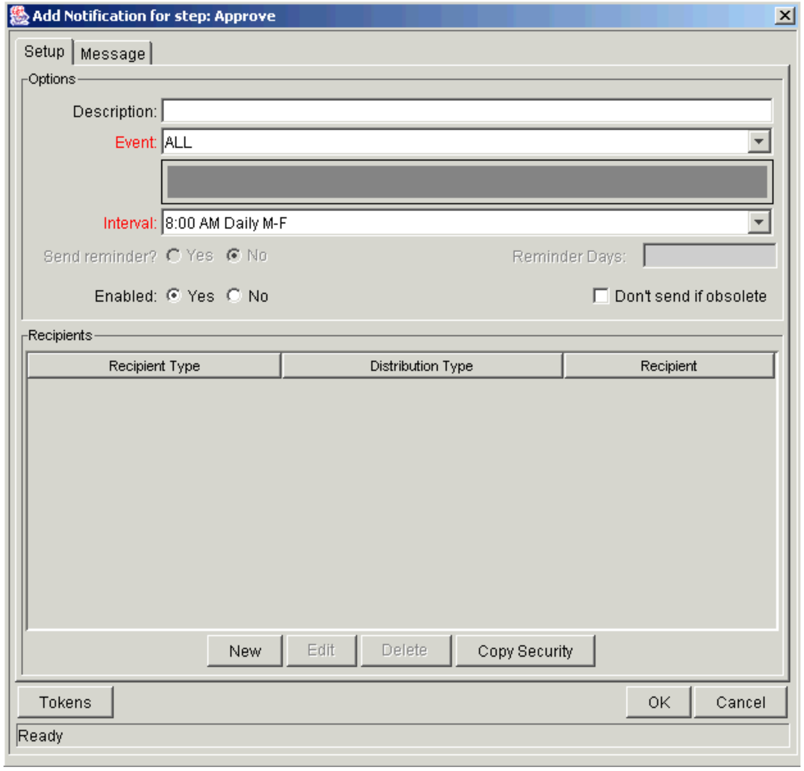
Each Workflow step can be configured to send an email when the step becomes eligible, has a specific result, or encounters an error.

This section covers the following topics:

- *Sending a Notification when a step becomes eligible*
- *Sending a Notification when a step has a specific result*
- *Sending a Notification when the step has a specific error*
- *Configuring multiple Notifications for a single step*
- *Specifying the Time the Notification is Sent*

Sending a Notification when a step becomes eligible

To send a Notification when a Workflow step becomes eligible, configure the Notification as indicated below.



Description:

Event: ALL

Interval: 8:00 AM Daily M-F

Send reminder? ○ Yes ● No   Reminder Days:

Enabled: ● Yes ○ No   ☐ Don't send if obsolete

Recipient Type	Distribution Type	Recipient

Table 9-1. Workflow step Notification configuration - send on eligible

Field	Value	Notes
Event	**Eligible**	
Interval	**Immediate**	A Notification can be sent at different intervals. For example, you might choose to send a Notification of a final approval step at midnight so that it is ready for approval in the morning. Note also that multiple Notifications to a single recipient can be batched and sent together. Selecting an interval other than “Immediate” will allow this batching to occur. See *“Configuring the Notification Intervals”* on page 214 for instructions on configuring this.

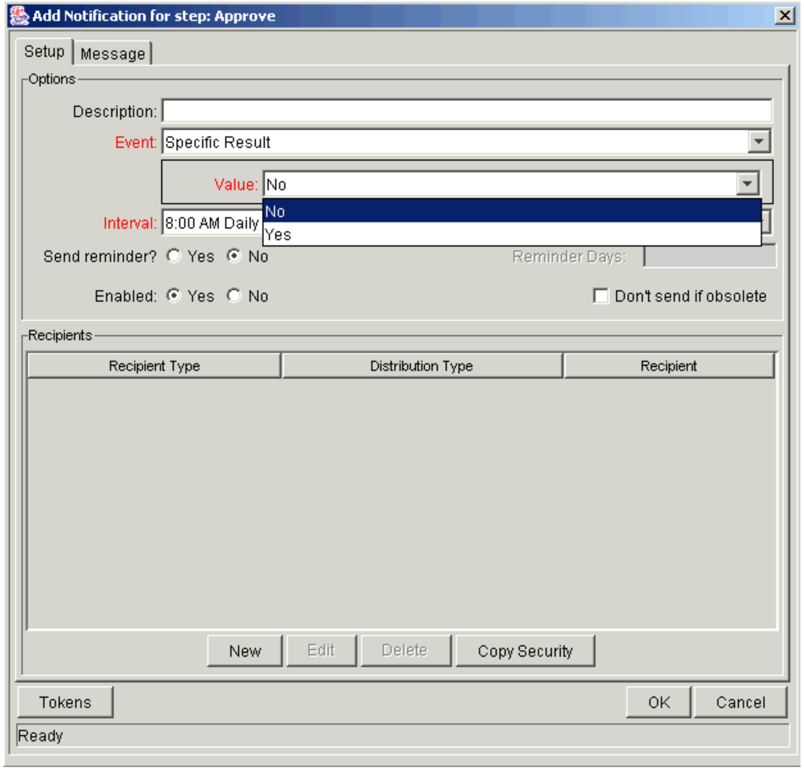
Table 9-1. Workflow step Notification configuration - send on eligible

Field	Value	Notes
Send Reminder	**Yes/No**	This field is optional. A reminder Notification can be sent if the Notification event is still true after a period of time. For example, a reminder can be sent if a step is still Eligible after a number of days. A reminder cannot be sent if the Notification event is 'All.'
Enabled	**Yes**	

Sending a Notification when a step has a specific result

It is possible to configure the Notification to be sent when a Workflow step results in a specific decision or execution result. The value for these events is determined by the Workflow step source's Validation.

To send a Notification when a Workflow has a specific result, configure the Notification as indicated below.



Add Notification for step: Approve

Options

Description:

Event: Specific Result

Value: No

Interval: 8:00 AM Daily

Send reminder? ○ Yes ● No

Reminder Days:

Enabled: ● Yes ○ No

□ Don't send if obsolete

Recipients

Recipient Type	Distribution Type	Recipient

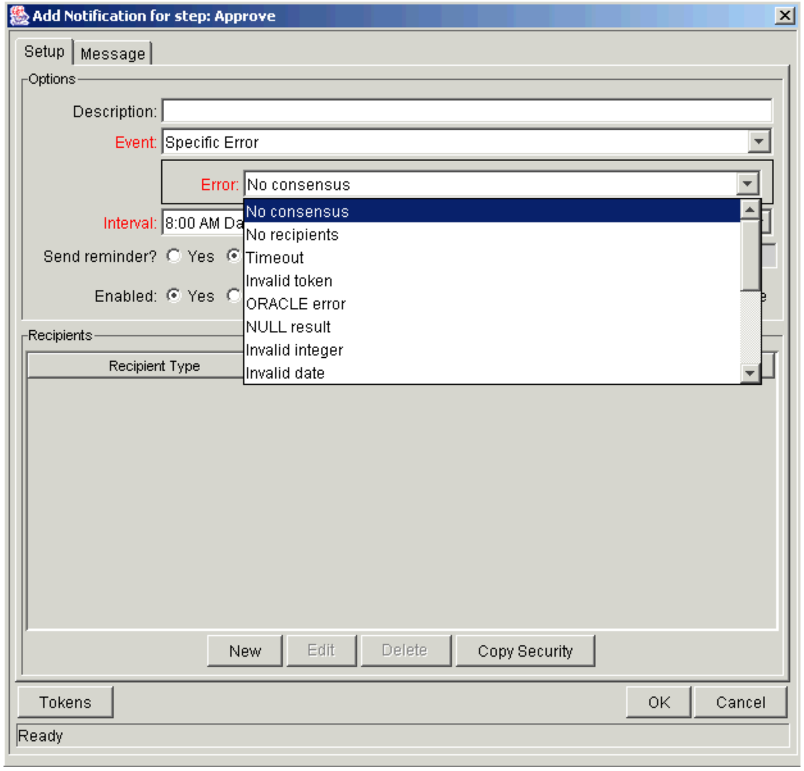
Ready

Table 9-2. Workflow step Notification configuration - send on step result

Field	Value	Notes
Event	**Specific Result**	
Value	Select the value to trigger the Notification.	The list of values is determined by the Workflow step source's Validation. Therefore, this selection will always be limited to the possible results of the step.
Interval	**Immediate**	A Notification can be sent at different intervals. For example, you might choose to send a Notification of a final approval step at midnight so that it's ready for approval in the morning. Note also that multiple Notifications to a single recipient can be batched and sent together. Selecting an interval other than "Immediate" will allow this batching to occur. See *"Configuring the Notification Intervals"* on page 214 for instructions on configuring this.
Send Reminder	**Yes/No**	This field is optional. A reminder Notification can be sent if the Notification event is still true after a period of time. For example, a reminder can be sent if a step is still Eligible after a number of days. A reminder cannot be sent if the Notification event is 'All.'
Enabled	**Yes**	

Sending a Notification when the step has a specific error

To send a Notification when a Workflow has a specific error, configure the Notification as indicated below.



Add Notification for step: Approve

Description:

Event: Specific Error

Error: No consensus

Interval: 8:00 AM Da

Send reminder? ○ Yes ●

Enabled: ● Yes ○

Table 9-3. Workflow step Notification configuration - send on error

Field	Value	Notes
Event	**Specific Error**	
Error	Select the value to trigger the Notification.	This is a standard set of errors. See *“Specific Errors for Workflow Steps”* on page 211.

Table 9-3. Workflow step Notification configuration - send on error

Field	Value	Notes
Interval	**Immediate**	A Notification can be sent at different intervals. For example, you might choose to send a Notification of a final approval step at midnight so that it's ready for approval in the morning. Note also that multiple Notifications to a single recipient can be batched and sent together. Selecting an interval other than "Immediate" will allow this batching to occur. See *"Configuring the Notification Intervals"* on page 214 for instructions on configuring this.
Send Reminder	**Yes/No**	This field is optional. A reminder Notification can be sent if the Notification event is still true after a period of time. For example, a reminder can be sent if a step is still Eligible after a number of days. A reminder cannot be sent if the Notification event is 'All.'
Enabled	**Yes**	

Specific Errors for Workflow Steps

The following errors can cause a Notification to be sent.

Table 9-4. Specific Errors for Workflow Steps

Specific Error	Meaning
No consensus	When all users of all Security Groups, or users linked to the Workflow Step need to vote, and there is no consensus.
No recipients	When none of the Security Groups linked to the Workflow Step has users linked to it, no user can act on the Workflow Step.
Timeout	When the Workflow Step times out. Used for Executions and Decisions.
Invalid token	Invalid Token used in the execution.

Table 9-4. Specific Errors for Workflow Steps

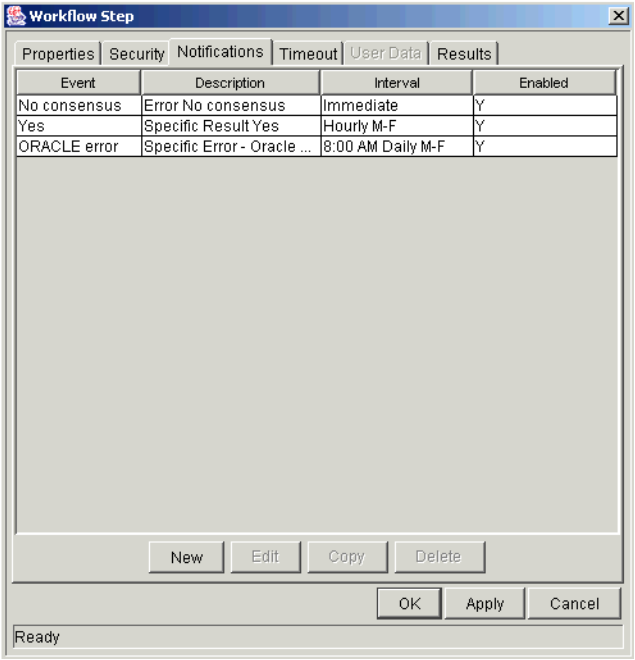
Specific Error	Meaning
ORACLE error	Failed PL/SQL Execution.
NULL result	No result is returned from the execution.
Invalid integer	Validation includes an invalid value in the Integer field.
Invalid date	Validation includes an invalid value in the Date field.
Command execution error	Execution engine has failed or has a problem.
Invalid Result	Execution or Subworkflow has returned a result not included in the Validation.
Parent closed	For wf_receive or wf_jump steps, a Request is expecting a message from a Package Line that is cancelled or closed.
Child closed	For wf_receive or wf_jump steps, a Package Line is expecting a message from a Request that is cancelled or closed.
No parent	For wf_receive or wf_jump steps, a Request is expecting a message from a Package Line that has been deleted.
No child	For wf_receive or wf_jump steps, a Package Line is expecting a message from a Request that has been deleted.
Multiple jump results	For wf_jump steps in a Package Line, different result values were used to transition to the step.
Multiple Return Results	When the Package Level Subworkflow receives multiple results from Package Lines that traversed through it.

Configuring multiple Notifications for a single step

It is possible to configure multiple Notifications for each Workflow step. This can be useful in the following sample situations:

- Sending a different message depending on the result of the step
- Sending a different message depending on the type error
- Sending the Notification to a different set of users depending on the step's result or error
- Specifying different intervals or reminders based on the type of step error

To configure multiple Notifications for a Workflow step, simply add multiple Notifications to the same Workflow step window, as shown in *Figure 9-1*.



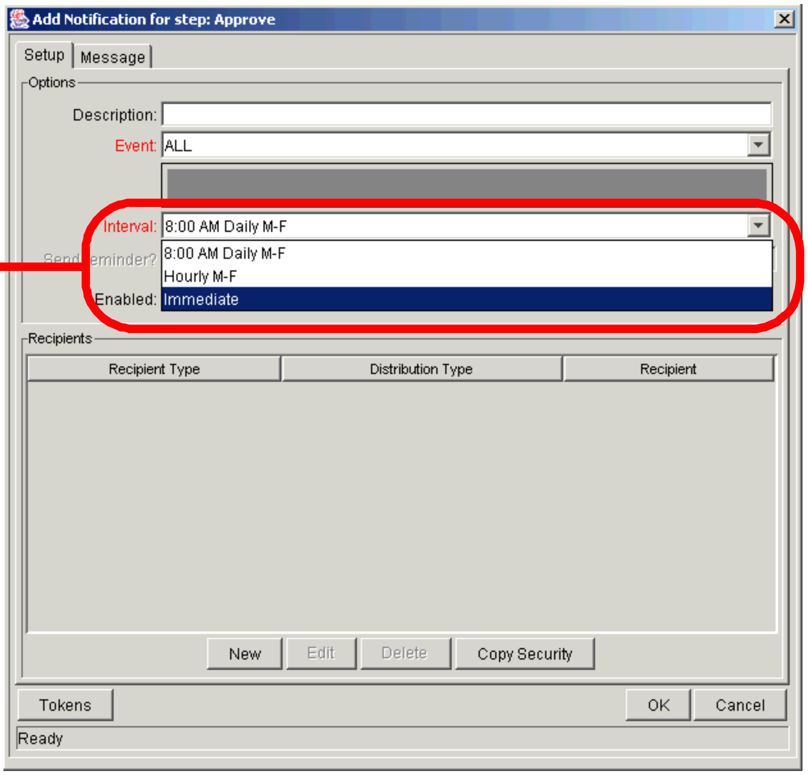
Event	Description	Interval	Enabled
No consensus	Error No consensus	Immediate	Y
Yes	Specific Result Yes	Hourly M-F	Y
ORACLE error	Specific Error - Oracle ...	8:00 AM Daily M-F	Y

Figure 9-1 Workflow Step with Multiple Notifications

In this example, one set of users is notified when the step becomes eligible. Then, depending on the outcome of the step, different groups are notified. If the step experiences a "Timeout" error event, then the user responsible for acting on the step is notified. If the step results in the specific result of "Not Approved," then a Notification is sent to the deployment manager.

Specifying the Time the Notification is Sent

Use the Interval field on the Workflow step to specify when the Notification will be sent.



Select an Interval to specify when and how often the Notification is sent.

The interval determines how frequently the Notification will be sent. The following pre-configured intervals are available:

- **8:00 AM Daily M-F:** This Notification is sent every 8:00 AM on the next available work day after the Notification event occurs.
- **Hourly M-F:** This Notification is sent every hour, starting on the next available work day after the Notification event occurs.
- **Immediate:** This Notification is sent immediately.

Configuring the Notification Intervals

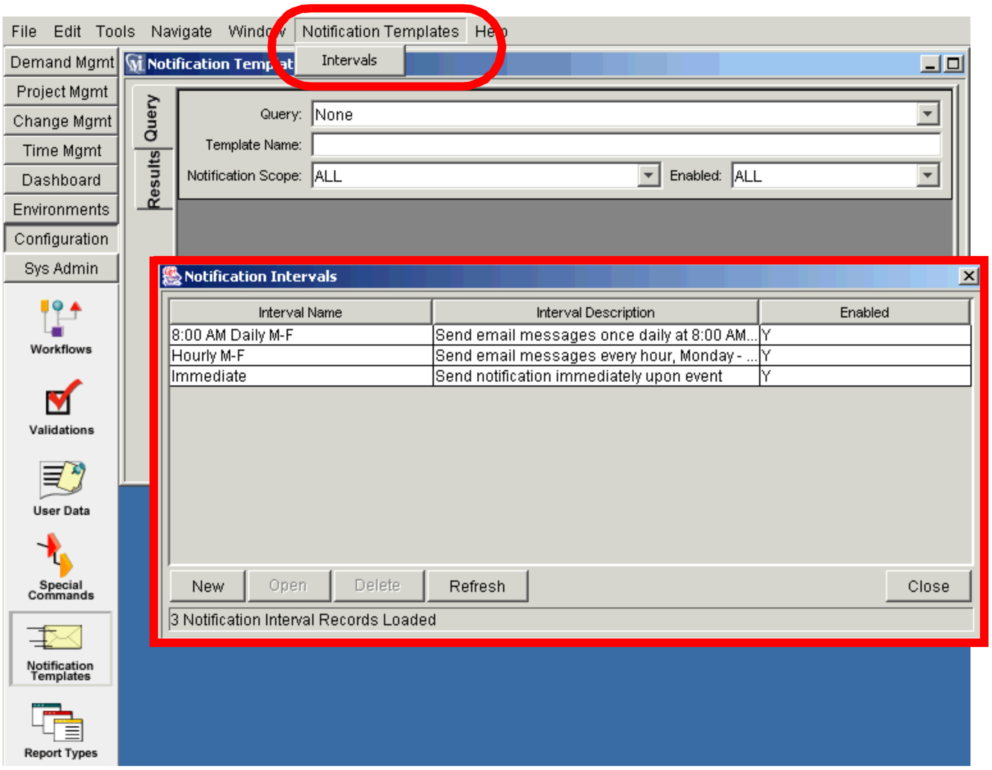
Notifications are configured on the Notification Templates Workbench.

To configure the Notification Intervals:

1. Click the **Configuration** screen group and click the **Notification Templates** icon.

2. From the menu, select **Notification Templates > Intervals**.

   The Notification Intervals window opens.



Query: None

Template Name:

Notification Scope: ALL

Enabled: ALL

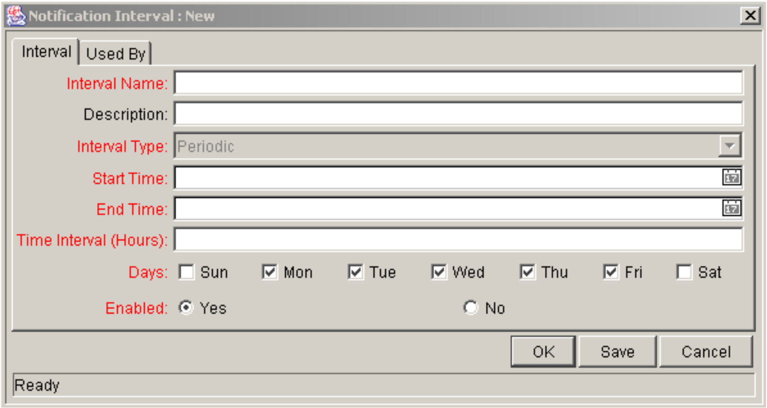
Interval Name	Interval Description	Enabled
8:00 AM Daily M-F	Send email messages once daily at 8:00 AM...	Y
Hourly M-F	Send email messages every hour, Monday - ...	Y
Immediate	Send notification immediately upon event	Y

3 Notification Interval Records Loaded

3. Click **New** or **Open** to access the Notification Interval: New window.

4. On the **Interval** tab, enter the required information.

   These fields are defined in *Table 9-5*.



Interval Name:

Description:

Interval Type: Periodic

Start Time:

End Time:

Time Interval (Hours):

Days: □ Sun ☑ Mon ☑ Tue ☑ Wed ☑ Thu ☑ Fri □ Sat

Enabled: ● Yes ○ No

Ready

Table 9-5. Notification Intervals

Field Name	Description
Interval Name	This is the name assigned to the interval.
Description	Free form description of this interval.
Interval Type	For internal use. This is always set to Periodic, unless Immediate Interval is used.
Start Time	Time to start sending out Notifications and to start counting down the time interval until the next batch.
End Time	Time to stop sending out Notifications.
Time Interval	Number of hours to wait after the Start Time or the last batch sent, before sending out the next batch of Notifications.
Days	Used to select which days this interval should execute on.
Enabled	If **Yes** is set, this interval is selectable. If **No** is set, this interval is unavailable.

5. Click **OK**.

   The new interval is added to the Notification Intervals window.

6. Click **Close** to close the window.

The new Notification Interval can now be used in any Workflow step Notification.

When Notifications are sent with an hourly or daily interval, there are sometimes several Notifications pending for a particular user. In this case, all Notifications are grouped together in one email message. The Subject of each individual Notification appears at the top of the email message in a Summary section.

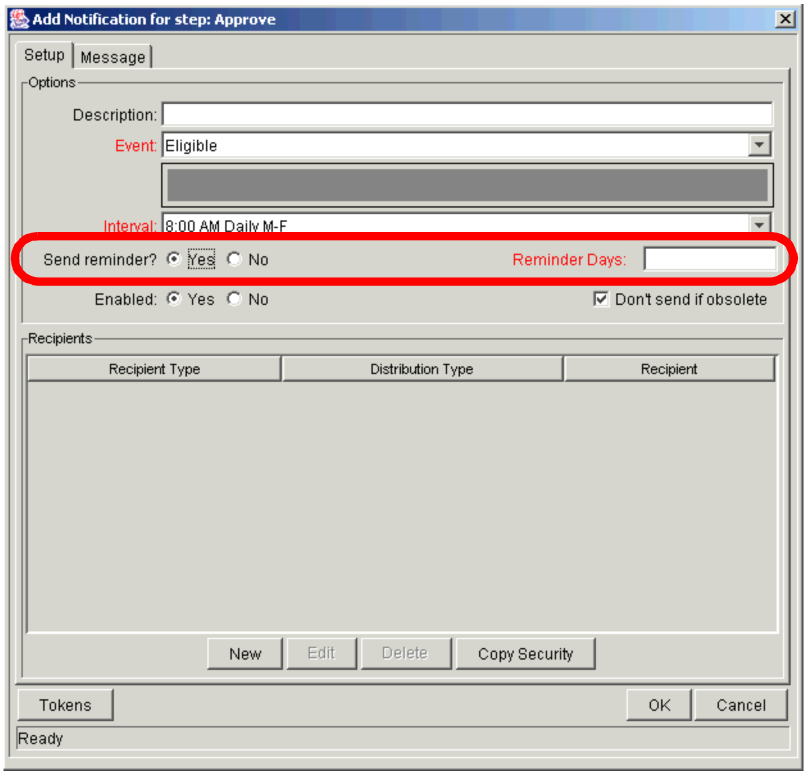
Sending a follow up Notification (reminder)

A reminder Notification can be sent if the Notification event is still true after a period of time. For example, a reminder can be sent if a step is still Eligible after a number of days. A reminder cannot be sent if the Notification event is **All**.

To configure a Notification to re-send after a period of time, configure the Notification as indicated below.

Table 9-6. Workflow step Notification configuration - send on error

Field	Value	Notes
Event		Select any event except for **All**.
Send Reminder	Yes	Selecting Yes enables the Reminder Days field.
Reminder Days	Enter the number of days.	The number of days to wait before sending a reminder Notification.



Description:

Event: Eligible

Interval: 8:00 AM Daily M-F

Send reminder? ● Yes ○ No

Reminder Days:

Enabled: ● Yes ○ No

☑ Don't send if obsolete

Recipient Type	Distribution Type	Recipient

Ready

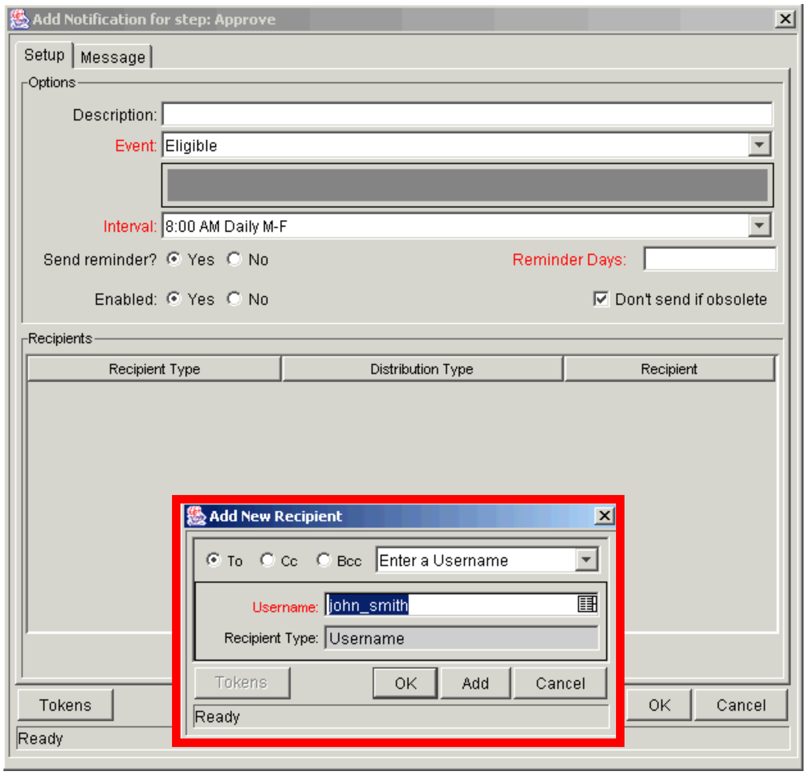
Configuring the Notification Recipients

When creating a Notification, at least one recipient must be added for the message. The recipient can be a specific user, all members of a Security Group, or any email address.

To add a recipient to a Notification:

1. In the Add Notification for Step window, click **New**.

The Add New Recipient window opens.



Description:

Event: Eligible

Interval: 8:00 AM Daily M-F

Send reminder? ● Yes ○ No

Reminder Days:

Enabled: ● Yes ○ No

☑ Don't send if obsolete

Recipient Type	Distribution Type	Recipient

● To ○ Cc ○ Bcc Enter a Username

Username: john_smith

Recipient Type: Username

2. Select how to specify the recipient from the drop down list:

   - **Enter a Security Group** – select a specific Security Group, and all enabled users in the group with email addresses will receive the Notification.

   - **Enter a Username** – select a specific User to receive the Notification. The User must have an email address.

   - **Enter an Email Address** – enter any email address to send the Notification to.

   - **Enter a Standard Token** – select from a list of system Tokens that corresponds to a User, Security Group, or Email Address.

   - **Enter a User Defined Token** – enter any field Token that corresponds to a User, Security Group, or Email Address.

Selecting a value will automatically update the field below. For example, selecting **Enter a Security Group** will change the field below to Security Group.

3. Enter the specific value that corresponds to the recipient type selected above. This can be a Username, Email Address, Security Group, or a Token.

Tip

Use Security Groups or dynamic access (Tokens) to define the Notification recipients whenever possible. Avoid specifying a list of users or an individual user's email address. If the list of users changes (due to a departmental or company reorganization), that list would have to be updated manually. By using a Security Group instead of a list of users, the Security Group can be updated once, and the changes will be propagated throughout the Workflow steps.

Tip

Use Tokens when sending a Notification to an undetermined party. For example, the Notification can be configured to be sent to the Assigned to User by specifying **[REQ.ASSIGNED_TO_USERID]** in the Add New Recipient window.

Configuring the Notification Message

It is possible to construct the Notification's message to ensure that it contains the correct information or instructions for the recipient. For example, if the Notification was sent to instruct the user that a Request requires his approval, the message should instruct him to log onto Mercury Demand Management and update the Request's status. Additionally, the Notification should include a link (URL) to the referenced Request.

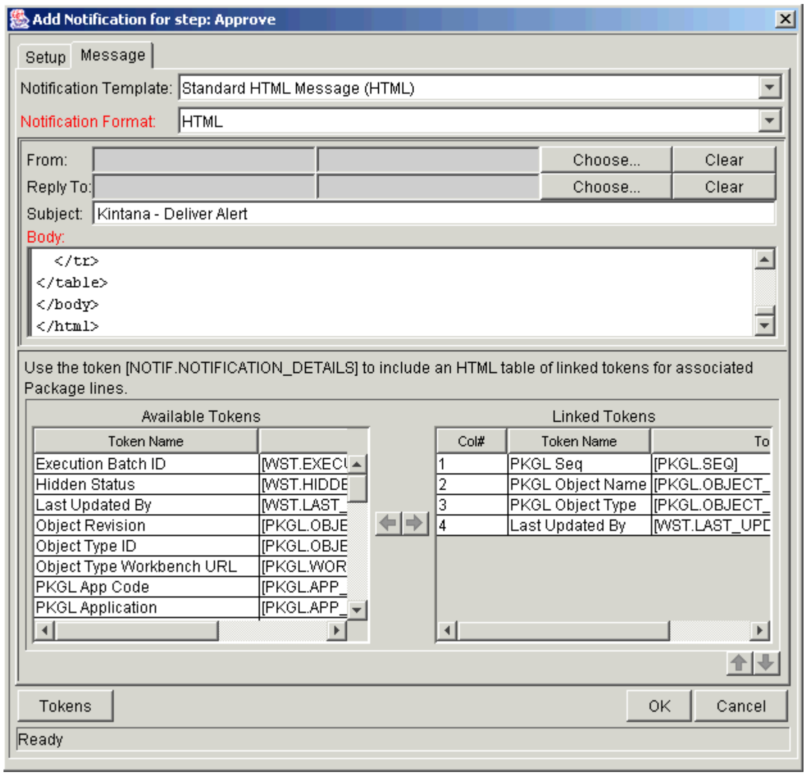
Notifications include the following features which make them easier to configure and use:

- Select from a number of pre-configured Notification templates to more quickly construct the body of your message.
- The body of the Notification can be plain text or HTML.
- Multiple Tokens can be included in the Notification. These Tokens will resolve to information relevant to the recipient. For example, you can

include Tokens for the URL to the Request approval page, information on Request status and priority, and emergency contacts.

To configure the message in a Notification:

1. On the Add Notification for Step window, click the **Message** tab.



Notification Template: Standard HTML Message (HTML)

Notification Format: HTML

From:

Reply To:

Subject: Kintana - Deliver Alert

Body:

```
  </tr>
</table>
</body>
</html>
```

Use the token [NOTIF.NOTIFICATION_DETAILS] to include an HTML table of linked tokens for associated Package lines.

Available Tokens

Token Name	
Execution Batch ID	[WST.EXECU
Hidden Status	[WST.HIDDE
Last Updated By	[WST.LAST_
Object Revision	[PKGL.OBJE
Object Type ID	[PKGL.OBJE
Object Type Workbench URL	[PKGL.WOR
PKGL App Code	[PKGL.APP_
PKGL Application	[PKGL.APP_

Linked Tokens

Col#	Token Name	To
1	PKGL Seq	[PKGL.SEQ]
2	PKGL Object Name	[PKGL.OBJECT_
3	PKGL Object Type	[PKGL.OBJECT_
4	Last Updated By	[WST.LAST_UPD

2. Select a Notification Template.

   This updates the contents in the Body section with the information defined for the selected template.

3. Select **HTML** or **Plain Text** from the Notification Format field.

   Selecting **HTML** allows more flexibility when formatting the look and feel of the Notification. The HTML code can be written and tested in any HTML editor and then pasted into the Body window.

4. Select values for the From and Reply to fields.

5. Construct the Body of the message.

   When constructing the body, consider utilizing the following:

- Token for the URL to the Request Detail page. See *Table 9-7* for a list of these Tokens.

- Token for the URL to the Package (Workbench or standard interface). See *Table 9-7* for a list of these Tokens.

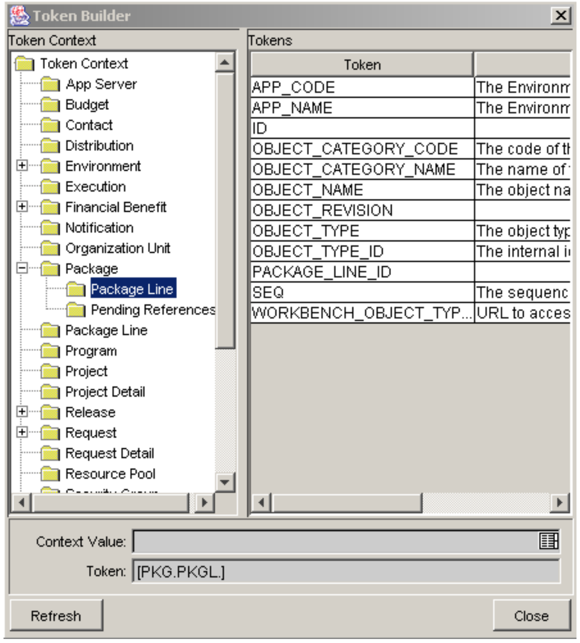
- Tokens in the Body of the message:
  Click the **Tokens** button to access the Token Builder window where Tokens can be added to the message body.

- Tokens related to specific Package Lines:
  Add Tokens to the Linked Token list to include Tokens that resolve information related to the individual Package Line.

6. Click **OK** to save the Notification specification.

Using Tokens in the Message Body

It is possible to select any of the available Tokens accessed through the Token builder to include in the body of your message. Note, however, that not all Tokens will resolve in all situations. As a general rule, Tokens associated with the Request or Workflow will resolve.



Token	
APP_CODE	The Environm
APP_NAME	The Environm
ID	
OBJECT_CATEGORY_CODE	The code of th
OBJECT_CATEGORY_NAME	The name of
OBJECT_NAME	The object na
OBJECT_REVISION	
OBJECT_TYPE	The object typ
OBJECT_TYPE_ID	The internal i
PACKAGE_LINE_ID	
SEQ	The sequenc
WORKBENCH_OBJECT_TYP...	URL to acces

Context Value:

Token: [PKG.PKGL.]

Figure 9-2 Token Builder Window

Including URLs to Open the Request (Smart URLs)

When a user receives a Notification, it is often helpful to include a link to the item that needs their attention. For example, John Smith receives a Notification stating that his Request is ready for final approval. He clicks the URL included in the body of the Notification and proceeds directly to the Request page.

Notifications can be configured in the body of the email Notification to include the Web address (URL) for the following entities:

- Packages
- Requests
- Request Types
- Projects
- Tasks
- Workflows
- Validations
- Object Types
- Environments

If end-users are viewing their mail with a Web-based mail reader (such as Microsoft Outlook or Netscape Messenger), they can then click the URL in the Notification and be taken directly to the referenced entity.

For Workflows, Request Types, Validations, Object Types and Environments the Notification can use the entity ID or the entity name as the parameter in the URL. This will bring the user to the correct window in the Workbench and open the detail window for the specified entity.

The most commonly used Smart URL Tokens for Packages and Requests are described in *Table 9-7*.

Table 9-7. Smart URL Tokens

Smart URL Token	Description
PACKAGE_URL	Provides a URL that loads the Package Details page in the standard interface.

Table 9-7. Smart URL Tokens [continued]

Smart URL Token	Description
WORKBENCH_PACKAGE_URL	Provides a URL that loads the Package window in the Workbench.
REQUEST_URL	Provides a URL that loads the Request Details page in the standard interface.

Smart URLs in HTML Formatted Messages

When using an HTML formatted message, an alternate Token must be used to provide a link to the Request.

Table 9-8. Smart URL Tokens in HTML Format

Smart URL Token	Description
REQUEST_ID_LINK	Provides a link that loads the Request Detail page in the standard interface.

The Token will resolve to the following format:

<a href=”http://URL”>Request Name</a>

In the Notification, the link would appear as:

Request Name

Note

These Tokens can also be used in plain-text formatted Notifications. They will appear with the HTML tags showing.

Using Notification Templates

Notification Templates are pre-configured Notifications that can be used to quickly construct the body of your message. Notification Templates can be used in Workflow steps, Report submissions, and Task State changes.

The following sections provide detailed instructions for:

- *Creating New Notification Templates*

- *Copying Notification Templates*
- *Deleting a Notification Template*

Creating New Notification Templates

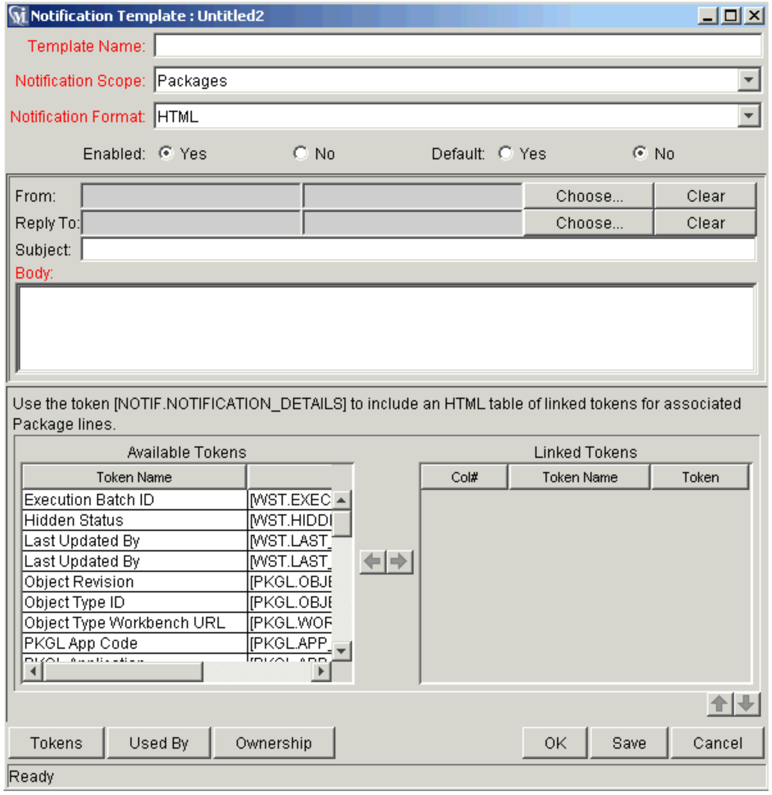
To create a new Notification Template:

1. Click the **Configuration** screen group and click the **Notification Templates** screen.

   The Notification Template Workbench opens.

2. Click **New Notification Template**.

   A Notification Template window opens.



Template Name:

Notification Scope: Packages

Notification Format: HTML

Enabled: ● Yes ○ No Default: ○ Yes ● No

From:

Reply To:

Subject:

Body:

Use the token [NOTIF.NOTIFICATION_DETAILS] to include an HTML table of linked tokens for associated Package lines.

Token Name	
Execution Batch ID	[WST.EXEC
Hidden Status	[WST.HIDDI
Last Updated By	[WST.LAST_
Last Updated By	[WST.LAST_
Object Revision	[PKGL.OBJE
Object Type ID	[PKGL.OBJE
Object Type Workbench URL	[PKGL.WOF
PKGL App Code	[PKGL.APP_

Ready

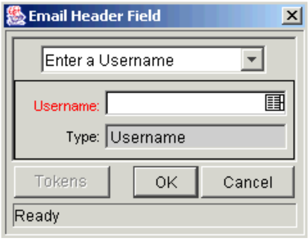
3. In the Template Name field, enter the name of the Notification Template.

4. In the Notification Scope field, select the product area where this Notification Template will be used.

5. Enter a From address.

   a. Click **Choose...**.

      The Email Header Field window opens.

   b. Select the recipient category from the drop down list (**Username**, **Email Address**, **Standard Token**, or **User Defined Token**).

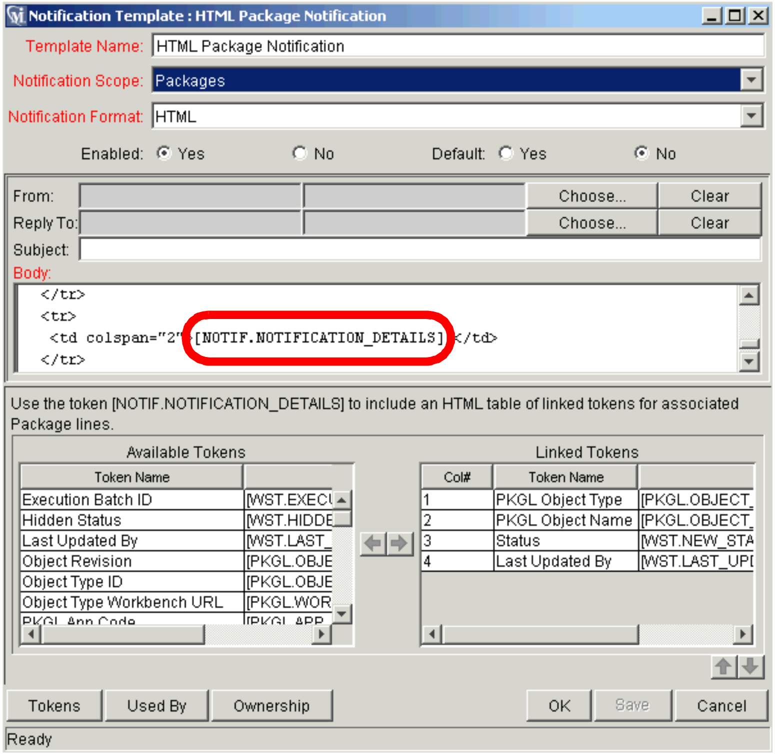
      The context-sensitive required field is dynamically updated to gather the necessary information for that category. For instance, if **Enter an Email Address** is selected from the drop down list, then it is necessary to enter an Email Address. If a **User Defined Token** is selected, click **Tokens** to bring up a full list of available Tokens or type in a specific Token.

   c. Enter the appropriate information in the required field.

   d. If a **User Defined Token** has been entered, select the type that corresponds with the evaluated Token value.

6. Enter the Notification text in the Body area.

   If using **HTML** for the Notification Format, put HTML code in the Body area. HTML Notifications for Mercury Change Management should include the Token '[NOTIF.NOTIFICATION_DETAILS]' within the <body> tags to incorporate linked Tokens.



Template Name: HTML Package Notification

Notification Scope: Packages

Notification Format: HTML

Enabled: ● Yes ○ No Default: ○ Yes ● No

From:

Reply To:

Subject:

Body:

```
  </tr>
  <tr>
   <td colspan="2">[NOTIF.NOTIFICATION_DETAILS]</td>
  </tr>
```

Use the token [NOTIF.NOTIFICATION_DETAILS] to include an HTML table of linked tokens for associated Package lines.

Available Tokens

Token Name	
Execution Batch ID	[WST.EXECU
Hidden Status	[WST.HIDDE
Last Updated By	[WST.LAST_
Object Revision	[PKGL.OBJE
Object Type ID	[PKGL.OBJE
Object Type Workbench URL	[PKGL.WOR
PKGL App Code	[PKGL.APP

Linked Tokens

Col#	Token Name	
1	PKGL Object Type	[PKGL.OBJECT_
2	PKGL Object Name	[PKGL.OBJECT_
3	Status	[WST.NEW_STA
4	Last Updated By	[WST.LAST_UPI

Ready

7. Click **OK** to save and close the window.

8. To enter a Reply To address, click **Choose...**

   The Email Header Field window opens. Follow the instructions described in step 5 for entering a From address.

9. Click **Save** to save the new Notification Template definition.

Copying Notification Templates

To copy a Notification Template:

1. Click the **Configuration** screen group and click the **Notification Templates** screen.

   The Notification Template Workbench opens.

2. Enter the search criteria and click **List**.

3. Select the Notification Template to be copied in the **Results** tab of the Notification Templates Workbench.
4. Click **Copy**.

   The Copy Notification Template window opens.
5. Enter the new Template Name.
6. Click **Copy**.

   A question dialog asks if you want to edit the new Notification Template.
7. Click **Yes** to edit or **No** to return to the Notification Template Workbench.

Deleting a Notification Template

To delete a Notification Template:

1. Click the **Configuration** screen group and click the **Notification Templates** screen.

   The Notification Template Workbench opens.
2. Click **List**.
3. Select the Notification Template to be deleted in the **Results** tab of the Notification Template Workbench.
4. Click **Delete**.

   A dialog box appears asking for confirmation that the Notification Template is to be deleted.
5. Click **Yes** to delete the Notification Template.

Note

> You can not delete Notification Templates that are referenced from a Mercury ITG Notification. To delete a Notification Template you must first remove these references. Referenced Notification Templates can be disabled.

Setting Notifications on Request Field Changes

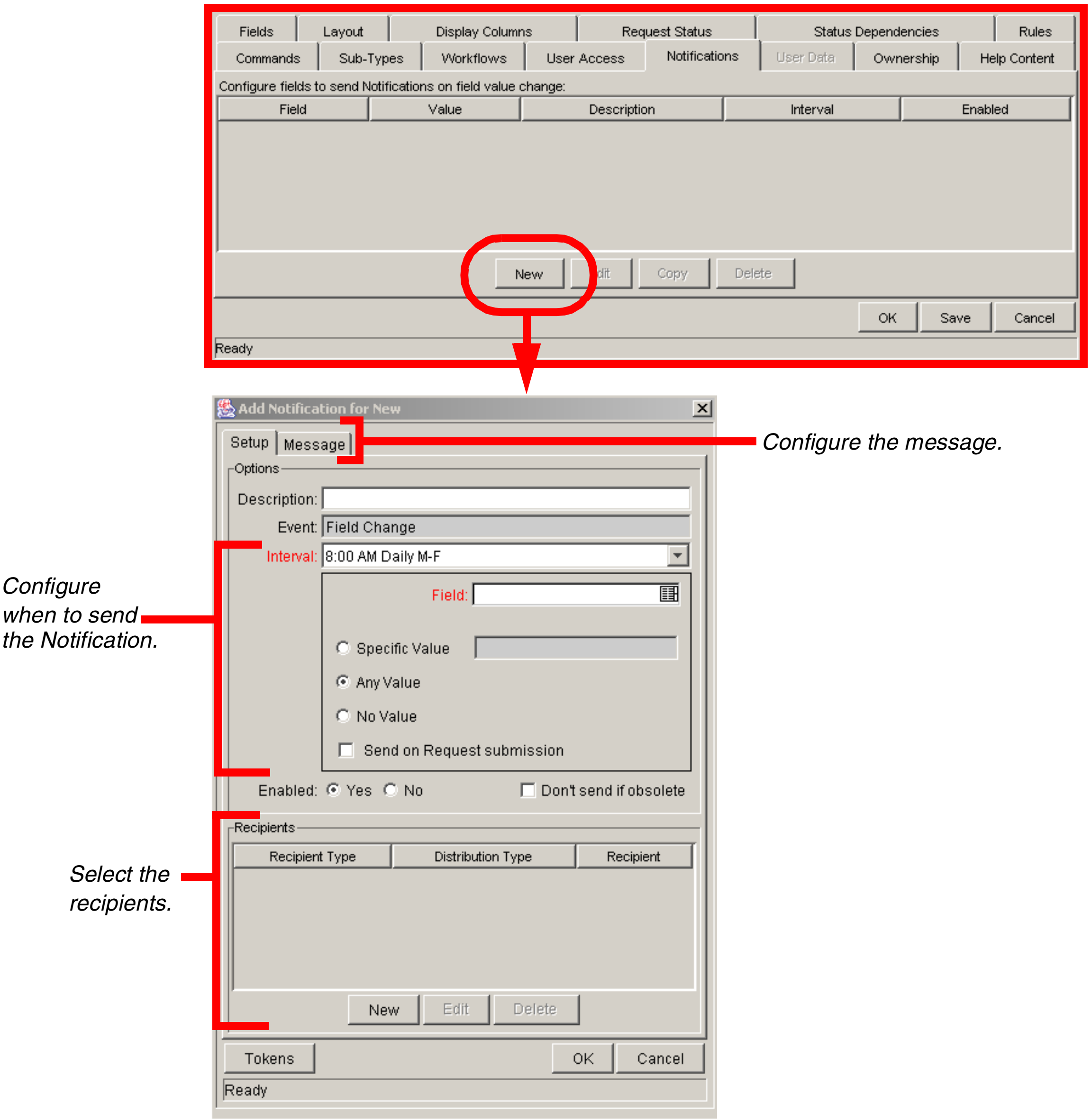
Request Types can be configured to send Notifications when a value in a particular field changes. For example, when the Assigned Group field changes to Managers, the manager group will receive a Notification that they need to review the Request.

The following fields are supported for Notification functionality:

- Priority
- Assigned To
- Contact Name
- Assigned Group
- Application
- Department
- Sub-type
- Workflow
- Request Group
- Company

Tip

Though Field Groups, Request custom fields, and User Data fields are not directly supported, use a Token evaluation Execution step to evaluate the content of these fields and trigger Notifications based on the results. Use this method to control when in the resolution process the Notification gets sent, rather than simply having a Notification sent every time a particular field changes.



Configure fields to send Notifications on field value change:

Ready

Description:

Event: Field Change

Interval: 8:00 AM Daily M-F

Field:

○ Specific Value

● Any Value

○ No Value

□ Send on Request submission

Enabled: ● Yes ○ No □ Don't send if obsolete

Ready

Configure the message.

Configure when to send the Notification.

Select the recipients.

Figure 9-3 Request Type - Notifications Tab and Add Notification Window

The Add Notification window for Request Types is nearly identical to the Add Notification for Step window for Workflow Steps. The only difference lies in the fields that configure when to send the Notification, described in *Table 9-9*.

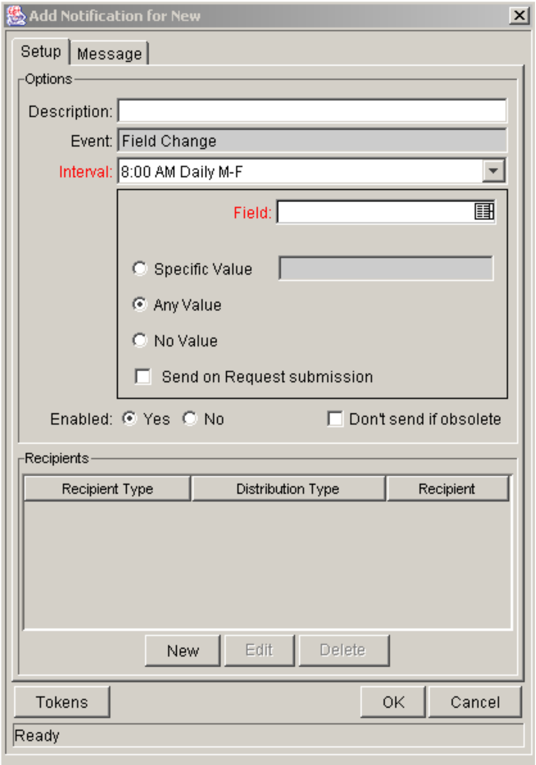
To add a Notification to a Request Type:

1. Open the Request Type.

2. Click the **Notifications** tab.

3. Click **New**.

   The Add Notification window opens.



Add Notification for New

Options

Description:

Event: Field Change

Interval: 8:00 AM Daily M-F

Field:

○ Specific Value

● Any Value

○ No Value

□ Send on Request submission

Enabled: ● Yes ○ No

□ Don't send if obsolete

Recipients

Recipient Type	Distribution Type	Recipient

Ready

4. Configure the following:

   - When the Notification is sent (Field and Value choice)
   - Who receives the Notification (Recipients)
   - The body of the Notification (Message)

   See *"Adding a Notification to a Workflow step - Overview"* on page 204 for additional details on configuring the Notification recipients and message. Table 1-9 defines the fields in the Add Notification window.

Table 9-9. Fields Controlling Request Type Field Notification

Field	Description
Event	Always set to **Field Change**. Cannot be edited.
Interval	Notifications can be sent at different intervals. For example, you might choose to send a Notification of a field change at midnight so that it's ready to be reviewed in the morning. Note also that multiple Notifications to a single recipient can be batched and sent together. Selecting an interval other than **Immediate** will allow this batching to occur. Mercury ITG includes the following preconfigured Intervals: • 8:00 AM Daily M-F: This Notification is sent every 8:00 AM on the next available work day after the Notification event occurs. • Hourly M-F: This Notification is sent every hour, starting on the next available work day after the Notification event occurs. • Immediate: This Notification is sent immediately (just once). See *"Configuring the Notification Intervals"* on page 214 for instructions on configuring this.
Field	Specifies the field that will trigger the Notification. The Field dynamically changes the Specific Value drop down list.
Specific Value	The specific value of the field triggering the Notification to be sent. If the Specific Value is checked, the drop down list is enabled. The values of the drop down list are dynamically updated by the Field value.
Any Value	Select to send the Notification when the chosen Field has any value at all.
No Value	Select to send the Notification when the chosen Field has no value.
Send on Request Submission	Checking this box means the Notification will be send only when the Request is submitted and the chosen Field matches the value specified.
Enabled	Turns the Notification on or off.
Don't send if obsolete	Checking this box means that when the Interval is not set to **Immediate**, the system will check if the chosen Field matches the value specified before sending the Notification.

5. Click **OK**.

The Notification specification is added to the **Notifications** tab. Additional specifications can be added to the step by clicking **New** and repeating the above process. A different Notification can therefore be sent to different recipients for different events.

6. Click **OK** to save and close the window.

Note

If you select another Request Header Type to use with the Request Type, you will need to reconfigure the field Notifications.

Configuring Your Dashboard

The Dashboard provides an interface through which the current state of Requests can be quickly assessed. Personalize the Dashboard to display status information that is most meaningful to your role. For example, Financial system users may only want to see the Requests that they submitted, whereas the Financial manager may want to have visibility into each critical Request currently in progress.

Each user can personalize their own Dashboard to display only information relevant to their role. When configuring a Request resolution system, consider the following configuration topics:

- *Controlling User Access to Portlets*
- *Creating Custom Portlets*
- *Creating and Distributing a Default Dashboard*

This section provides an overview of configuring the Dashboard for your Request resolution process. See *Configuring the Dashboard* for additional details.

Related Documents:

- *Using the Dashboard*
- *Configuring the Dashboard*
- *Security Model Guide and Reference*

Controlling User Access to Portlets

Control Portlet user access at two levels:

- *Disabling Portlets*
- *Restricting User Access*

Disabling Portlets

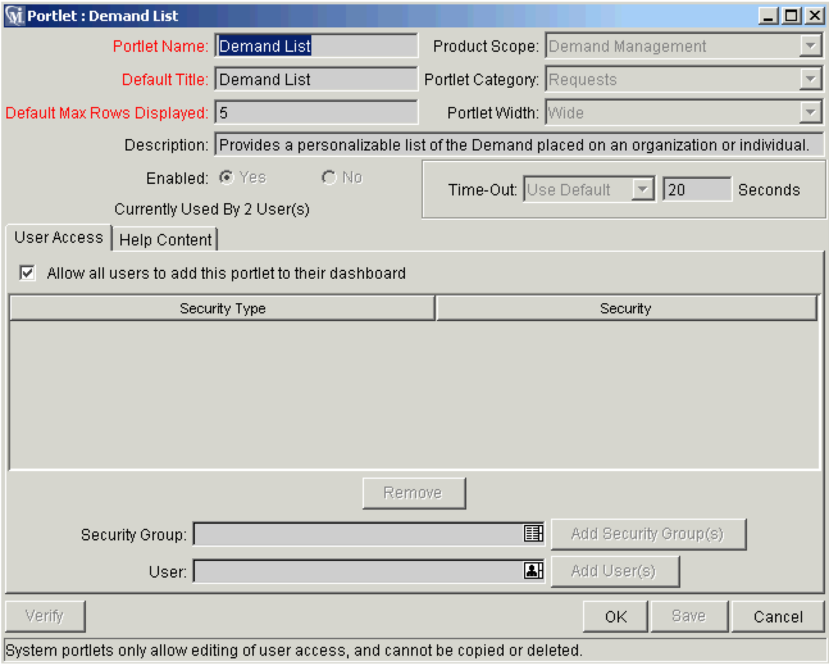
It is possible to disable custom-built Portlets at your site.

To disable a Portlet:

1. Click the **Dashboard** screen group and click the **Portlets** icon.

2. Search for and open the custom Portlet to disable.

   System Portlets can not be disabled. To control access to these Portlets, restrict user access. See *"Restricting User Access"* on page 234 for details.



Portlet : Demand List

Portlet Name: Demand List
Product Scope: Demand Management
Default Title: Demand List
Portlet Category: Requests
Default Max Rows Displayed: 5
Portlet Width: Wide
Description: Provides a personalizable list of the Demand placed on an organization or individual.
Enabled: ● Yes ○ No
Time-Out: Use Default 20 Seconds
Currently Used By 2 User(s)

☑ Allow all users to add this portlet to their dashboard

Security Type	Security

Security Group:
User:

System portlets only allow editing of user access, and cannot be copied or deleted.

3. Click Enabled = **No**.

Note

If there are any users currently using the Portlet on their Dashboard, disabling the Portlet will delete it from their Dashboards.

4. Click **Save**.

Related Topics:

- *Using the Dashboard*
- *Configuring the Dashboard*

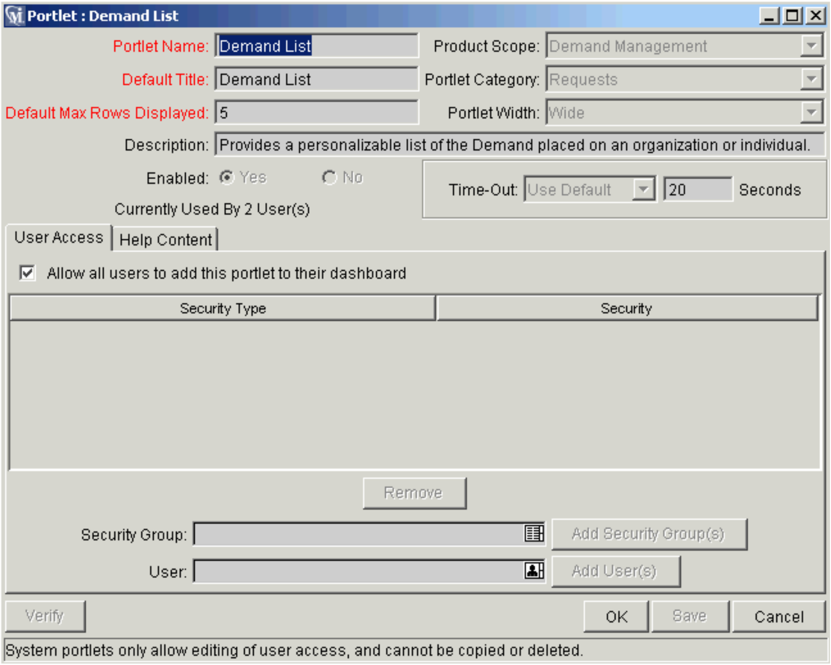
Restricting User Access

It is possible to control which users can add a Portlet to their Dashboard. For example, you may want to restrict the Request-related Portlets to only members involved in resolution processes. Enabling only the Portlets that a specific user needs will make it easier for that user to personalize their Dashboard, because there are less (non-relevant) Portlets to choose from.

To specify which users can use the Portlet on their Dashboard:

1. Click the **Dashboard** screen group and click the **Portlets** icon.
2. Search for and open the Portlet to configure.
3. Click the **User Access** tab.

   For system Portlets (such as My Packages), the **User Access** tab is the only displayed tab.



Portlet : Demand List

Portlet Name: Demand List

Product Scope: Demand Management

Default Title: Demand List

Portlet Category: Requests

Default Max Rows Displayed: 5

Portlet Width: Wide

Description: Provides a personalizable list of the Demand placed on an organization or individual.

Enabled: ● Yes ○ No

Time-Out: Use Default 20 Seconds

Currently Used By 2 User(s)

☑ Allow all users to add this portlet to their dashboard

Security Type	Security

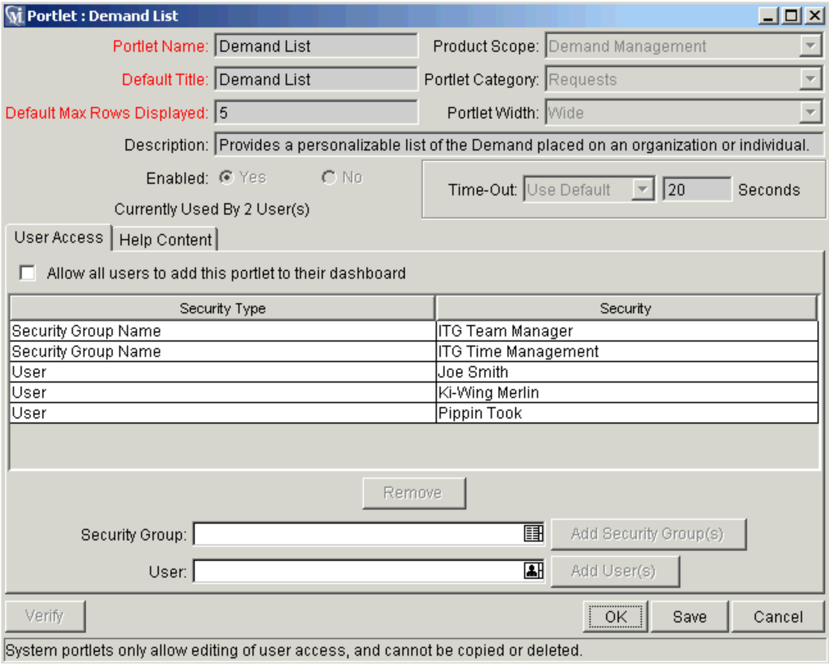
System portlets only allow editing of user access, and cannot be copied or deleted.

4. Un-check the Allow all users to add this Portlet to their dashboard field.

   The Security Group and User fields are enabled.

5. Select the desired Security Groups or Users and click the respective **Add** button.

   They are added to the **User Access** tab.



Portlet : Demand List

Portlet Name:	Demand List	Product Scope:	Demand Management
Default Title:	Demand List	Portlet Category:	Requests
Default Max Rows Displayed:	5	Portlet Width:	Wide

Description: Provides a personalizable list of the Demand placed on an organization or individual.

Enabled: ● Yes ○ No

Time-Out: Use Default 20 Seconds

Currently Used By 2 User(s)

User Access | Help Content

□ Allow all users to add this portlet to their dashboard

Security Type	Security
Security Group Name	ITG Team Manager
Security Group Name	ITG Time Management
User	Joe Smith
User	Ki-Wing Merlin
User	Pippin Took

Security Group:

User:

System portlets only allow editing of user access, and cannot be copied or deleted.

6. Click **Save**.

Access can be restricted by specifying multiple Security Groups and Users for each Portlet. Only members of the specified Security Group or the specified users can add this Portlet to their Dashboard.

Note

It is possible to restrict user access for both custom and system Portlets.

Creating Custom Portlets

Portlets are visual displays that act as windows into the Demand Management data. While Mercury ITG’s system Portlets (provided at the time of installation) are personalizable by end-users and provide wide data access. Custom Portlets can also be created to access additional information in the system or in other databases. These custom Portlets behave the same as the system Portlets, using filter fields to limit the displayed data. Textual or graphical Portlets can be created.

Since custom Portlets are data-driven entities and require extracting information stored in the database, knowledge of SQL is required for users who wish to create or configure Portlets.

See the *Configuring the Dashboard* for detailed instructions on creating custom Portlets.

Note

Before creating custom Portlets, consider using one of the Mercury ITG system Portlets. Review the business and data presentation Portlet requirements and compare against the following list of system Portlets.

Creating and Distributing a Default Dashboard

To ensure that all participants in your Request processes have visibility into the most relevant information, configure and distribute Dashboard pages for those users. When distributing Dashboard pages, use the following approaches:

- *Publishing Required Dashboard Pages*
- *Distributing Optional Dashboard Pages*
- *Creating a Default Dashboard*

For instructions on implementing these functions, see *Configuring the Dashboard*.

Publishing Required Dashboard Pages

Publishing involves the dissemination of required Dashboard pages and Portlets to one or many user Dashboards at one time. Publishing is a regimented approach to disseminating Dashboard pages and Portlets. The Dashboard pages and Portlets of a published Module can not be edited or removed by the owner of the Dashboard.

Distributing Optional Dashboard Pages

Distributing involves the dissemination of optional Dashboard pages and Portlets to one or many Dashboards at one time. Distributing provides a more flexible approach to disseminating Portlets and Dashboard pages than publishing. Once distributed, the Portlets and Dashboard pages can be edited by the user.

Creating a Default Dashboard

The Default Dashboard allows first-time users quick and easy integration of the Dashboard into their business processes. The Default Dashboard can be published or distributed. Published Default Dashboard pages are disseminated to Dashboards as read-only pages. Distributed Default Dashboard pages can be edited by the owner of the Dashboard.

Note

Users with the Dashboard, Configure Module and Dashboard, Distribute Module Access Grants can configure and distribute the Default Dashboard to all users.

Configuring Reports

Mercury Demand Management includes a pre-defined set of HTML-based reports that are accessed through a Web browser. The reports allow users to view the current detailed status of their data at any point in time.

Demand Management also provides a Reporting Meta Layer, which allows users to build their own custom reports using third-party reporting tools.

See *Reports Guide and Reference* for a full list of available reports and information on configuring them.

Creating Custom Help for Requests

Custom Help content can be defined for each Request in the system. This provides end users with accessible information related to a Request, a section on the Request, or a specific field on the Request. For example, one of the Request's includes a Priority field to specify the urgency of a particular Request. The values in the Priority field are **Critical**, **High**, **Medium**, and **Low**. Help content can be configured for that field to instruct the user when to mark a Request as **Critical** rather than **High**.

See *"Adding Custom Online Help to the Request Type"* on page 166 for details.

Chapter 10 Rolling Out a Request Resolution System

This chapter provides checklists and instructions for rolling out the Request resolution process to your users. This includes information on using Migrators, enabling user access, and training the users to use the system.

This chapter covers the following topics:

- *Test the Request Resolution System - Checklists*
- *Migrate Your Configuration Data into Production*
- *Enable Entities and User Access*
- *Educating Your Users*

Test the Request Resolution System - Checklists

This section provides a series of high-level checklists to help you validate the system before rolling it out to the users.

The following topics are discussed:

- *General Configuration Checklist*
- *Workflow Checklist*
- *Request Type Checklist*
- *Security / User Access Checklist*
- *Dashboard / Portlet Checklist*
- *Cross-Entity Checklist*

General Configuration Checklist

The following items have to be configured to enable your Request resolution system. See the referenced sections in the Notes column for additional details and instructions on configuring each of the entities.

Table 10-1. General Configuration Checklist

	Entity Defined?	Notes
	Workflow	One or more Workflows that will be used to process the Requests must be defined. See the following sections for details on Workflow construction: • *"Mapping your Process into a Workflow"* on page 53 • *"Advanced Workflow Topics"* on page 251
	Request Types	A Request Type must be defined for each type of Request to be resolved. This includes creating fields that describe the Request and decisions and field logic required to process it during resolution. See the following sections for details on Request Type construction: • *"Constructing the Request Type"* on page 111 • *"Validations"* on page 283 • *Commands and Tokens Guide and Reference*
	Security Groups / User Access	Define the Security Groups used to control different aspects of the deployment process: Request creation, Request processing, and Request resolution system configuration. See the following sections for details on Security Group and User Participant definition: • *"Integrating Participants into Your Request Resolution System"* on page 177 • *Security Model Guide and Reference*
	Dashboard / Portlets	Decide which Portlets can be added to the Dashboard. If none of the default system Portlets suit your business needs, construct your own custom Portlets. See the following sections for details on Portlet construction and default Dashboard creation: • *"Setting Up Communication Paths"* on page 203 • *Configuring the Dashboard*

Workflow Checklist

Table 10-2. Workflow configuration checklist

	Workflow Check Item	Notes
	Business process is modeled on the Workflow	Execution, decision and condition steps have been added to the **Layout** tab on the Workflow window. See the following sections for details: • *"Building the Workflow Skeleton - Overview"* on page 53
	Decision steps set	See the following sections for details: • *"Create the Required Step Source"* on page 56
	Timeouts are set	Timeout values have been placed on how long Workflow steps can remain in a single state, and timeouts have added to command executions. This ensures that the process is not delayed from lack of user action or complications during executions. See the following sections for details: • *"Creating a Decision Type Step"* on page 60 • *"Create an Execution Type Step"* on page 65
	Automatic transitions are properly set	Ensure that the Request will not become "stuck" in a step. This can happen when the results of an execution or query yield a result that is not linked to a transition out of the step. See the following sections for details: • *"Adding Transitions Between Steps"* on page 97
	Manual transitions are set	Ensure that the step has a transition path for each available decision result. See the following sections for details: • *"Adding Transitions Between Steps"* on page 97
	Notifications are set on appropriate Workflow steps	Configure Notifications to be sent at specific points in the process. See the following sections for details: • *"Adding a Notification to a Workflow step - Overview"* on page 204
	Includes a Close step.	The process should conclude with a "Closed" Request.
	Verify the Workflow	Use the Workflow's Verify tool to ensure that serious configuration errors were not made. The Workflow verification tool checks for the possible configuration errors described in *Table 10-3*.

Table 10-3. Workflow Logical Guidelines

Guideline	Returns	Reason
Workflow should have at least one step.	Error	No processing can be done if the Workflow has no steps.
Workflow should have at least one Close step.	Error	The Request cannot be closed without a Close step in the Workflow.
Each enabled Workflow step should have at least one incoming transition	Error	It is not possible to flow to a Workflow Step without an incoming transition.
Each decision step should have at least one Security Group, user or Token defined in the **Security** tab.	Error	No one is authorized to act on the step without a Security Group.
Each manual execution step should have at least one Security Group, user or Token defined in the **Security** tab.	Error	No one is authorized to act on the step without a Security Group.
First Workflow step should not be a condition.	Error	Workflow processing may not be correct if the first step is a condition.
A condition step should not have a transition to itself.	Error	A condition with a transition to itself could cause the Workflow to run indefinitely.
Transition value is not a valid Validation value (error).	Error	The Validation value has changed since the transition has been made.
Close steps should not have a transition on 'Success' or 'Failure.' Return steps should have no outgoing transitions.	Error	The Request will not close if a transition exists on Success.
An immediate execution step should not have a transition to itself on 'Success' or 'Failure.'	Error	The Workflow could loop indefinitely.
'Other Values' and 'All Values' transitions should not exist at the same step.	Warning	Other Values transition is always ignored if an All Values transition exists.
Each Workflow Step should have at least one outbound transition.	Warning	The branch of the Workflow stops indefinitely without closing the Request.
Each value from a list-validated Validation should have an outbound transition.	Warning	There are Validation values that do not have transitions defined.
Step with text or numeric Validation should have an 'Other Values' or 'All Values' transition.	Warning	Since text and numeric Validations are not limited, an Other Values or All Values transition should be defined.
All steps should be enabled.	Warning	Disabled steps cannot be used by a Request.

Table 10-3. Workflow Logical Guidelines

Guideline	Returns	Reason
AND or OR condition step should have at least two incoming transitions.	Warning	An AND or OR condition with only one incoming transition will always immediately be true and have no effect.
Subworkflow should have at least one Return step.	Error	Should include a Return step.
Notifications with reminders should not be set on results that have transitions.	Error	Transition into the Return Step does not match the Validation.
Close step in Subworkflow will close entire Request.	Warning	Has a Close step.
Top-level Workflow should not have a Return step.	Error	Only Subworkflows have a Return step.
Request status for a step not linked to a Request Type that uses this Workflow.	Warning	Request cannot handle status.

Request Type Checklist

Table 10-4. Request Type configuration checklist

	Request Type Check Item	Notes
	Request Header Type associated	A Request Header Type should be associated with the Request Type. If no satisfactory Request Header Type exists, a new one can be created. See the following sections for details: • *"Choosing a Request Header Type"* on page 114
	Fields defined	Fields are required to define the Request. Ensure the correct parameters are used to describe the Request to be processed. See the following sections for details: • *"Creating a Field"* on page 133 • *"Validations"* on page 283
	Request Rules defined	Rules can be set for the automatic population of fields in the Request. See the following sections for details: • *"Configuring Request Type Defaulting Behavior (Rules)"* on page 145

Table 10-4. Request Type configuration checklist

	Request Type Check Item	Notes
	Request Statuses defined	The Statuses the Request can take on should be defined and associated with the Request Type. New Statuses can be added to the list if necessary. See the following sections for details: • *“Creating Request Statuses”* on page 157
	Status Dependencies set	Request fields can be configured to be hidden, required, read-only, cleared, or reconfirmed based on the Status of the Request. See the following sections for details: • *“Configuring Field Behavior Using Status Dependencies”* on page 157
	Request security set	It is possible to exercise a great deal of control over who can participate in your Request resolution process. See the following sections for details: • *“User Security and Participation - Overview”* on page 177
	Request field security set	Request fields can be configured to be hidden to particular users or Security Groups. See the following sections for details: • *“User Security and Participation - Overview”* on page 177

Security / User Access Checklist

Table 10-5. Security / User Access Configuration Checklist

	Security / User Access Check Item	Notes
	Created Security Groups (for access to screens and functions)	Security groups have been created to grant access to certain screens and functions. See the following sections for details: • *“Establishing Security Groups”* on page 178 • *Security Model Guide and Reference*
	Created Security Groups (for association with Workflow steps)	Security Groups have been created to allow users to act on a specific Workflow step. See the following sections for details: • *“Establishing Security Groups”* on page 178 • *Security Model Guide and Reference*
	Set security on Request Creation	All available options have been set for restricting who can create and submit Requests. See the following sections for details: • *“Setting Request Creation Security”* on page 186

Table 10-5. Security / User Access Configuration Checklist

	Security / User Access Check Item	Notes
	Set security on Request processing	All available options have been set for restricting who can process Requests. See the following sections for details: • *"Setting Request Processing Security"* on page 190
	Set security on Request field visibility	All available options have been set for restricting who can view or edit particular Request fields. See the following sections for details: • *"Creating a Field"* on page 133
	Set security on Request resolution system configuration	Specified who can modify the Request resolution process. This includes editing the Workflow, Request Type, Security Groups, etc. See the following sections for details: • *"Setting Configuration Security"* on page 197

Dashboard / Portlet Checklist

Table 10-6. Dashboard / Portlet Configuration Checklist

	Dashboard Check Item	Notes
	Created custom Portlets to display desired data	Advanced users with a knowledge of SQL programming can create their own Dashboard. See the following sections for details: • *"Configuring Your Dashboard"* on page 232
	Enable Portlets for use on the Dashboard	See the following sections for details: • *"Configuring Your Dashboard"* on page 232 • *Configuring the Dashboard*
	Specify which users can add use certain Portlets	See the following sections for details: • *"Configuring Your Dashboard"* on page 232 • *Configuring the Dashboard*
	Create a default Dashboard or distribute a Dashboard to users.	See the following sections for details: • *Configuring the Dashboard*

Cross-Entity Checklist

Table 10-7. Cross Entity Configuration Checklist

	Entities	Configuration Considerations
	Request Header Type and Request Type	The following items should be coordinated between the Request Header Type and Request Type: • Decide which Request Header Type will be used with the Request Type.
	Workflow and Request Type	The following items should be coordinated between the Workflow and Request Type: • Decide which Request Type statuses correspond to which Workflow steps. • Decide which Workflow steps will execute the Request Type commands, if any. • Workflow step source Validations and Request Type field Validations are in agreement. This is required when transitioning based on a field value (using Token, SQL or PL/SQL execution types) • Allow the Request Type use for the Workflow (set in the Workflow window - **Request Types** tab). • Allow the Workflow to be used by the Request Type (set in the Request Type window - **Workflows** tab).
	Workflow and Security Groups	The following items should be coordinated between the Workflow and Security Groups: • Associate Security Groups with Workflow steps. Users in the included groups can act on the step. • Set Workflow and Workflow step ownership.
	Security Groups and other entities (Request Types, Environments, etc.)	Set Ownership Groups for these entities. Members of the Ownership Group (determined by associating Security Groups) are the only users who can edit the entities.

Migrate Your Configuration Data into Production

“Developing Your Configurations (Using Migrators)” on page 23 discusses using multiple Mercury Change Management instances to configure and deploy your configuration data. After all entities have been validated in a

Development or Testing environment, perform the final migration into production. The following entities can be transferred using the migrators:

- Workflows
- Object Types
- Validations
- User Data Context
- Special Commands
- Report Types
- Request Types
- Request Header Types
- Project Template
- Portlet Migrator

Note

When migrating the above entities, related configuration information is also migrated. For example, when migrating the Object Type the following information is also migrated: Validations referenced by the Object Type fields, Environments referenced by the Validations, and Special Commands referenced by Commands or Validations.

All migrations can be processed in a single Package. Each individual entity (such as Workflow A, Workflow B, Object Type 1, or Object Type 2) is added as a separate Package Line. See *Migrators Guide and Reference* for detailed instructions on using the migrators.

The entity will inherit the Enabled or Disabled status. For example, if the Object Type is disabled in the Test instance, then it will be disabled in the Production instance upon migration. Ensure that each entity has the desired Enabled or Disabled status in the production instance.

Enable Entities and User Access

Each entity used in the Request resolution process includes an Enabled parameter. This parameter needs to set to **Yes** in order to provide general access. Ensure that the following entities are enabled in the system:

- Workflows (including Subworkflows)
- Request Types
- Request Header Types
- Security Groups
- Users
- Portlets

Educating Your Users

The final step in rolling out the system is training the users. This includes the educating the users on the following activities:

- Basic product use: creating, processing, and reporting on Requests
- Process-specific training:
  Ensure that each of the users understands the general Request process. Plan a formal roll-out meeting or publish documents on the configurations and processes at the site.
- User Responsibilities:
  Ensure that each user understands their individual role in the process. Also, take advantage of the product's email Notification functionality. The Notifications can be very specific, instructing individual users with their required actions.

Additional Resources for Education and Roll-Out

Consider using the following resources to assist in the education and roll-out activities:

Mercury Education and Online courses

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

Online Help

The Online Help provides details on creating and processing Requests, Packages, Projects and Tasks. This Help system features information on using Mercury Change Management's main HTML interface and the Dashboard.

Power Users can access other product documentation from the Workbench. From the **Help** menu, select **Mercury IT Governance Library**. A single page opens and provides access to all user, configuration and administration documentation.

Appendix A
Advanced Workflow Topics

This chapter provides instructions for implementing advanced Workflow features. This includes configuring Subworkflows, integrating Workflows used in different Mercury IT Governance products, and using Workflow parameters.

This appendix covers the following topics:

Using Subworkflows

A Subworkflow is any Workflow that is referenced from within another Workflow. Use Subworkflows to model complex business processes into logical, more manageable and reusable sub-processes.

A Subworkflow can be selected from the Workflow Step Sources window and dragged onto the **Layout** tab. When the Package, Request, or Release Distribution reaches the Subworkflow step, it follows the path defined in that Subworkflow. The Subworkflow will either close within that Workflow or return to the parent Workflow.

Subworkflows are defined in the Workbench using the same process as when configuring a Workflow. When creating a Subworkflow, be sure to set the following:

- The Workflow window contains a Subworkflow radio button which should be set to **Yes.**
- The Validation for the step leaving the Subworkflow layout should match the Subworkflow step in the parent Workflow.

Note

Subworkflows can also be generated by copying and renaming an existing Subworkflow.

This section covers the following topics:

- *Transitioning to a Subworkflow*
- *Transitioning From a Subworkflow*

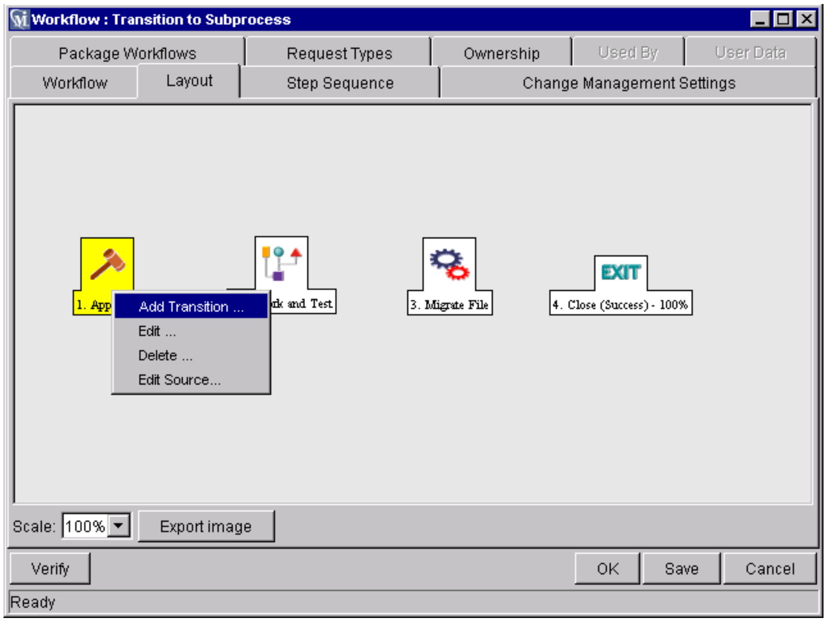
Transitioning to a Subworkflow

A transition to a Subworkflow Step is made in the same way as a transition to any other Workflow Step (Execution, Decision or Condition).

To transition to a Subworkflow:

1. Open a Workflow and click the **Layout** tab.
2. Right-click the source Workflow Step and select **Add Transition** from the pop-up menu.

   This creates an arrow from the source Workflow Step.



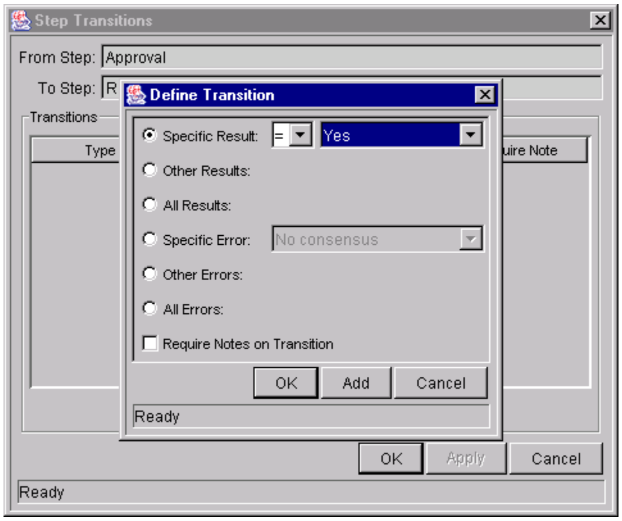
Scale: 100%

3. Drag the arrow to the destination Step and left-click.

   The Step Transitions window opens.

4. Click **New**.

   The Define Transition window opens.



From Step: Approval

To Step: R

● Specific Result: = Yes

○ Specific Error: No consensus

□ Require Notes on Transition

5. Define the desired transaction result by:

    a. Clicking one of the radio buttons for results (Specific Result, Other Results, etc.).

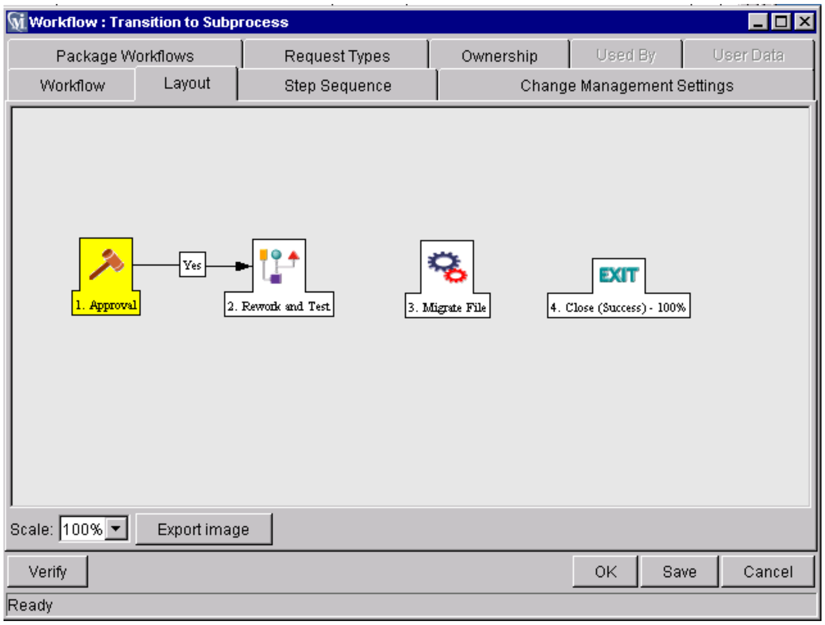
    b. Selecting **=** or **!=** (does not equal) from the drop down list.

    c. Selecting **Yes** or **No** from the drop down list.

    If the Workflow Step results in this value, the Request or Package proceeds along this path.

6. Click **OK** to close the Define Transition window.

7. Click **OK** to finalize the step transition definition.

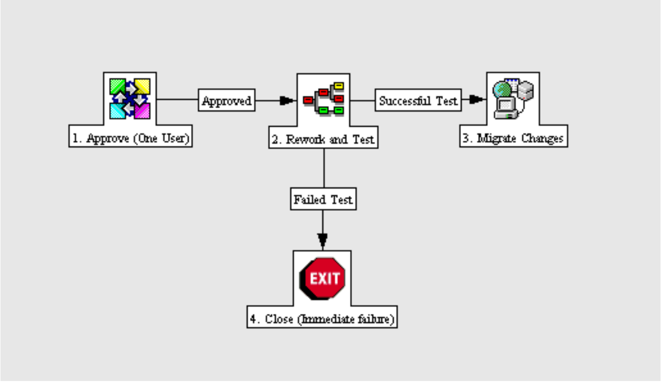


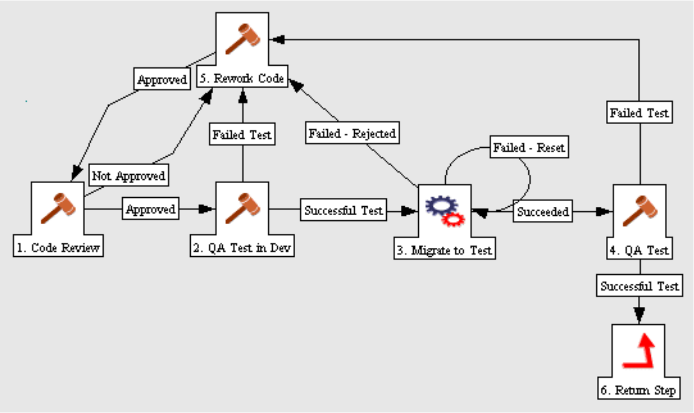
The transition is graphically represented by an arrow between the two steps. The Package Line or Request proceeds to the First Step designated in the Subworkflow definition.

Transitioning From a Subworkflow

When the Package or Request reaches the Subworkflow Step, it follows the path defined in that Subworkflow. It either closes within that Workflow (at a Close step) or returns to the parent Workflow.

For a Package Line or Request to transition back to the parent Workflow, the Subworkflow must contain a Return step. The transitions leading into the Return step must match the Validation established for the Subworkflow Step. In the following example, the transitions exiting the Rework and Test step (**Successful Test** and **Failed Test**) match the possible transitions entering the Subworkflow's Return Step.





Users must verify that the Validation defined for the Subworkflow Step is synchronized with the transitions entering the Return Step. The Subworkflow Validation is defined in the Workflow window.

Users typically define the possible transitions from the Subworkflow Step during the Subworkflow definition.

Note

The Subworkflow Step validation cannot be edited if the Subworkflow is used in another Workflow definition.

The Subworkflow field cannot be edited if the Subworkflow is used in another Workflow definition.

Package - Request Workflow Integration

Request and Package Workflows can be configured to work together, communicating at key points in the Request and Package processes. A Request Workflow Step can actually jump to a preselected Package Workflow Step. The Package Workflow step receives the Request Workflow Step and acts on it to go to the next step in the process.

Note

Packages and Requests can also be integrated at a level that does not rely on the Workflow configuration. Attach Packages and Requests to each entity as References. Dependencies can then be set on these reference to control the behavior of the Request or Package. For example, you can specify that a Request is a “Predecessor” to the Package. This means that the Package will not continue until the Request closes.

Two built-in Workflow Events facilitate this cross-product Workflow integration. These steps are **wf_jump** and **wf_receive**. Jump (**wf_jump**) and Receive (**wf_receive**) steps are used at the points of interaction between Workflows. Each Jump step must be coupled with a Receive step. Workflows can communicate through these Jump and Receive pairs.

As an example of when this kind of communication is useful:

1. A Request spawns a Package for migrating new code to the Production environment.
2. The newly spawned Package must go through an Approval step in Mercury Change Management.

3. When the Approval step is successful, the process jumps back to and is received by the Request. The Request then undergoes more testing and changes in the QA Environment.

4. After successfully completing the QA Test, the process jumps from the Request and is received by the Package.

5. Since the step has succeeded, the process can now migrate the code changes to the Production Environment.

This process is graphically represented in *Figure A-1*.

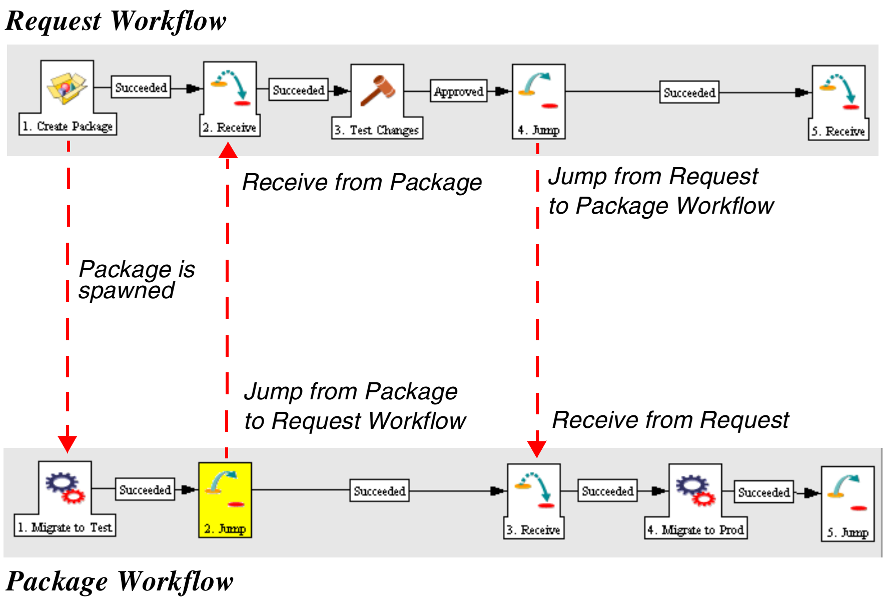


Figure A-1 Jump/Receive Workflow Steps

The Jump and Receive pair must be carefully coordinated. Each Jump step must have an associated Receive step, linked together by a common Jump/Receive Step Label defined in the Workflow Step window. The transition values for entering into and exiting the Jump and Receive steps must also be coordinated.

This section details the process for setting up a successful Request - Package Workflow integration.

To establish communication between Requests and Packages:

1. Set up the 'WF - Jump/Receive Step Labels' Validation for use in the Workflow Step window.

This validation is used to group a Jump and Receive step. The selected Jump/Receive Step Label must match in the paired Jump and Receive Workflow Step windows. See *"Setting Up the 'WF - Jump/Receive Step Labels' Validation"* on page 258.

2. Create a Jump step using the **wf_jump** Built-in Workflow Event.

   See *"Generating a Jump Step Source"* on page 260.

3. Create a Receive step using the **wf_receive** Built in Workflow Event.

   See *"Generating a Receive Step Source"* on page 262.

4. Verify that both the Jump and Receive steps specify the same Jump/Receive Step Label.

   See *"Including the Jump/Receive pair in Workflows"* on page 264.

5. Verify that the transitions exiting the Jump and Receive steps match the possible values entering the Jump step.

Setting Up the 'WF - Jump/Receive Step Labels' Validation

To set up the WF - Jump/Receive Step Labels Validation:

1. Click the **Configuration** screen group and click the **Validations** icon.

   The Validation Workbench opens.

2. On the **Query** tab, enter **WF - Jump/Receive Step Labels** in the Validation Name field.

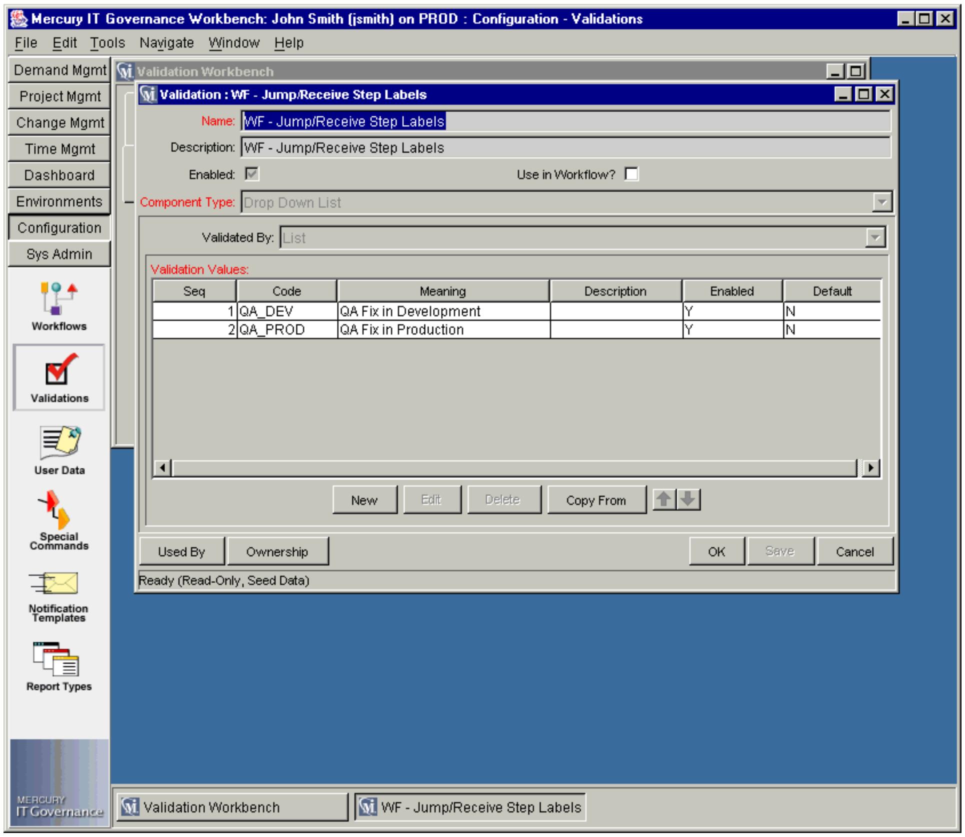
3. Click **List**.

4. Click the **Results** tab.

   The WF - Jump/Receive Step Labels is listed in the **Results** tab.

5. Click **Open**.

   The Validation window opens.



Validation : WF - Jump/Receive Step Labels

Name: WF - Jump/Receive Step Labels

Description: WF - Jump/Receive Step Labels

Enabled: ☑ Use in Workflow? □

Component Type: Drop Down List

Validated By: List

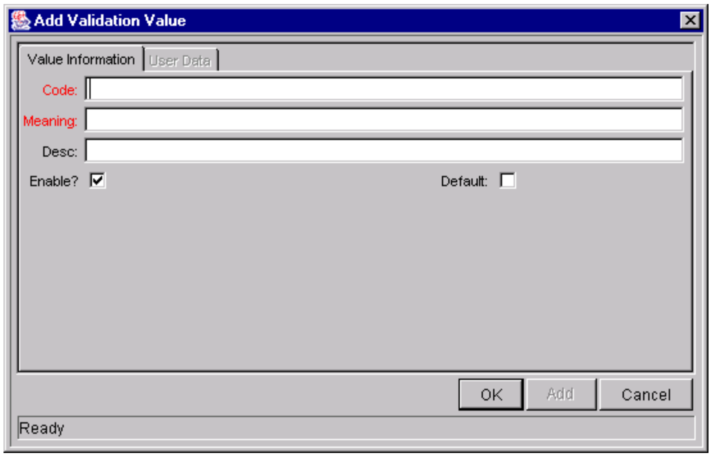
Validation Values:

Seq	Code	Meaning	Description	Enabled	Default
1	QA_DEV	QA Fix in Development		Y	N
2	QA_PROD	QA Fix in Production		Y	N

Ready (Read-Only, Seed Data)

6. Click **New** to define a new Validation Value that is used to link two Workflows together.

   The Add Validation Value window opens.



Code:

Meaning:

Desc:

Enable? ☑ Default: □

Ready

7. Enter the desired Code, Meaning and Description in the appropriate fields.

8. Click **OK** to close the Add Validation Value window.

9. Click **Ownership** to select which Ownership Groups will have the ability to edit this Validation.

10. Click **OK** to close the Validation window.

The new Validation Value is now included in the Jump/Receive Step Label drop down list in the Workflow Step window.

Generating a Jump Step Source

To create a Jump step using the wf_jump Built-in Workflow Event:

1. Click the **Configuration** screen group and click the **Workflows** icon.

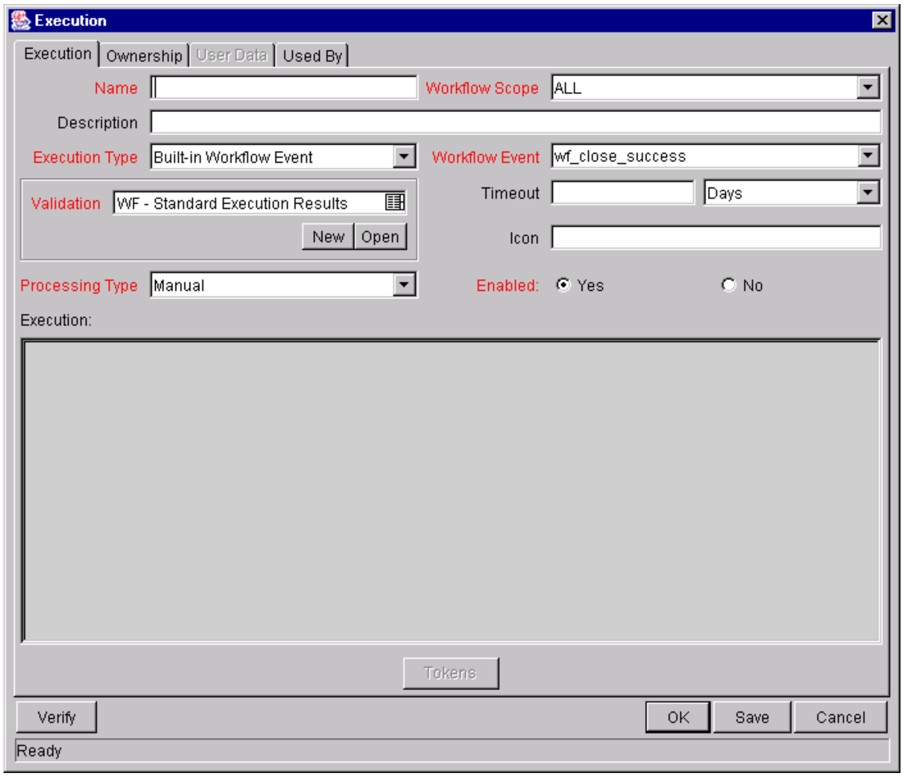
   The Workflow Workbench and Workflow Step Sources window open.

2. Select the Workflow Step Sources window.

3. Select the Executions folder.

4. Click **New**.

   The Execution window opens.



Name:

Workflow Scope: ALL

Description:

Execution Type: Built-in Workflow Event

Workflow Event: wf_close_success

Validation: WF - Standard Execution Results

Timeout: Days

Icon:

Processing Type: Manual

Enabled: ● Yes ○ No

Execution:

Ready

5. Select either **Packages** or **Requests** from the Workflow Scope drop down list, depending on the desired application of the Workflow.

   Package Level Subworkflows can not include jump and receive steps.

6. From the Execution Type drop down list, select **Built-in Workflow Event**.

7. From the Workflow Event drop down list, select **wf_jump**.

8. From the Validation drop down list, select or create a Validation which will be used to transition out of this Workflow Step.

Note

The Validation values exiting the Jump step must match the possible Validation values entering the Jump step.

9. Fill in any other required or optional information in the Execution window (such as Name, Description or Processing Type).

10. Click the **Ownership** tab to select which Ownership Groups will have the ability to edit this Execution step.

11. Click **OK**.

    The Workflow Step is added to the Workflow Step Sources window.

This Workflow Step can now be used in any new or existing Workflow within the step's defined Workflow Scope. Remember that every Jump step must have a paired Receive step in another Workflow.

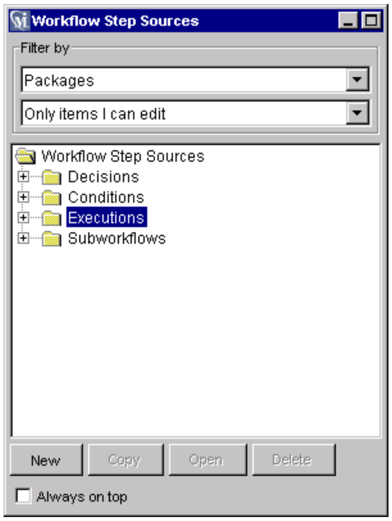
Generating a Receive Step Source

To create a Receive step using the wf_receive Built-in Workflow Event:

1. Click the **Configuration** screen group and click the **Workflows** icon.

   The Workflow Workbench and Workflow Step Sources window open.

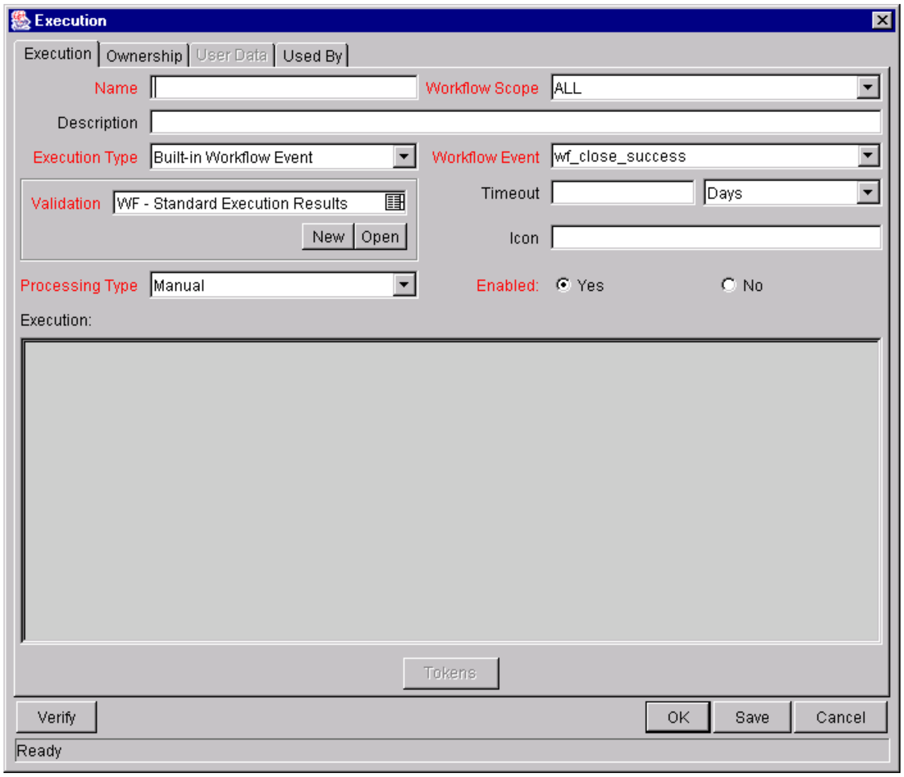
2. Select the Workflow Step Sources window.



3. Select the Executions folder.

4. Click **New**.

   The Execution window opens.



Name		Workflow Scope	ALL
Description			
Execution Type	Built-in Workflow Event	Workflow Event	wf_close_success
Validation	WF - Standard Execution Results	Timeout	Days
		Icon	
Processing Type	Manual	Enabled:	● Yes ○ No

Execution:

Ready

5. From the Workflow Scope drop down list, select either **Packages** or **Requests** depending on the desired application of the Workflow.

6. From the Execution Type drop down list, select **Built-in Workflow Event**.

7. From the Workflow Event drop down list, select **wf_receive**.

8. Select or create a Validation which will be used to transition out of this Workflow Step.

Note

The Validation values exiting the Receive step must match the possible Validation values entering and exiting the Jump step.

9. Fill in any other required or optional information (such as Name, Description or Processing Type).

10. Click the **Ownership** tab to select which Ownership Groups will have the ability to edit this Execution step.

11. Click **OK**.

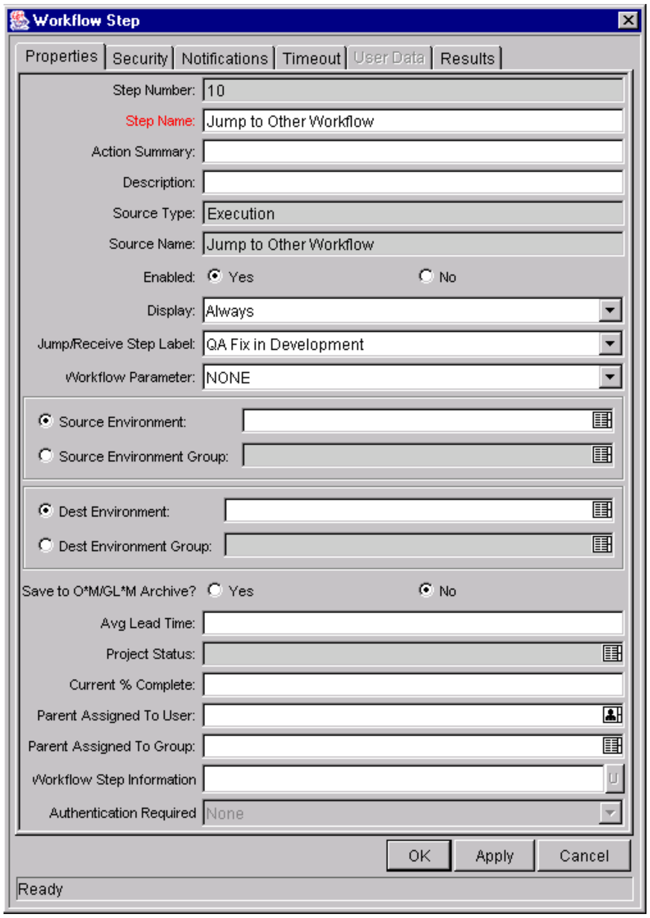
    The Workflow Step is added to the Workflow Step Sources window.

This Workflow Step can now be used in any new or existing Workflow within the step's defined Workflow Scope. Remember that every Receive step must have a paired Jump step in another Workflow.

Including the Jump/Receive pair in Workflows

1. Drag either the Jump or Receive step from the Workflow Step Sources window into the Workflow's **Layout** tab.

   The Workflow Step window opens.



Step Number: 10
Step Name: Jump to Other Workflow
Action Summary:
Description:
Source Type: Execution
Source Name: Jump to Other Workflow
Enabled: ● Yes ○ No
Display: Always
Jump/Receive Step Label: QA Fix in Development
Workflow Parameter: NONE
● Source Environment:
○ Source Environment Group:
● Dest Environment:
○ Dest Environment Group:
Save to O*M/GL*M Archive? ○ Yes ● No
Avg Lead Time:
Project Status:
Current % Complete:
Parent Assigned To User:
Parent Assigned To Group:
Workflow Step Information
Authentication Required None
Ready

2. Select an item from the Jump/Receive Step Label drop down list.

   This item must be the same for a paired Jump/Receive Step.

Note

The Jump/Receive Step Label is the key communication link between separate Workflows. The communicating Jump and Receive Workflow Steps must have a matching Jump/Receive Step Label. It is also important that the Jump/Receive Step Label is unique for any Jump and Receive pair.

3. Enter any additional Workflow Step information.
4. Click **OK.**
5. Repeat the above process for the other paired Workflow Step (Jump or Receive), depending on which one was configured first

Using Condition Steps

It is possible to perform complex routing based on the status of multiple Workflow Steps using Condition steps. *Table 10-8* lists the five available Condition steps:

Table A-1. Workflow Condition Steps

Condition	Description
AND	Waits until all parallel Workflow branches of a single Package Line or Request have reached this point before continuing processing.
FIRST LINE	Only used for Package Workflows and has a validation of True/False. Evaluates to true for the first Package Line to reach the condition. Used for multiple Package Lines.
LAST LINE	Only used for Package Workflows. Has a validation of True/False. Evaluates to true for the last Package Line to reach the condition. Used for multiple Package Lines.
OR	Allows the first branch of the Workflow that reaches this point to continue and terminates the other branches upon their arrival at this point so only one will continue.

Table A-1. Workflow Condition Steps [continued]

Condition	Description
SYNC	Only used for Package Workflows. This condition halts further processing of each Package Line until all Lines of the Package reach this step. This is used to synchronize all the Lines of a Package at one point. Used for multiple Package Lines.

The following sections provide examples of the following:

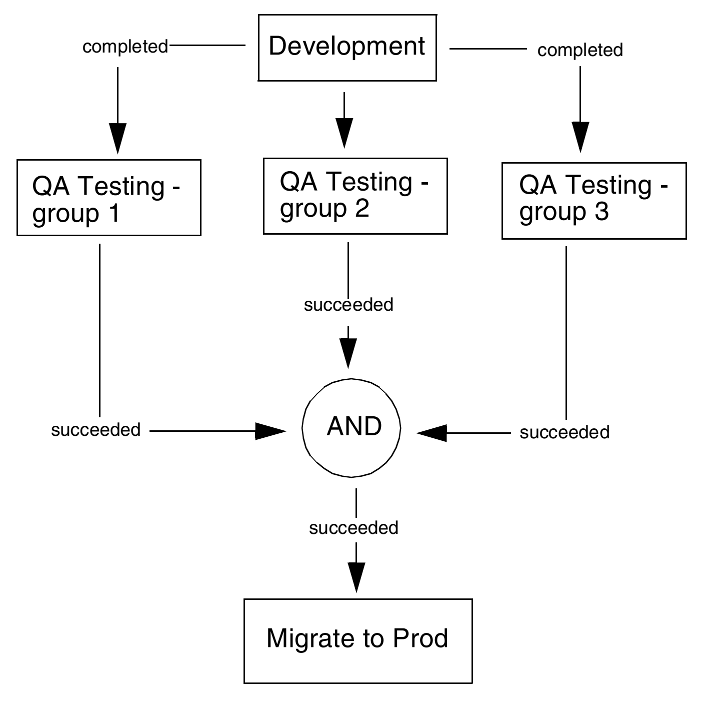
- *AND*
- *OR*
- *SYNC*
- *FIRST LINE*
- *LAST LINE*

AND

An AND condition is satisfied only if all steps leading to it reach the status they are supposed to attain.

Example

The AND step becomes successful only if ‘QA Testing - group 1,’ ‘QA Testing - group 2’ and ‘QA Testing - group 3’ are successful. At that point, the following step ‘Migrate to Prod’ becomes eligible.

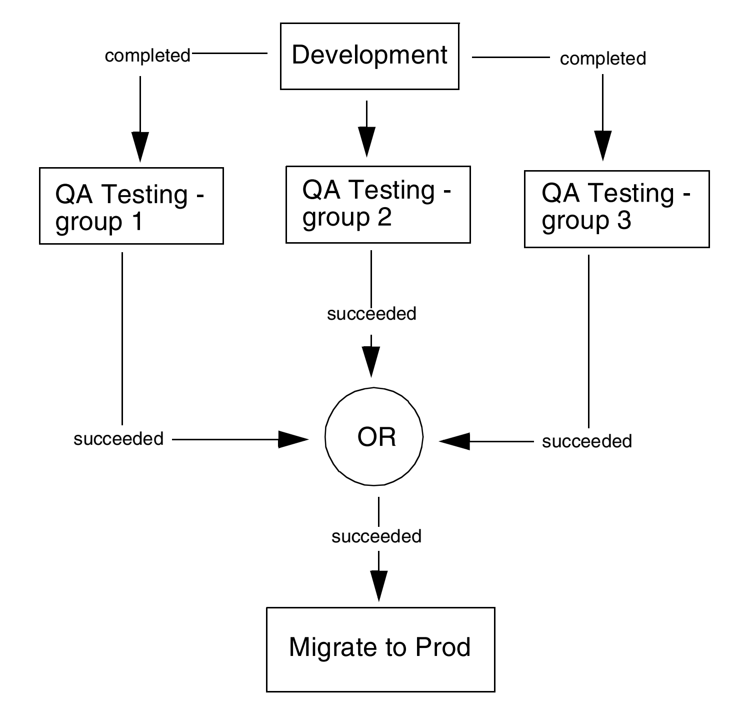


OR

An OR condition is successful when at least one of the steps leading to it reaches the status it is supposed to attain.

Example

The OR step becomes successful if any one of ‘QA Testing - group 1,’ ‘QA Testing - group 2’ and ‘QA Testing - group 3’ is successful. At that point, the following step ‘Migrate to Prod’ becomes eligible.



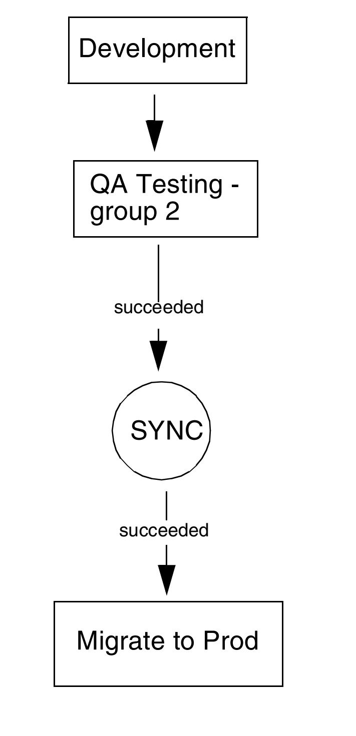
SYNC

A SYNC step is valid only for Mercury Change Management Packages. A SYNC step is successful only if all the Package Lines of that Package reach the status expected for the Workflow Step right before the SYNC step.

Example

Consider the business process outlined in the following flow chart. According to the flow chart, when ‘QA Testing’ is successful for all Package Lines, SYNC becomes successful and the next step, ‘Migrate to Prod’ becomes eligible.

This business process could be part of a software development life cycle. Consider a case where three Java files are being processed on three respective Package Lines in a single Package. By including a SYNC step, even if the first two Java files pass ‘QA Testing,’ they must wait for the third Java file to succeed ‘QA Testing’ before ‘Migrate to Prod’ becomes eligible for any of these Package Lines.



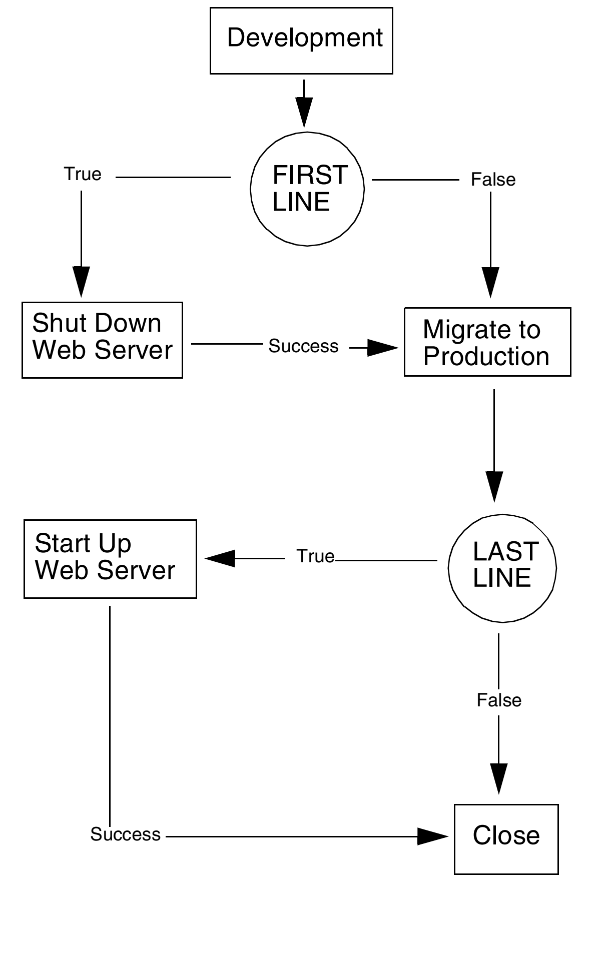
FIRST LINE

A FIRST LINE step is valid only for Mercury Change Management Packages. Only the first line to reach the Condition step takes the ‘True’ transition. All successive lines take the ‘False’ transition.

Example

Consider the business process outlined by the following flow chart. This business process could be part of a Website maintenance life cycle. As part of this life cycle, three HTML files are being processed on three respective Package Lines in a single Package. The Website updates are large enough to warrant shutting down the Web server while migrating the changes.

By including a FIRST LINE step, only the first line causes the server to shut down. The server remains down while the rest of the changes are migrated to production. By including a LAST LINE step, the server remains down until the last active line reaches the condition step. The last active line takes the True transition and the Web server starts up and the maintenance is complete.



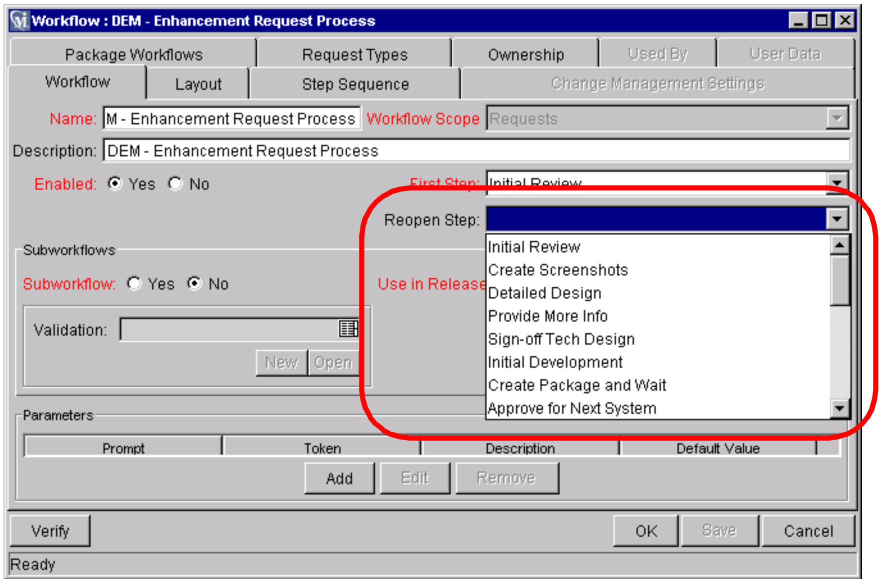
LAST LINE

A LAST LINE step is valid only for Mercury Change Management. Only the last active line to reach the Condition step takes the ‘True’ transition. All previous lines take the ‘False’ transition. See the example of a LAST LINE step shown in the previous flow chart.

Setting the Reopen Step for Request Workflows

Closed Requests can be re-opened by users with the proper access grants. A re-opened Request begins at the pre-defined Reopen Step in its Workflow, and begins processing normally.

The Reopen Step is defined from the Workflow window.



Workflow : DEM - Enhancement Request Process

Name: M - Enhancement Request Process

Workflow Scope: Requests

Description: DEM - Enhancement Request Process

Enabled: ● Yes ○ No

First Step: Initial Review

Reopen Step:

Subworkflow: ○ Yes ● No

Validation:

Ready

Figure 10-1 Workflow Window Reopen Step Drop Down

To specify the Reopen Step for the Workflow, select the desired step from the Reopen Step drop down list.

Modifying Workflows in Use

Workflows can be modified while they are going through their Workflow steps in a Package Line or Request that has been initiated. These modifications include adding new Workflow Steps, as well as changing the transitions, security assignments and notifications from within the Workflow.

It is possible to make changes to Workflows currently in use with the same procedures and windows that you used to define the Workflows. All of these procedures are performed in the Workflow Workbench.

When modifying Workflows that are being used, rules exist for which entities can be added, changed, deleted or renamed. These rules are described in *Table 10-8*.

Table 10-8. Rules for Modifying Production Workflows

Entity	Procedure
Transitions Security Notifications Workflow Steps Workflow Parameters	All of these entities can be modified or added to a Workflow in use.
Transitions Security Notifications Workflow Parameters	All of these entities can be deleted from a Workflow in use.
Workflow Steps	This entity cannot be deleted from a Workflow in use, but can be renamed. Transitions coming into or going out of a Workflow Step can be deleted, effectively removing it from the Workflow.

Note

When a Workflow that is in use is modified and saved, the changes take effect immediately. Any changes made to Workflow Steps are applied to all open Package Lines, Requests, and Distributions.

Changes to a Workflow can have undesirable effects on Requests or Packages currently in progress and are using that Workflow.

Note

The information included here also applies when migrating Workflows between installations (instances) of Mercury IT Governance products.

When modifying a Workflow that is in use, this can disrupt the normal flow in and out of the Workflow and prevent it from reaching completion. For example, removing a transition from a Workflow Step may result in the Requests or Package Lines being stuck in that Step. While no one solution covers all situations, the following sections describe possible solutions for common problems when modifying Workflows:

- *Copying and Testing the Workflow*
- *Moving Requests Out of a Step*
- *Disabling a Workflow Step*
- *Setting Up Execution Steps*
- *Verifying Workflow Logic*

Copying and Testing the Workflow

To modify a Workflow that is being used, make a copy of the original Workflow in a Development environment. Then modify the copied version of the Workflow. Test the copied version of the Workflow to make sure it works correctly.

After verifying that the modified Workflow functions as it is supposed to, make the same changes to the original Workflow and move it through the same cycle of Development -> Test -> Production environments.

Moving Requests Out of a Step

If the Requests are stuck in a step after a transition has been removed from a Workflow in use, add the deleted transition back to the Workflow. After the Requests have flowed out of the step, delete the transition again.

To determine when the Requests have flowed out of the step, run the Workflow Detail Report. This report indicates if the step to delete is eligible for user action or has been completed.

Note

To determine if any Package Lines are Eligible for user action in a Workflow, run the Packages Pending Report.

Disabling a Workflow Step

As mentioned in *Table 10-8*, a step can not be deleted from a Workflow when it is in use. It can only be disabled. However, you may want to change the process. Any changes to the process must be reflected in the Workflow. This may require disabling existing steps and adding new steps.

To disable a and add a new step:

1. Remove transitions to the existing Workflow step you no longer want to use.
2. Add a new Workflow step to the Workflow.
3. Redirect the transitions to the new Workflow step.

Redirecting the Workflow

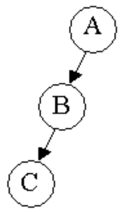
When disabling a Workflow step that is currently ‘Eligible’ for user action, the Requests or Package Lines in that step will become stuck. Since the step is now disabled, the user cannot take action on it and will not be able to progress any further through the Workflow.

To determine which steps are currently Eligible, remove the incoming transition to the step that will be deleted and run the Packages Pending Report in Mercury Change Management or the Workflow Detail Report in Mercury Demand Management. The reports will indicate if the step to be deleted is ‘Eligible’ for action by Package Lines or Requests.

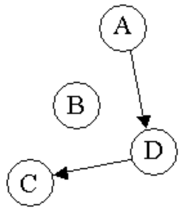
The outgoing transition to be deleted is still intact, so the eligible Package Lines and Requests will eventually be acted upon and flow out of the Workflow step.

Add a new Workflow step to the Workflow and redirect the transitions to that new Workflow step so that the movement of Package Lines and Requests avoids the disabled step and is not interrupted.

For example, consider a Workflow where you wanted to disable Workflow step B in the sequence shown below.



After removing the incoming and outgoing transitions to B, add a new Workflow step D which would connect steps A and C and let the Workflow continue to process Requests or Package Lines. See the sequence shown below.



Run the appropriate report(s) again to be sure there are no entities Eligible for action by the user in the step that was disabled.

Setting Up Execution Steps

When setting up Execution steps in a Workflow Step, be sure to include Workflow Events for both **Success** and **Failure**. If a Workflow Step has failed and users cannot select **Failure** as one of the Workflow Events, the Workflow will not be able to proceed.

Modifying Workflow Step Security – Performance Consideration

Updating an existing Workflow's step security with a specific configuration can impact system performance. When adding dynamic security to a step (for example, based on a Standard or User Defined Token) in the Workflow Step window on the **Layout** tab, product database tables are updated to handle this new configuration. Due to the scope of these database changes, the Database Statistics need to be re-run on your database. Instructions for this operation are included in the *System Administration Guide*. Contact the System Administrator for help with this procedure.

Note

This also applies when migrating a Workflow with these types of changes into an instance of the Mercury IT Governance Center.

Verifying Workflow Logic

A Workflow can also become stuck if the logic behind it is faulty. Plan the steps of the Workflow process carefully before actually defining it. After configuring the Workflow, click the **Verify** button in the Workflow window to ensure that the logic of the Workflow is correct. Any mistakes in the Workflow's logic will be highlighted.

Using Workflow Parameters

Use Workflow parameters to store the result of a workflow step. This value can then be used later to define a transition.

This section covers the following topics:

- *Creating a Workflow Parameter*
- *Example: Building a Loop Counter Using Workflow Parameters*

Note

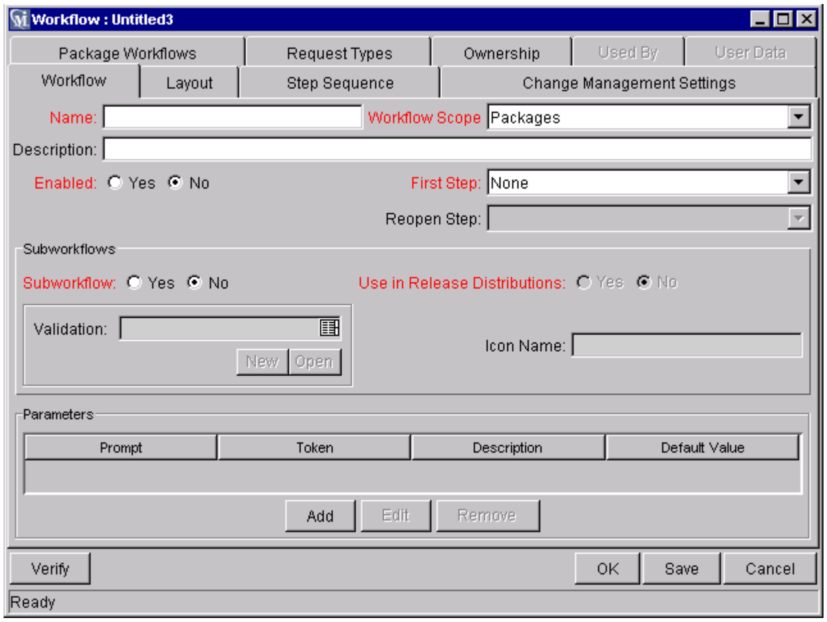
Workflow Parameters:

- Can be referenced using the WFI.P Token prefix.
- Can be used in PL/SQL and SQL Workflow Step executions.

Creating a Workflow Parameter

To create a Workflow parameter:

1. From the Workflow Workbench, query and open the Workflow to be modified.



Name:

Workflow Scope: Packages

Enabled: ○ Yes ● No

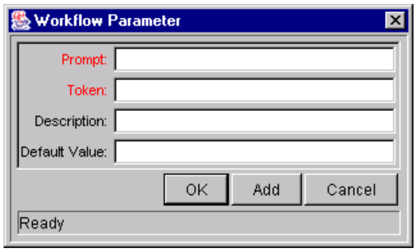
First Step: None

Subworkflow: ○ Yes ● No

Use in Release Distributions: ○ Yes ● No

2. In the **Workflow** tab, click **Add**.

   The Workflow Parameter window opens.



3. Enter information in the required fields.

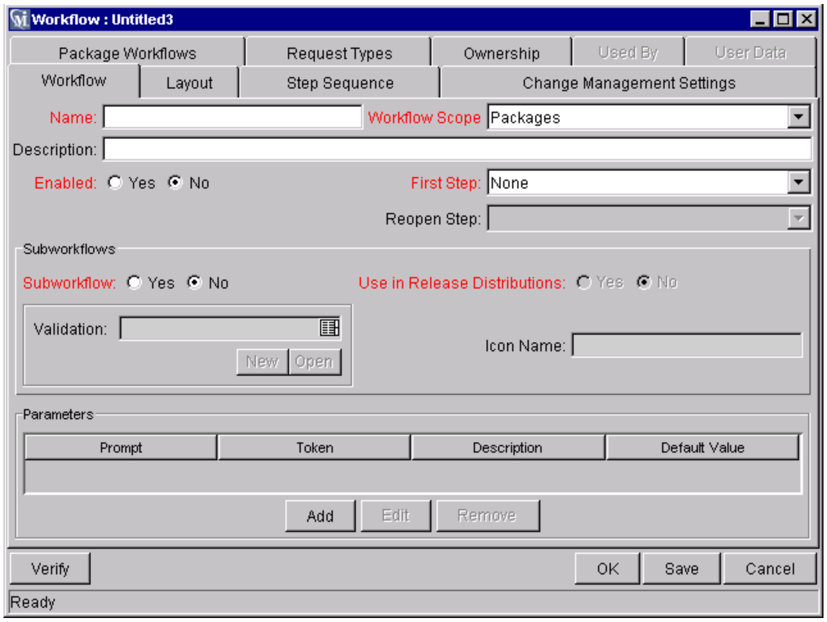
4. Click **OK.**

Example: Building a Loop Counter Using Workflow Parameters

Workflow parameters can be used to generate a counter for the number of times a Workflow Step is in a certain state.

To build a loop counter using Workflow parameters:

1. From the Workflow Workbench, open the Workflow to which the loop counter is to be added.



Name:

Workflow Scope: Packages

Description:

Enabled: ○ Yes ● No

First Step: None

Reopen Step:

Subworkflow: ○ Yes ● No

Use in Release Distributions: ○ Yes ● No

Validation:

Icon Name:

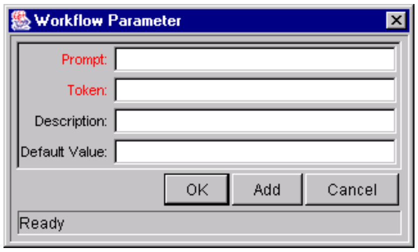
Parameters

Prompt	Token	Description	Default Value

Ready

2. In the **Workflow** tab, click **Add**.

   The Workflow Parameter window opens.



Prompt:

Token:

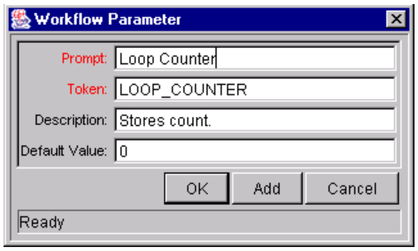
Description:

Default Value:

Ready

3. Generate the Workflow parameter by entering information in the fields of the Workflow Parameter window.

   In this example, the parameter is named **LOOP_COUNTER**.



Prompt: Loop Counter

Token: LOOP_COUNTER

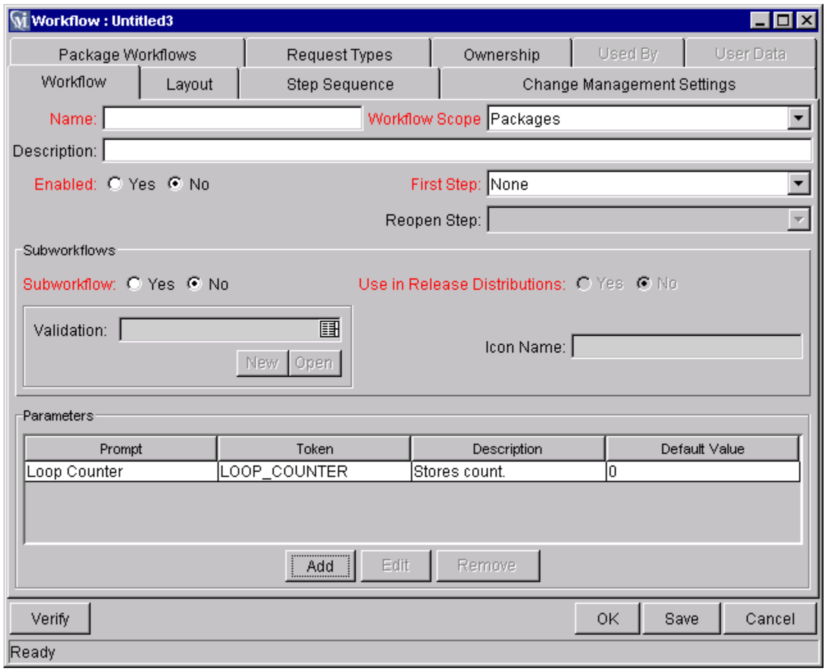
Description: Stores count.

Default Value: 0

Ready

4. Click **OK**.

The **Loop Count** parameter is added to the Workflow window.



Name:
Workflow Scope: Packages
Description:
Enabled: ○ Yes ● No
First Step: None
Reopen Step:
Subworkflow: ○ Yes ● No
Use in Release Distributions: ○ Yes ● No

Parameters

Prompt	Token	Description	Default Value
Loop Counter	LOOP_COUNTER	Stores count.	0

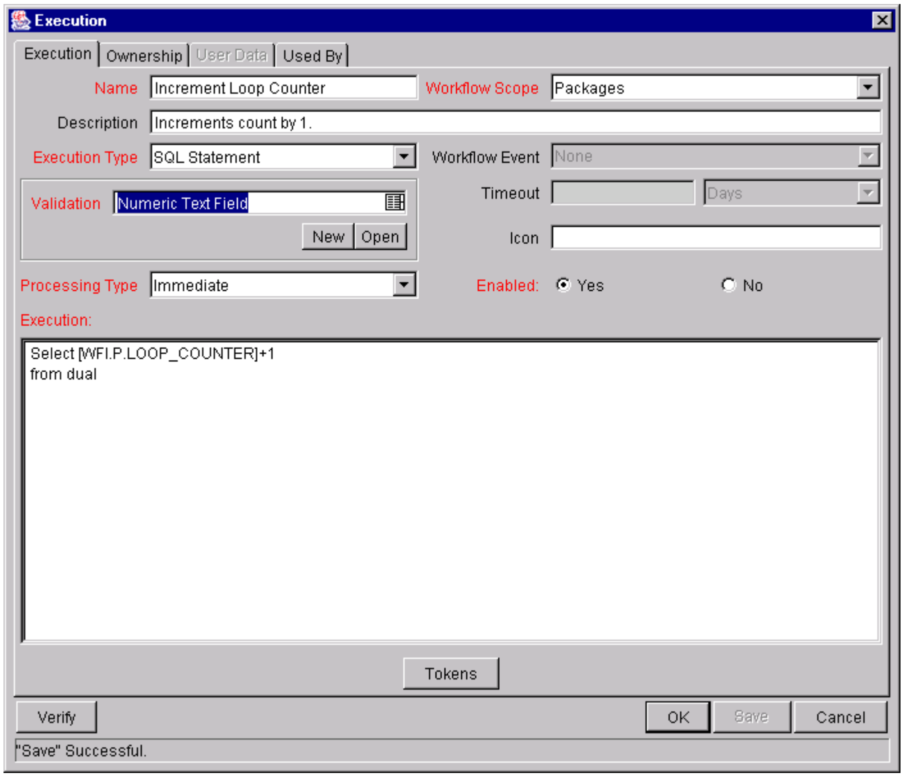
Ready

5. Generate a new Immediate SQL Execution Step.

There are two key concepts to note about the new step definition.

- The result of the SQL Execution step returns the result **LOOP_COUNTER + 1**. This return value is linked back into the parameter when the Workflow Step is generated on a Workflow.

- A Validation for a **Numeric** Text Field is used. This allows <=, <, >=, and > comparisons to be used in transitions off this step.



Name: Increment Loop Counter

Workflow Scope: Packages

Description: Increments count by 1.

Execution Type: SQL Statement

Workflow Event: None

Validation: Numeric Text Field

Timeout: Days

Icon:

Processing Type: Immediate

Enabled: ● Yes ○ No

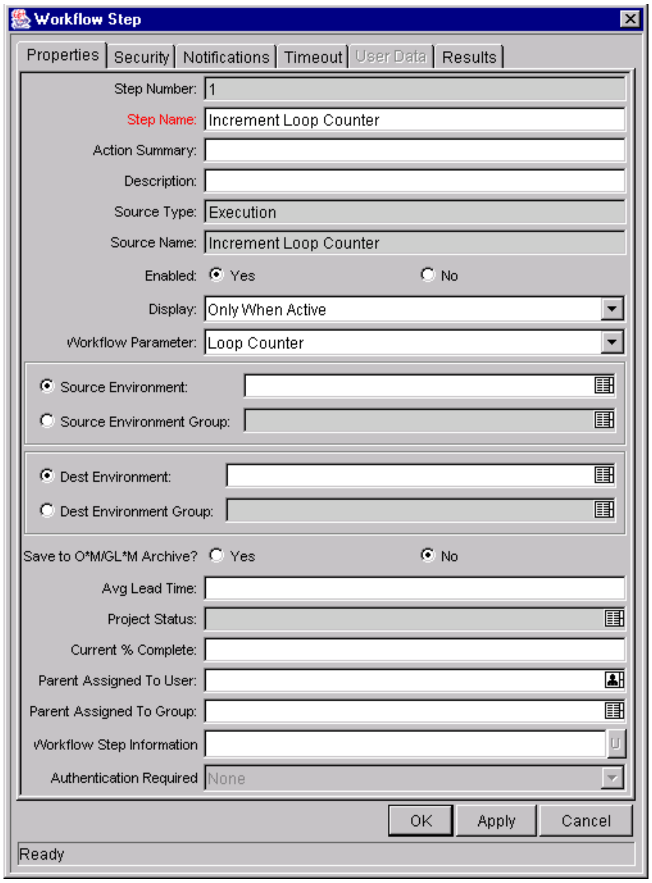
Execution:

```
Select [WFI.P.LOOP_COUNTER]+1
from dual
```

"Save" Successful.

6. Add the Workflow Step to a Workflow and choose the new Workflow Parameter **LOOP_COUNTER**.

   By choosing **Loop Count**, the Workflow Engine is told to assign the result of `select loop counter val + 1 from dual` back into the loop counter parameter.



Step Number: 1

Step Name: Increment Loop Counter

Action Summary:

Description:

Source Type: Execution

Source Name: Increment Loop Counter

Enabled: ● Yes ○ No

Display: Only When Active

Workflow Parameter: Loop Counter

● Source Environment:

○ Source Environment Group:

● Dest Environment:

○ Dest Environment Group:

Save to O*M/GL*M Archive? ○ Yes ● No

Avg Lead Time:

Project Status:

Current % Complete:

Parent Assigned To User:

Parent Assigned To Group:

Workflow Step Information

Authentication Required None

Ready

It is now possible to add transitions to and from the new loop counter step.

Example

The loop counter can be incremented each time an execution fails. If the execution fails three times, a notification can be sent to the user. If the execution fails five times, management can be notified.

Appendix B Validations

This appendix provides an overview for how to use Validations in your Mercury IT Governance system. Validations determine the acceptable input values for user-defined fields (such as Object Type or Request Type fields). Validations also determine the possible results that a Workflow step can return.

This appendix covers the following topics:

What are Validations

Validations are used for two primary functions:

- Fields:
  Validations determine the field's component type (text field, drop down list, etc.) and the fields possible values. Fields can be created for a number of product entities: Object Types, Request Types, Request Header Types, and User Data.

- Workflow step results:
  Validations determine the possible results exiting a Workflow step. For example, the validation WF - Standard Execution Results contains the possible execution step results of **Succeeded** or **Failed**.

Pre-seeded (system) Validations are included with every product installation or upgrade. When configuring your system, you can select to use these system Validations. If no Validation exists that meets your specific requirements, you can create a new Validation using the Validation Workbench. See *"Creating a Validation"* on page 287 for details.

Validation Component Types - Overview

The following table summarizes the available types of field components. Note that only certain component types can be used in a Workflow step source's Validation.

Table B-1. Component Types

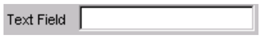
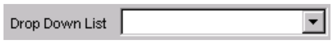
Component Type	Use In Workflow?	Example**	Description
Text Field	Yes		Text entry fields displayed on a single line. Text fields can be configured to display the data according to a certain format. For example, you can configure a text field to accept and format a hyphenated nine-digit social security number or a ten digit telephone number.
Drop down list	Yes		Field showing a column of choices.

Table B-1. Component Types

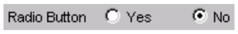
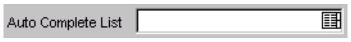
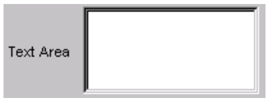
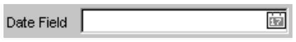
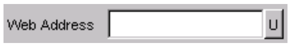
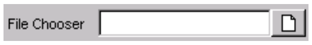
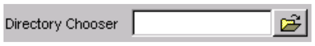
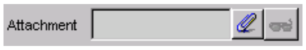
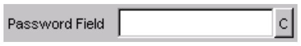
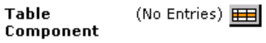
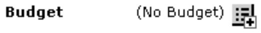
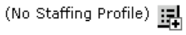
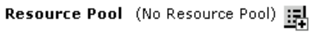
Component Type	Use In Workflow?	Example**	Description
Radio Button	No		Field providing a Yes/No input.
Auto-complete list	Yes		Field showing list of choices with multiple columns.
Text Area	No		Text entry field that can span multiple lines.
Date Field	No		Supports a variety of date and time formats: long, medium, and short.
Web Address (URL)	No		Text entry field for entering a URL. Pressing the U button opens a browser window to the specified web address.
File Chooser	No		Used only in Object Types. Requires that two fields be defined with the following Tokens: P_FILE_LOCATION and P_SUB_PATH. See *"Using Directory and File Choosers"* on page 334 for configuration details.
Directory Chooser	No		Used only in Object Types. Requires that a parameter field be defined with the Token P_FILE_LOCATION.
Attachment	No		Field for indicating file attachments. Comes with buttons for locating files for previewing contents of the selected file.
Password field	No		Field for capturing passwords.

Table B-1. Component Types

Component Type	Use In Workflow?	Example**	Description
Table Component	No	 Table Component (No Entries)	Used to enter multiple records into a single component. The table component can be configured to include multiple columns of varied data types. Additionally, this component supports rules for populating elements within the table and provides functionality for capturing column totals. See *"Configuring the Table Component"* on page 339 for details. Fields of this component can only be added to Request Types, Request Header Types and Request User Data.
Budget	No	 Budget (No Budget)	Field that can be added to the Request Type to enable access to view, edit or create Budgets associated with a Request or Project. Fields of this component can only be added to a Request Type.
Staffing Profile	No	 (No Staffing Profile)	Field that can be added to the Request Type to enable access to view, edit or create Staffing Profiles associated with a Request or Project. Fields of this component can only be added to a Request Type.
Resource Pool	No	 Resource Pool (No Resource Pool)	Field that can be added to the Request Type to enable access to view, edit or create Resource Pools associated with a Request or Project. Fields of this component can only be added to a Request Type.

Creating a Validation

Generating certain Workflow steps may require specific validations to ensure that business procedures are being followed. It is necessary to have both the Validation Editor and the Validation Values Editor access grants to add a new validation. See *Security Model Guide and Reference* for a discussion of security groups and access grants.

To define a new Validation:

1. Click **New Validation** on the Validation Workbench or select **File > New > Validation** from the menu.

   The Validation window opens.

2. Enter the name of the new Validation in the Name field.

3. Enter a description of the new Validation in the Description field.

4. Select whether the Validation is enabled or not in the Enabled check box.

5. In the Use in Workflow checkbox, specify whether or not this Validation can be used in a Workflow step source.

   You can only use Text Field, Drop Down List and Auto-Complete component types within Workflow step sources.

6. Select the desired type of Validation from the Component Type drop down list.

   Choices are **Text Field**, **Drop Down List**, **Radio Buttons (Y/N)**, **Auto Complete List**, **Text Area**, **Date Field**, **Web Address (URL)**, **File Chooser**, **Directory Chooser**, **Password Field**, **Attachment**, **Table Component**, Budget, **Staffing Profile**, and **Resource Pool**. Selecting a value from this field will dynamically update the Validation window to display fields used to configure that type of Validation.

7. Enter any additional information required for the component type selected.

8. Click **Ownership** to select which users will be able to edit, copy and delete this validation.

9. To save changes to the Validation without closing the window, click **Save**. To save changes and close the window, click **OK**

User Data on the Validation Value

You can enable the **User Data** tab to capture more information related to an individual Validation value within a specific Validation. For example, you can create a Description user data field that is associated with the Departments Validation. When you add new values to the validation, you can click on the **User Data** tab and enter a description for that value.

The **User Data** tab can only be used when creating a drop down or an auto-complete validated by a list.

To enable the User Data tab in the Edit Validation Value window:

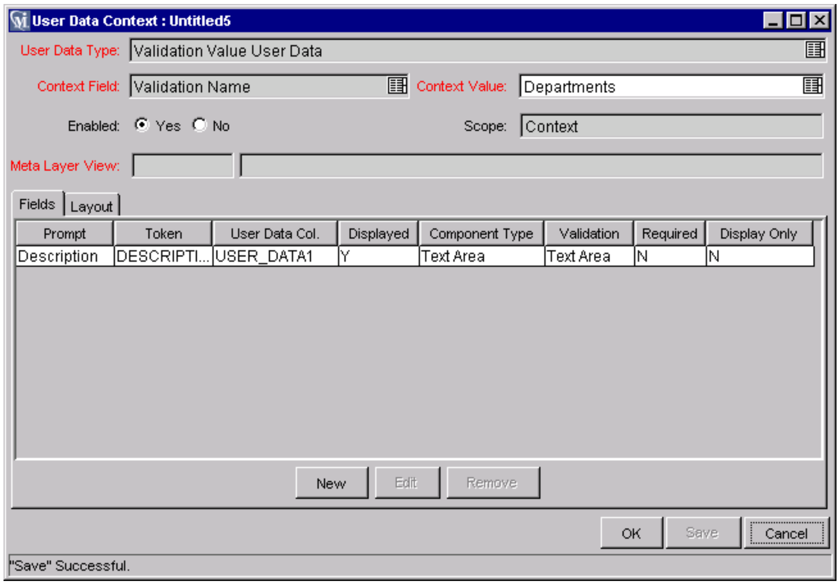
1. Create the Validation and note its name.

2. Open the User Data workbench.

3. Click **New User Data Context**.

4. Select **Validation Value User Data** from the User Data Type field.

5. Click **New** to create a User Data field.



User Data Context : Untitled5

User Data Type: Validation Value User Data

Context Field: Validation Name

Context Value: Departments

Enabled: ● Yes ○ No

Scope: Context

Meta Layer View:

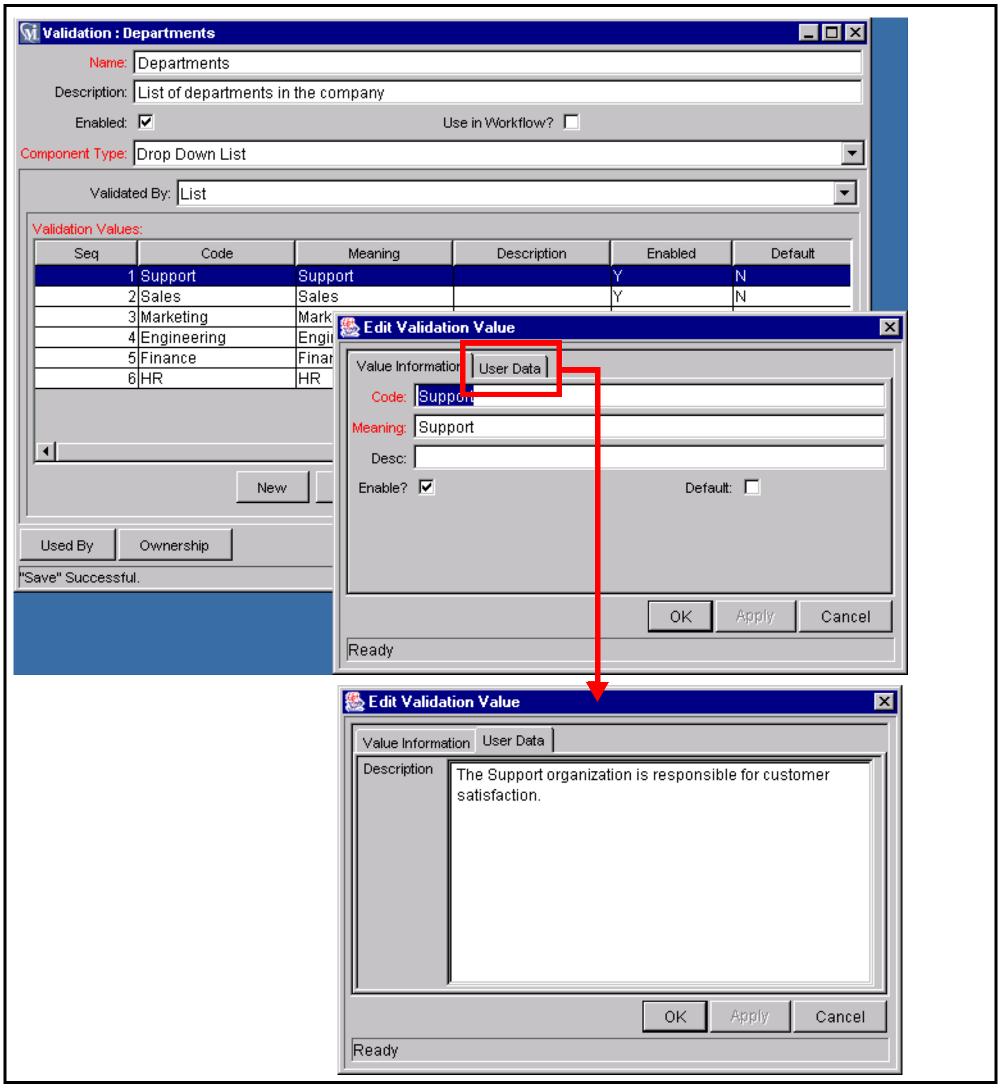
Prompt	Token	User Data Col.	Displayed	Component Type	Validation	Required	Display Only
Description	DESCRIPTI...	USER_DATA1	Y	Text Area	Text Area	N	N

"Save" Successful.

6. Save the settings in the User Data window.

7. On the Validation window, add or edit a Validation value.

The **User Data** tab is now enabled. You can select the tab and enter information in the newly defined user data field.



Name: Departments

Description: List of departments in the company

Enabled: ☑ Use in Workflow? ☐

Component Type: Drop Down List

Validated By: List

Validation Values:

Seq	Code	Meaning	Description	Enabled	Default
1	Support	Support		Y	N
2	Sales	Sales		Y	N
3	Marketing	Mark			
4	Engineering	Engi			
5	Finance	Finar			
6	HR	HR			

"Save" Successful.

Code: Support

Meaning: Support

Desc:

Enable? ☑ Default: ☐

Description: The Support organization is responsible for customer satisfaction.

Editing Validations

You can open and edit Validations using the Workbench. You should exercise caution when editing Validations that are currently used by fields or Workflow step sources. Both field and Workflow step validations can be tied to Workflow logic. Changing the Validation values can invalidate a process.

For example, ACME changes the Priority field Validation to include a new value **Very Easy**. ACME uses a deployment system Workflow that has an Evaluate Priority step that routes the Package based on the value in the Priority field (using a Token execution type). ACME, however, did not update the Workflow to enable a transition out of the step for the case when Priority = **Very Easy**. When a **Very Easy** Package enters the Evaluate Priority step, it will get stuck.

The following restrictions apply to editing Validations:

- User must have the following Access Grants:
  - Edit Validations
  - Edit Validation Values
- User must be a member of the Ownership Group for the Validation
- You can not change which Validation is associated with a Workflow step source after a Package has traversed that step. You can, however, still edit the values within that Validation.

Creating a URL to Open the Validation Window

You can create a URL that opens a specific Validation in the Workbench. This can provide a quick link to the configuration screen for a Validation that is expected to change frequently. This URL can be included on your internal or external Web pages or a list of browser Favorites to provide convenient access to the Validation’s definition.

Use the following URL format to access a specific Validation window:

```
http://host:port/kintana/servlet/SmartURL?screen=VAL&pkname=<Va
lidationName>
```

Note

The following URL opens the Validation window for the Validation named “Development Priorities.”

```
http://host:port/kintana/servlet/SmartURL?screen=VAL&pkname=
   Development+Priorities
```

Deleting Validations

Validations can be deleted from the Workbench. To delete a Validation, you must be a member of the Validation's Ownership Group and have the Edit Validations access grant.

A Validation can not be deleted when:

- It is a system Validation (a Validation that is delivered with the product as seed-data)
- It is being used by a Workflow step source. Validations referenced by Workflow step sources can only be disabled. A disabled Validation continues to function in existing Workflow steps, but can not be used when defining a new step source.
- It is being used by a field in a product entity (Object Type, Request Type, User Data, Report Type, or Project Template field). Validations referenced by entity fields can only be disabled. A disabled Validation continues to function in existing fields, but can not be used when defining a new field.

Tip

Although you may not be able to delete a custom Validation in all cases, you can disable it. This will allow the Validation to be used in any active Workflows or product entities, but will keep it from being used in any new Workflow or entity definitions.

Static List Validations

You can create Validations that provide a static list of options to the user. For example, ACME, Inc. can create a Validation for their engineering teams. They create a Validation called Engineering Teams, consisting of the following values: **New Product Introduction**, **Product One**, and **Product Two**.

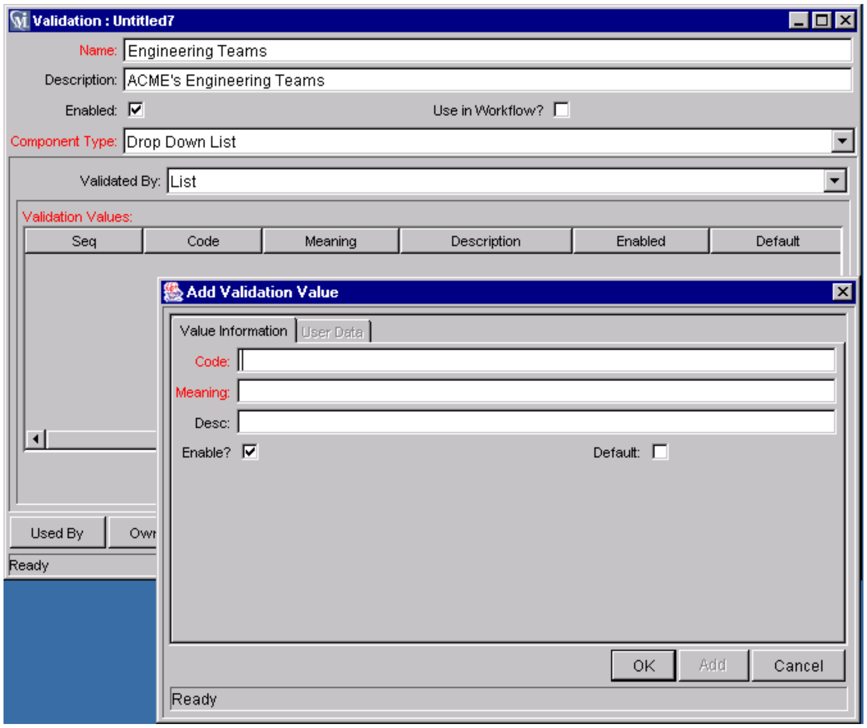
A static list validation can be a drop down or an auto-complete list component.

To add values to the Validation list:

1. In the Validation window, select **Drop Down List** or **Auto Complete List** from the Component Type field.
2. Select **List** from the Validated By field.

3. Click **New** and add a value.

   The Add Validation Window opens.



Validation : Untitled7

Name: Engineering Teams

Description: ACME's Engineering Teams

Enabled: ☑ Use in Workflow? ☐

Component Type: Drop Down List

Validated By: List

Validation Values:

Seq	Code	Meaning	Description	Enabled	Default

Add Validation Value

Code:

Meaning:

Desc:

Enable? ☑ Default: ☐

Ready

4. Enter information for the Validation value as described in the following table.

Field	Definition
Code	The underlying code for the Validation value. The code is the value stored in the database or passed to any internal functions, and is rarely displayed.
Meaning	The displayed meaning for the Validation value in the drop down list or auto-complete.
Default	The default value for the list. This value is initially displayed in drop down lists (this is not used for auto-complete lists). There can be only one default value per list.

5. Optionally set the Validation value as the default by checking the Default field.

The default option is only available for drop down lists.

6. Click **OK** to close the window and add the value to the Validation. Click **Add** to add the value and keep the Add Validation Value open.

Validation values can be re-ordered using the up and down arrow buttons. The sequence of the Validation values determines the order that the values are displayed in the list.

Tip

You can copy existing values defined in other Validations using the **Copy From** button. Click **Copy From** and query an existing list-validated Validation and choose any of the Validation values. Click **Add** or **OK** in the Copy From window and the selected value or values are added to the list.

Note

Be careful when creating Validations (drop down lists and auto-complete lists) that are validated by lists. Each time the set of values changes, you will be forced to update the Validation. Consider, instead, validating using a SQL query or PL/SQL function to obtain the values from a database table.

Dynamic List Validations

You can create Validations that provide a dynamic list to the user. This is often a better approach than defining static list validations. Each time a static list Validation needs to be updated, a manual update has to occur. Dynamic list Validations can often be constructed in such a way as to automatically pick up and display the altered values.

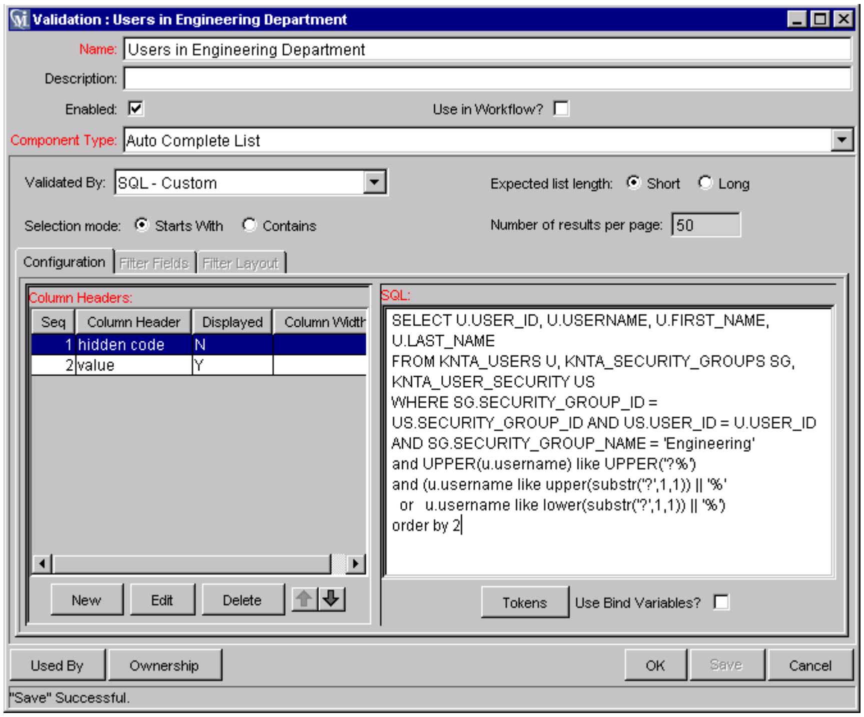
For example, ACME needs a field validation that will list all users who are on their Support Team. They could construct a Validation that is validated by a list of users, but any time the Support Team changed (members join or leave the department) the list would have to be manually updated. ACME decides instead to create a dynamic list validation. They create an auto-complete list validation that is validated by a SQL statement. The SQL statement returns all users who are a member of the **Support Team** Security Group. When the Security Group membership is altered, the Validation is automatically updated with the correct values.

A dynamic list validation can be created using a drop down or an auto-complete list component. The lists are dynamically generated using either:

- *SQL Validation*
- *Command Validation*

SQL Validation

You can use a SQL statement to generate the values in a Validation. SQL can be used as a validation method for drop down lists and auto-complete lists. To define a dynamic list of choices, set a drop down list or auto-complete list to Validated By - **SQL**. Then in the SQL area, enter the Select statement that queries the necessary database.



Name: Users in Engineering Department

Description:

Enabled: ☑ Use in Workflow? □

Component Type: Auto Complete List

Validated By: SQL - Custom

Expected list length: ● Short ○ Long

Selection mode: ● Starts With ○ Contains

Number of results per page: 50

Column Headers:

Seq	Column Header	Displayed	Column Width
1	hidden code	N	
2	value	Y	

SQL:

```
SELECT U.USER_ID, U.USERNAME, U.FIRST_NAME,
U.LAST_NAME
FROM KNTA_USERS U, KNTA_SECURITY_GROUPS SG,
KNTA_USER_SECURITY US
WHERE SG.SECURITY_GROUP_ID =
US.SECURITY_GROUP_ID AND US.USER_ID = U.USER_ID
AND SG.SECURITY_GROUP_NAME = 'Engineering'
and UPPER(u.username) like UPPER('?%')
and (u.username like upper(substr('?',1,1)) || '%'
  or   u.username like lower(substr('?',1,1)) || '%')
order by 2
```

Use Bind Variables? □

"Save" Successful.

If an auto-complete list is being used, you can define headers for the selected columns. These column headers are used in the window that opens when a value from an auto-complete list is selected. Click **New** under Column Headers. *Table 10-9* shows the fields that can be entered for a column header. If a column header is not defined for each column in a SQL query, a default name is used.

Table 10-9. Column Headers

Field	Definition
Column Header	The name of the column that is displayed in the auto-complete window.
Display	Determines whether or not the column is displayed. The first column is never displayed and the second column is always displayed.

Example

ACME, Inc. creates an auto-complete field that lists all users in the "Engineering" department. They choose to validate the list by SQL.

```
SELECT U.USER_ID, U.USERNAME, U.FIRST_NAME, U.LAST_NAME
FROM KNTA_USERS U, KNTA_SECURITY_GROUPS SG,
   KNTA_USER_SECURITY US
WHERE SG.SECURITY_GROUP_ID = US.SECURITY_GROUP_ID AND
   US.USER_ID = U.USER_ID
AND SG.SECURITY_GROUP_NAME = 'Engineering'
and UPPER(u.username) like UPPER('?%')
and (u.username like upper(substr('?',1,1)) || '%'
  or   u.username like lower(substr('?',1,1)) || '%')
order by 2
```

When a new user account is created and is added to the "Engineering" Security Group, that user will automatically be included in the auto-complete list.

Tip

A Validation may already exist that meets your process requirements. If it does, consider using that Validation in your process. Also consider copying and modifying Validations that are similar to the desired Validation. See *"System Validations"* on page 355 for a complete list of Validations that are delivered with the product.

SQL Validation Tips

The following guidelines are helpful when writing a SQL statement for a SQL-validated Validation:

- The SQL statement must query at least two columns. The first column is a hidden value which is never displayed, and is often stored in the database or passed to internal functions. The second column is the value that is displayed in the field. All other columns are for information purposes and are only displayed in the auto-complete window. Extra columns are not displayed for drop down lists.

- When something is typed into an auto-complete list field, the values in the auto-complete window that appear are constrained by what was first typed in the field. Generally, the constraint is case insensitive. This is accomplished by writing the SQL statement to query only values that match what was typed.

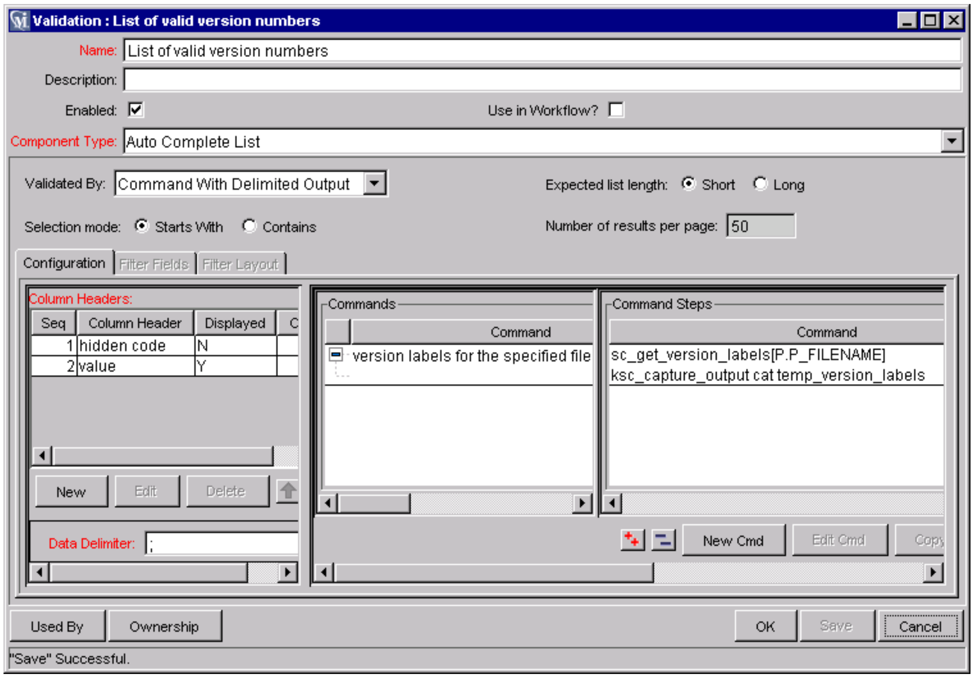
  Before the auto-complete window is displayed, all question marks in the SQL statement are replaced by the text that the user typed. In general, if the following conditions are added to the WHERE clause in a SQL statement, the values in the auto-complete window are constrained by what the user typed.

  ```
  where UPPER(<displayed_column>) like UPPER('?%')
  and (<displayed_column> like upper(substr('?',1,1)) || '%'
  or   <displayed_column> like lower(substr('?',1,1)) || '%')
  ```

  Any column aliases included directly in the SQL statement are not used. The names of the columns, as displayed in auto-complete lists, are determined from the Column Headers. Drop down lists do not have column headers

Command Validation

An auto-complete list can contain command line executions that return and display a list of values. To define a dynamic list of choices, set an auto-complete list to Validated By - **Command with Delimited Output** or **Command with Fixed Width Output**. Then enter commands the Commands area. See *“Configuring the Auto-Complete Values”* on page 311 for detailed instructions.



Validation : List of valid version numbers

Name: List of valid version numbers

Description:

Enabled: ☑ Use in Workflow? □

Component Type: Auto Complete List

Validated By: Command With Delimited Output

Expected list length: ● Short ○ Long

Selection mode: ● Starts With ○ Contains

Number of results per page: 50

Column Headers:

Seq	Column Header	Displayed	C
1	hidden code	N	
2	value	Y	

Data Delimiter: ;

Commands

Command
version labels for the specified file

Command Steps

Command
sc_get_version_labels[P.P_FILENAME] ksc_capture_output cat temp_version_labels

"Save" Successful.

Figure B-1 Auto Complete Using Command Validation

Configuring Auto-Complete Validations

Auto-complete fields are used throughout the Mercury IT Governance Center to provide users with an efficient way to select field values from a set of valid choices. Configuring auto-complete fields consists of two activities:

- Specifying general auto-complete behavior
- Configuring the validation values

Configuring General Auto-complete Behavior

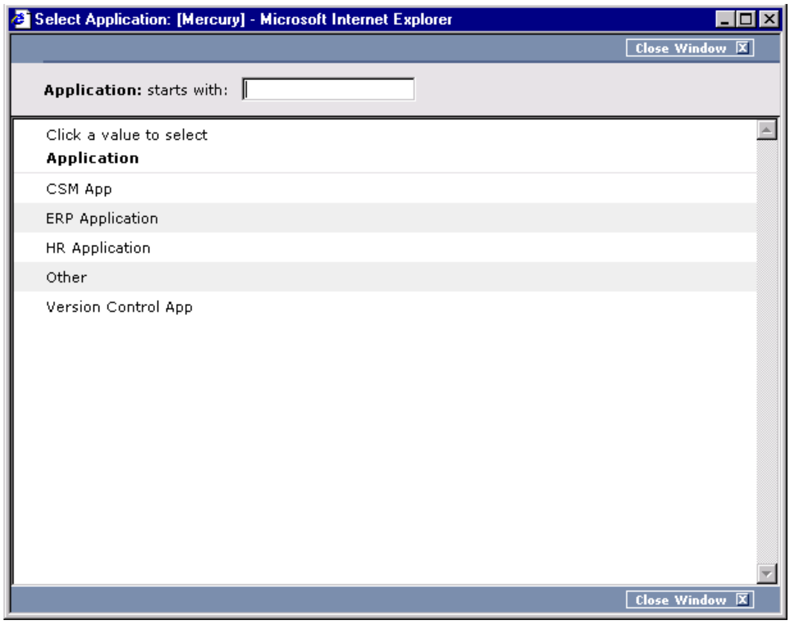
Auto-complete fields can be used for validations with a small or large number of choices. The auto-complete can be configured to behave differently depending on the expected number of values. For example, if you expect a large number of entries, the auto-complete window will include an interface that allows you to page through your results. Additionally, you can configure how the “auto-complete” feature of the field behaves. For example, you can configure the auto-complete field to automatically complete entries that either start with or contain a text string.

This section covers the following topics related to configuring auto-complete field behavior:

- *Configuring Short List Auto-Complete Fields*
- *Configuring Long List Auto-Complete Fields*
- *Configuring the Automatic Value Matching and the Interactive Select Page*
- *Adding Search Fields to the Auto-Complete Window*

Configuring Short List Auto-Complete Fields

Auto-complete fields configured to display a short list of entries, displaying all of the values on a single page. *Figure B-2* shows the Select window for a short list auto-complete field.



Application: starts with:

Click a value to select

Application

CSM App

ERP Application

HR Application

Other

Version Control App

Figure B-2 Short List Auto-Complete

Tip

Auto-completes configured as short lists will load all values when the window is opened. This can lead to a slower load time and an unfavorable user experience. For fields with many possible values, consider formatting the auto-complete using the long list format.

To configure a short list auto-complete field:

1. Create a new Validation or open an existing Validation.

   The Validation window opens.

2. From the Component Type field, select **Auto Complete List**.

3. In the Expected list length field, select **Short**.

4. Click **Save**.

Configuring Long List Auto-Complete Fields

Auto-complete fields configured to display a long list of entries, dividing the results between multiple pages. By default, 50 results are shown per page. End users can page through the results or further limit the results by specifying text in one of the available filter fields at the top of the page. *Figure B-3* shows the Select window for a long list auto-complete field.

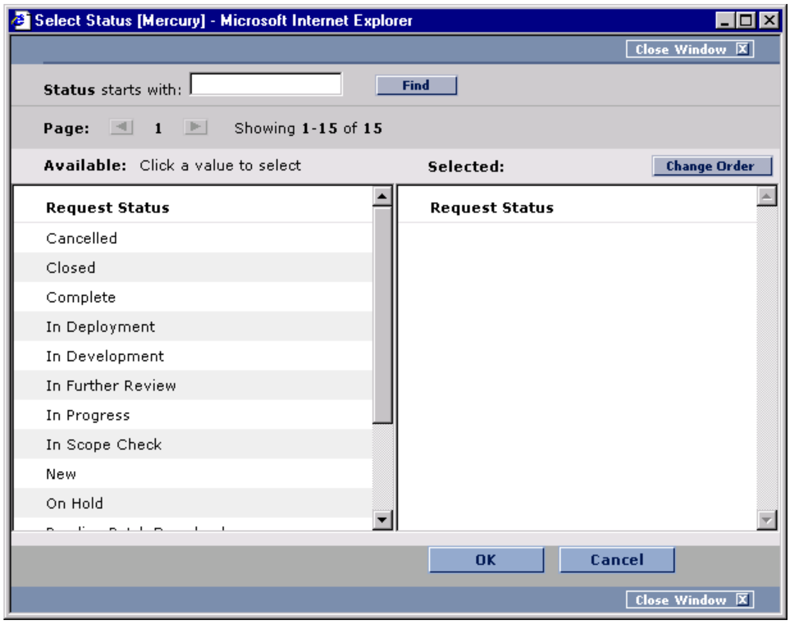


Figure B-3 Long List Auto-Complete

Tip

Auto-completes configured as long lists only load a limited set of values when the window is opened. For extremely long lists or lists that are at risk of loading slowly (for example the values are obtained from an alternate database), consider using the long list format.

To configure a long list auto-complete field:

1. Create a new Validation or open an existing Validation.

   The Validation window opens.

2. In the Expected list length field, select **Long**.

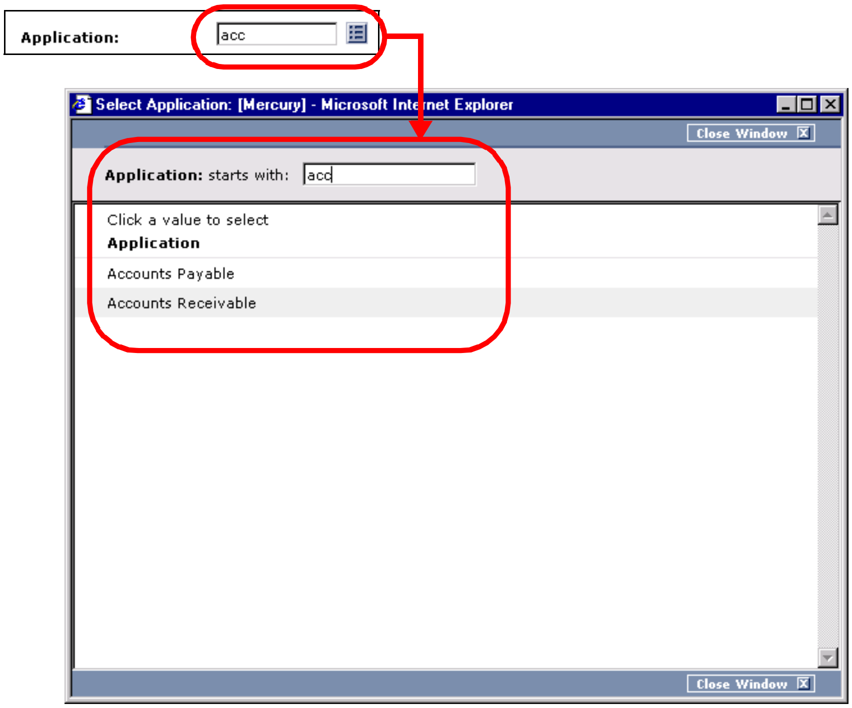
3. Click **Save**.

Note

All auto-completes that are validated by **SQL - User** are required to use the long list auto-complete format. This selection is automatically defaulted when the user selects **SQL - User** from the Validated By field on the Validation window.

Configuring the Automatic Value Matching and the Interactive Select Page

This section provides instructions for configuring auto-complete fields to filter a list of possible values based on a matching character string. It also provides instructions for configuring the automatic value-limiting that occurs on the auto-complete's Select page. *Figure B-4* shows an auto-complete field that has opened to display matching values.



Application: acc

Application: starts with: acc

Click a value to select

Application

Accounts Payable

Accounts Receivable

Figure B-4 Auto-complete field and matching values in the Select page

This section discusses the following topics:

- *Functional Overview: Matching for "Starts with" or "Contains"*
- *Configuration Instructions*
- *Configuration Tips*

Functional Overview: Matching for "Starts with" or "Contains"

Auto-complete field behavior can be divided into the following areas:

- Field behavior—A user types a character in the field and presses the Tab key. If an exact match is not available, the Select page opens.

- The Select page behavior—For lists that are configured appropriately, when a user types a character or characters into the field at the top of the page, the results are automatically limited to display only matching entries.

For both the field and Select page behaviors, automatic value matching can be based on either “starts with” character matching or “contains” character matching. The following table summarizes this behavior:

Table 10-10. Automatic character matching field behavior

Character matching mode	Description of Behavior
Starts with	Type characters and press tab. The selection window will open and list entries that begin with the specified characters.
Contains	Type characters and press tab. The selection window will open and list entries that contain the specified character string. This is the same behavior as a wild card search, which uses the % character at the beginning of the search text.

Table 10-11. Automatic character matching Select page behavior

Character matching selection mode	Description of Behavior
Starts with	Type characters and the list will automatically be filtered for entries that begin with the specified characters.
Contains	Type characters and the list will automatically be filtered for entries that contain the character string.

Configuration Instructions

The field and the Select page behavior are configured distinctly in the Validation window for the specific auto-complete list. This section provides instructions for configuring the “starts with” and “contains” functionality in the field and Select page, as described in *“Functional Overview: Matching for “Starts with” or “Contains””* on page 301.

To configure "starts with" matching from the auto-complete window to the selection window, add the following to the SQL WHERE clause:

```
UPPER(value) like UPPER('?%') and (value like
upper(substr('?',1,1)) || '%' or value like
lower(substr('?',1,1)) || '%')
```

To configure "contains" matching from the auto-complete window to the selection window, add the following to the SQL WHERE clause:

```
UPPER(value) like UPPER('%?%') and (value like '%' ||
upper(substr('?',1,1)) || '%' or value like '%' ||
lower(substr('?',1,1)) || '%')
-
```

To configure "starts with" matching within the interactive selection window:

1. Open the auto-complete's Validation window.
2. From the Expected list length field, select **Short**.

   This feature is only available for short lists.
3. From the Selection mode radio button, select **Starts With**.
4. Save the Validation.

Note

This setting only controls the matching in the Select page. Matching in the auto-complete field is controlled by including specific clauses in the auto-complete's SQL. See above for details.

To configure "contains" matching within the interactive selection window:

1. Open the auto-complete's Validation window.
2. From the Expected list length field, select **Short**.

   This feature is only available for short lists.
3. From the Selection mode radio button, select **Contains**.
4. Save the Validation.

Note

This setting only controls the matching in the Select page. Matching in the auto-complete field is controlled by including specific clauses in the auto-complete's SQL. See above for details.

Configuration Tips

Consider the following tips when configuring the “starts with” versus “contains” functionality for auto-complete fields and the Select page.

Tip

Auto-completes should be configured such that the field matching behavior works the same way as the Select page matching behavior. Specifically, if the auto-complete field uses the "starts with" clauses in the SQL, then the selection window should use the "Starts With" Selection Mode. See *“Configuration Instructions”* on page 302 for details.

Tip

Consider using the "Contains" Selection Mode for fields with multi-word values. For example, consider the possible values for the Request Type auto-complete field:

```
Development Bug
Development Enhancement
Development Issue
Development Change Request
IS Bug
IS Enhancement
IS Issue
IS Change Request
Support Issue
Support Change Request
```

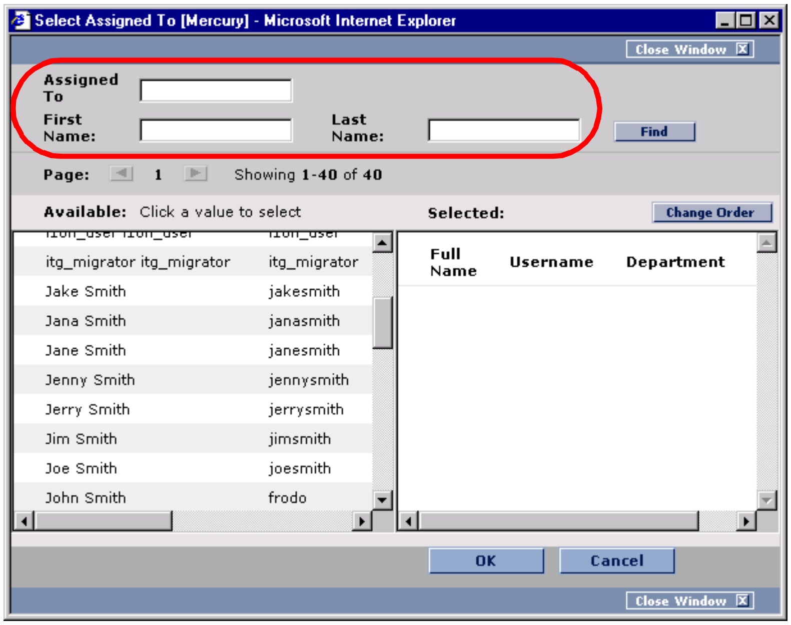
Using "contains" can be useful here. The user knows that he needs to log a bug against one of the IS-supported Financial applications. The user types "bug" into the auto-complete field and presses the Tab key. The following items are returned:

```
Development Bug
IS Bug
```

The user selects "IS Bug." Without the "contains" feature enabled, typing "bug" would have returned the entire list. He might have also typed "Financial," thinking that there might be a separate Request Type used for each type of supported application. This, too, would have returned the entire list. At that point, the user would be forced to try another "starts with" phrase or simply read the entire (potentially long) list.

Adding Search Fields to the Auto-Complete Window

Auto-completes with a long list of values can be configured to display additional filter fields in the Select window. These fields can be used to search other properties than the primary values in the list. Users can enter values in the filter fields and click **Find** to display only the values that match the search criteria. *Figure B-3* shows the Select window with additional filter fields.


Select Assigned To [Mercury] - Microsoft Internet Explorer
Close Window
Assigned To
First Name:
Last Name:
Find
Page: 1 Showing 1-40 of 40
Available: Click a value to select
Selected:
Change Order
OK
Cancel
Close Window

itg_migrator itg_migrator	itg_migrator
Jake Smith	jakesmith
Jana Smith	janasmith
Jane Smith	janesmith
Jenny Smith	jennysmith
Jerry Smith	jerrysmith
Jim Smith	jimsmith
Joe Smith	joesmith
John Smith	frodo

Full Name	Username	Department

Figure B-5 Filter Fields in the Auto-Complete Select Window

Note

Filter fields can not be configured when validating your list by **List**, **Command With Delimited Output** or **Command With Fixed Width Output**.

To add a filter field to the auto-complete validation:

1. Open the Validation for the auto-complete.

   Auto-complete validations must display **Auto Complete List** in the Component Type field.

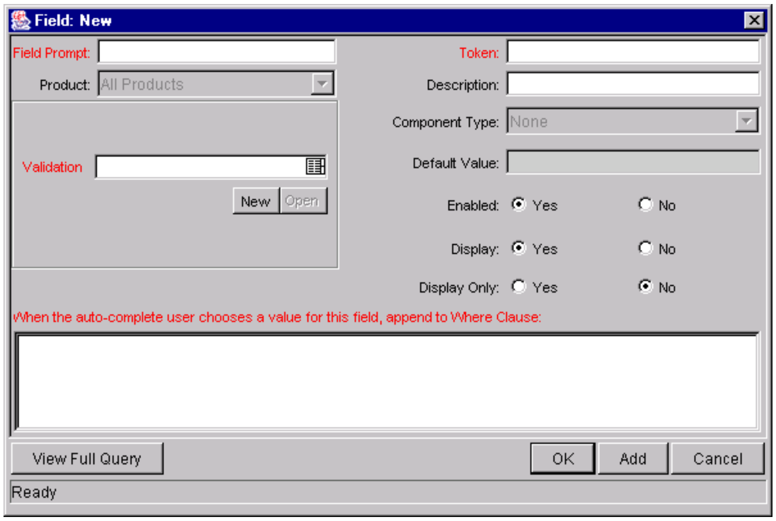
2. In the Expected list length field, select **Long**.

   Only long formatted auto-complete lists can include filter fields.

3. Click the **Filter Fields** tab.

4. Click **New**.

The Field: New window opens.



Field: New

Field Prompt:

Product: All Products

Validation

Token:

Description:

Component Type: None

Default Value:

Enabled: ● Yes ○ No

Display: ● Yes ○ No

Display Only: ○ Yes ● No

When the auto-complete user chooses a value for this field, append to Where Clause:

Ready

5. Enter the required information.

   Table B-2 defines all of the fields on this window.

Table B-2. Fields in the Fields:New Window

Field	Description
Field Prompt	The name that is displayed for the field in the auto-complete Select window.
Product	
Validation	The Validation for the filter field. You can select any type of Validation, except for auto-complete type Validations. The values accepted by this Validation will be appended to the WHERE clause in the SQL query that determines the ultimate auto-complete list display.
New	Opens the Validation window where you can construct a new Validation for the filter field. Note that you can not use an auto-complete type Validation for the filter field.
Open	Opens the Validation window and displays the definition of the Validation specified in the Validation field.
Token	The Token for the field value. The Token value will be appended to the WHERE clause in the SQL query that determines the ultimate auto-complete list display.

Table B-2. Fields in the Fields:New Window [continued]

Field	Description
Description	The description of the filter field.
Component Type	The component type for the filter field, determined by its Validation.
Default Value	The default value for the filter field, determined by its Validation.
Enabled	Determines whether the filter field is enabled.
Display	Determines whether the filter field is visible to the user in the auto-complete's Select window.
Display Only	Determines whether the filter field is updatable. When Display Only is set to **Yes**, the field can not be updated.
When the auto-complete user chooses a value for this field, append to WHERE clause:	The AND clause that is appended to the portlet's WHERE clause if the user enters a value in this filter field. Each filter field will append its term to the portlet query when a value is entered by the end user in the Select window. For example, if the filter field uses the CRT-Priority-Enabled Validation and a filter field Token of P_PRIORITY, enter the following into this field: `AND R.PRIORITY_CODE = '[P.P_PRIORITY]'` Note: The value in this field must start with "AND".
View Full Query	Opens a window showing the full query.

6. Click **OK**.

Tip

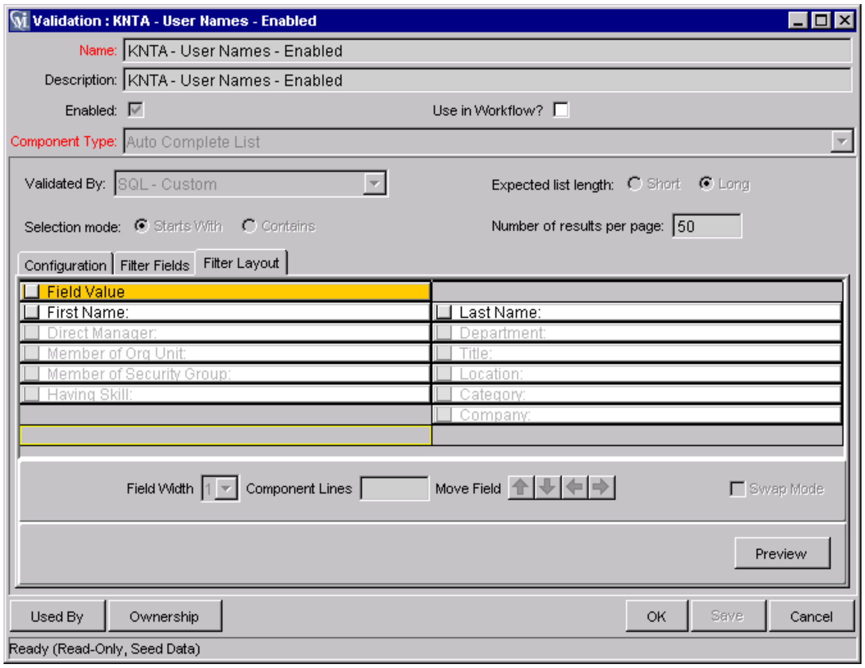
Filter fields can offer a powerful method for enabling users to efficiently locate specific values in large lists. When adding filter fields to an auto-complete Validation, consider the following tips:

- Ensure that the filter fields are functionally related to the list of values. For example, a validation that provides a list of Request Types can include a filter field for a specific Department associated with the Request Types.
- Consider reusing (copying) an auto-complete Validation and modifying the filter fields to display a subset of the entire list. Using the Displayed, Display Only, and Default fields in the Filter Field window, you can configure the auto-complete values to automatically limit the results.
- Performance can degrade if joining tables over database links.
- Only use this functionality for complex fields.

To modify the filter field layout:

1. Open the auto-complete Validation that includes filter fields on the **Filter Fields** tab.
2. Click the **Filter Layout** tab.

   The tab lists the primary field and all of the filter fields that have been defined for the auto-complete. The primary field is named Field Value. This is the field that holds the eventual selected value.



Validation : KNTA - User Names - Enabled

Name: KNTA - User Names - Enabled

Description: KNTA - User Names - Enabled

Enabled: ☑ Use in Workflow? □

Component Type: Auto Complete List

Validated By: SQL - Custom

Expected list length: ○ Short ● Long

Selection mode: ● Starts With ○ Contains

Number of results per page: 50

Configuration | Filter Fields | Filter Layout

□ Field Value	
□ First Name:	□ Last Name:
□ Direct Manager:	□ Department:
□ Member of Org Unit:	□ Title:
□ Member of Security Group:	□ Location:
□ Having Skill:	□ Category:
	□ Company:

Field Width 1 Component Lines Move Field □ Swap Mode

Ready (Read-Only, Seed Data)

3. Select the field that you would like to move.

   To select more than one field, use the Shift key while selecting a range. It is only possible to select a continuous set of fields (i.e. the Ctrl-Select functionality is not supported).

4. Use the arrow buttons to move the fields to the desired location in the layout builder.

Note

A field or a set of fields cannot be moved to an area where other fields already exist. The other field(s) must be moved out of the way first.

5. To switch the positions of two fields:

   a. Select the first field and check the Swap Mode check box.

      An “S” appears in the check box area of the selected field.

   b. Double-click the second field that you want to switch positions with the first.

This causes the two fields to change positions. Following the switch, the Swap Mode check box is turned off. To swap another set of fields, repeat this procedure.

6. To check what the layout looks like in actual use, click **Preview**.

   This opens a small window that shows the fields as they will appear. It is important to note that:

   - Any rows with no fields are ignored. They do not show up as a blank line.
   - Any non-displayed fields do not affect the layout. They are considered the same as a blank field.

Special Case: Configuring an Auto-Complete List of Users

User auto-completes or Validations (Validated by: SQL-User) have three filter fields by default:

- Primary field—this field takes the name of the auto-complete field
- First Name
- Last Name

The user auto-complete always appears in the long list format, which uses the paging interface to display the items. Additionally, user auto-completes display a different icon (👤) in the auto-complete field.

To configure a user auto-complete Validation:

1. Create a new Validation.

   The Validation window opens.

2. From the Component Type field, select **Auto Complete List**.
3. From the Validated By field, select **SQL - User**.
4. Configure the SQL query that will determine the users listed in the Validation.

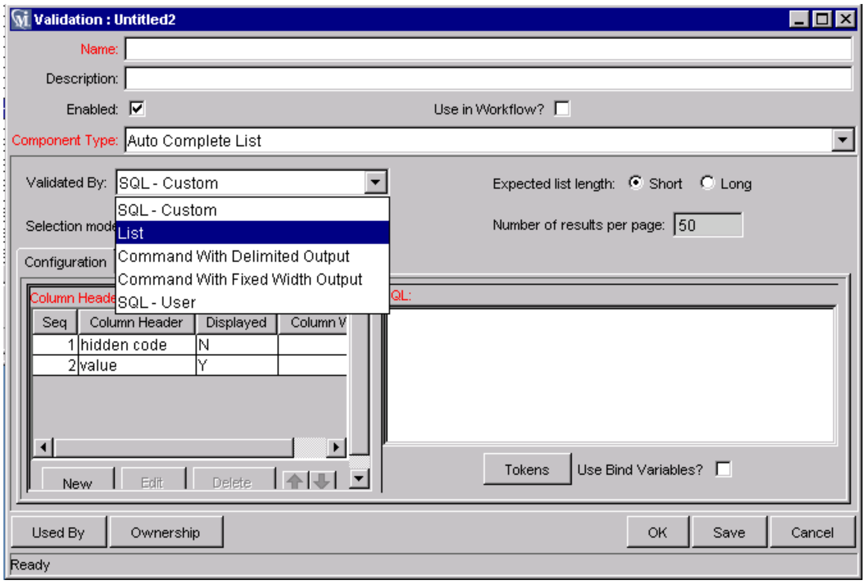
   See *"Configuring the Auto-Complete Values"* on page 311 for details.

5. Click **Save**.

Configuring the Auto-Complete Values

The values in an auto-complete list can be specified in the following ways. In the Validate By field, select one of the following:

- **List**: Used to enter specific values.
- **SQL**: Uses a SQL statement to build the contents of the list.
- **SQL - User**: Identical to SQL configuration, but includes a few additional preconfigured filter fields.
- **Command With Delimited Output**: Uses a system command to produce a character-delimited text string and uses the results to define the list.
- **Command With Fixed Width Output**: uses a system command to produce a text file and parses the result on the basis of the width of columns, as well as the headers.



Name:

Description:

Enabled: ☑ Use in Workflow? ☐

Component Type: Auto Complete List

Validated By: SQL - Custom

Expected list length: ● Short ○ Long

Number of results per page: 50

Seq	Column Header	Displayed	Column V
1	hidden code	N	
2	value	Y	

Use Bind Variables? ☐

Figure B-6 Auto-Complete List

The following sections discuss the following topics:

- *Validation by Command With Delimited Output*
- *Validation by Command With Fixed Width Output*
- *User-Defined Multi-Select Auto-Complete Fields*
- *Example: Token Evaluation and Validation by Command with Delimited Output*

For more information on creating auto-completes validated by List or SQL, refer to the following sections:

- *"Static List Validations"* on page 291
- *"Dynamic List Validations"* on page 293

Validation by Command With Delimited Output

Validations that are validated by commands with delimited output can be used to get data from an alternate source, and use that data to populate an auto-complete field. This functionality provides additional flexibility when designing auto-complete lists.

Many enterprises need to use alternate sources of data within their applications. Examples of these sources are a flat file, an alternate database source, or output from a command line execution. Special commands may be used in conjunction with these alternate data sources, in the context of a Validation, to provide a list of values.

To configure a validation by command with delimited output:

1. In the Validation Workbench, under Validated By, choose **Command With Delimited Output** and input the delimiting character.

2. Under New Command, enter in the command steps to be executed.

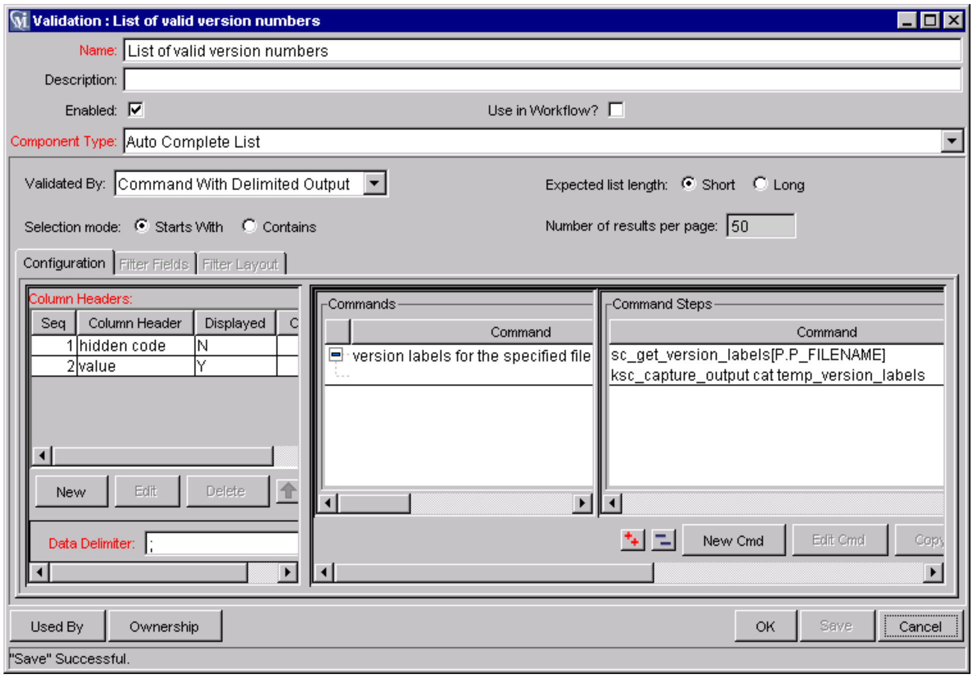
   These can include Mercury ITG Special Commands. Your commands should include the Special Command `ksc_capture_output`, which captures and parses the delimited command output. If the `ksc_capture_output` Special Command is surrounded by the `ksc_connect` and `ksc_disconnect` commands, the command will be run on the remote system. Otherwise, the command will be run locally on the Mercury ITG server (similar to `ksc_local_exec`).

Example

The simple example below uses a comma for a delimiter and has the validation values red, blue and green. The script places the validations into the newfile.txt file, and then uses the Special Command ksc_capture_output to process the text of the file.

```
ksc_begin_script[AS.PKG_TRANSFER_PATH]newfile.txt
red,red
blue,blue
green,green
ksc_end_script
ksc_capture_output cat[AS.PKG_TRANSFER_PATH]newfile.txt
```

Table B-3 shows the Validation window for Command with Delimited Output.



Validation : List of valid version numbers

Name: List of valid version numbers

Description:

Enabled: ☑

Use in Workflow? □

Component Type: Auto Complete List

Validated By: Command With Delimited Output

Expected list length: ● Short ○ Long

Selection mode: ● Starts With ○ Contains

Number of results per page: 50

Column Headers:

Seq	Column Header	Displayed	C
1	hidden code	N	
2	value	Y	

Data Delimiter: ;

Commands

Command
version labels for the specified file

Command Steps

Command
sc_get_version_labels[P.P_FILENAME]
ksc_capture_output cat temp_version_labels

"Save" Successful.

Figure B-7 Validation by Command with Delimited Output

Table B-3. Validation by Command With Delimited Output

Field	Definition
Command Panel	Panel where new commands can be added to capture Validation values.
Data Delimiter	Indicates the character or key by which the file will be separated into the Validation columns.

Headers can also be defined for the columns selected. These column headers are used in the window that opens when a value is selected from an auto-complete list. To define a new header, click **New** under Column Header. Table B-4 shows the fields that can be entered for a column header. If a column header is not defined for each column in a Command, a default name is used.

Table B-4. Column Headers

Field	Definition
Column Header	The name of the column that is displayed in the auto-complete window.
Display	Determines whether or not the header is displayed in the Validation.

Validation by Command With Fixed Width Output

Validations by Command with Fixed Width Output can be used to obtain data from an alternate source, and use that data to populate an auto-complete field. This functionality provides additional flexibility when designing auto-complete lists.

Many enterprises need to use alternate sources of data within their applications. Examples of these sources are a flat file, an alternate database source, or output from a command line execution. Special commands may be used in conjunction with these alternate data sources, in the context of a Validation, to provide a list of values on the fly.

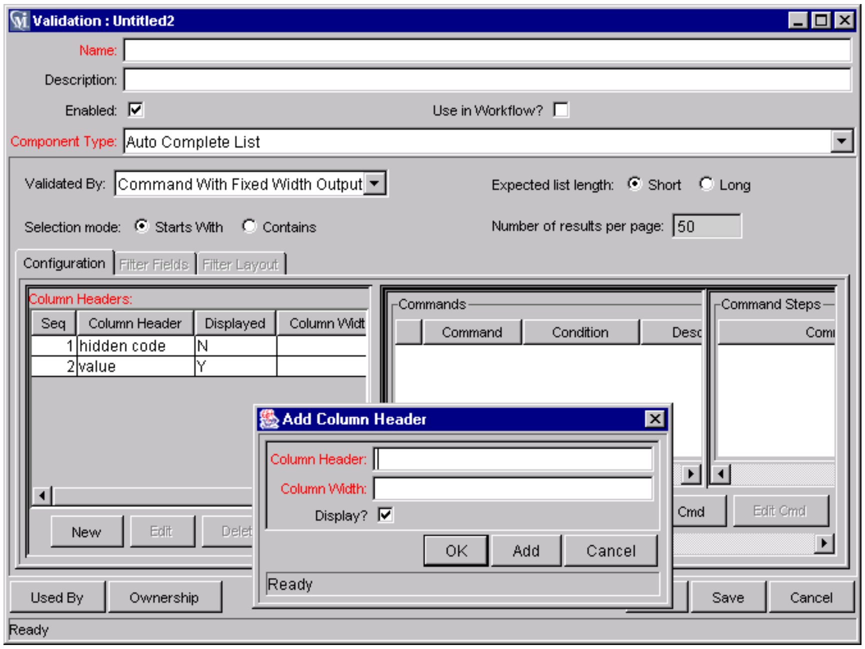
In the Validation Workbench, under Validated By, choose **Command With Fixed Width Output** and input the appropriate width information.

Then, under New Command, enter in the command steps to be executed. These can include Special Commands. Your commands should include the Special Command `ksc_capture_output`, which captures and parses the delimited command output. If the `ksc_capture_output` Special Command is surrounded by the `ksc_connect` and `ksc_disconnect` commands, the command will be run on the remote system. Otherwise, the command will be run locally on the Mercury ITG server (similar to `ksc_local_exec`).

Example

The example below has the validations red, blue and green. The column width is set to a value of 6. The script places the validations into the newfile.txt file.

```
ksc_begin_script[AS.PKG_TRANSFER_PATH]newfile.txt
red    red
blue   blue
green  green
ksc_end_script
ksc_capture_output cat[AS.PKG_TRANSFER_PATH]newfile.txt
```



Validation : Untitled2

Name:

Description:

Enabled: ☑ Use in Workflow? □

Component Type: Auto Complete List

Validated By: Command With Fixed Width Output

Expected list length: ● Short ○ Long

Selection mode: ● Starts With ○ Contains

Number of results per page: 50

Column Headers:

Seq	Column Header	Displayed	Column Widt
1	hidden code	N	
2	value	Y	

Add Column Header

Column Header:

Column Width:

Display? ☑

Ready

Ready

Figure 10-2 Validation by Command with Fixed Width Output

Table B-5. Validation by Command With Fixed Width Output

Field	Definition
Command Panel	The panel where new commands can be added to capture Validation values.

Headers can also be defined for the columns selected. These column headers are used in the window that opens when a value is selected from an auto-complete list. To define a new column header, click **New** under Column Header. *Table B-6* shows the fields can be entered for a column header. If a column header is not defined for each column in a Command, a default name is used.

Table B-6. Column Headers

Field	Definition
Column Header	The name of the column that is displayed in the Auto Complete dialog.

Table B-6. Column Headers

Field	Definition
Display	Whether or not the column is displayed. The first column is never displayed and the second column is always displayed.
Column Width	The number of characters in each column of the output generated as a result of the command.

User-Defined Multi-Select Auto-Complete Fields

A number of auto-complete fields in the Workbench have been pre-configured to allow users to open a separate window for selecting multiple values from a list. Users can also define custom auto-complete fields to have multi-select capability when creating various product entities.

The user-defined multi-select capability is supported for:

- User Data fields
- Report Type fields
- Request Type fields
- Project Template fields

The user-defined Multi-Select capability is **not** supported for:

- Request Header Types
- Object Types

In order to use this feature when creating a new entity, users must:

- Select a Validation for the new entity that has **Auto-Complete List** as the Component Type. This enables the Multi-Select Enabled field in the Field: New window.
- In the Field: New window, users must click **Yes** for the Multi-Select Enabled radio button.

The step-by-step procedure for defining multi-select capability in User Data, Report Type, Request Type Project Template fields is very similar. The procedure for enabling this capability for Request Type field is shown below as an example.

To define a multi-select auto-complete field for a Request Type:

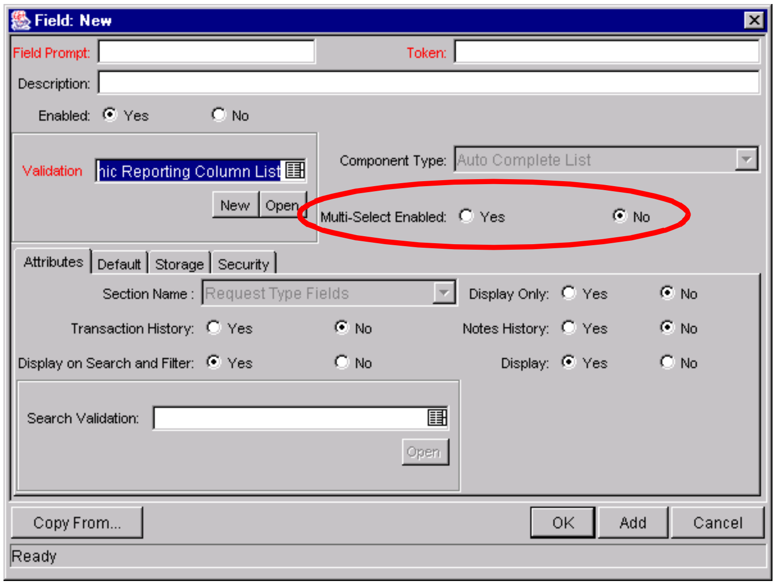
1. Click the **Create** screen group and click the **Request Types** screen.

   The Request Type Workbench opens.

2. Click **New Request Type**.

   The Request Type window opens.

3. Click **New**. The Field: New window opens.



Field Prompt:
Token:
Description:
Enabled: ● Yes ○ No
Validation: hic Reporting Column List
Component Type: Auto Complete List
Multi-Select Enabled: ○ Yes ● No
Section Name : Request Type Fields
Display Only: ○ Yes ● No
Transaction History: ○ Yes ● No
Notes History: ○ Yes ● No
Display on Search and Filter: ● Yes ○ No
Display: ● Yes ○ No
Search Validation:
Ready

4. Click the auto-complete icon for the **Validation** field.

   The Validate window opens.

5. In the Validate window, select a Validation that has **Auto-Complete List** as the Component Type.

6. Click **OK** in the Validate window.

   The Validate window closes. The **Validation** field is populated with the selection from the Validate window. The Multi-Select Enabled option is now enabled.

7. Click the **Yes** radio button for the Multi-Select Enabled option.

   The Possible Conflicts window opens. It warns you not to use a multi-select auto-complete for Advanced Queries, Workflow Transitions and Reports.

If this field is not going to be used in Advanced Queries, Workflow Transitions or Reports, click **Yes** to continue.

8. Configure the other options in this window for the new Request Type.

9. Click **OK**.

The field is now enabled for multi-select auto-complete.

Example: Token Evaluation and Validation by Command with Delimited Output

The Validation functionality can be extended to include field dependent token evaluation. Validations can be configured to dynamically change, depending on the client-side value entered in another field.

To use field dependent token evaluation, it is necessary to configure a Validation in conjunction with an Object Type, Request Type, Report Type, Project Template, or User Data definition. Consider the following example for setting up an Object Type using field dependent tokens.

1. Generate a Validation and set the following parameters as shown:

   a. Name: **demo_client_token_parsing**

   b. Component Type: **Auto Complete List**

   c. Validated By: **Command With Delimited Output**

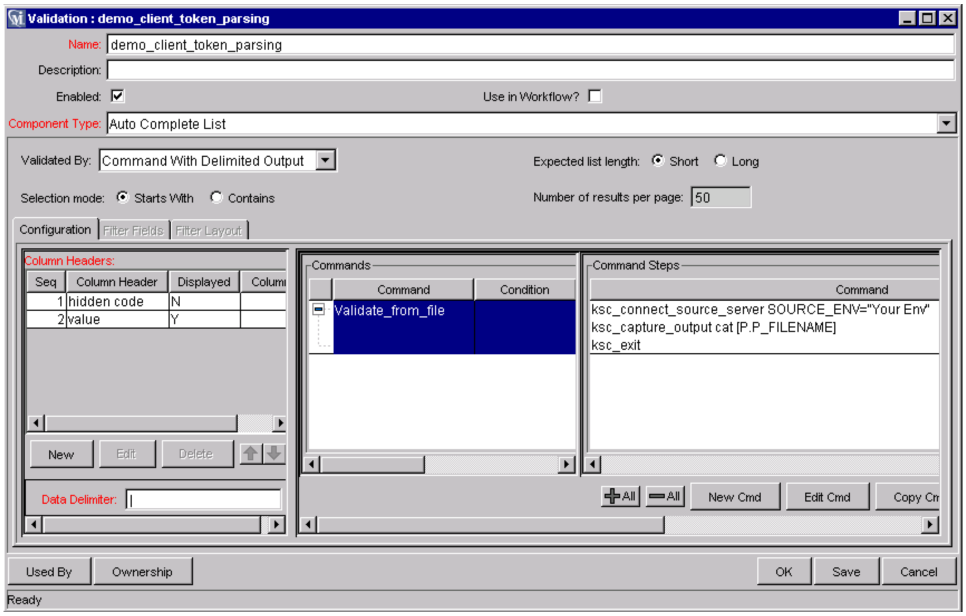
   d. Data Delimiter: | (bar)

   e. Command

      o Command: **Validate_from_file**

      o Steps

```
ksc_connect_source_server SOURCE_ENV="Your Env"
ksc_capture_output cat [P.P_FILENAME]
ksc_exit
```



Validation : demo_client_token_parsing

Name: demo_client_token_parsing

Description:

Enabled: ☑

Use in Workflow? □

Component Type: Auto Complete List

Validated By: Command With Delimited Output

Expected list length: ● Short ○ Long

Selection mode: ● Starts With ○ Contains

Number of results per page: 50

Column Headers:

Seq	Column Header	Displayed	Colum
1	hidden code	N	
2	value	Y	

Data Delimiter: |

Commands

Command	Condition
Validate_from_file	

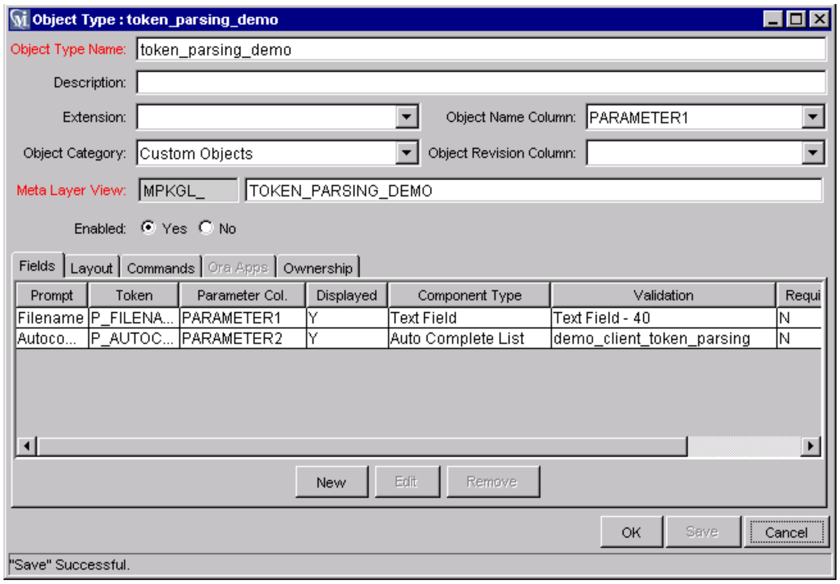
Command Steps

Command
ksc_connect_source_server SOURCE_ENV="Your Env"
ksc_capture_output cat [P.P_FILENAME]
ksc_exit

Ready

When called, this Validation will connect to an Environment called ‘Your Env’ and retrieve data from a file specified by the token P_FILENAME. The file should be located in the directory specified in the Base Path in the Environment window.

2. Generate an Object Type named **token_parsing_demo**.



Object Type : token_parsing_demo

Object Type Name: token_parsing_demo

Description:

Extension:

Object Name Column: PARAMETER1

Object Category: Custom Objects

Object Revision Column:

Meta Layer View: MPKGL_ TOKEN_PARSING_DEMO

Enabled: ● Yes ○ No

Prompt	Token	Parameter Col.	Displayed	Component Type	Validation	Requi
Filename	P_FILENA...	PARAMETER1	Y	Text Field	Text Field - 40	N
Autoco...	P_AUTOC...	PARAMETER2	Y	Auto Complete List	demo_client_token_parsing	N

"Save" Successful.

a. Generate a new field with the following parameters:

- Name: **Filename**
- Token: **P_FILENAME**
- Validation: **Text Field - 40**

b. Generate a new field with the following parameters:

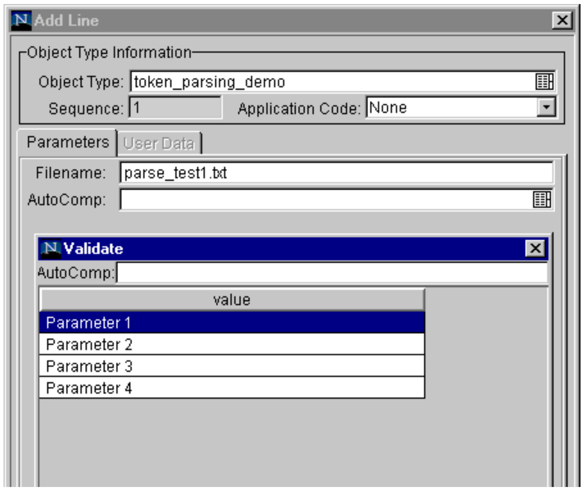
- Name: **AutoComp**
- Token: **P_AUTOCOMP**
- Validation: **demo_client_token_parsing** (this is the Validation that was defined above)

3. For this example to return any values in the auto-complete, a file must be generated in the directory specified in the Base Path in the Environment Detail of ‘Your Env’ Environment. Generate a file named ‘parse_test1.txt’ with the following delimited data:

```
DELIMITED_TEXT1|Parameter 1
DELIMITED_TEXT2|Parameter 2
DELIMITED_TEXT3|Parameter 3
DELIMITED_TEXT4|Parameter 4
```

The Object Type ‘token_parsing_demo’ is now enabled to use this token evaluation. To test the above configuration sample:

1. Generate a new Package.
2. Select a Workflow and click **Add Line.**
3. Select **token_parsing_demo** from the Object Type drop down list. The following fields are displayed:
    - Filename
    - AutoComp
4. Type ‘parse_test1.txt’ in the Filename field.
5. Click on the auto-complete box in the AutoComp field. The following Validation window opens, displaying the contents of the ‘parse_test1.txt’ file.



Object Type Information

Object Type: token_parsing_demo

Sequence: 1

Application Code: None

Filename: parse_test1.txt

AutoComp:

AutoComp:

value
Parameter 1
Parameter 2
Parameter 3
Parameter 4

Configuring Text Fields

Text fields displayed on a single line. Text fields can be configured to display the data according to a certain format. For example, you can configure a text field to accept and format a ten digit telephone number or display a specific number of decimal places for a percentage.

This section covers the following topics:

- *Creating a Text Field Validation Overview*
- *Available Text Data Masks*
- *Customizing the System Text Data Masks*
- *Creating a Custom Data Mask*

Creating a Text Field Validation Overview

To create a text field Validation:

1. Open the Validation window in the Workbench.
2. In the Name field, enter the name of the Validation.

3. From the Component Type field, select **Text Field**.

4. From the Data Mask field, select the data mask that represents the desired format for the field.

   See *"Available Text Data Masks"* on page 322 for additional details.

5. (Optional) Configure the selected data mask.

   See *"Customizing the System Text Data Masks"* on page 324 for additional details.

6. Click **OK**.

Available Text Data Masks

The Mercury IT Governance Center includes a number of preconfigured data masks that can be used when creating text field Validations. Each of these data masks can be configured to meet your specific data requirements. *Table B-7* defines the data masks delivered with Mercury ITG.

Table B-7. Data Mask Formats

Data Mask	Description
Alphanumeric	Field allows all alphanumeric characters. The maximum field length for fields using this Validation can be specified.
Alphanumeric Uppercase	Field allows alphanumeric characters and formats all characters as uppercase text. The maximum field length for fields using this Validation can be specified.
Numeric	Field allows only numeric characters. The following characteristics can be specified for this data mask: • Range of values (maximum and minimum) allowed for this field • Whether or not a zero is displayed when data is not entered into the field • Whether or not group separators (such as a comma) are used within large numbers • How negative numbers are displayed • Number of decimal places See *"Customizing the Numeric Data Mask"* on page 324 for details.

Table B-7. Data Mask Formats [continued]

Data Mask	Description
Currency	Field allows only numeric characters and is used to display currency data. The following characteristics can be specified for this data mask: • Range of values (maximum and minimum) allowed for this field • Whether or not a zero is displayed when data is not entered into the field • Whether or not group separators (such as a comma) are used within large numbers • How negative numbers are displayed • Number of decimal places See *"Customizing the Currency Data Mask"* on page 326 for details.
Percentage	Field allows only numeric characters and is used to display percentages. The following characteristics can be specified for this data mask: • Range of values (maximum and minimum) allowed for this field • Whether or not a zero is displayed when data is not entered into the field • Whether or not group separators (such as a comma) are used within large numbers • How negative numbers are displayed • Number of decimal places See *"Customizing the Percentage Data Mask"* on page 328 for details.
Telephone	Field allows only numeric characters and is used to display telephone numbers. The following characteristics can be specified for this data mask: • Format—specify how many digits are included, and what delimiter should be used between groups of numbers. For example, you can select to use dashes (-) rather than periods (.) between numbers: 555-555-5555 or 555.555.5555. • Maximum and minimum number of digits See *"Customizing the Telephone Data Mask"* on page 330 for details.

Table B-7. Data Mask Formats [continued]

Data Mask	Description
Custom	Field allows a range of custom inputs. You can customize the field to accept digits, letters, spaces, and custom delimiters. See *"Creating a Custom Data Mask"* on page 332 for details.

Customizing the System Text Data Masks

Each data mask that is included in Mercury ITG can be customized. The following sections provide additional details on modifying the data masks:

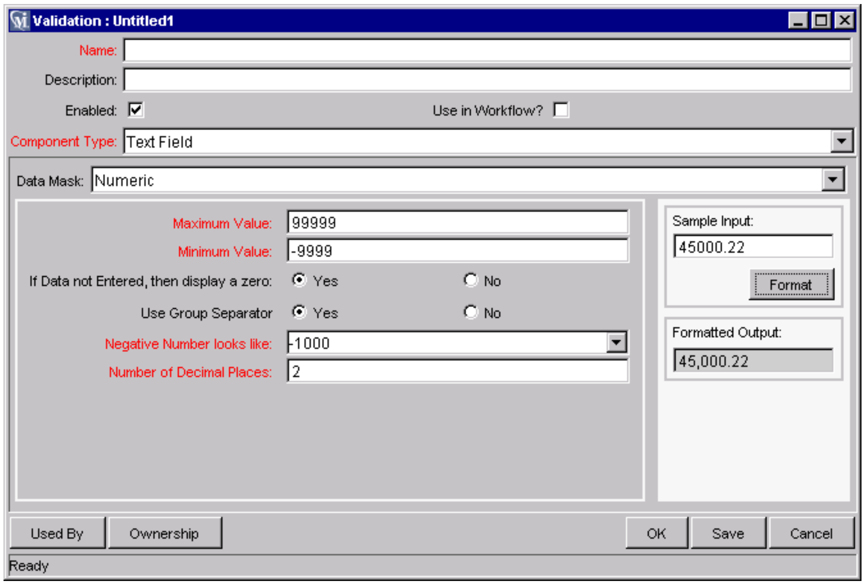
- *Customizing the Numeric Data Mask*
- *Customizing the Currency Data Mask*
- *Customizing the Percentage Data Mask*
- *Customizing the Telephone Data Mask*

Customizing the Numeric Data Mask

The numeric data mask allows only numeric characters. When creating a validation using this data mask, the following characteristics can be specified:

- Range of values (maximum and minimum) allowed for this field
- Whether or not a zero is displayed when data is not entered into the field
- Whether or not group separators (such as a comma) are used within large numbers
- How negative numbers are displayed
- Number of decimal places

Figure B-8 shows the fields that can be configured for this data mask. *Table B-8* defines these fields.



Name:

Description:

Enabled: ☑

Use in Workflow? □

Component Type: Text Field

Data Mask: Numeric

Maximum Value: 99999

Minimum Value: -9999

If Data not Entered, then display a zero: ● Yes ○ No

Use Group Separator ● Yes ○ No

Negative Number looks like: -1000

Number of Decimal Places: 2

Sample Input: 45000.22

Formatted Output: 45,000.22

Figure B-8 Validation window for the numeric data mask

Table B-8. Fields for configuring the numeric data mask for text fields

Field	Description
Maximum Value	Largest value allowed for this field. This can be a positive or negative number.
Minimum Value	Smallest value allowed for this field. This can be a positive or negative number.
If Data not Entered, then display a zero	Determines if the field should display a zero when no data is entered.
Use Group Separator	Determines if the field should use a group separator (such as a comma) to divide characters within large numbers. For example: 1000000 versus 1,000,000. The character used for the separator defaults based on the machine's local, but can be configured in the Regional Settings dialog in the Workbench. Select **Edit > Regional Settings** to access this window.

Table B-8. Fields for configuring the numeric data mask for text fields

Field	Description
Negative Number looks like	Determines the appearance of negative numbers. There are four options available: • (1000)—parenthesis and black text • (1000)—parenthesis and red text • -1000—minus sign (-) and black text • -1000—minus sign (-) and red text
Number of Decimal Places	Determines the number of allowed decimal places. Users will only be able to enter up to this number of digits beyond the decimal place.

To view your customized data mask:

1. In the Sample Input field, enter the digits that you would like to see formatted.
2. Click **Format**.

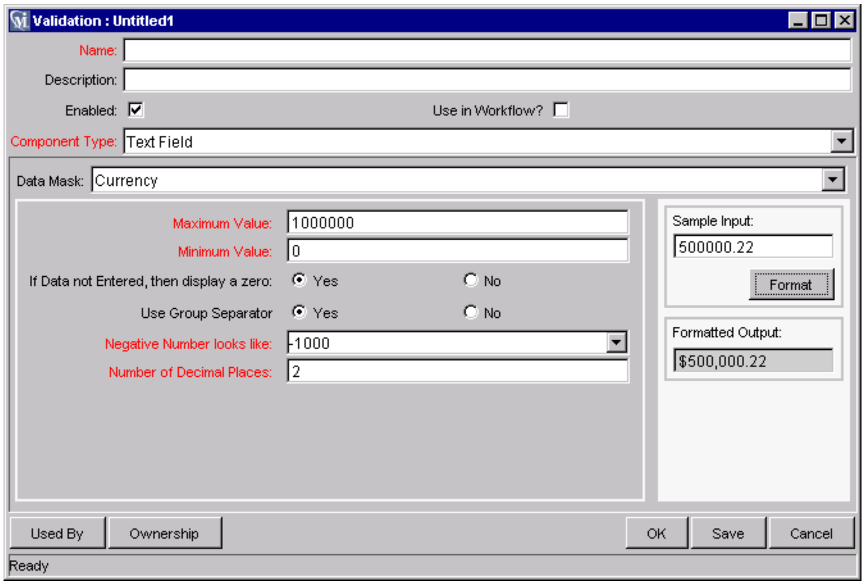
The digits are formatted according to your settings and displayed in the Formatted Output field.

Customizing the Currency Data Mask

The currency data mask allows only numeric characters and is used to display currency data. When creating a validation using this data mask, the following characteristics can be specified:

- Range of values (maximum and minimum) allowed for this field
- Whether or not a zero is displayed when data is not entered into the field
- Whether or not group separators (such as a comma) are used within large numbers
- How negative numbers are displayed
- Number of decimal places

Figure B-9 shows the fields that can be configured for this data mask. *Table B-9* defines these fields.



Name:

Description:

Enabled: ☑ Use in Workflow? □

Component Type: Text Field

Data Mask: Currency

Maximum Value: 1000000

Minimum Value: 0

If Data not Entered, then display a zero: ● Yes ○ No

Use Group Separator ● Yes ○ No

Negative Number looks like: -1000

Number of Decimal Places: 2

Sample Input: 500000.22

Formatted Output: $500,000.22

Figure B-9 Validation window for the currency data mask

Table B-9. Fields for configuring the currency data mask for text fields

Field	Description
Maximum Value	Largest value allowed for this field. This can be a positive or negative number.
Minimum Value	Smallest value allowed for this field. This can be a positive or negative number.
If Data not Entered, then display a zero	Determines if the field should display a zero when no data is entered.
Use Group Separator	Determines if the field should use a group separator (such as a comma) to divide characters within large numbers. For example: 1000000 versus 1,000,000. The character used for the separator defaults based on the machine's local, but can be configured in the Regional Settings dialog in the Workbench. Select **Edit > Regional Settings** to access this window.

Table B-9. Fields for configuring the currency data mask for text fields

Field	Description
Negative Number looks like	Determines the appearance of negative numbers. There are four options available: • (1000)—parenthesis and black text • (1000)—parenthesis and red text • -1000—minus sign (-) and black text • -1000—minus sign (-) and red text
Number of Decimal Places	Determines the number of allowed decimal places. Users will only be able to enter up to this number of digits beyond the decimal place.

To view your customized data mask:

1. In the Sample Input field, enter the digits that you would like to see formatted.

2. Click **Format**.

The digits are formatted according to your settings and displayed in the Formatted Output field.

Note

> The INSTALLATION_CURRENCY server parameter dictates which currency symbol is displayed in the field. This parameter also dictated the position of the text in the field. For example:
>
> ```
> INSTALLATION_CURRENCY=$;RIGHT
> ```
>
> will right-align the text using a dollar sign.
>
> Contact your system administrator for help with changing this setting.

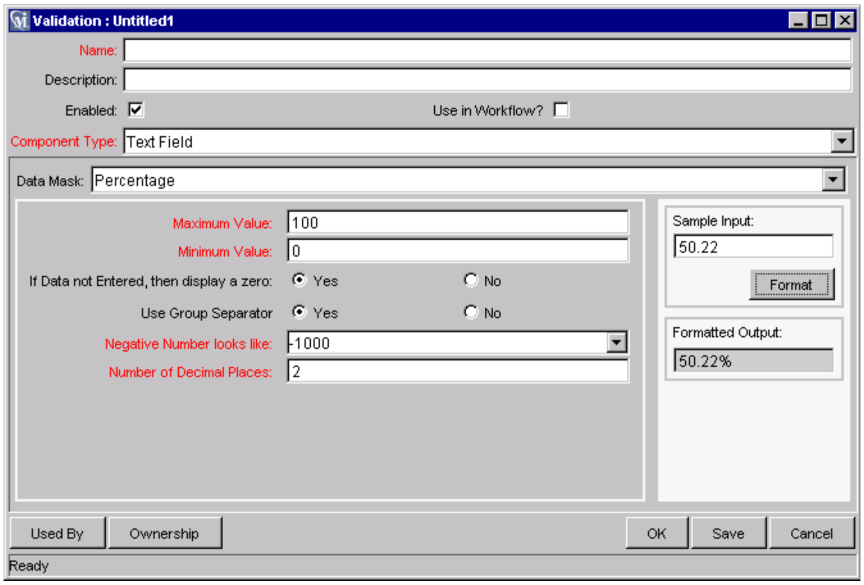
Customizing the Percentage Data Mask

The percentage data mask allows only numeric characters and is used to display percentages. When creating a validation using this data mask, the following characteristics can be specified:

- Range of values (maximum and minimum) allowed for this field
- Whether or not a zero is displayed when data is not entered into the field

- Whether or not group separators (such as a comma) are used within large numbers
- How negative numbers are displayed
- Number of decimal places

Figure B-10 shows the fields that can be configured for this data mask. *Table B-10* defines these fields.



Validation : Untitled1

Name:

Description:

Enabled: ☑ Use in Workflow? ☐

Component Type: Text Field

Data Mask: Percentage

Maximum Value: 100

Minimum Value: 0

If Data not Entered, then display a zero: ● Yes ○ No

Use Group Separator ● Yes ○ No

Negative Number looks like: -1000

Number of Decimal Places: 2

Sample Input: 50.22

Formatted Output: 50.22%

Ready

Figure B-10 Validation window for the percentage data mask

Table B-10. Fields for configuring the percentage data mask for text fields

Field	Description
Maximum Value	Largest value allowed for this field. This can be a positive or negative number.
Minimum Value	Smallest value allowed for this field. This can be a positive or negative number.
If Data not Entered, then display a zero	Determines if the field should display a zero when no data is entered.

Table B-10. Fields for configuring the percentage data mask for text fields

Field	Description
Use Group Separator	Determines if the field should use a group separator (such as a comma) to divide characters within large numbers. For example: 1000000 versus 1,000,000. The character used for the separator defaults based on the machine's local, but can be configured in the Regional Settings dialog in the Workbench. Select **Edit > Regional Settings** to access this window.
Negative Number looks like	Determines the appearance of negative numbers. There are four options available: • (1000)—parenthesis and black text • (1000)—parenthesis and red text • -1000—minus sign (-) and black text • -1000—minus sign (-) and red text
Number of Decimal Places	Determines the number of allowed decimal places. Users will only be able to enter up to this number of digits beyond the decimal place.

To view your customized data mask:

1. In the Sample Input field, enter the digits that you would like to see formatted.

2. Click **Format**.

The digits are formatted according to your settings and displayed in the Formatted Output field.

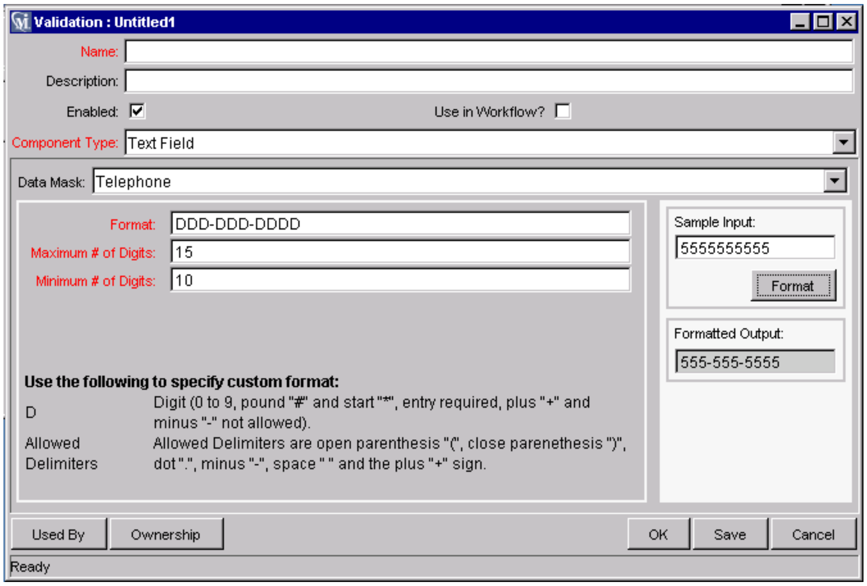
Customizing the Telephone Data Mask

The percentage data mask allows only numeric characters and is used to display telephone numbers. When creating a validation using this data mask, the following characteristics can be specified:

- Format—specify how many digits are included, and what delimiter should be used between groups of numbers. For example, you can select to use dashes (-) rather than periods (.) between numbers. For example, 555-555-5555 or 555.555.5555.

- Maximum and minimum number of digits

Figure B-11 shows the fields that can be configured for this data mask. *Table B-11* defines these fields.



Name:

Description:

Enabled: ☑ Use in Workflow? ☐

Component Type: Text Field

Data Mask: Telephone

Format: DDD-DDD-DDDD

Maximum # of Digits: 15

Minimum # of Digits: 10

Sample Input: 5555555555

Formatted Output: 555-555-5555

Use the following to specify custom format:

D Digit (0 to 9, pound "#" and start "*", entry required, plus "+" and minus "-" not allowed).

Allowed Delimiters Allowed Delimiters are open parenthesis "(", close parenethesis ")", dot ".", minus "-", space " " and the plus "+" sign.

Ready

Figure B-11 Validation window for the telephone data mask

Table B-11. Fields for configuring the telephone data mask for text fields

Field	Description
Format	The rule that dictates how the digits are formatted, including any spaces or delimiters. The following delimiters are allowed in the format definition: • Open and close parentheses ( ) • Period (.) • Dash (-) • Space • Plus sign (+) See *Table B-12 on page 332* for a few examples of different Telephone formats.
Maximum # of Digits	Largest number of digits that will be accepted in this field.
Minimum # of Digits	Smallest number of digits that will be accepted in this field. If the user enters fewer than this number of digits in the field and then tries to move from the field, he will receive an error.

Table B-12. Sample telephone data mask formats

Format Rule	Text Entered By User	Sample Formatted Output
D-DDD-DDD-DDDD	15555555555	1-555-555-5555
DDD DDD DDDD	5555555555	555 555 5555
(DDD) DDD-DDDD	5555555555	(555) 555-5555

To view your customized data mask:

1. In the Sample Input field, enter the digits that you would like to see formatted.

2. Click **Format**.

The digits are formatted according to your settings and displayed in the Formatted Output field.

Note

Special behavior applies to the extra characters, if your format is defined to allow a range of entries. Extra characters will always be grouped with the first set of characters. For example, if the telephone data mask is configured with a minimum of 10 characters and a maximum of 15 characters, then the following behavior is expected:

```
Format: DDD-DDD-DDDD
Min: 10
Max: 15
```

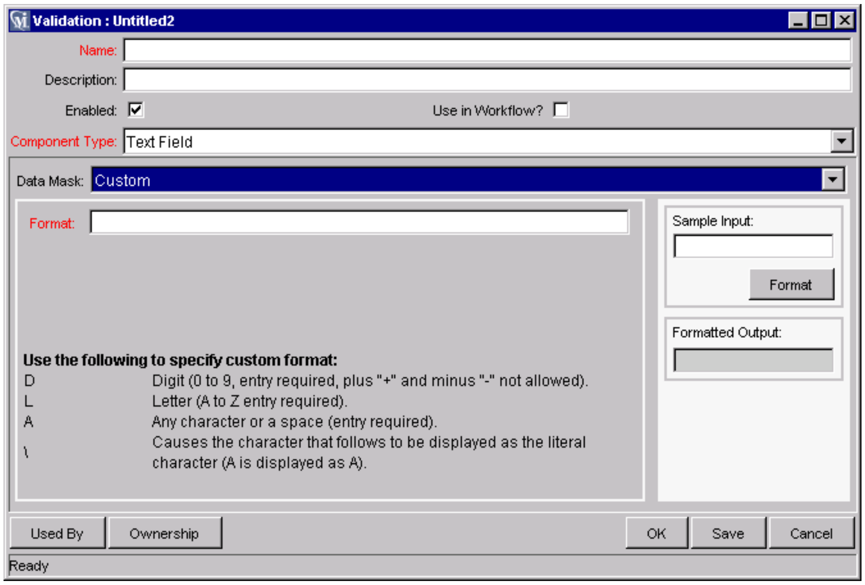
Input: `1234567890`
Output: `123-456-7890`

Input 2: `12345678901`
Output 2: `1234-567-8901`

Creating a Custom Data Mask

A custom data mask can be defined that will allow a range of inputs and format them to your specification. You can customize the field to accept digits, letters, spaces, and custom delimiters.

Figure B-11 shows the fields that can be configured for this data mask.



Validation : Untitled2

Name:

Description:

Enabled: ☑ Use in Workflow? □

Component Type: Text Field

Data Mask: Custom

Format:

Use the following to specify custom format:

D	Digit (0 to 9, entry required, plus "+" and minus "-" not allowed).
L	Letter (A to Z entry required).
A	Any character or a space (entry required).
\	Causes the character that follows to be displayed as the literal character (A is displayed as A).

Sample Input:

Formatted Output:

Ready

Figure B-12 Validation window for the custom data mask

To configure a custom format, enter a combination of symbols into the Format field. This field can accept the following entries:

- D—Specifies a required digit between 0 and 9.
- L—Specifies a required letter between A and Z.
- A—Specifies a required character or space.
- \ (backslash)—Causes the character that follows to be displayed as the literal character. For example: "\A" will be displayed as "A"

Table B-13 displays some examples of custom formats.

Table B-13. Sample custom data mask formats

Format Rule	Text Entered By User	Formatted Output
DDD\-DD\-DDDD	555555555	555-55-5555
AA\-DDD	BC349	BC-349

To view your customized data mask:

1. In the Sample Input field, enter the digits that you would like to see formatted.

2. Click **Format**.

The digits are formatted according to your settings and displayed in the Formatted Output field.

Using Directory and File Choosers

Directory and File Choosers are only used with Mercury Change Management Object Types. The following sections discuss them in more detail:

- *Directory Chooser*
- *File Chooser*

Directory Chooser

The Directory Chooser field can be used to select a valid directory from an Environment. Mercury Change Management connects to the first Source Environment on a Workflow and allows navigation through the directory structure and the selection of a directory from the list.

When implementing the Directory Chooser, note the following:

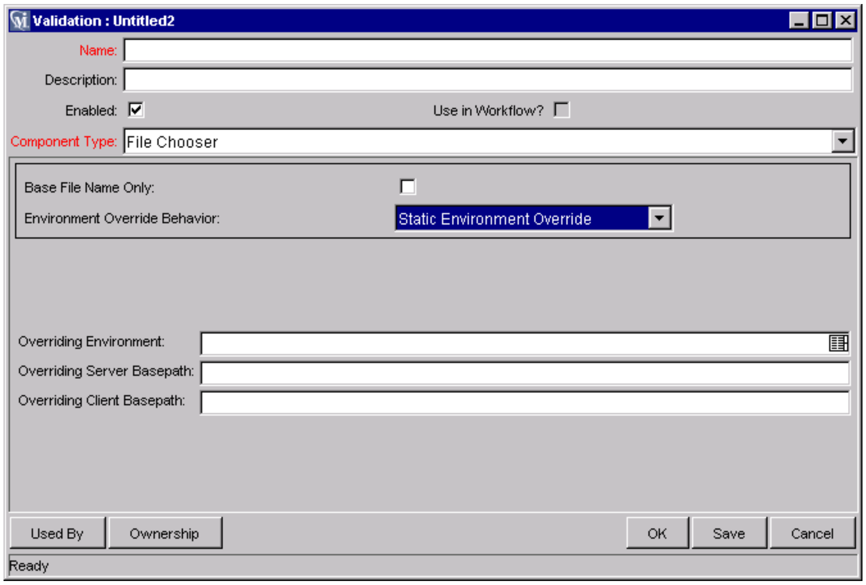
- The Directory Chooser field can only be used on an Object Type.
- On every Object Type that a Directory Chooser is chosen, it is also necessary to have a field whose token is **P_FILE_LOCATION** and whose validation is **DLV - File Location**. The possible values for this field are **Client** and **Server**. If **Client** is chosen, the Directory Chooser connects to the Client Base Path of the Source Environment. If **Server** is chosen, the Directory Chooser connects to the Server Base Path of the Source Environment.

File Chooser

A File Chooser field can be used by Object Types to select a valid file from an Environment. Mercury Change Management connects to the first Source Environment on a Workflow and provides the ability to view all files within a specific directory and select one from the list.

On every Object Type that a File Chooser is chosen, it is necessary to define the following fields:

- The first is a field for the File Location for the directory chooser, described in the previous section.
- The second is a field whose token is ‘P_SUB_PATH’. This field is the directory from which the file is selected and is usually a Directory Chooser field.



Enabled: ☑ Use in Workflow? □

Component Type: File Chooser

Base File Name Only: □

Environment Override Behavior: Static Environment Override

Figure B-13 Validation Window for Static Environment Override in File Chooser.

Table B-14. File Chooser Field

Field	Definition
Base File Name Only	Defines whether the base file name only (without its suffix) or the complete name is displayed.
Environment Override Behavior	Used to select files from a specific environment other than the default environment.

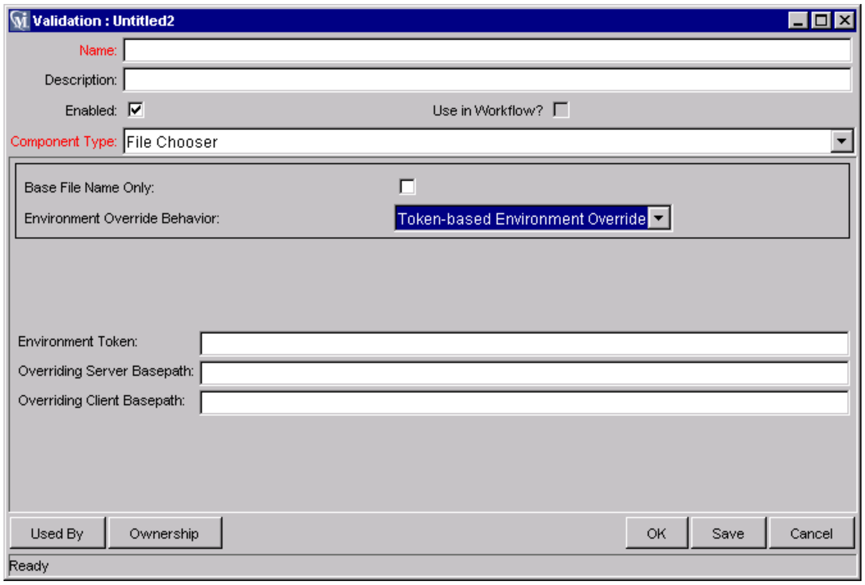
The Environment Override Behavior drop down list contains three options: **Default Behavior**, **Static Environment Override**, and **Token-Based Environment Override**.

Static Environment Override provides the ability to override one Environment at a time. The fields for Static Environment Override are pictured in *Figure B-13* and described in *Table B-15*.

Table B-15. Static Environment Override

Field	Definition
Overriding Environment	Selects the Environment to be overridden.
Overriding Server Basepath	The server basepath of the Environment may be overridden.
Overriding Client Basepath	The client basepath of the Environment may be overridden.

Token-based Environment Override provides the ability to select a token that will resolve to the overriding Environment. The fields for **Token-based Environment Override** are shown in *Figure B-14* and defined in *Table B-15*.



Validation : Untitled2

Name:

Description:

Enabled: ☑ Use in Workflow? □

Component Type: File Chooser

Base File Name Only: □

Environment Override Behavior: Token-based Environment Override

Environment Token:

Overriding Server Basepath:

Overriding Client Basepath:

Figure B-14 Validation Window for Token-Based Environment Override in File Chooser.

Figure B-15 Token-Based Environment Override

Field	Definition
Environment Token	Select the token that will resolve to the overriding Environment.
Overriding Server Basepath	The server basepath of the Environment that is to be resolved by the token may be overridden.
Overriding Client Basepath	The client basepath of the Environment that is to be resolved by the token may be overridden.

Date Field Formats

Date fields can accept a variety of formats. The current date field Validations are separated into two categories: all systems, and systems using only the English language. These formats are defined in Table 3-14.

Table 10-12. Date Field

Field		Definitions
Name	Systems	
Date Format	All	The format for the date part of the field. Choices are: · Long - “January 2, 1999". · Medium - “02-Jan-99”. · Short - “1/2/99”. · None - no date is displayed.
Date Format	English Only	The format for the date part of the field. Choices are: · MM/DD/YY (6/16/99). · DD-MON-YY (16-Jun-99). · MONTH DD, YYYY (June 16, 1999). · Day, Month DD, YYYY (Monday, June 16, 1999). · DD-MON (16-JUN) - Defaults to current year. · DD-MON-YYYY (16-JUN-1999). · MM-DD-YYYY (06-16-1999). · MM-DD-YY (06-16-99). · DD [Defaults to the current month and year]. · MM/DD (06/16) - Defaults to current year. · MM/DD/YYYY (06/16/1999).

Table 10-12. Date Field

Field		Definitions
Name	Systems	
Time Format	All	The format for the time part of the field. Choices are: · Long - the time is displayed as “12:00:00 PM PST”. · Medium - the time is displayed as “12:00:00 PM”. · Short - the time is displayed as “12:00 PM”. · None - no time is displayed.

Creating 1800 Character Text Areas

Standard Text Areas are either 40 or 200 characters. You can, however, create a Text Area Validation with a character length of 1800.

To create a Validation with a character length of 1800:

1. Open the Validation Workbench.

2. Search for “**Text Area - 1800**.”

3. In the results tab, select **Text Area - 1800.**

4. Click **Copy**.

5. Rename the Validation.

The new Text Area Validation (with a length of 1800) can be used when defining a custom field in the product.

Note

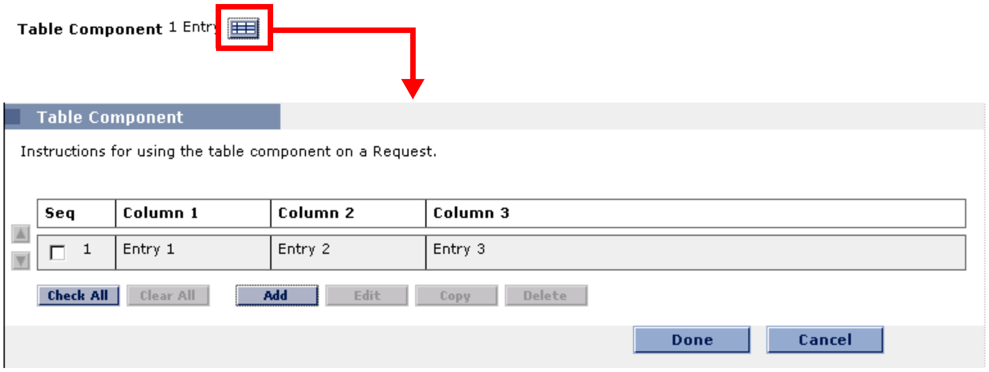
You can only create a Text Field or Area of length 40, 200, or 1800.

Configuring the Table Component

The table component is used to enter multiple records into a single field on a Request. The table component can be configured to include multiple columns of varied data types. Additionally, this component supports rules for populating elements within the table and provides functionality for capturing column totals.

1. Click the Table Component icon to open the Table Component entry page.

2. Add, edit, or delete entries in the list.



Instructions for using the table component on a Request.

Seq	Column 1	Column 2	Column 3
□ 1	Entry 1	Entry 2	Entry 3

Fields of this component can only be added to Request Types, Request Header Types and Request User Data.

To configure and use a Table Component:

1. *Define the Table Component in the Validation Workbench*

2. *Add the Table Component to a Request Type*

Example

ACME creates a Request Type to request quotes and parts for hardware. Each entry of this type has four elements: Part, Sub-Type, Part Number, and Unit Price. ACME creates a Table Component field called Hardware Information to collect this information.

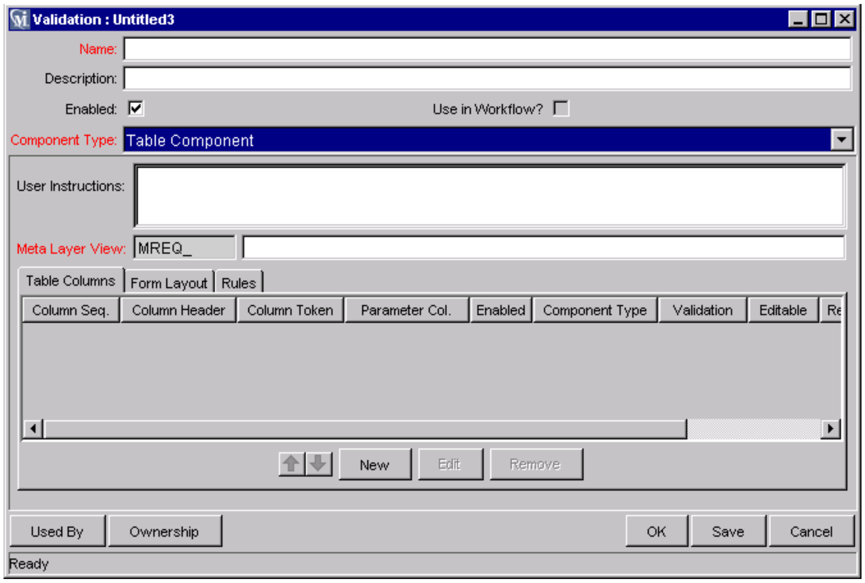
When the user logs a request for new hardware, the Request displays the Hardware Information field. The user opens the field. He selects a Part, which triggers a rule to populate the Part Number and Unit Price. He submits the Request, which now contains all of the information required to successfully order the hardware.

Define the Table Component in the Validation Workbench

To create a Table Component field:

1. Open the Validation Workbench in the Configuration screen group.
2. Click **New Validation**.

   The Validation window opens.
3. Select **Table Component** from the Component Type drop down list.



Validation : Untitled3

Name:

Description:

Enabled: ☑ Use in Workflow? □

Component Type: Table Component

User Instructions:

Meta Layer View: MREQ_

4. Enter a Validation Name and Description.
5. Enter any User Instructions.

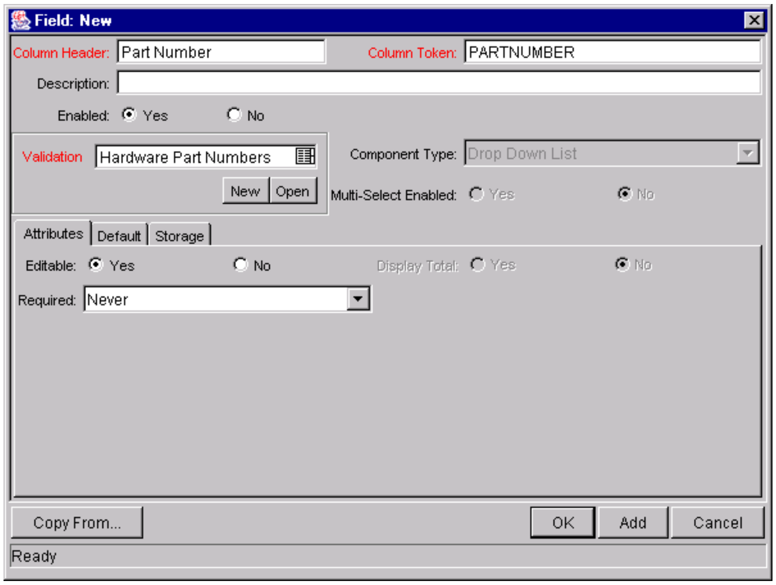
   This text will appear on the top of the table entry page.
6. Create the Table Columns.

   a. Click **New** in the **Table Columns** tab. The Field window opens.

   b. Define the type of information that will be stored in that column’s entries. This may require you to create a new Validation for the column.

Note

File attachments can not be used in a Table component column.



Field: New

Column Header: Part Number

Column Token: PARTNUMBER

Description:

Enabled: ● Yes ○ No

Validation: Hardware Part Numbers

Component Type: Drop Down List

Multi-Select Enabled: ○ Yes ● No

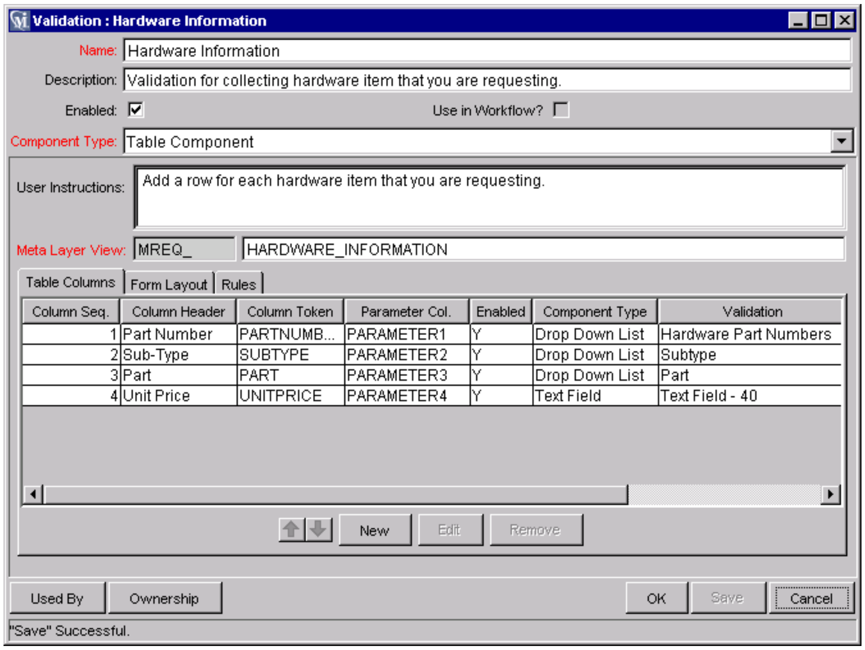
Editable: ● Yes ○ No

Display Total: ○ Yes ● No

Required: Never

c. Specify the **Attributes** (Editable or Required) and any **Default** behavior.

d. Click **Add** to save the column information and add another column. When you are finished adding columns, click **OK** to close the Field window.



Name: Hardware Information

Description: Validation for collecting hardware item that you are requesting.

Enabled: ☑ Use in Workflow? □

Component Type: Table Component

User Instructions: Add a row for each hardware item that you are requesting.

Meta Layer View: MREQ_ HARDWARE_INFORMATION

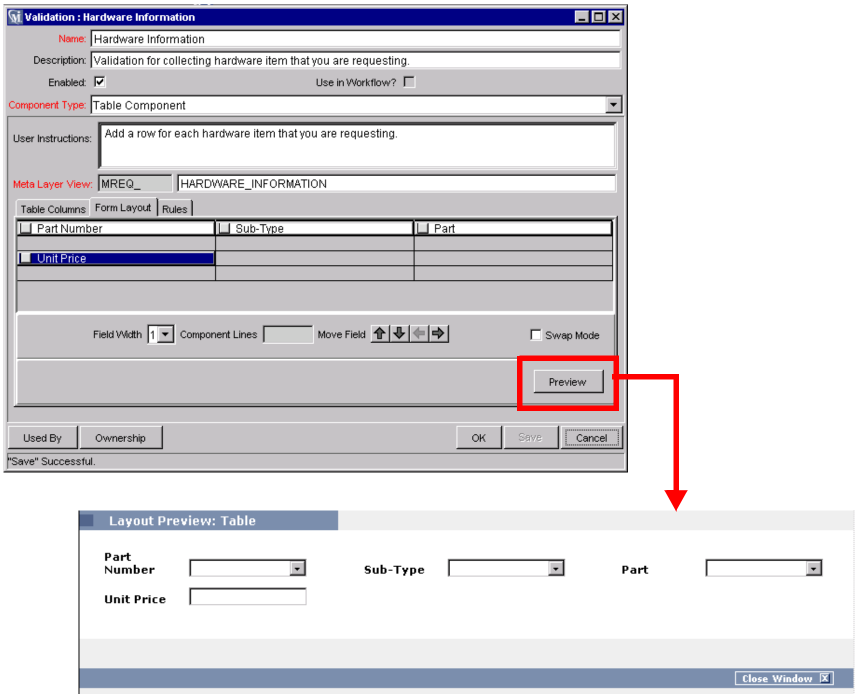
Column Seq.	Column Header	Column Token	Parameter Col.	Enabled	Component Type	Validation
1	Part Number	PARTNUMB...	PARAMETER1	Y	Drop Down List	Hardware Part Numbers
2	Sub-Type	SUBTYPE	PARAMETER2	Y	Drop Down List	Subtype
3	Part	PART	PARAMETER3	Y	Drop Down List	Part
4	Unit Price	UNITPRICE	PARAMETER4	Y	Text Field	Text Field - 40

"Save" Successful.

7. Configure the Form Layout.

   a. Click the **Form Layout** tab.

   b. Select the fields and move their positions using the arrow buttons.



Validation : Hardware Information

Name: Hardware Information

Description: Validation for collecting hardware item that you are requesting.

Enabled: ☑ Use in Workflow? □

Component Type: Table Component

User Instructions: Add a row for each hardware item that you are requesting.

Meta Layer View: MREQ_ HARDWARE_INFORMATION

Field Width 1 Component Lines Move Field □ Swap Mode

"Save" Successful.

Layout Preview: Table

c. Click Preview to see a representation of the final positioning.

   Note that the Preview loads a window in the Workbench, but the actual table component will only be available to users in the standard interface (HTML).

8. Configure any Table logic in the **Rules** tab.

   Rules are used for advanced defaulting behavior and calculating column totals.

   a. Click the **Rules** tab.

   b. Click **New** to define a new rule.

      See *"Creating a Table Rule"* on page 344 for detailed instructions.

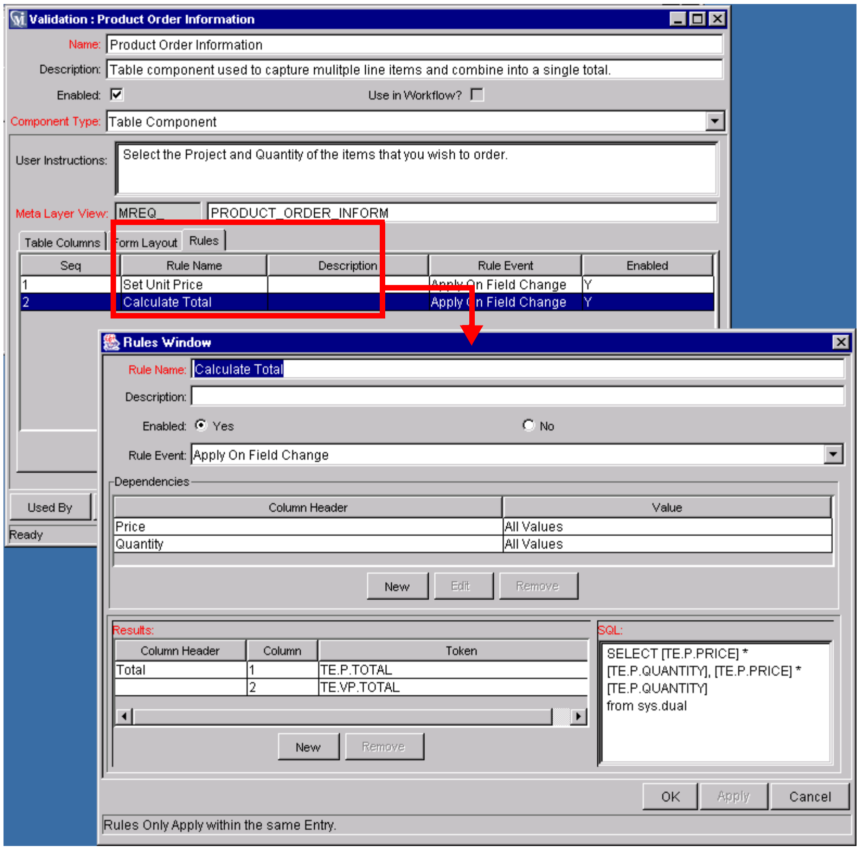
9. Click **OK** to save the Validation.

The new Table Component field can be included on a Request Type, Request Header Type or Request User Data field.

Creating a Table Rule

Table rules are configured in the same manner as advanced Request Type rules. Essentially, you can configure fields (columns) in the table to default to certain values based on an event or value in another field in the table. Because the table component rules are configured using a SQL statement, you are given enormous flexibility for the data that is populated in the table cells.

Table rules are configured using the **Rules** tab on the Validation window.



Name: Product Order Information

Description: Table component used to capture mulitple line items and combine into a single total.

Enabled: ☑ Use in Workflow? □

Component Type: Table Component

User Instructions: Select the Project and Quantity of the items that you wish to order.

Meta Layer View: MREQ_ PRODUCT_ORDER_INFORM

Seq	Rule Name	Description	Rule Event	Enabled
1	Set Unit Price		Apply On Field Change	Y
2	Calculate Total		Apply On Field Change	Y

Rule Name: Calculate Total

Description:

Enabled: ● Yes ○ No

Rule Event: Apply On Field Change

Dependencies

Column Header	Value
Price	All Values
Quantity	All Values

Results:

Column Header	Column	Token
Total	1	TE.P.TOTAL
	2	TE.VP.TOTAL

SQL:

```
SELECT [TE.P.PRICE] *
[TE.P.QUANTITY], [TE.P.PRICE] *
[TE.P.QUANTITY]
from sys.dual
```

Rules Only Apply within the same Entry.

Figure B-16 Rules window accessed from the Rules tab

Example: Using a Table Component on an Order Form

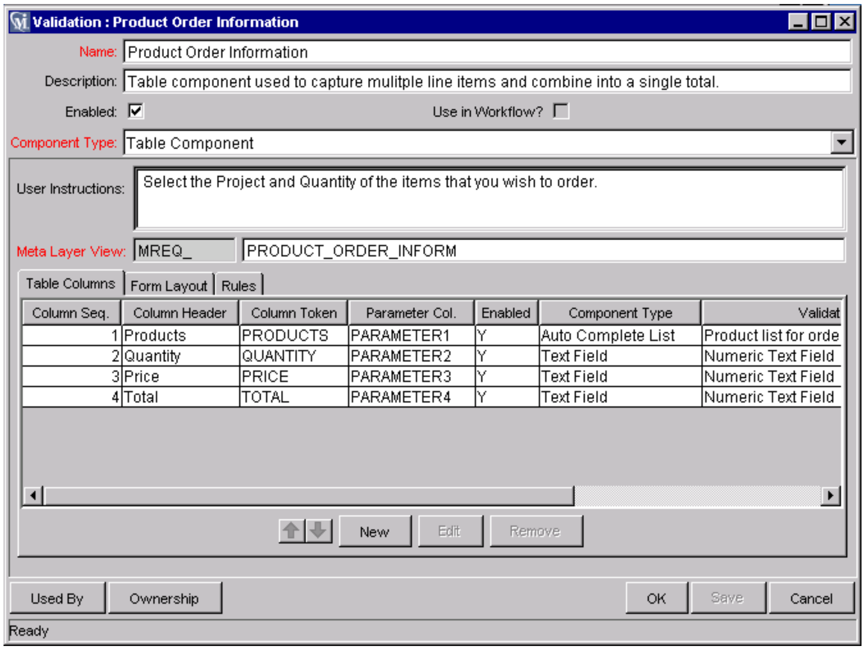
The following example illustrates the table component rules functionality.

ACME, Inc. uses a Request for creating and tracking employee computer hardware equipment orders. ACME has included a table component field on their Request Type for gathering the order information. When the employee selects a Product, the Unit Price is automatically updated. Then, when they update the Quantity, the total line cost is automatically calculated and displayed in the table.

To enable this functionality, ACME first has to configure a new Validation with the following specifications:

Table B-16. Example - Table Component Validation Settings

Setting	Value / Description
Validation Name	Product Order Information
Component Type	Table Component
Column 1	Column Header = Products Column Token = PRODUCTS Validation = Auto complete list with the following list values: PC, MOUSE, MONITOR, KEYBOARD
Column 2	Column Header = Quantity Column Token = QUANTITY Validation = Numeric Text Field
Column 3	Column Header = Price Column Token = PRICE Validation = Numeric Text Field
Column 4	Column Header = Total Column Token = TOTAL Validation = Numeric Text Field



Validation : Product Order Information

Name: Product Order Information

Description: Table component used to capture mulitple line items and combine into a single total.

Enabled: ☑ Use in Workflow? □

Component Type: Table Component

User Instructions: Select the Project and Quantity of the items that you wish to order.

Meta Layer View: MREQ_ PRODUCT_ORDER_INFORM

Column Seq.	Column Header	Column Token	Parameter Col.	Enabled	Component Type	Validat
1	Products	PRODUCTS	PARAMETER1	Y	Auto Complete List	Product list for orde
2	Quantity	QUANTITY	PARAMETER2	Y	Text Field	Numeric Text Field
3	Price	PRICE	PARAMETER3	Y	Text Field	Numeric Text Field
4	Total	TOTAL	PARAMETER4	Y	Text Field	Numeric Text Field

Ready

Once the Validation’s columns have been defined, the Rules can be configured:

Rule 1: Set Unit Price.

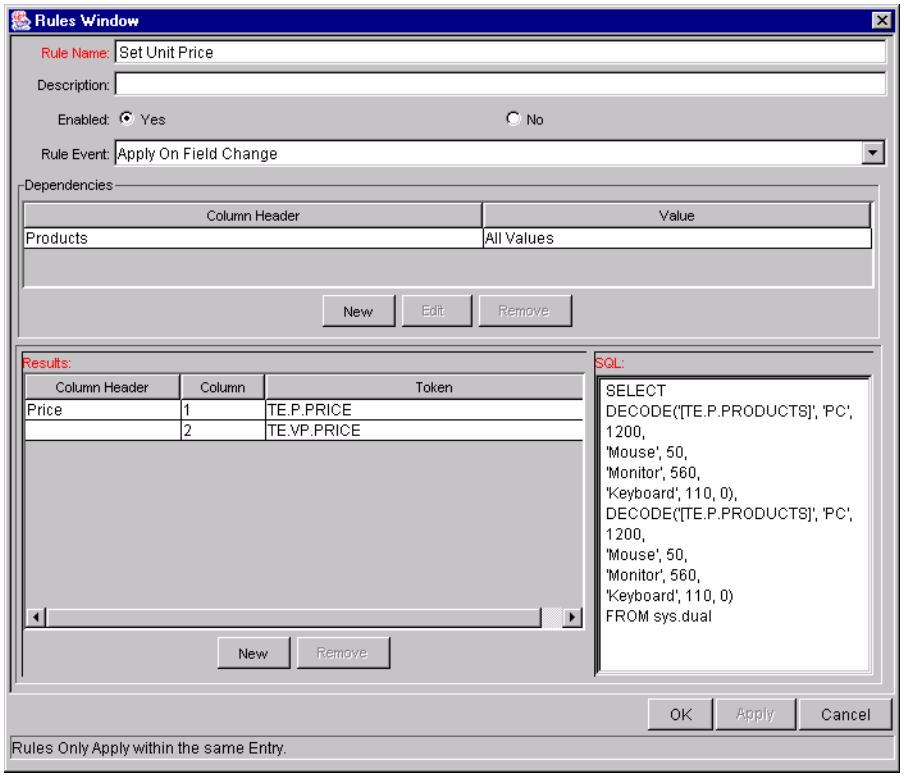
ACME uses the following rule to set the default unit price in the Price cell based on the Product selection.

Table B-17. Example - Set Unit Price Rule Settings

Setting	Value / Description
Rule Name	Set Unit Price
Rule Event	Apply on Field Change
Dependencies	Column = Products All Values = Yes
Results	Column Header = Price

Table B-17. Example - Set Unit Price Rule Settings

Setting	Value / Description
SQL	SELECT DECODE('[TE.P.PRODUCTS]', 'PC', 1200, 'Mouse', 50, 'Monitor', 560, 'Keyboard', 110, 0), DECODE('[TE.P.PRODUCTS]', 'PC', 1200, 'Mouse', 50, 'Monitor', 560, 'Keyboard', 110, 0) FROM sys.dual



Rule Name: Set Unit Price

Description:

Enabled: ● Yes ○ No

Rule Event: Apply On Field Change

Dependencies

Column Header	Value
Products	All Values

Results:

Column Header	Column	Token
Price	1	TE.P.PRICE
	2	TE.VP.PRICE

SQL:

```
SELECT
DECODE('[TE.P.PRODUCTS]', 'PC',
1200,
'Mouse', 50,
'Monitor', 560,
'Keyboard', 110, 0),
DECODE('[TE.P.PRODUCTS]', 'PC',
1200,
'Mouse', 50,
'Monitor', 560,
'Keyboard', 110, 0)
FROM sys.dual
```

Rules Only Apply within the same Entry.

Rule 2: Set Unit Price.

ACME uses the following rule to set the calculate and display the total line price in the Total column based on the values in the Products and Quantity cells.

Table B-18. Example - Calculate Total Rule Settings

Setting	Value / Description
Rule Name	Calculate Total
Rule Event	Apply on Field Change

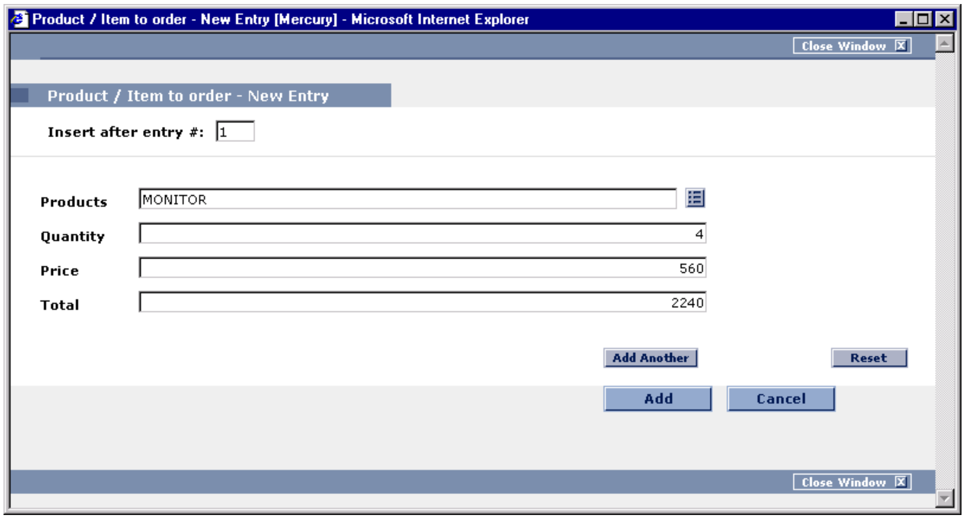
Dependencies	Column = Price [All Values = Yes]
	Column = Quantity [All Values = Yes]

Results	Column Header = Total

SQL	SELECT [TE.P.PRICE] * [TE.P.QUANTITY],
	[TE.P.PRICE] * [TE.P.QUANTITY]
	from sys.dual

Using the Table Component

Add a field to a Request Type that is validated by this Table Component Validation. When a user opens the field to enter information, the table rules will be applied to each row that is created.



Product / Item to order - New Entry [Mercury] - Microsoft Internet Explorer

Product / Item to order - New Entry

Insert after entry #: 1

Products: MONITOR

Quantity: 4

Price: 560

Total: 2240

Tokens in the Table Components

Each column included in the table component has an associated Token. These Tokens can be used in the same manner as other field tokens, such as for commands, notifications or advanced field defaulting. See *Commands and*

Tokens Guide and Reference for details on referencing Tokens related to Table Components.

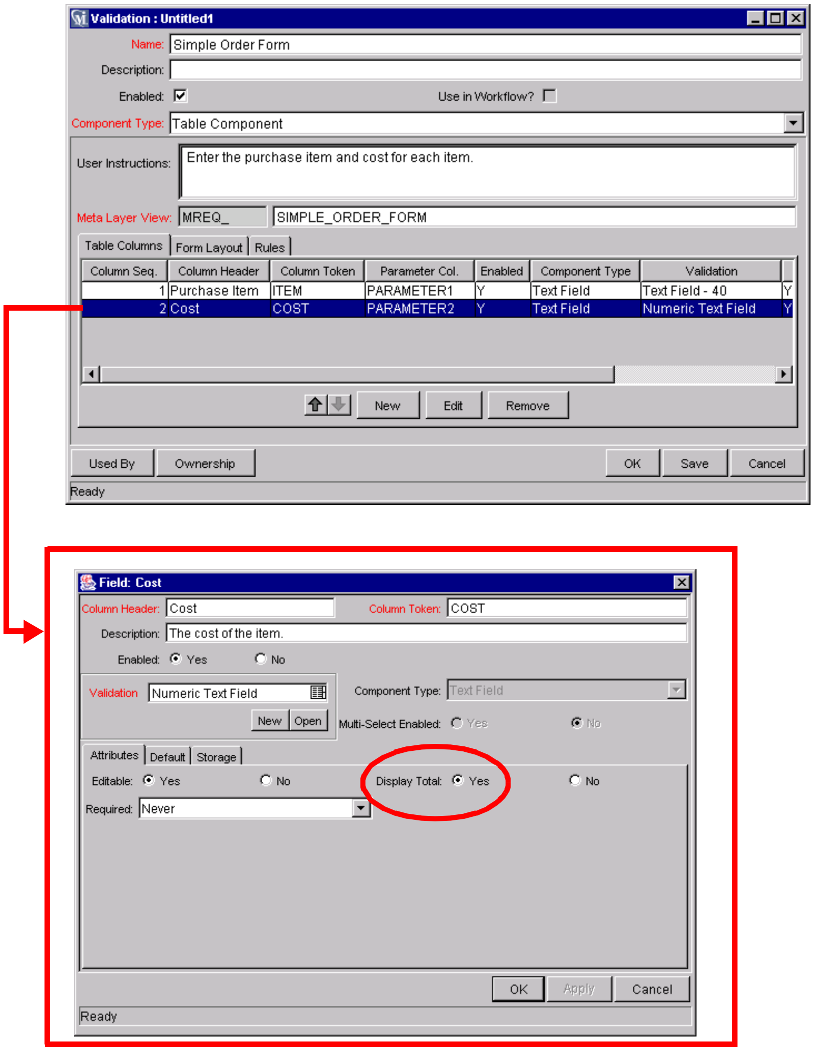
Calculating Column Totals

You can configure columns that are validated by a number to calculate the total for that column. This is configured in the Validation's Field window. The following example illustrates how to configure a column to calculate and display the column total.

ACME, Inc. uses a Request for creating and tracking simple employee equipment orders. ACME has included a table component field on their Request Type for gathering the order information. Employee enter the Purchase Items and Cost for each item. The table component automatically calculates the total cost for the Cost column.

ACME creates a Validation with the following settings:

- Component Type = **Table Component**.
- Column 1 = Purchase Item (text field)
- Column 2 = Cost (number). In the Field window for the Cost column, select Display Total = **Yes**. The Display Total field is only enabled if the field's validation is a number.



Name: Simple Order Form

Description:

Enabled: ☑ Use in Workflow? □

Component Type: Table Component

User Instructions: Enter the purchase item and cost for each item.

Meta Layer View: MREQ_ SIMPLE_ORDER_FORM

Column Seq.	Column Header	Column Token	Parameter Col.	Enabled	Component Type	Validation	
1	Purchase Item	ITEM	PARAMETER1	Y	Text Field	Text Field - 40	Y
2	Cost	COST	PARAMETER2	Y	Text Field	Numeric Text Field	Y

Ready

Column Header: Cost Column Token: COST

Description: The cost of the item.

Enabled: ● Yes ○ No

Validation: Numeric Text Field

Component Type: Text Field

Multi-Select Enabled: ○ Yes ● No

Editable: ● Yes ○ No

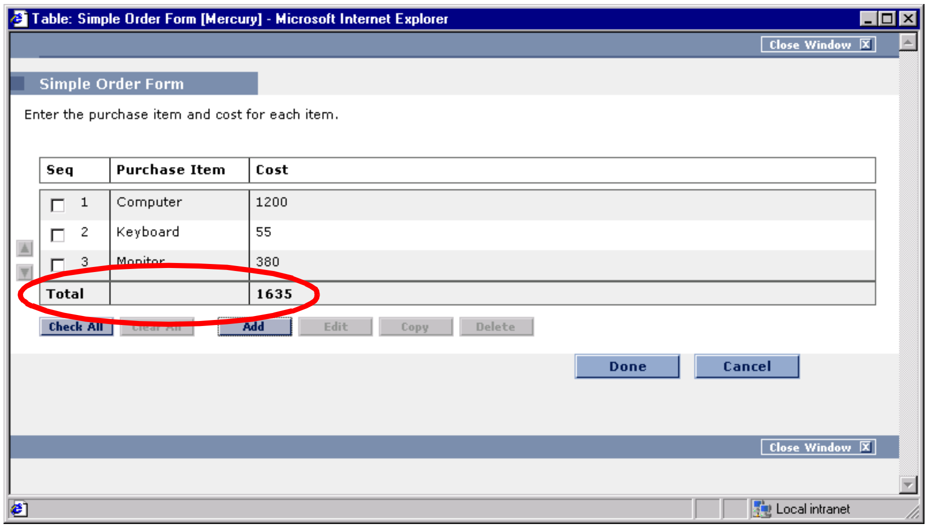
Display Total: ● Yes ○ No

Required: Never

Ready

Figure B-17 Sample validation for a Simple Order table component.

ACME includes adds a field to their Order Request Type that uses this Validation. When a user creates a Request using that Request Type, he can click on the table component icon next to the field to open the order form. The total for the Cost column is displayed at the bottom of the table.



Simple Order Form

Enter the purchase item and cost for each item.

Seq	Purchase Item	Cost
□ 1	Computer	1200
□ 2	Keyboard	55
□ 3	Monitor	380
Total		1635

Figure B-18 Sample table component displaying a column total.

Add the Table Component to a Request Type

Table Component fields can be included on a Request Type, Request Header Type or Request User Data field.

To add a Table Component field to a Request Type:

1. Open the Request Type window.

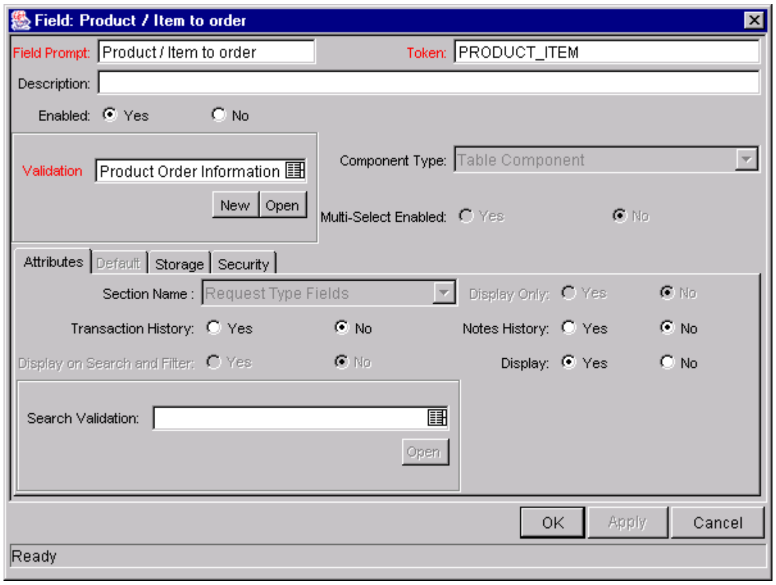
2. Click New in the **Fields** tab.

   The Field window opens.

3. Enter the Field Prompt, Token, and Description.

4. In the Validation field, select a table component Validation.

   If you have not created a table component Validation, click **New** to create one. See *"Define the Table Component in the Validation Workbench"* on page 340 for instructions.



Field Prompt: Product / Item to order

Token: PRODUCT_ITEM

Description:

Enabled: ● Yes ○ No

Validation: Product Order Information

Component Type: Table Component

Multi-Select Enabled: ○ Yes ● No

Section Name : Request Type Fields

Display Only: ○ Yes ● No

Transaction History: ○ Yes ● No

Notes History: ○ Yes ● No

Display on Search and Filter: ○ Yes ● No

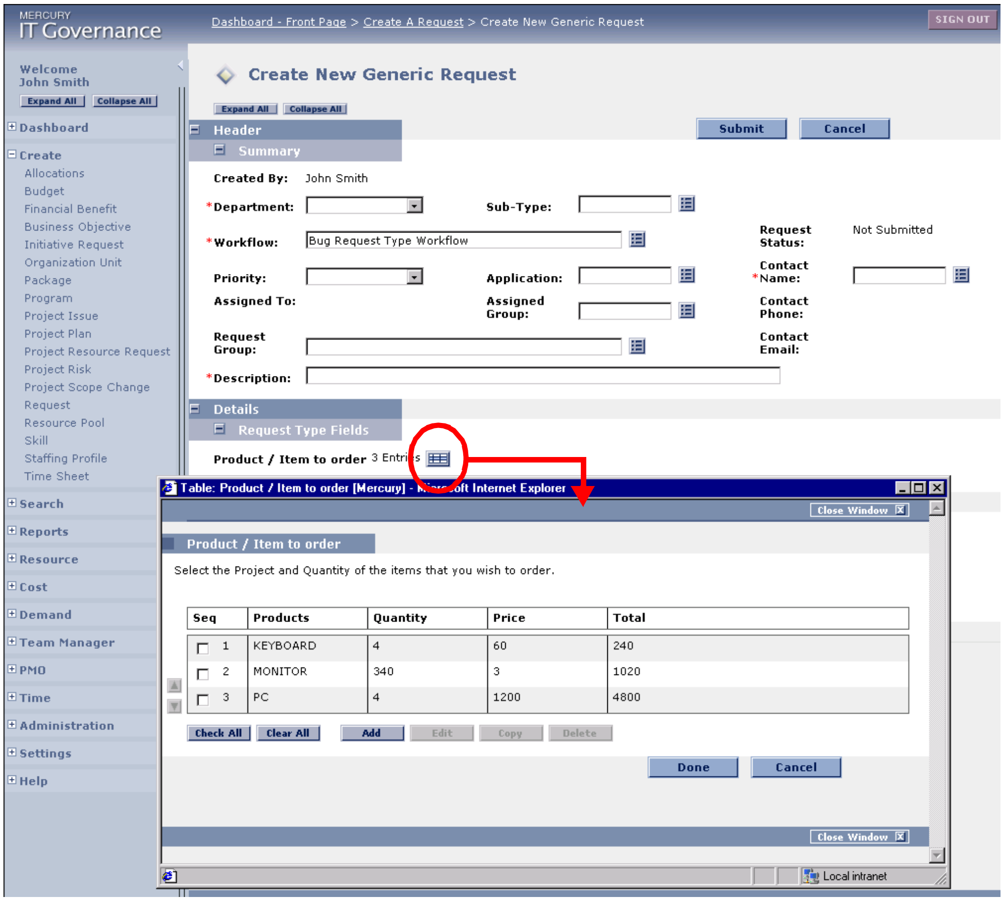
Display: ● Yes ○ No

Search Validation:

5. Click **OK** to add the field to the Request Type.

6. Save the Request Type.

The table component field will now appear on Requests of this Request Type.



Created By: John Smith

*Department:

Sub-Type:

*Workflow: Bug Request Type Workflow

Request Status: Not Submitted

Priority:

Application:

*Contact Name:

Assigned To:

Assigned Group:

Contact Phone:

Request Group:

Contact Email:

*Description:

Product / Item to order 3 Entries

Select the Project and Quantity of the items that you wish to order.

Seq	Products	Quantity	Price	Total
□ 1	KEYBOARD	4	60	240
□ 2	MONITOR	340	3	1020
□ 3	PC	4	1200	4800

Package and Request Group Validations

Two particular entity-specific Validations can be accessed in the Workbench without entering the Validations screen group:

- *Package and Request Groups*
- *Request Type Category*

Package and Request Groups

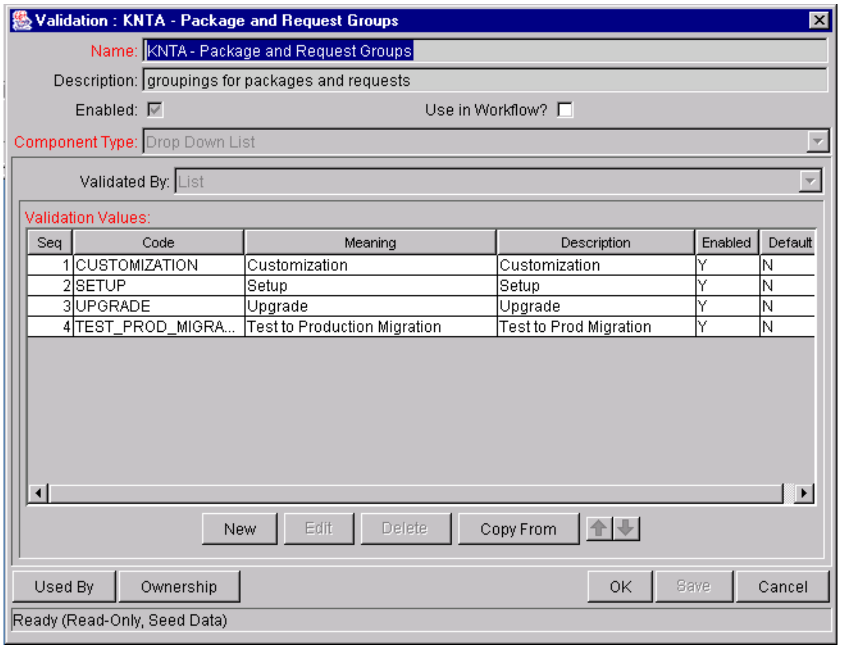
The KNTA-Package and Request Groups Validation can be accessed directly from the **Package** screen. To specify that a Package belongs to a new or unique Package Group that is not named in the auto-complete Validation list, it is not necessary to proceed through the Validation Workbench.

To access the KNTA-Package and Request Groups Validation window from the **Package** screen:

Select **New Package Group** from the **Package** menu. The Validation window will appear, listing the existing Mercury Change Management Package Groups.

Note

All users are granted read access to this screen, but only users with appropriate security privileges can alter the KNTA-Package and Request Groups Validation list.



Validation : KNTA - Package and Request Groups

Name: KNTA - Package and Request Groups

Description: groupings for packages and requests

Enabled: ☑ Use in Workflow? □

Component Type: Drop Down List

Validated By: List

Validation Values:

Seq	Code	Meaning	Description	Enabled	Default
1	CUSTOMIZATION	Customization	Customization	Y	N
2	SETUP	Setup	Setup	Y	N
3	UPGRADE	Upgrade	Upgrade	Y	N
4	TEST_PROD_MIGRA...	Test to Production Migration	Test to Prod Migration	Y	N

Ready (Read-Only, Seed Data)

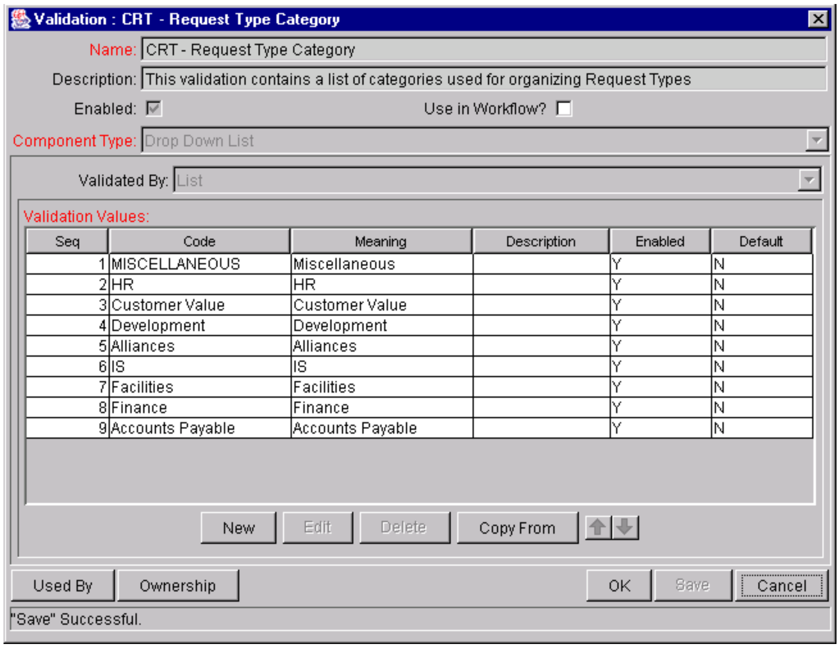
Request Type Category

The CRT - Request Type Category Validation can be accessed directly from the **Request Types** screen.

Access the CRT - Request Type Category Validation window from the **Request Types** screen by selecting **Request Type Category Setup** from the **Request Type** menu. The Validation window will appear, listing the existing Request Type Categories.

Note

All users are granted read access to this screen, but only users with appropriate security privileges can alter the CRT - Request Type Category Validation list.



Validation : CRT - Request Type Category

Name: CRT - Request Type Category

Description: This validation contains a list of categories used for organizing Request Types

Enabled: ☑ Use in Workflow? □

Component Type: Drop Down List

Validated By: List

Validation Values:

Seq	Code	Meaning	Description	Enabled	Default
1	MISCELLANEOUS	Miscellaneous		Y	N
2	HR	HR		Y	N
3	Customer Value	Customer Value		Y	N
4	Development	Development		Y	N
5	Alliances	Alliances		Y	N
6	IS	IS		Y	N
7	Facilities	Facilities		Y	N
8	Finance	Finance		Y	N
9	Accounts Payable	Accounts Payable		Y	N

"Save" Successful.

Validation Special Characters

The Validation Name field for all Validations cannot contain a question mark (‘?’). The Workbench prevents this character from being entered into the field, but all previously configured Validation Names (Validations entered before Kintana release 4.5) should be checked and corrected.

System Validations

There are a number of Validations that are provided with Mercury IT Governance Center. Note that many of these validations may have been altered to better match your company’s specific business needs. Use the Validations

Report to get a list of all validations currently in your system. The report includes information on Validation values and commands.

Appendix C Tokens

While configuring certain features, it is often necessary to reference information that is undefined until the product is actually used a particular context. Instead of generating objects that are valid only in specific contexts, you can create variable objects that can be applied to a variety of contexts. These variables are called Tokens.

There are two types of Tokens: custom Tokens and standard Tokens. Standard Tokens are provided with the product. Custom Tokens are generated to suit specific needs. Each field of the following entities can be referenced as a custom Token:

- Object Types
- Request Types and Request Header Types
- Report Types
- User Data
- Workflow Parameters

In addition, numerous standard Tokens are available that provide other useful pieces of information related to the system. For example, there is one Token that represents the users currently logged onto the system.

For instructions on using Tokens and for a list of available system Tokens, see *Commands and Tokens Guide and Reference*.

Appendix D User Data Creation and Processing

This appendix provides instructions for creating User Data fields for supported product entities. It also includes information on configuring advanced User Data field behavior and maintaining the User Data field definitions.

This appendix covers the following topics:

- *"User Data Overview"* on page 359
- *"Creating and Editing User Data"* on page 361
- *"Creating and Editing Context Sensitive User Data"* on page 375
- *"Project/Task User Data Roll-Up"* on page 388
- *"Referring to User Data"* on page 402
- *"Migrating User Data"* on page 403

User Data Overview

Product entities such as Packages, Workflows, Requests and Projects include a set of standard fields that provide information about those entities. While these fields are normally sufficient for day to day processing, User Data fields provide the ability to capture additional information specific to each organization.

For every major entity, up to twenty User Data fields can be defined. These fields are displayed in the **User Data** tab for the specific entity. The major attributes of each of these fields, such as their graphical presentation, the validation method, and whether or not they are required can be configured.

User Data fields are available for each entity instance generated; the fields are available globally. For example, you can configure a Manager field to appear on the **User Data** tab in the User window, and then specify each user's manager when setting up their account.

For some entities, context-sensitive custom fields can be set up. For example, User Data fields could be defined for the Request entities that are only available when the priority of a Request is **Critical**.

The following entities support User Data functionality:

- Budgets
- Organizations
- Resource Pools
- Staffing Profiles
- Packages
- Package Lines
- Environments
- Environment Application
- Environment Refresh
- Requests
- Request Types
- Projects
- Tasks
- Security Groups
- Users
- Validation Values
- Workflows
- Workflow Steps
- Workflow Executions
- Workflow Decisions

Creating and Editing User Data

The following sections provide detailed instructions for creating and editing User Data:

- *Creating User Data Fields*
- *Copying a Field's Definition*
- *Editing User Data Fields*
- *Configuring User Data Field Dependencies*
- *Removing Fields*
- *Modifying the User Data Layout*

Note

To configure User Data, you must have the **Config: Edit User Data** Access Grant.

Creating User Data Fields

To create a new User Data field:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

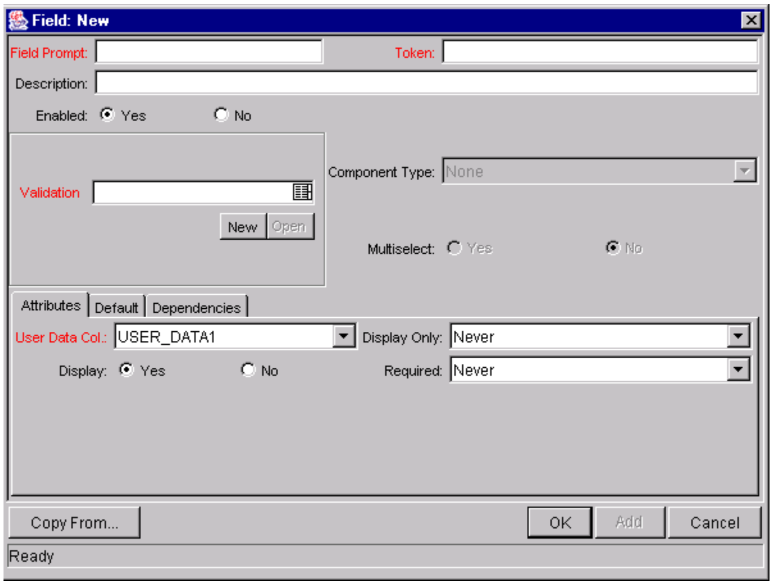
2. Search for the desired User Data Type from the **Query** tab and select it from the **Results** tab of the User Data Workbench.

3. Click **Open**.

   The User Data window opens.

4. Click **New** in the **Fields** tab.

   The Field: New window opens.



Field: New

Enabled: ● Yes ○ No

Component Type: None

Multiselect: ○ Yes ● No

User Data Col.: USER_DATA1

Display Only: Never

Display: ● Yes ○ No

Required: Never

Ready

5. Enter the general information region fields for the User Data field.

   This includes entering a name in the Field Prompt field and selecting a Validation.

6. Click the **Attributes** tab to define the field's basic properties. This includes entering information described in the following table.

Field	Description
User Data Col	Determines the internal column that the field value will be stored in. These values will then be stored in the corresponding column in the table for the given entity (such as KNTA_USERS for the Users entity). User Data provides the ability to store information in up to 20 columns, therefore allowing up to 20 fields. No two fields in User Data can use the same column.
Display Only	Determines whether the field is only displayed and cannot be updated. Select **Use Dependency Rules** to use the logic defined in the **Dependencies** tab.
Display	Determines if the user sees this field in the **User Data** tab.
Required	Determines if the user is required to specify a value for this field. Select **Use Dependency Rules** to use the logic defined in the **Dependencies** tab.

Field	Description
Workbench Only	Determines whether or not the field is seen in the standard (HTML) interface.
Multi-Select Enabled	Determines whether or not the field allows users to select more than one entry. Only valid for fields with an auto-complete component for the Validation.
Display in Search and Filter Pages	Determines whether or not the field will be displayed in Search and Filter pages in the standard interface.

7. Click the **Defaults** tab to define the default value for that field. This includes entering information described in the following table.

Field	Description
Default Type	Defines if the field will have a default value. Either default the field with a constant value or default it from the value in another User Data field.
Visible Value	If a Default Type of **Constant** is selected, the constant value can be entered here.
Depends On	To default from another field, choose the token name of that field. When using this User Data, every time a value is entered or updated in the source field, it will automatically be entered or updated in this destination field.

8. Click the **Dependencies** tab to define the field dependent properties of the field. This includes entering information described in the following table.

Field	Description
Clear When ___ Changes	Indicates that the current field should be cleared when the specified field changes.
Display Only When	Indicates that the current field should only be editable when certain logical criteria are satisfied. The field functions with two adjacent fields, a drop down list containing logical qualifiers, and a text field. To use this functionality, select **Use Dependency Rules** on the **Attributes** tab.

Field	Description
Required When	Indicates that the current field should be required when certain logical criteria are satisfied. The field functions with two adjacent fields, a drop down list containing logical qualifiers, and a text field. To use this functionality, select **Use Dependency Rules** on the **Attributes** tab.

9. Click the **Security** tab to define which users can view or update this field. Only a few User Data types support the use of the **Security** tab.

   a. On the Security tab, click Edit.

   b. In the Edit Field Security window define the field access. This includes entering information described in the following table.

Field / Button	Description
Visible to all users	Checking this checkbox allows all users to see the field. If this checkbox is not checked, you can set who can see the field. The default is for all users to be able to see a field. If this checkbox is not checked, the Select User/ Security Group that can view this field area is activated. De-selecting the **Visible to all users** or **Editable by all users** checkboxes enables the Select Users/Security Groups that can view this field area of the Edit Field Security window.
Editable by all users	Checking this checkbox allows all users to edit the field. If this checkbox is not checked, you can set who can edit the field. The default is for all user to be able to edit a field. De-selecting the **Visible to all users** or **Editable by all users** checkboxes enables the Select Users/Security Groups that can view this field area of the Edit Field Security window.

Field / Button	Description
Enter a Security Group (drop down list)	To select the format for specifying users to grant visibility and editability permission, use the **Enter a Security Group** drop down list. The drop down lists the formats to choose users. The drop down list dynamically updates the Security Group Validate autocomplete window list. The choices are: • **Enter a Username** – select a specific user to grant visibility and/or editability to the field. The user must have an email address. • **Enter a Security Group** – select a specific Security Group to grant visibility and/or editability to the field. • **Enter a Standard Token** – select a standard token to grant visibility and/or editability to the field. • **Enter a User Defined Token** – select a user defined token to grant visibility and/or editability to the field. Selecting the **Enter a User Defined Token** format enables the Tokens button. Selecting an item from the Enter a Security Group drop down list dynamically updates the Security Group field.
Security Group	Provides a field for specifying the recipient. If the Enter a Security Group drop down list is**:** • **Enter a Username** – then the Validate: Username window is returned. • **Enter a Security Group** – then the Validate: Security Group window is returned. • **Enter a Standard Token** – then the Validate: Standard Token window is returned. • **Enter a User Defined Token** – then the Validate: User Defined Token window is returned.

10. Click **OK** to add the User Data field to the entity.

Copying a Field's Definition

The **Copy From** functionality can also be utilized to streamline the process of adding Fields by copying the definition of existing Fields.

To copy a field's definition:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Search for the desired User Data Type from the **Query** tab and select it from the **Results** tab of the User Data Workbench.

3. Click **Open**.

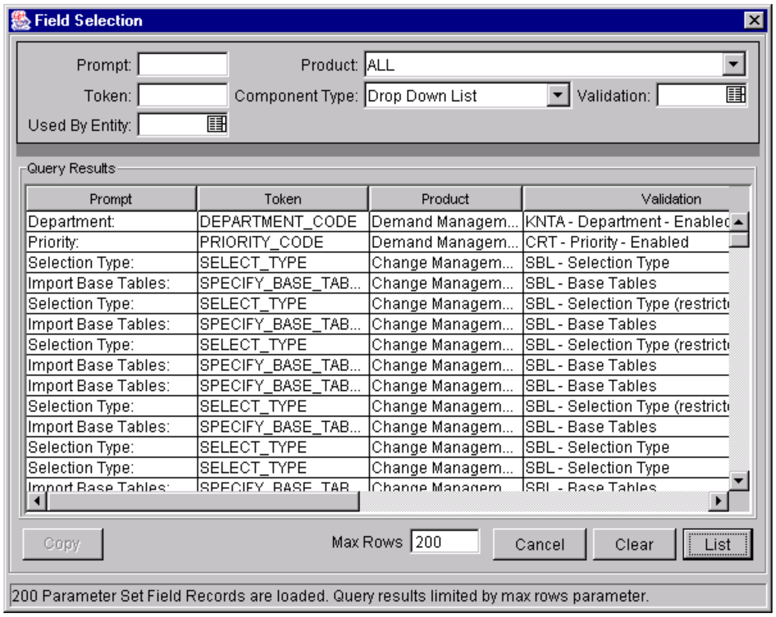
   The User Data window opens.

4. In the **Fields** tab, click **New**.

   The Field: New window opens.

5. Click **Copy From**.

   The Field Selection window opens.



Prompt:
Product: ALL
Token:
Component Type: Drop Down List
Validation:
Used By Entity:

Prompt	Token	Product	Validation
Department:	DEPARTMENT_CODE	Demand Managem...	KNTA - Department - Enablec
Priority:	PRIORITY_CODE	Demand Managem...	CRT - Priority - Enabled
Selection Type:	SELECT_TYPE	Change Managem...	SBL - Selection Type
Import Base Tables:	SPECIFY_BASE_TAB...	Change Managem...	SBL - Base Tables
Selection Type:	SELECT_TYPE	Change Managem...	SBL - Selection Type (restrict
Import Base Tables:	SPECIFY_BASE_TAB...	Change Managem...	SBL - Base Tables
Selection Type:	SELECT_TYPE	Change Managem...	SBL - Selection Type (restrict
Import Base Tables:	SPECIFY_BASE_TAB...	Change Managem...	SBL - Base Tables
Import Base Tables:	SPECIFY_BASE_TAB...	Change Managem...	SBL - Base Tables
Selection Type:	SELECT_TYPE	Change Managem...	SBL - Selection Type (restrict
Import Base Tables:	SPECIFY_BASE_TAB...	Change Managem...	SBL - Base Tables
Selection Type:	SELECT_TYPE	Change Managem...	SBL - Selection Type
Selection Type:	SELECT_TYPE	Change Managem...	SBL - Selection Type
Import Base Tables:	SPECIFY_BASE_TAB...	Change Managem...	SBL - Base Tables

Max Rows 200

200 Parameter Set Field Records are loaded. Query results limited by max rows parameter.

6. Search for a field to copy.

   Query fields by a number of criteria, such as the Token Name or field prompt. It is also possible to perform more complex queries such as listing all fields that reference a certain Validation or are used by a certain entity.

Note

Due to the large number of fields in the Mercury IT Governance Center, limit the list of fields by one or more of the query criteria.

7. Select the desired field.

8. Click **Copy**.

   This closes the window and copies the definition of the selected field into the Field: New window.

9. Make any necessary modifications and click **OK**.

Editing User Data Fields

To edit an existing field:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Search for the desired User Data Type from the **Query** tab and select it from the **Results** tab of the User Data Workbench.

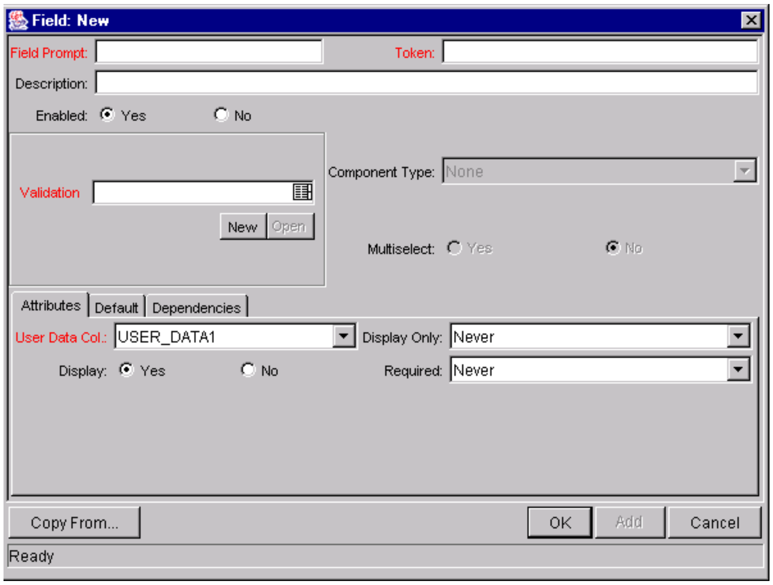
3. Click **Open**.

   The User Data window opens.

4. Select the field to edit.

5. Either double-click on the Field in the **Fields** tab or select the field and click **Edit**.

   The Field window opens.



Field Prompt:

Token:

Description:

Enabled: ● Yes ○ No

Component Type: None

Multiselect: ○ Yes ● No

User Data Col.: USER_DATA1

Display Only: Never

Display: ● Yes ○ No

Required: Never

6. Make the desired changes in the header region, **Attributes** tab, **Default** tab, and **Dependencies** tab.

7. Click **Apply** to save the changes to the **Fields** tab without closing the Field window, or click **OK** to save the changes and close the Field window.

The field has been updated with the changes.

Configuring User Data Field Dependencies

Field behavior and properties can be linked to the value of other fields defined for that entity.

Example

A Report Type field can become required when the value in another field in that Report Type is **Critical**.

A field can be configured to:

- Clear when another field changes.
- Become read only when another field meets a logical condition, defined in *Table 10-13*.

- Become required when another field meets a logical condition, defined in *Table 10-13*.

Table 10-13. Field Dependency Logical Qualifiers

Logical qualifier	Definition
like	A 'like' condition looks for close matches of the value to the contents of the field chosen.
not like	A 'not like' condition looks for contents in the selected field that are not close matches to the Value field.
is equal to	An 'is equal to' condition looks for an exact match of the Value to the contents of the Field chosen.
is not equal to	An 'is not equal to' condition is true when there are no results exactly matching the value of the field contents.
is null	An 'Is null' condition is true when the field selected is blank.
is not null	An 'Is not null' condition is true when the field selected is not blank.
is greater than	An 'Is greater than' condition looks for a numerical value larger than the value entered in the Value field.
is less than	An 'Is less than' condition looks for a numerical value below the value entered in the Value field.
is less than equal to	An 'Is less than equal to' condition looks for a numerical value below or the same as the value entered in the Value field.
is greater than equal to	An 'Is greater than equal to' condition looks for a numerical value larger than or the same as the value entered in the Value field.

To configure a User Data field dependency:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Search for the desired User Data Type from the **Query** tab and select it from the **Results** tab of the User Data Workbench.

3. Click **Open**.

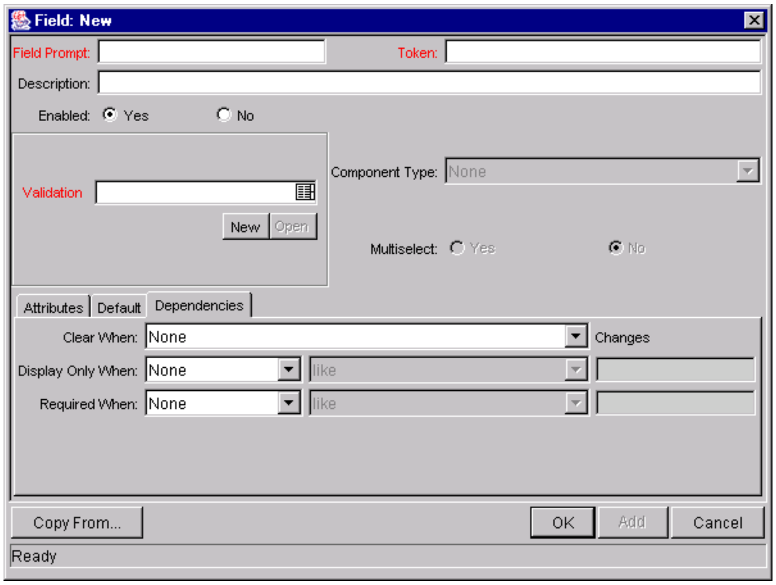
The User Data window opens.

4. Select the field you wish to edit.

5. Either double-click on the Field in the **Fields** tab or select the field and click **Edit**.

   The Field window opens.

6. Click the **Dependencies** tab.



Field: New

Field Prompt:

Token:

Description:

Enabled: ● Yes ○ No

Validation

Component Type: None

Multiselect: ○ Yes ● No

Clear When: None

Display Only When: None like

Required When: None like

Ready

7. Set the field dependencies. It is possible to:

   - Select a field name from the Clear When drop down list to indicate that the current field should be cleared when the selected field changes.

   - Select a field name from the Display Only When drop down list to indicate that the current field should for display only (for example, not editable) when certain logical criteria are satisfied. This field functions with two adjacent fields. These are a drop down list containing logical qualifier and another field which dynamically changes to a date field, drop down list, or text field, depending on the selected field's validation.

   - Select a field name from the Required When drop down list to indicate that the current field should be required when certain logical criteria are satisfied. This field functions with two adjacent fields. These are a drop down list containing logical qualifier and another field which dynamically changes to a date field, drop down list, or text field, depending on the selected field's Validation.

8. Click **OK**.

Removing Fields

To remove a field permanently from a User Data Type:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Search for the desired User Data Type from the **Query** tab and select it from the **Results** tab of the User Data Workbench.

3. Click **Open**.

   The User Data window opens.

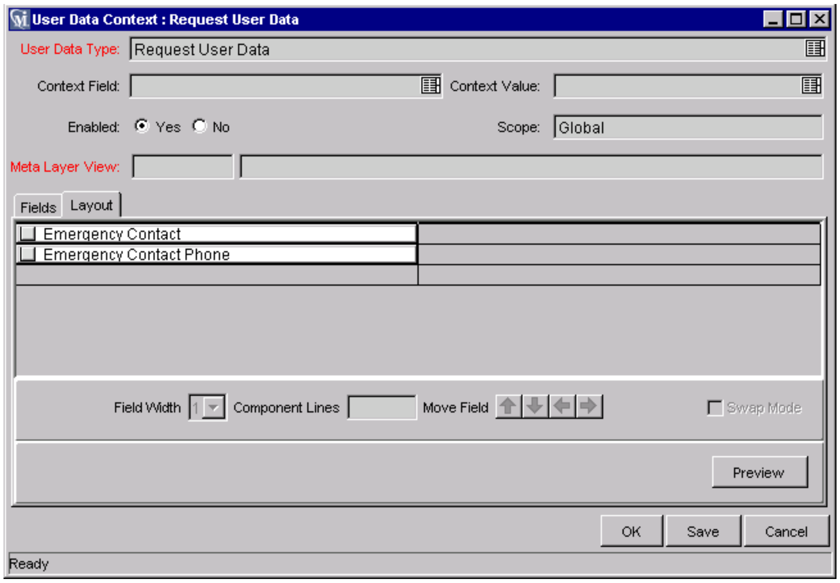
4. In the **Fields** tab, select the field.

5. Click **Remove**.

6. To save the change to the database and close the window, click **OK**.

Modifying the User Data Layout

The layout of User Data fields can be changed in the **Layout** tab of the User Data window.



User Data Context : Request User Data

User Data Type: Request User Data

Context Field:

Context Value:

Enabled: ● Yes ○ No

Scope: Global

Meta Layer View:

□ Emergency Contact

□ Emergency Contact Phone

Field Width 1

Component Lines

□ Swap Mode

Ready

Figure 10-3 User Data Window - Layout Tab

The following sections discuss modifying User Data field layout in more detail:

- *Changing Column Width*
- *Moving a Field*
- *Swapping Positions of Two Fields*

Changing Column Width

To change the column width of a Field:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Search for the desired User Data Type from the **Query** tab and select it from the **Results** tab of the User Data Workbench.

3. Click **Open**.

   The User Data window opens.

4. Click the **Layout** tab.

5. Select the Field.

6. The Field Width radio button, select either **1** or **2** in.

Note

The Layout editor will not allow changes to be made if it conflicts with another field in the layout (for example, a field's width cannot be changed from one to two if another field exists in column two on the same row).

Additionally, for fields of component type **Text Area**, it is possible to determine the number of lines the text area will display. Select the **Text Area** type field and change the value in the Component Lines attribute. If the selected field is not of type **Text Area**, this attribute will be blank and non-updateable.

Moving a Field

To move a field or a set of fields:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Search for the desired User Data Type from the **Query** tab and select it from the **Results** tab of the User Data Workbench.

3. Click **Open**.

   The User Data window opens.

4. Click the **Layout** tab.

5. Select the field(s).

   To select more than one field, press the Shift key while selecting the last field in a set. It is only possible to select a continuous set of fields.

6. Use the directional arrow buttons to move the fields to the desired location in the layout builder.

Note

A field, or a set of fields, cannot be moved to an area where other fields already exist. Those other fields must be moved out of the way first.

Swapping Positions of Two Fields

To swap the positions of two fields:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Search for the desired User Data Type from the **Query** tab and select it from the **Results** tab of the User Data Workbench.

3. Click **Open**.

   The User Data window opens.

4. Click the **Layout** tab.

5. Select the first field.

6. Select the Swap Mode check box.

   This causes an **S** to appear in the check box area of the selected field.

7. Once the **S** appears, double-click on the field to be swapped with.

   This causes the two fields to change positions. Following the swap, the Swap Mode is turned off.

To swap another set of fields, repeat the above procedure.

Previewing the Layout

To check what the layout will look like in actual use, click **Preview**. This opens a small window that shows the fields as they will appear in the window, shown in *Figure 10-4*.

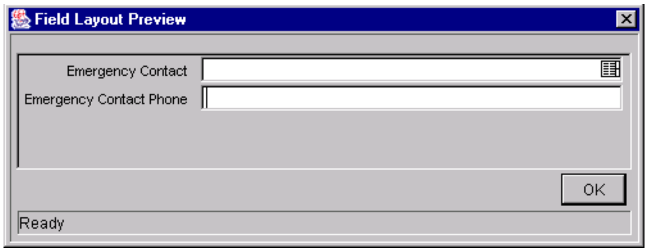


Figure 10-4 Preview Mode

Note

If all fields have a width of one column, all displayed columns will automatically span the entire available area when an entity of the given User Data is being viewed or generated.

Non-displayed fields do not affect the layout. The layout engine considers them the same as a blank field.

Creating and Editing Context Sensitive User Data

Certain User Data Types support an additional level of detail. The fields displayed in the entity's **User Data** tab can be dynamically linked to a field value in the information region of the respective entity's window. Context Sensitivity can be used with the following User Data Types:

- Package
- Request
- Validation Value

For example, if **Critical** was selected from the drop down list for the Priority field of a Request, then the Assigned To field could be automatically set to one of the top level support personnel, such as the manager of the technical support team.

The following sections provide detailed instructions for creating and editing Context Sensitive User Data:

- *Creating Context Sensitive User Data*
- *Editing Context Sensitive User Data*
- *Deleting Context Sensitive User Data*
- *Copying Context Sensitive User Data*
- *Example - Using Context Sensitive User Data for a Field in a Request Header Type*

Creating Context Sensitive User Data

Context Sensitive User Data can be defined for the Request, Package, and Validations (Validation value region) windows in the User Data Workbench.

To define Context Sensitive User Data:

1. Define a Context Field in the Global User Data scope.

   See *"Defining the Context Field"* on page 376.

2. Define a Context Value.

   See *"Defining a Context Value"* on page 378.

3. Define and configure the fields which appear under certain contexts.

   See *"Defining the Context Sensitive Fields"* on page 378.

Note

When defining or editing Context Sensitive User Data fields for the same User Data Type, be sure to save any changes to the Global User Data fields before working on the Context User Data fields.

Defining the Context Field

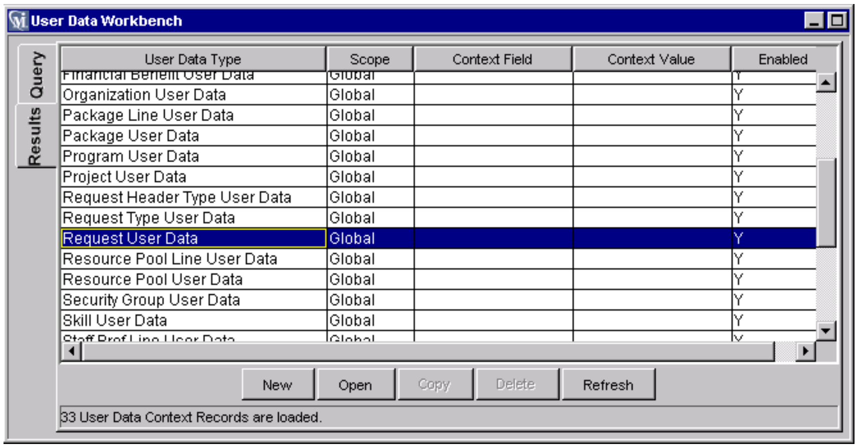
Only one field can be defined as the Context Field at any given time.

To specify the Context Field:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Click **List** to display all of the existing User Data Types.



User Data Type	Scope	Context Field	Context Value	Enabled
Financial Benefit User Data	Global			Y
Organization User Data	Global			Y
Package Line User Data	Global			Y
Package User Data	Global			Y
Program User Data	Global			Y
Project User Data	Global			Y
Request Header Type User Data	Global			Y
Request Type User Data	Global			Y
Request User Data	Global			Y
Resource Pool Line User Data	Global			Y
Resource Pool User Data	Global			Y
Security Group User Data	Global			Y
Skill User Data	Global			Y
Staff Prof Line User Data	Global			Y

33 User Data Context Records are loaded.

3. Select the desired User Data Type (Package, Request or Validation) with a **Global** scope.

4. Click **Open**.

   The User Data Context window opens.

5. Select the Context Field from the auto-complete list.

> Note
>
> The Context Field is disabled if any specific contexts for the User Data has been defined. In order to change the Context Field, all specific contexts for the User Data Type must first be deleted. For more information on editing Context Sensitive User Data, see *"Editing Context Sensitive Fields"* on page 380.

6. Verify that the global context is enabled (Enabled=**Yes**).

7. To save and close the window, click **OK**.

The Context Field has now been specified.

> Note
>
> The Context Field for the Validations Value User Data Type is always **Validation Name**.

Defining a Context Value

Context Values are the predefined possible values for the selected Context Field. Different User Data fields can be defined to be displayed for each of the possible Context Values.

To define a Context Value:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Click **New** in the **Results** tab or **New User Data Context** in the **Query** tab.

   The User Data Context window opens.

3. Select a User Data Type from the auto-complete list.

   The Context Field is displayed as read-only.

> Note: Only User Data Types with a defined Context Field appear in the list. To define a Context Field for Request, Package, or Validation Values, see *"Defining the Context Field"* on page 376.

4. Select a Context Value from the drop down list or auto-complete list.

Defining the Context Sensitive Fields

User Data fields to be used with the specified Context Value are defined and configured just as in other areas of the product. For details on defining the field content and layout, see one of the following sections:

- *"Creating User Data Fields"* on page 361
- *"Editing User Data Fields"* on page 367
- *"Removing Fields"* on page 371

To define different fields based on a different Context Value, see *"Defining a Context Value"* on page 378.

Editing Context Sensitive User Data

Context Sensitive User Data can be edited for the Request, Package, Environment, and Validations (Validation value region) windows in the User Data Workbench. For details on editing Context Sensitive User Data, see one of the following sections:

- *Changing the Context Field*
- *Changing the Context Value*
- *Editing Context Sensitive Fields*

Changing the Context Field

In order to change the Context Field, all specific contexts for the User Data Type must first be deleted.

To change the Context Field for a particular User Data Type:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. To display all of the existing User Data Types, click **List**.

3. Locate the desired User Data Type.

4. Select the rows where the desired User Data Type's Scope=**Context**.

5. Click **Delete.**

6. Select the desired User Data Type.

   It will have a **Global** Scope.

7. Click **Open**.

   The User Data Context window opens.

8. From the auto-complete list, select the Context Field.

9. Verify that the Global Context is enabled (Enabled=**Yes**).

10. To save and close the window, click **OK**.

The User Data Type's Context Field has now been changed.

Note

> The Context Field for the Validations Value User Data Type is always **Validation Name** and cannot be changed.

Changing the Context Value

To change an existing User Data Type's Context Value:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Click **List** to display all of the existing User Data Types.

3. Select the desired User Data Type that has the Context Value to be changed.

4. Click **Open**.

   The User Data Context window opens.

5. From the drop down list or auto-complete list, select a new Context Value.

6. Click **OK** to save the changes and close the window.

The User Data's Context Value has been changed.

Editing Context Sensitive Fields

User Data fields to be used with the specified Context Value are edited just as in other areas of the product. For details on editing the field content and layout, see one of the following sections:

- *"Creating User Data Fields"* on page 361
- *"Editing User Data Fields"* on page 367
- *"Removing Fields"* on page 371

Deleting Context Sensitive User Data

To delete a Context Sensitive User Data Type:

1. Click the **Configuration** screen group and click the **User Data** screen.

The User Data Workbench opens.

2. Click **List** to display all of the existing User Data Types.
3. Select the User Data Type to be deleted.
4. Click **Delete**.

   A Question dialog opens with the message “Delete 1 User Data Context[s]?”

5. Click **Yes** to confirm deletion.

Copying Context Sensitive User Data

To copy a Context Sensitive User Data Type:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Click **List** to display all of the existing User Data Types.
3. Select the User Data Type with Scope=**Context** that is to be copied.
4. Click **Copy**.

   The Copy User Data window opens.

5. From the New Context Value list, select the new value.
6. Click **OK**.

   A Question dialog opens.

7. Click **Yes** to edit the context sensitive fields or **No** to accept.

Example - Using Context Sensitive User Data for a Field in a Request Header Type

Different values can appear in the Request’s Applications field depending on which Request Header Type is used. For the Applications field in an ERP Request, the following distinct set of fields are available:

- Accounts Receivable
- Accounts Payable
- General Ledger
- Inventory

For an eCommerce Request, the following fields are available:

- Registration
- User Preferences
- Order Entry
- Order Tracking

Note

Changing the Validation of a field in a Request Header Type can affect how information is returned from queries and reports. Use a context sensitive User Data approach when setting up such a system.

The following procedure provides an example for setting up the Applications Validation for the ERP Request introduced above:

Setting Up the Context Sensitive User Data

First, the Context Sensitive User Data must be configured.

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.

2. Click **New User Data Context**.

   The User Data Context window opens.

3. From the User Data Type auto-complete list, select **Validation Value User Data**.

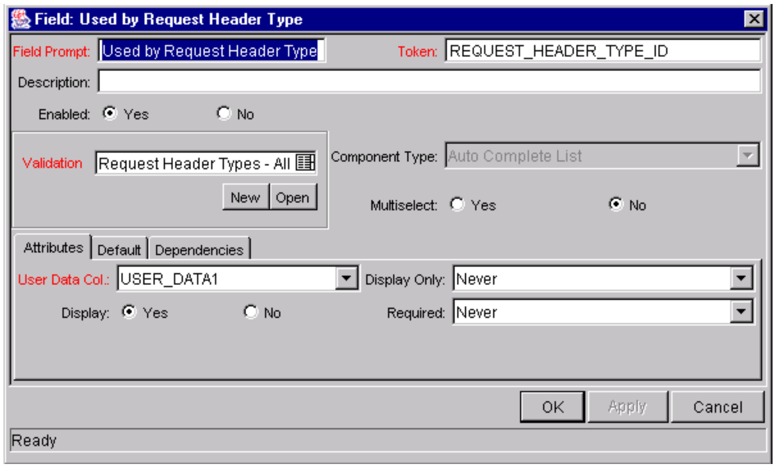
4. From the Context Value auto-complete list, select **KNTA - Application - Enabled**.

5. Click **New**.

   The Field: New window opens.

6. Create a new field with the following specifications:

- Field Prompt = **Used by Request Header Type**
- Token = **REQUEST_HEADER_TYPE_ID**
- Validation = **CRT - Request Header Types - All**



Field: Used by Request Header Type

Field Prompt: Used by Request Header Type

Token: REQUEST_HEADER_TYPE_ID

Description:

Enabled: ● Yes ○ No

Validation: Request Header Types - All

Component Type: Auto Complete List

Multiselect: ○ Yes ● No

User Data Col.: USER_DATA1

Display Only: Never

Display: ● Yes ○ No

Required: Never

Ready

7. Click **OK**.

Example: Configuring the Validations

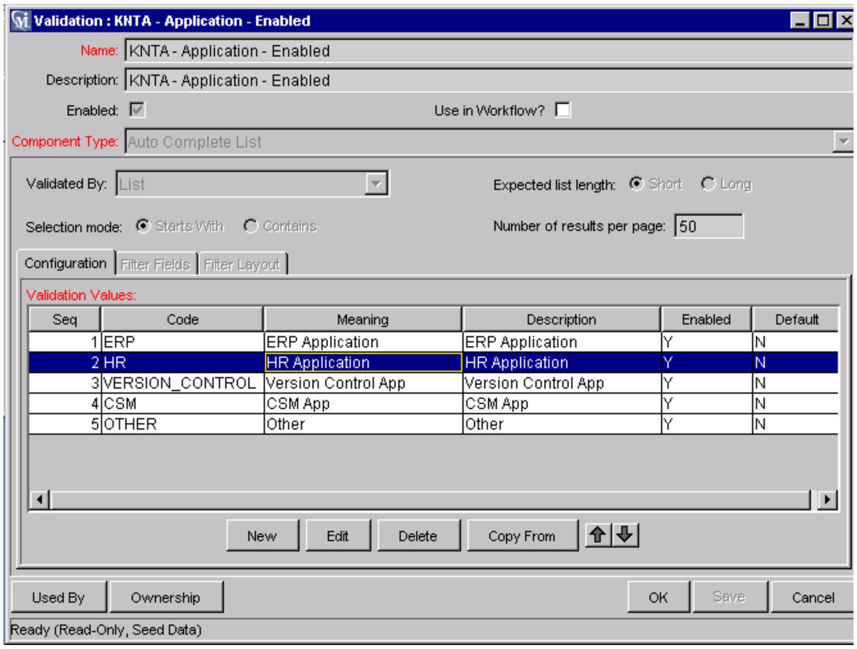
The Validation can now be configured:

1. Click the **Configuration** screen group and click the **Validations** screen.

   The Validation Workbench opens.

2. Search for and open the global KNTA - Application - Enabled Validation.

   The Validation window opens.



Validation : KNTA - Application - Enabled

Name: KNTA - Application - Enabled

Description: KNTA - Application - Enabled

Enabled: ☑ Use in Workflow? □

Component Type: Auto Complete List

Validated By: List

Expected list length: ● Short ○ Long

Selection mode: ● Starts With ○ Contains

Number of results per page: 50

Validation Values:

Seq	Code	Meaning	Description	Enabled	Default
1	ERP	ERP Application	ERP Application	Y	N
2	HR	HR Application	HR Application	Y	N
3	VERSION_CONTROL	Version Control App	Version Control App	Y	N
4	CSM	CSM App	CSM App	Y	N
5	OTHER	Other	Other	Y	N

Ready (Read-Only, Seed Data)

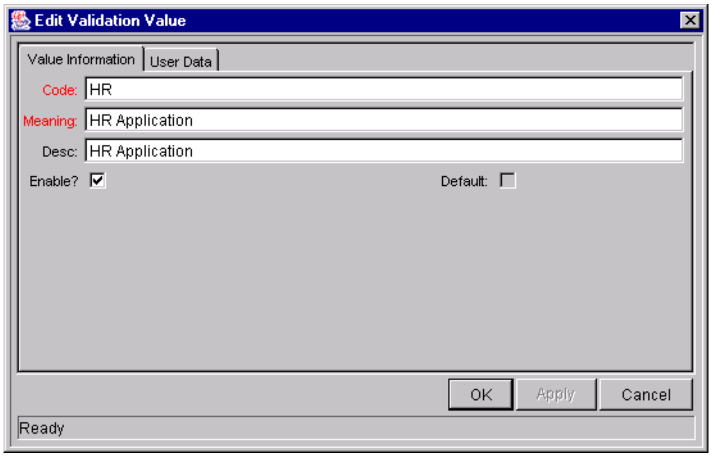
3. Associate each Validation Value with the appropriate Request Header Type shown in *Table 10-14*.

   To associate a Validation Value with a Request Header Type:

   a. Select a Validation Value from the Validation Values list.

   b. Click **Edit**.

      The Edit Validation Value window opens.



Edit Validation Value

Code: HR

Meaning: HR Application

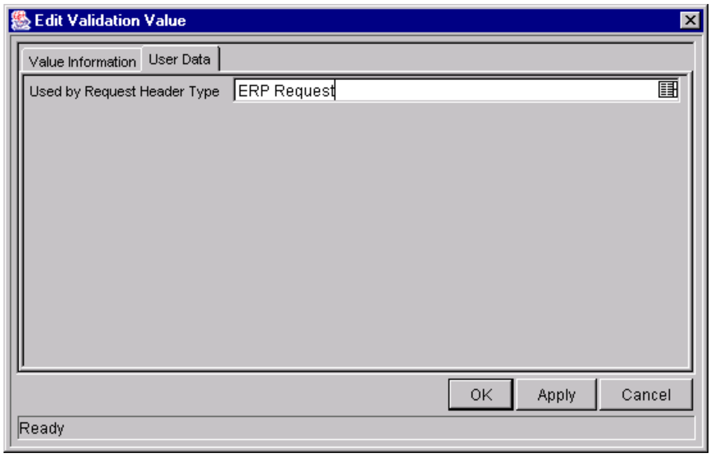
Desc: HR Application

Enable? ☑ Default: □

Ready

c. Click the **User Data** tab.

d. Select the Request Header Type (**ERP Request** in this example) from the Used by Request Header Type auto-complete list.



Used by Request Header Type: ERP Request

e. Click **OK**.

f. Repeat these steps for each row in *Table 10-14*.

Table 10-14. Validation values associated with Request Header Types

Validation Value	Used by Request Header Type
Accounts Receivable	ERP Request
Accounts Payable	ERP Request
General Ledger	ERP Request
Inventory	ERP Request
Registration	eCommerce Request
User Preferences	eCommerce Request
Order Entry	eCommerce Request
Order Tracking	eCommerce Request

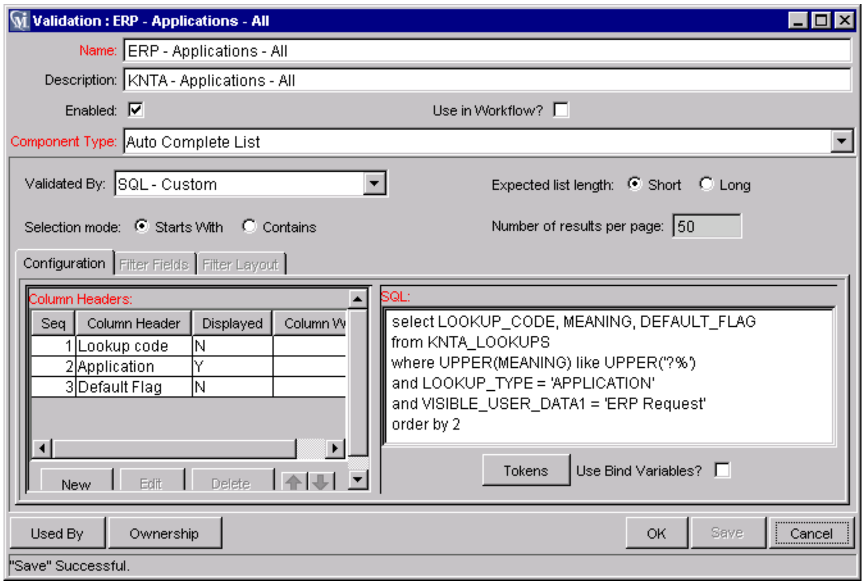
Example: Modifying the SQL

It is possible to now create variants of the standard Application Validation which vary depending on which Request Header Type is being used.

1. From the Validations Workbench, copy the KNTA - Application - All Validation.

2. Rename the Validation to ERP Applications - All.

3. Edit the Validation's SQL as shown below.



Name: ERP - Applications - All
Description: KNTA - Applications - All
Enabled: ☑ Use in Workflow? □
Component Type: Auto Complete List
Validated By: SQL - Custom
Expected list length: ● Short ○ Long
Selection mode: ● Starts With ○ Contains
Number of results per page: 50

Column Headers:

Seq	Column Header	Displayed	Column W
1	Lookup code	N	
2	Application	Y	
3	Default Flag	N	

SQL:

```
select LOOKUP_CODE, MEANING, DEFAULT_FLAG
from KNTA_LOOKUPS
where UPPER(MEANING) like UPPER('?%')
and LOOKUP_TYPE = 'APPLICATION'
and VISIBLE_USER_DATA1 = 'ERP Request'
order by 2
```

Use Bind Variables? □

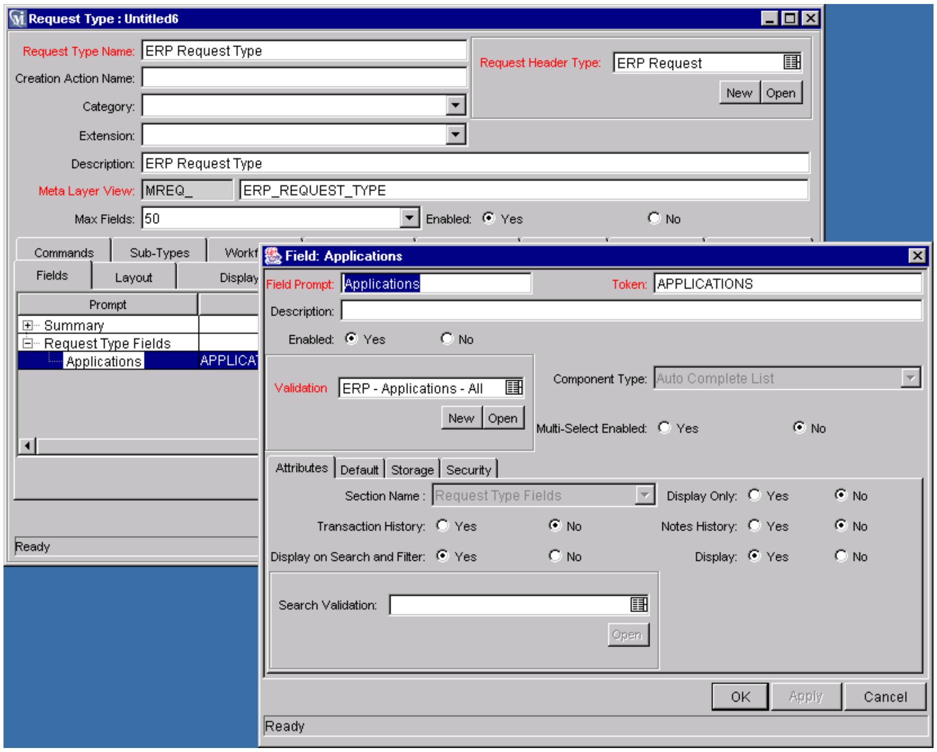
"Save" Successful.

The specific variant for the ERP Request Header Type would be (assuming that the context-specific user data field was captured in the USER_DATA1 column):

```
select LOOKUP_CODE, MEANING, DEFAULT_FLAG
from KNTA_LOOKUPS
where UPPER(MEANING) like UPPER('?%')
and LOOKUP_TYPE = 'APPLICATION'
and VISIBLE_USER_DATA1 = 'ERP Request'
order by 2
```

Example: Resulting Behavior

It is now possible to create a context-sensitive Applications field for any Request Type. In this example, simply create a new field with the Validation ERP - Applications - All.



Request Type : Untitled6

Request Type Name: ERP Request Type

Creation Action Name:

Category:

Extension:

Description: ERP Request Type

Meta Layer View: MREQ_ ERP_REQUEST_TYPE

Max Fields: 50

Enabled: ● Yes ○ No

Request Header Type: ERP Request

Prompt	
⊞ Summary	
⊟ Request Type Fields	
Applications	APPLICA

Ready

Field: Applications

Field Prompt: Applications

Token: APPLICATIONS

Description:

Enabled: ● Yes ○ No

Validation: ERP - Applications - All

Component Type: Auto Complete List

Multi-Select Enabled: ○ Yes ● No

Section Name : Request Type Fields

Display Only: ○ Yes ● No

Transaction History: ○ Yes ● No

Notes History: ○ Yes ● No

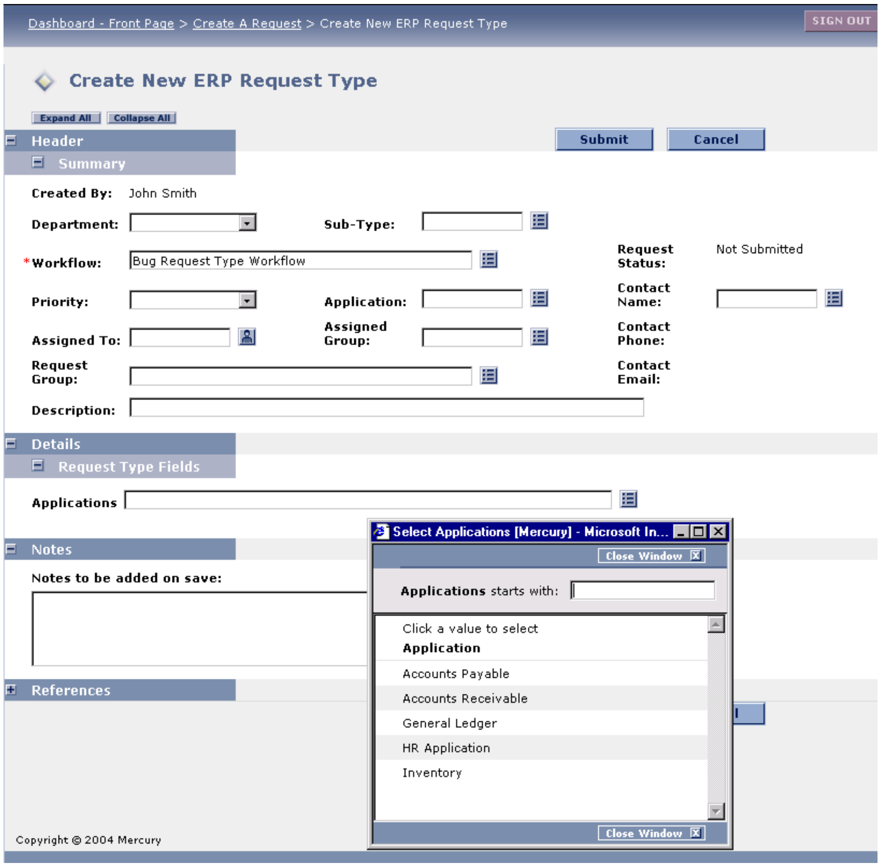
Display on Search and Filter: ● Yes ○ No

Display: ● Yes ○ No

Search Validation:

Ready

When a user creates a Request with this Request Type (which references the ERP Request Header Type), the following Validations are associated with the **Applications** field.



Create New ERP Request Type

Created By: John Smith

Department:

Sub-Type:

*Workflow: Bug Request Type Workflow

Request Status: Not Submitted

Priority:

Application:

Contact Name:

Assigned To:

Assigned Group:

Contact Phone:

Request Group:

Contact Email:

Description:

Applications

Notes to be added on save:

Copyright © 2004 Mercury

Select Applications [Mercury] - Microsoft In...

Applications starts with:

Click a value to select

Application

Accounts Payable

Accounts Receivable

General Ledger

HR Application

Inventory

Project/Task User Data Roll-Up

Values from Task User Data fields can be configured to “roll up” (combine/process values in a meaningful way) into parent Project User Data fields. The following types of Task User Data can roll up into Project User Data:

- Numeric fields (Text Field component type with Numeric data mask)
- Date fields

For each Project, a Project User Data field can show a roll-up of Task User Data values using one of the following methods:

- **Average** — Shows the average of all values of a specified Task User Data field for every Task under the Project (Numeric fields).

- **Maximum** — Shows the largest of all values of a specified Task User Data field for every Task under the Project (Numeric and Date fields).
- **Minimum** — Shows the smallest of all values of a specified Task User Data field for every Task under the Project (Numeric and Date fields).
- **Sum** — Shows the summation of all values of a specified Task User Data field for every Task under the Project (Numeric fields).

Project/Task User Data Roll-Up can be used to capture various important aspects of a Project.

Example

Using the **Average** Roll-Up Method, the average cost of all a Project's Tasks can be easily determined and automatically recalculated each time a Task is updated.

Using the **Maximum** Roll-Up Method, the latest date out of a Project's Tasks can be captured.

Using the **Minimum** Roll-Up Method, the earliest date out of a Project's Tasks can be captured.

Using the **Sum** Roll-Up Method, the total cost of a Project's Tasks can be easily determined and automatically recalculated each time a Task is updated.

Creating Project/Task User Data Roll-Up

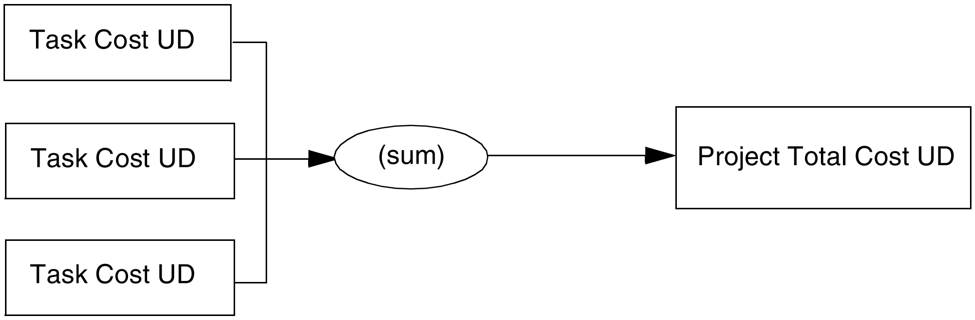
User Data must be configured for Projects and Tasks before specifying Roll-Up Methods.

1. Create and configure Project and Task User Data fields.
2. Link Project and Task User Data fields with Roll-Up Method.

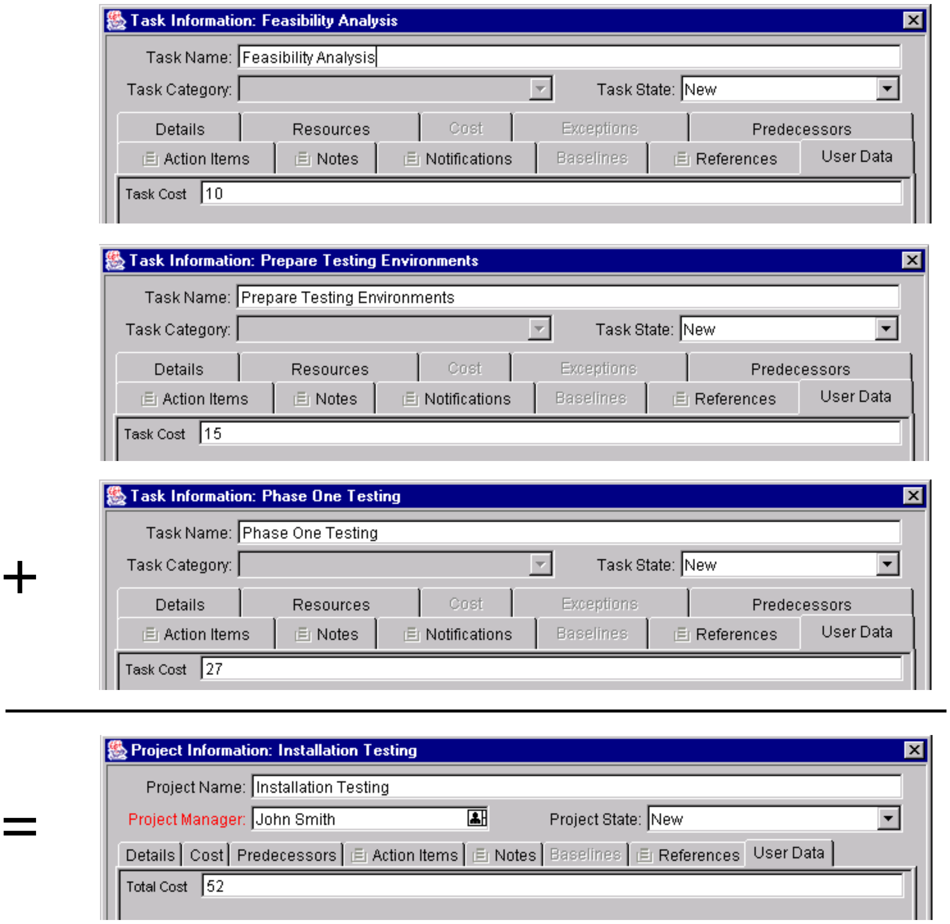
For more detailed information on configuring User Data, see *"Creating and Editing User Data"* on page 361.

Example: Using Project/Task User Data Roll-Up

Company X would like to capture total cost for its Projects. Total Project cost in this case is to be calculated by adding the costs of individual Tasks. User Data fields for Task cost and Project total cost are each defined. The relationship is illustrated below:



Each Task has its own Cost User Data field. The values for each Task Cost User Data field are rolled up using the **Sum** Roll-Up Method into the Project Total Cost User Data field.



Task Information: Feasibility Analysis

Task Name: Feasibility Analysis

Task Category:

Task State: New

Task Cost 10

Task Information: Prepare Testing Environments

Task Name: Prepare Testing Environments

Task Category:

Task State: New

Task Cost 15

Task Information: Phase One Testing

Task Name: Phase One Testing

Task Category:

Task State: New

Task Cost 27

Project Information: Installation Testing

Project Name: Installation Testing

Project Manager: John Smith

Project State: New

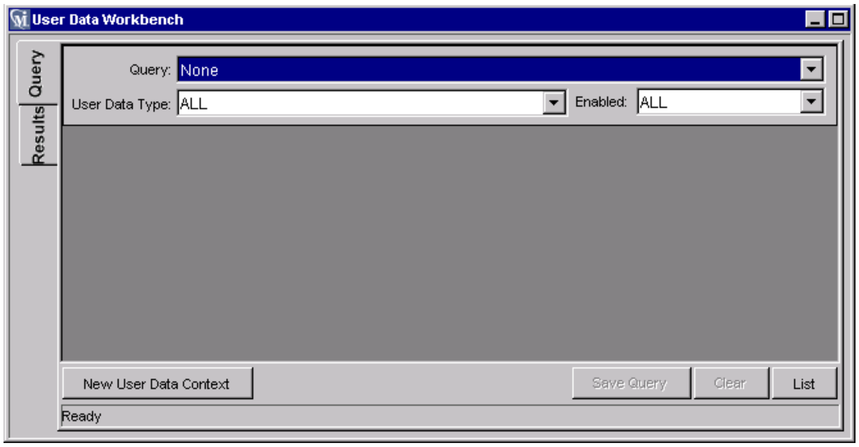
Total Cost 52

Once Project and Task User Data fields have been configured and saved, the Roll-Up relationship can be specified.

To specify the Roll-Up Method:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.



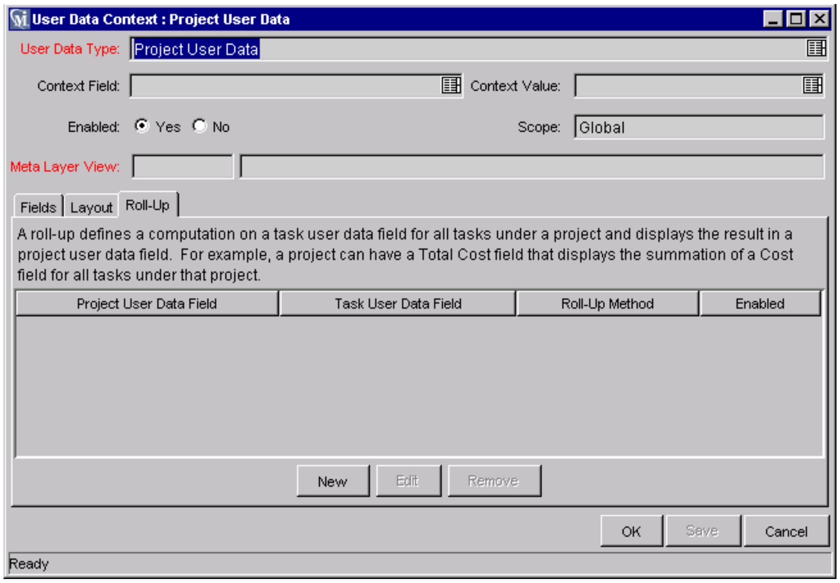
Query: None

User Data Type: ALL

Enabled: ALL

2. From the User Data Type drop down list, select **Project User Data.**
3. Click **List**.

   The **Results** tab opens with the Project User Data Type loaded.
4. Select the Project User Data and click **Open**.
5. Click the **Roll-Up** tab.



User Data Context : Project User Data

User Data Type: Project User Data

Context Field:

Context Value:

Enabled: ● Yes ○ No

Scope: Global

Meta Layer View:

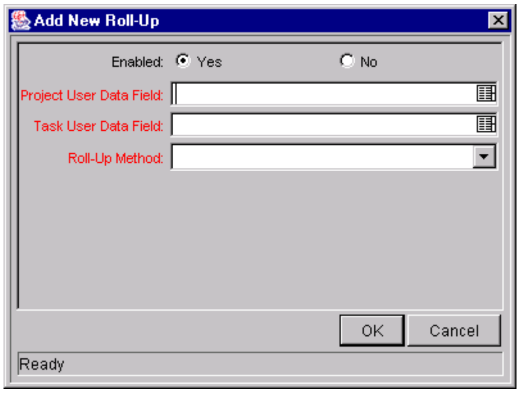
A roll-up defines a computation on a task user data field for all tasks under a project and displays the result in a project user data field. For example, a project can have a Total Cost field that displays the summation of a Cost field for all tasks under that project.

Project User Data Field	Task User Data Field	Roll-Up Method	Enabled

Ready

6. Click **New**.

   The Add New Roll-Up window opens.



Add New Roll-Up

Enabled: ● Yes ○ No

Project User Data Field:

Task User Data Field:

Roll-Up Method:

Ready

7. Select the Project User Data Field that will contain rolled-up Task User Data values.

8. Select the Task User Data Field whose values will roll up into the chosen Project User Data Field.

9. Select the Roll-Up Method from the drop down list.

   The drop down list will only display valid options for the data types of the Project and Task User Data fields.

10. To enable the Roll-Up, select **Yes**.

11. Click **OK**.

    The Roll-Up relationship is added to the **Roll-Up** tab.

12. Click **Save**.

Note

Only two User Data fields of the same type can be selected for Roll-Up (for example, a Numeric field cannot roll up into a Date field, nor vice versa).

While a Task User Data field can have multiple Roll-Up relationships associated with it, a Project User Data field cannot have more than one Roll-Up relationship defined.

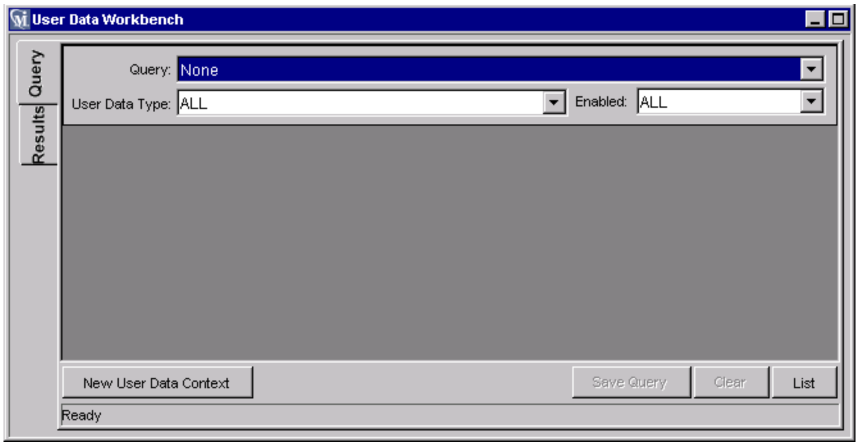
Editing Project/Task User Data Roll-Up

Project/Task User Data Roll-Up can be edited from the Workbench once it has been created.

To edit a Project/Task User Data Roll-Up relationship:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.



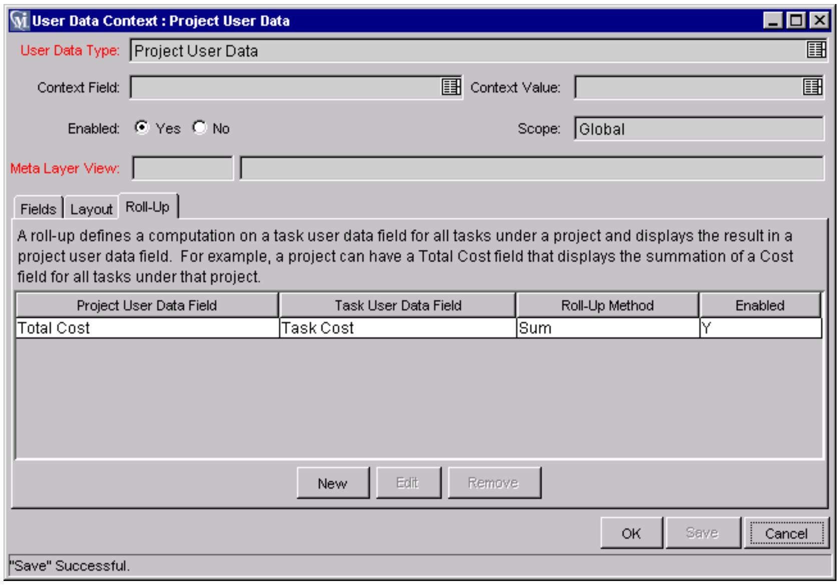
2. From the User Data Type drop down list, select **Project User Data**.

3. Click **List**.

The **Results** tab opens with the Project User Data Type loaded.

4. Select the Project User Data and click **Open**.

5. Click the **Roll-Up** tab.



User Data Type: Project User Data

Context Field:

Context Value:

Enabled: ● Yes ○ No

Scope: Global

Meta Layer View:

A roll-up defines a computation on a task user data field for all tasks under a project and displays the result in a project user data field. For example, a project can have a Total Cost field that displays the summation of a Cost field for all tasks under that project.

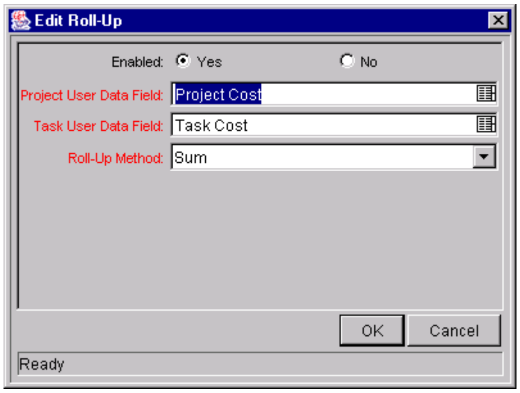
Project User Data Field	Task User Data Field	Roll-Up Method	Enabled
Total Cost	Task Cost	Sum	Y

"Save" Successful.

6. Select the Roll-Up relationship you wish to edit.

7. Click **Edit**.

   The Edit Roll-Up window opens.



Enabled: ● Yes ○ No

Project User Data Field: Project Cost

Task User Data Field: Task Cost

Roll-Up Method: Sum

Ready

8. Make any desired changes to the Project/Task User Data field or Roll-Up Method.

9. Click **OK**.

   The Roll-Up definition is updated in the **Roll-Up** tab.

10. Click **Save**.

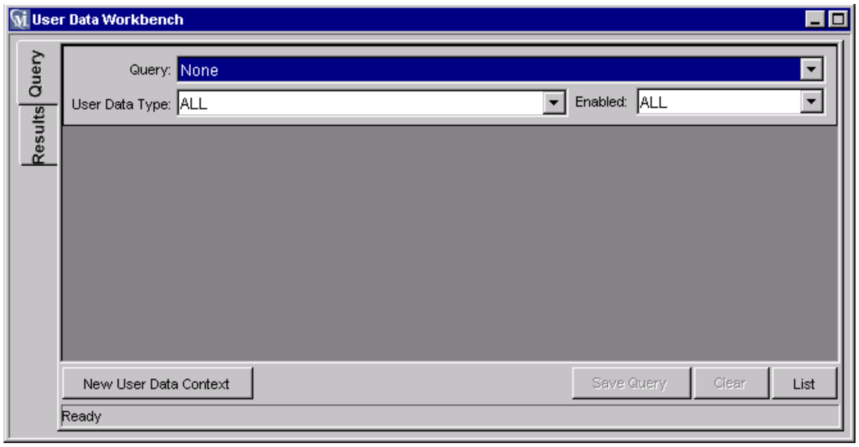
Deleting Project/Task User Data Roll-Up

Project/Task User Data Roll-Up can be deleted. This deletion only removes the Roll-Up relationship; it does not delete the referenced User Data fields.

To delete a Project/Task User Data Roll-Up relationship:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.



User Data Workbench

Query: None

User Data Type: ALL

Enabled: ALL

Ready

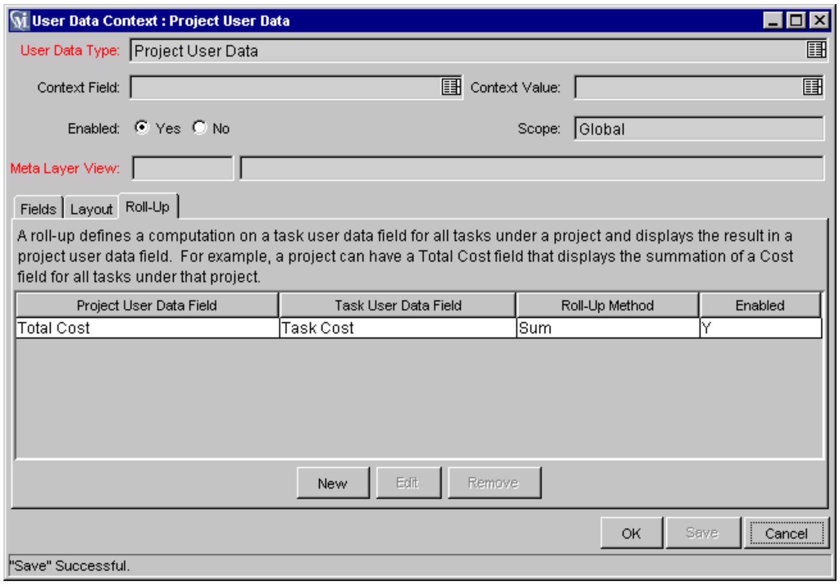
2. From the User Data Type drop down list, select **Project User Data.**

3. Click **List**.

   The **Results** tab opens with the Project User Data Type loaded.

4. Select the Project User Data and click **Open**.

5. Click the **Roll-Up** tab.



User Data Type: Project User Data

Context Field:

Context Value:

Enabled: ● Yes ○ No

Scope: Global

Meta Layer View:

A roll-up defines a computation on a task user data field for all tasks under a project and displays the result in a project user data field. For example, a project can have a Total Cost field that displays the summation of a Cost field for all tasks under that project.

Project User Data Field	Task User Data Field	Roll-Up Method	Enabled
Total Cost	Task Cost	Sum	Y

"Save" Successful.

6. Select the Roll-Up relationship you wish to remove.

7. Click **Remove**.

8. Click **Save**.

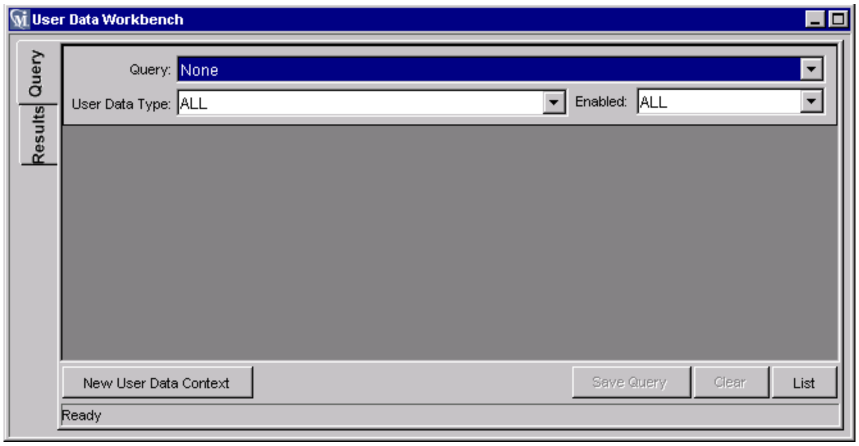
Example: Creating and Using Project/Task User Data Roll-Up

ACME wants to capture the total cost of any Project. This value will be calculated as the sum of all Task costs. They also want the calculated cost to be updated every time a Task cost is changed. They will accomplish this using Project/Task User Data Roll-Up.

To create Project/Task User Data Roll-Up that will calculate total Project cost:

1. Click the **Configuration** screen group and click the **User Data** screen.

   The User Data Workbench opens.



2. Create the Project User Data field.

   a. Select **Project User Data** from the User Data Type drop down list.

   b. Click **List**.

      The **Results** tab opens with the Project User Data Type loaded.

   c. Open the Project User Data Type.

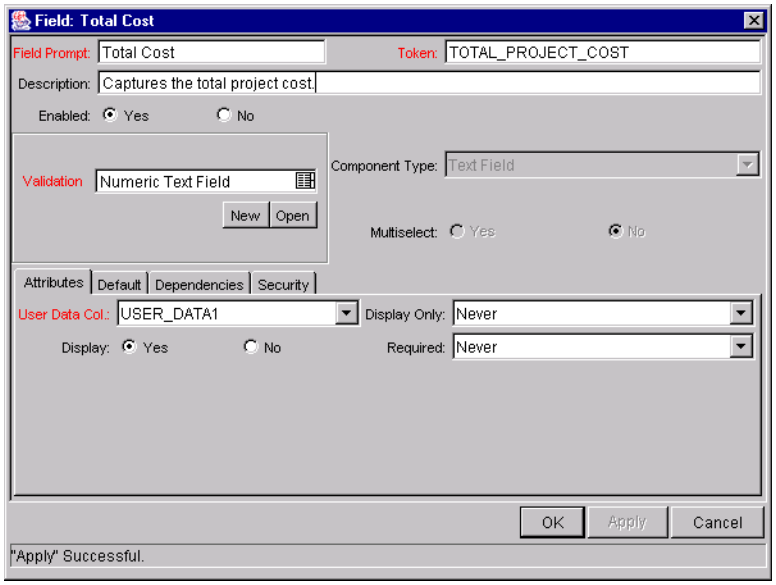
   d. Click **New** in the **Fields** tab.

      The Field: New window opens.

   e. Fill in the following information:

Field	Value
Field Prompt	Total Cost
Token	(any useful token)
Description	(any useful description)
Validation	Numeric Text Field
Enabled	Yes
User Data Col	(any available User Data column)
Display Only	Never
Display	Yes
Required	Never

Field	Value
Workbench Only	No



Field: Total Cost

Field Prompt: Total Cost

Token: TOTAL_PROJECT_COST

Description: Captures the total project cost.

Enabled: ● Yes ○ No

Validation: Numeric Text Field

Component Type: Text Field

Multiselect: ○ Yes ● No

User Data Col.: USER_DATA1

Display Only: Never

Display: ● Yes ○ No

Required: Never

"Apply" Successful.

f. In the Field: New window, click **OK**.

g. In the Project User Data window, click **OK** to save the new field.

3. Create the Task User Data field.

a. In the User Data Workbench **Query** tab, select **Task User Data** from the User Data Type drop down list.

b. Click **List**.

The **Results** tab opens with the Task User Data Type loaded.

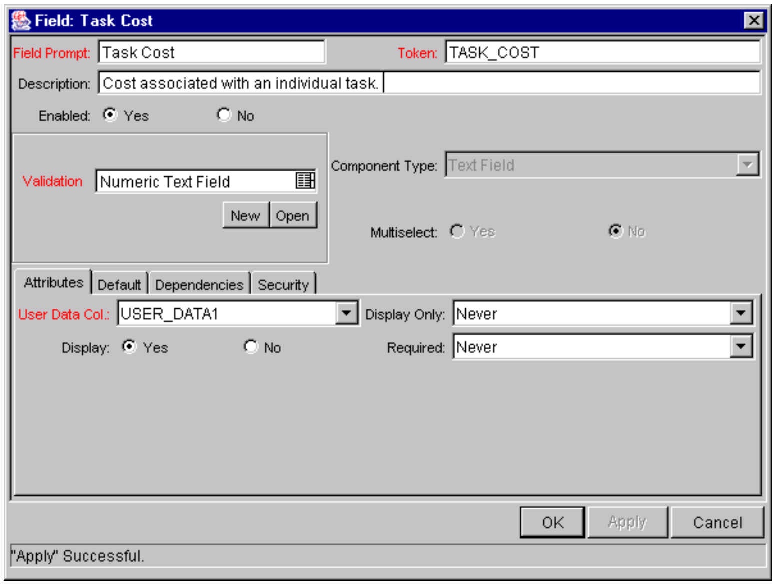
c. Open the Task User Data Type.

d. Click **New** in the **Fields** tab.

The Field: New window opens.

e. Fill in the following information:

Field	Value
Field Prompt	Task Cost
Token	(any useful token)
Description	(any useful description)
Validation	Numeric Text Field
Enabled	Yes
User Data Col	(any available User Data column)
Display Only	Never
Display	Yes
Required	Never
Workbench Only	No



Field Prompt: Task Cost

Token: TASK_COST

Description: Cost associated with an individual task.

Enabled: ● Yes ○ No

Validation: Numeric Text Field

Component Type: Text Field

Multiselect: ○ Yes ● No

User Data Col.: USER_DATA1

Display Only: Never

Display: ● Yes ○ No

Required: Never

"Apply" Successful.

f. In the Field: New window, click **OK**.

g. In the Task User Data window, click **OK** to save the new field.

4. Create the Roll-Up relationship between the Task and Project User Data fields.

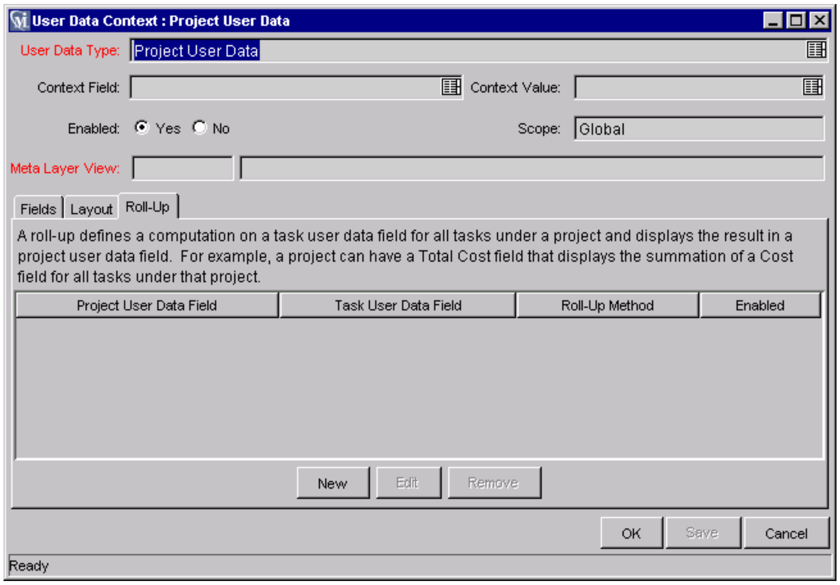
a. In the User Data Workbench **Query** tab, select **Project User Data** from the User Data Type drop down list.

b. Click **List**.

The **Results** tab opens with the Project User Data Type loaded.

c. Open the Project User Data Type.

d. Click the **Roll-Up** tab.



User Data Context : Project User Data

User Data Type: Project User Data

Context Field:

Context Value:

Enabled: ● Yes ○ No

Scope: Global

Meta Layer View:

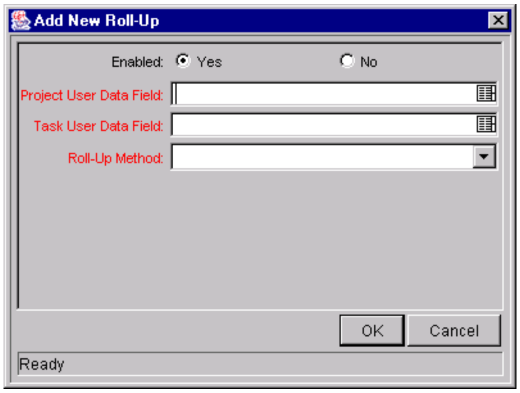
A roll-up defines a computation on a task user data field for all tasks under a project and displays the result in a project user data field. For example, a project can have a Total Cost field that displays the summation of a Cost field for all tasks under that project.

Project User Data Field	Task User Data Field	Roll-Up Method	Enabled

Ready

e. Click **New**.

The Add New Roll-Up window opens.



Add New Roll-Up

Enabled: ● Yes ○ No

Project User Data Field:

Task User Data Field:

Roll-Up Method:

Ready

f. In the Project User Data Field, select **Total Cost**.

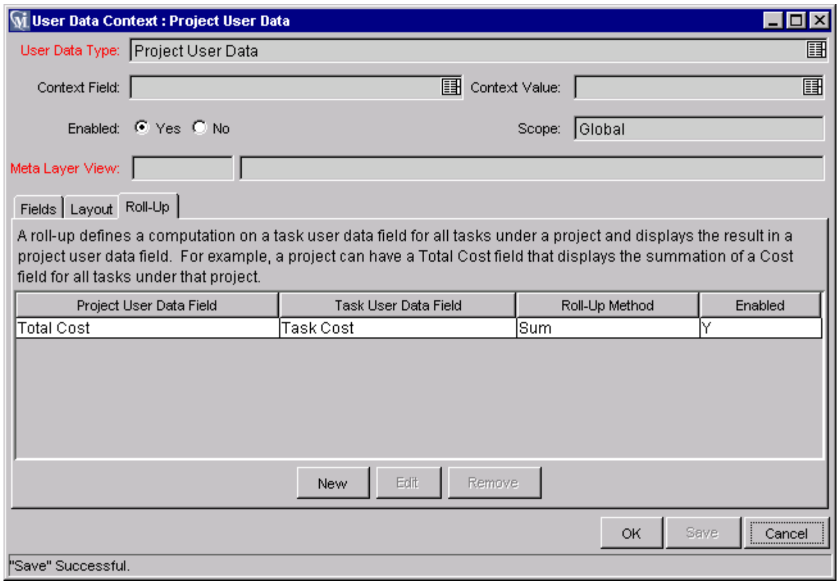
g. For the Task User Data Field, select **Cost**.

h. From the Roll-Up Method drop down list, select **Sum**.

i. Click **OK**.

The Roll-Up relationship is added to the **Roll-Up** tab.

5. Click **Save**.



User Data Type: Project User Data

Context Field:

Context Value:

Enabled: ● Yes ○ No

Scope: Global

Meta Layer View:

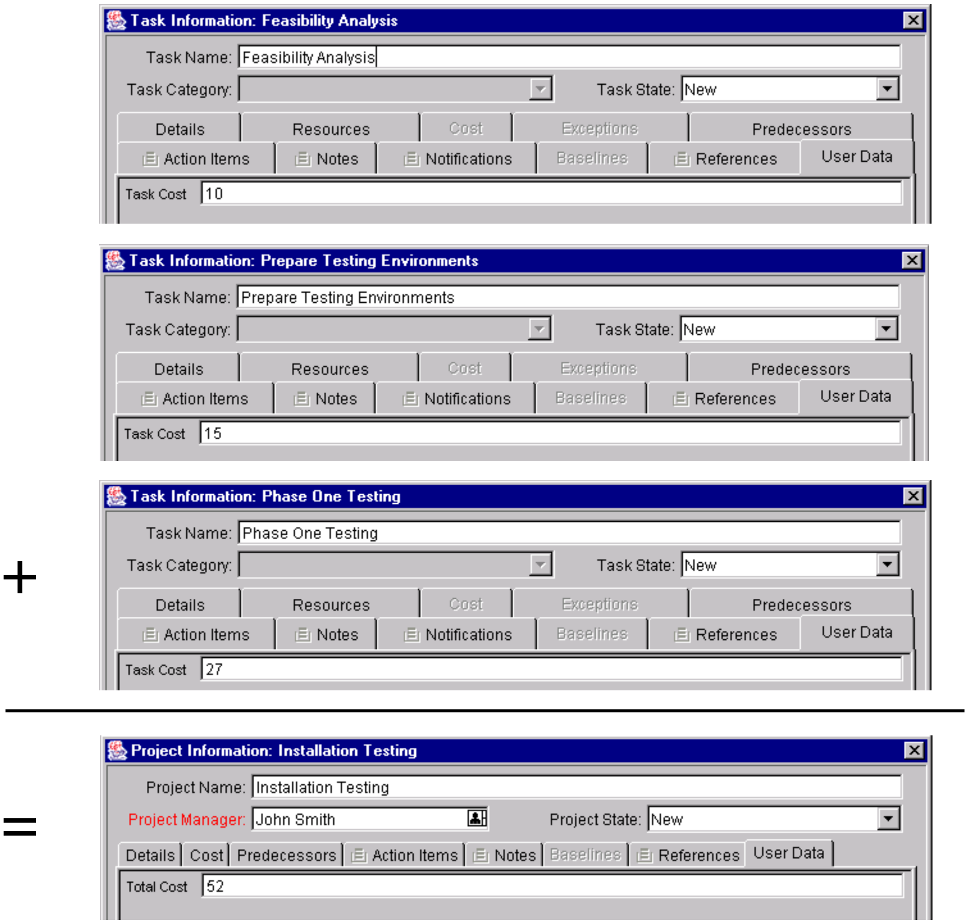
A roll-up defines a computation on a task user data field for all tasks under a project and displays the result in a project user data field. For example, a project can have a Total Cost field that displays the summation of a Cost field for all tasks under that project.

Project User Data Field	Task User Data Field	Roll-Up Method	Enabled
Total Cost	Task Cost	Sum	Y

"Save" Successful.

The rolled-up fields can now be accessed from the **User Data** tab of the respective Project or Task.

Each Task has its own Cost User Data field. The values for each Task Cost User Data field are rolled up using the **Sum** Roll-Up Method into the Project Total Cost User Data field.



Task Information: Feasibility Analysis

Task Name: Feasibility Analysis

Task Category:

Task State: New

Task Cost 10

Task Information: Prepare Testing Environments

Task Name: Prepare Testing Environments

Task Category:

Task State: New

Task Cost 15

+

Task Information: Phase One Testing

Task Name: Phase One Testing

Task Category:

Task State: New

Task Cost 27

=

Project Information: Installation Testing

Project Name: Installation Testing

Project Manager: John Smith

Project State: New

Total Cost 52

Referring to User Data

Once a User Data field has been created, it is possible to refer to it from other parts of the product by its Token name, proceeded by the entity abbreviation and the UD qualifier.

Example

It is possible to define a custom field for the Users entity to store the Department each user is in. This custom field would be defined using the user's User Data and would generate a field with a token of 'DEPARTMENT.' Then, in an Object Type command, a Workflow Step, or in a Report, refer to this new field as the Token [USR.UD.DEPARTMENT]. The Mercury IT Governance Center would then look into the custom field for the value of this Token.

Migrating User Data

Product configuration data such as Workflows, Validations, and Request Types can be migrated between product instances (installations). User Data values and configurations can also be migrated between instances. The following sections contain more detailed information:

- *Migrating User Data Values*
- *Migrating User Data Contexts*

Migrating User Data Values

For any configuration entity with User Data fields (such as Request Type, Object Type, or Workflow) the data in the User Data fields is migrated along with the entity.

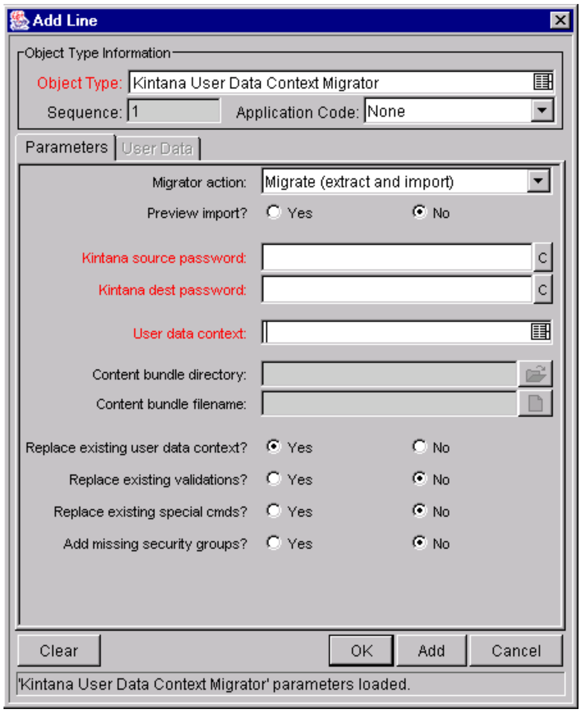
- If two instances have identical User Data configurations, then the User Data will be migrated correctly.
- If two instances do not have identical User Data configurations, then the User Data will be mapped into the data model according to the storage configuration in the source instance. For this reason, the two instances should be configured with the same User Data fields, or the User Data should be corrected after migration.
- If the User Data is Context Sensitive, then a corresponding Context Sensitive configuration must exist in the destination instance, or the migration will fail.

Note

User Data fields that have different hidden and visible values may be problematic. When the hidden value of a User Data field refers to a primary key (example: Security Group ID), that can be different in the source and destination instances, then the migrator does NOT correct the hidden value. The User Data should be corrected manually after migration.

Migrating User Data Contexts

User Data field contexts can also be migrated between product instances using the User Data Context Migrator Object Type. This Migrator Object Type can migrate Global as well as Context Sensitive User Data Contexts.



Object Type Information

Object Type: Kintana User Data Context Migrator

Sequence: 1

Application Code: None

Migrator action: Migrate (extract and import)

Preview import? ○ Yes ● No

Kintana source password:

Kintana dest password:

User data context:

Content bundle directory:

Content bundle filename:

Replace existing user data context? ● Yes ○ No

Replace existing validations? ○ Yes ● No

Replace existing special cmds? ○ Yes ● No

Add missing security groups? ○ Yes ● No

'Kintana User Data Context Migrator' parameters loaded.

For more detailed information on the User Data Context Migrator, see *Migrators Guide and Reference*.

Appendix E
Configuration Worksheets

This appendix provides worksheets that can be printed out and used to capture data required for configuring a Request resolution system. Worksheets are provided for the following entities:

- Workflows
- Workflow Steps
- Request Header Types
- Request Types
- Request Type Fields and Commands
- Security Groups

Table E-1. Workflow Skeleton

Step No.	Step Name	Description	Type (Execution, Decision, Condition, or Subworkflow	Transition Values	Validation
1					
2					
3					
4					
5					
6					
7					
8					
9					
10					
11					
12					
13					
14					
15					
16					
17					
18					
19					

Table E-2. Workflow Step [Execution] -- Step Number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Execution Type**	
Processing Type	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): • User Name • Standard Token • User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: • Username • Email Address • Security Group • Standard Token • User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Validation Information*	Value
Existing Validation?	
New Validation?	
Validation Type: *(text field, auto-complete, drop down list, etc.)*	
Validation Definition (list of values or SQL)	

Execution Type**	Value
Built-in Workflow Event: • Execute Commands • Close • Jump / Receive • Ready for Release • Return from Subworkflow	
PL/SQL Function	
Token	
SQL Statement	
Workflow step commands	

Table E-3. Workflow Step [Decision] -- Step Number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Decisions Required (Vote on Step's outcome?)	• One • At Least One • All
Timeout (Days)	
Security (who can act on step): • Security Group • User Name • Standard Token • User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: • Username • Email Address • Security Group • Standard Token • User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N)	
Authentication Type (if Y)	

Validation Information*	**Value**
Existing Validation?	
New Validation?	
Validation Type: *(text field, auto-complete, drop down list, etc.)*	
Validation Definition (list of values or SQL)	

Table E-4. Workflow Step [Sub-Workflow] -- Step Number ____.

	Value
Step Name	
Goal / Result of Step	
Validation*	
Vote on Step's outcome?	
Timeout (Days)	
Source Environment (Group)	
Dest Environment (Group)	
Security (who can act on step): • Security Group • User Name • Standard Token • User Defined Token	
Include Notification (Yes/No)	
Notification Event	
Notification Recipient: • Username • Email Address • Security Group • Standard Token • User Defined Token	
Notification Message	
Request Status at Step	
Request % Complete at Step	
Authentication Required (Y/N) Authentication Type (if Y)	

Validation Information*	**Value**
Existing Validation?	
New Validation?	
Validation Type: *(text field, auto-complete, drop down list, etc.)*	
Validation Definition (list of values or SQL)	

See *"Using Subworkflows"* on page 251 for notes on Validations for transitions into and out of Subworkflows.

Table E-5. Request Type Information.

	Value
Request Type Name	
Associated Request Header Type	
Description	

#	Field Names	Description
1		
2		
3		
4		
5		
6		
7		
8		
9		
10		
11		
12		
13		
14		
15		
16		

Table E-6. Request Type Commands

Goal of Commands	
Command Steps	
Conditions (When to execute)	

Table E-7. Request Type Statuses

Status	Corresponds to Workflow Step

Table E-8. Request Type Field Information

Field Name	
Validation*	
Field Behavior:	
Attributes (select one):	• Display • Editable • Display Only • Required
Default Value	
Users/Security Groups allowed to View Field	
Users/Security Groups allowed to Edit Field	
Status Dependencies:	
Clear field when Status = ?	
Display only when Status = ?	
Reconfirm only when Status = ?	
Required when Status = ?	
Auto-Population Behavior:	
Auto-Population triggered by (Depends on) Field:	
Value to populate Field with:	

Table E-9. Field Validation Information

Validation Information*	**Value**
Existing Validation?	
New Validation?	
Validation Type: *(text field, auto-complete, drop down list, etc.)*	
Validation Definition (list of values or SQL)	
Notes on Validation *(data masks, auto-complete behavior, etc.)*	

Table E-10. Request Header Type Information

	Value
Request Header Type Name	
Associated Request Type(s)	
Description	
Associated Field Group(s)	

For creating new Request Header fields, use the worksheets for *"Request Type Field Information"* on page 412.

Table E-11. Existing Request Header Type Field Information

Prompt	Display	Display Only?	Transaction History?	Notes History ?	Search/ Filter Page?
Request No					
Request Type					
Created By					
Department					
Sub-Type					
Created On					
Workflow					
Request Status					
Priority					
Application					
Contact Name					
Assigned To					
Assigned Group					
Contact Phone					
Request Group					
Contact Email					
Description					
Company					
% Complete					

Participant and Security

Table E-12. Security Groups.

Security Group Name	Members	Act on Workflow Steps	View/Edit Request Fields	Description

Index