

Mercury IT Governance Center™

Mercury Deployment Management™

User's Guide

Version: 7.0



This manual, and the accompanying software and other documentation, is protected by U.S. and international copyright laws, and may be used only in accordance with the accompanying license agreement. Features of the software, and of other products and services of Mercury Interactive Corporation, may be covered by one or more of the following patents: United States: 5,511,185; 5,657,438; 5,701,139; 5,870,559; 5,958,008; 5,974,572; 6,137,782; 6,138,157; 6,144,962; 6,205,122; 6,237,006; 6,341,310; 6,360,332; 6,449,739; 6,470,383; 6,477,483; 6,549,944; 6,560,564; 6,564,342; 6,587,969; 6,631,408; 6,631,411; 6,633,912; 6,694,288; 6,738,813; 6,738,933; 6,754,701; 6,792,460 and 6,810,494. Australia: 763468 and 762554. Other patents pending. All rights reserved.

U.S. GOVERNMENT RESTRICTED RIGHTS. This Software Documentation is a “commercial item” as defined at 48 C.F.R. 2.101 (October 1995). In accordance with 48 C.F.R. 12.212 (October 1995), 48 C.F.R. 27.401 through 27.404 and 52.227-14 (June 1987, as amended) and 48 C.F.R. 227.7201 through 227.7204 (June 1995), and any similar provisions in the supplements to Title 48 of the C.F.R. (the “Federal Acquisition Regulation”) of other entities of the U.S. Government, as applicable, all U.S. Government users acquire and may use this Documentation only in accordance with the restricted rights set forth in the license agreement applicable to the Computer Software to which this Documentation relates.

Mercury, Mercury Interactive, the Mercury logo, the Mercury Interactive logo, LoadRunner, WinRunner, SiteScope and TestDirector are trademarks of Mercury Interactive Corporation and may be registered in certain jurisdictions. The absence of a trademark from this list does not constitute a waiver of Mercury's intellectual property rights concerning that trademark.

All other company, brand and product names may be trademarks or registered trademarks of their respective holders. Mercury disclaims any responsibility for specifying which marks are owned by which companies or which organizations.

Mercury provides links to external third-party Web sites to help you find supplemental information. The content and availability may change without notice. Mercury makes no representations or warranties whatsoever as to the content or availability.

Mercury
379 North Whisman Road
Mountain View, CA 94043
<http://www.mercury.com>

© 1997–2006 Mercury Interactive Corporation. All rights reserved.

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

Table of Contents

List of Figures	vii
List of Tables	ix
Chapter 1: Getting Started with Mercury Deployment Management.....	11
Introduction to Mercury Deployment Management.....	12
Overview of a Deployment Management Process.....	13
Deployment Management Terms and Concepts	15
Standard Interface, Workbench, and Deployment Management	15
Packages and Object Types	16
Workflows and Workflow Steps	18
Releases.....	19
Deployment Management Environments	20
Integrating Deployment Management	21
Related Documents.....	23
Chapter 2: Understanding Packages.....	25
About Packages	26
Packages in the Standard Interface	29
Package Detail Page, Header Section	31
Package Detail Page, Status Section	31
Package Detail Page, Notes Section	32
Package Detail Page, References Section	32
Packages in the Workbench.....	33
Package Window, Package Information Section	34
Package Window, Package Lines Tab	35
Package Window, Status Tab	35

Accessing the Line Execution Log.....	37
Accessing the Package Execution History	37
Graphical View.....	37
Viewing Line Permissions	38
Viewing Line Transaction History	38
Viewing the Decision Detail.....	38
Viewing the Workflow Step Information URL.....	39
Package Window, Notes Tab.....	39
Package Window, References Tab.....	39
Reference Relationships.....	40
Predecessor Relationships.....	43
Successor Relationships.....	43
Chapter 3: Creating Packages.....	45
Overview of Creating Packages.....	46
Before You Create Packages	46
Selecting Workflows.....	46
Verifying Environments	49
Verifying Object Types	50
Creating Packages.....	52
Creating Packages Using the Workbench.....	52
Creating Packages by Copying Existing Packages.....	54
Creating Packages as References from Packages.....	56
Creating Packages as References from Requests.....	58
Entering General Information for Packages.....	59
Adding Lines to Packages	60
Adding Notes to Packages.....	63
Adding References to Packages from the Package Workbench.....	64
Adding Attachments as Package References	64
Adding Existing Packages as Package References.....	66
Adding New Packages as Package References.....	67
Attaching Programs as Package References.....	69
Attaching Projects as Package References.....	70
Adding Releases as References	71
Adding Existing Requests as Package References	72
Adding New Requests as Package References.....	73
Adding Tasks as Package References.....	75
Adding URLs as Package References.....	76
Submitting Packages.....	77
Chapter 4: Processing Packages.....	79
Overview of Processing Packages	80
Using the Standard Interface to Process Packages	81

Opening Packages from the Standard Interface	81
Opening Packages from the Search Packages Page	81
Opening Packages from the My Packages Portlet.....	82
Opening Packages Using Notification Links.....	82
Approving Packages in the Standard Interface	82
Adding Notes to Packages in the Standard Interface	86
Adding References to Packages from the Standard Interface.....	86
Adding Attachments as References.....	87
Adding Existing Packages as Package References.....	88
Adding New Packages as Package References	89
Attaching Programs as Package References.....	92
Attaching Projects as Package References.....	93
Adding Releases as Package References.....	94
Adding Existing Requests as Package References.....	95
Adding New Requests as Package References.....	96
Adding Tasks as Package References.....	97
Adding URLs as Package References.....	99
Processing Packages Using the Workbench	100
Opening Packages from the Workbench.....	100
Checking Package Status.....	100
Acting on Eligible Steps.....	101
Acting on Decision Workflow Steps.....	104
Acting on Execution Workflow Steps.....	106
Adding Notes.....	108
Adding, Updating, and Deleting References.....	109
Adding References.....	109
Updating Reference Relationships	110
Deleting References	110
Adding a Package Line to an In-Progress Package.....	111
Chapter 5: Managing Packages	113
Managing Packages in Mercury Deployment Management	114
Merging Packages.....	114
Creating New Package Groups.....	116
Editing Package Groups	118
Deleting Packages	119
Cancelling Package Lines.....	120
Index	123

List of Figures

Figure 1-1	Mercury Deployment Management overview.....	13
Figure 1-2	A package.....	14
Figure 1-3	Sample package in the Package window.....	14
Figure 1-4	Package workflow.....	15
Figure 1-5	Sample package in the Package window.....	17
Figure 1-6	Package workflow.....	19
Figure 1-7	Deployment Management environment.....	20
Figure 1-8	Sample environment.....	21
Figure 2-1	Package processing in Deployment Management.....	26
Figure 2-2	Package workflow and workflow steps.....	27
Figure 2-3	How a package comes together.....	28
Figure 2-4	Package detail page.....	30
Figure 2-5	Package detail page, Header section.....	31
Figure 2-6	Package detail page, Status section.....	31
Figure 2-7	Package detail page, Notes section.....	32
Figure 2-8	Package detail page, References section.....	33
Figure 2-9	Package window.....	33
Figure 2-10	Package window, Package Information section.....	34
Figure 2-11	Package window, Package Lines tab.....	35
Figure 2-12	Package window, Status tab.....	36
Figure 2-13	Graphical View window for a package.....	38
Figure 2-14	Package window, Notes tab.....	39
Figure 2-15	Package window, References tab.....	40

List of Figures

Figure 3-1	Package window, Notes section.....	63
Figure 4-1	Notes section.....	86
Figure 4-2	Status tab eligible steps.....	101
Figure 4-3	Notes tab in the Package window.....	108

List of Tables

Table 2-1	References and relationships	41
Table 3-1	Object types.....	50
Table 3-2	Copy package options.....	56

Chapter

1

Getting Started with Mercury Deployment Management

In This Chapter:

- *Introduction to Mercury Deployment Management*
 - *Overview of a Mercury Deployment Management Process*
 - *Mercury Deployment Management Terms and Concepts*
 - *Related Documents*
-

Introduction to Mercury Deployment Management

Mercury Deployment Management™ is a Mercury IT Governance Center™ product that automates the migration and deployment of software code, configurations, and content. Mercury Deployment Management groups these objects into packages and routes the packages along business processes modeled in Mercury's configurable workflow. The workflow automatically moves each package through required steps, which typically include system build, testing, quality assurance, staging, and final deployment. Mercury Deployment Management automatically deploys the application components (such as XML content, HTML files, Java programs, Oracle application configurations, PeopleSoft panels, and so on) that each package requires.

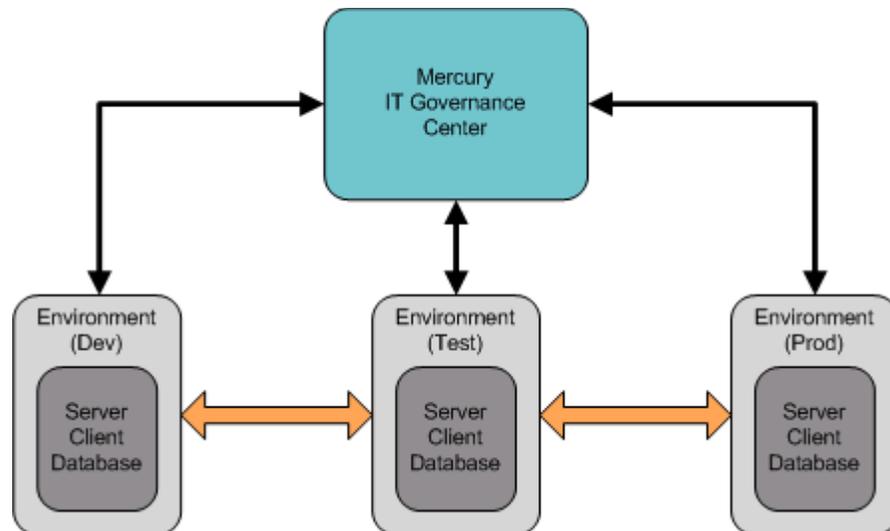
For example, Mercury Deployment Management connects to the development environment, copies Java files from the version control repository, and transfers the files to the quality assurance environment. It then uses the Java compiler to compile and pack all of the files into a single Java program. The customer can use a built-in scheduler to schedule deployments. Mercury Deployment Management maintains an audit trail for all activities, including package information, application components altered, approvals obtained, and deployments performed.

Overview of a Deployment Management Process

Mercury Deployment Management is an enterprise application designed to deploy and audit software changes to mission-critical applications. It does this by gathering all information required for a successful deployment (for example, information on environments and objects to be migrated) into a single logical unit called the package.

The *package* is the fundamental work unit in Mercury Deployment Management. A package consists of the objects that are processed through a business workflow. This results in a successful, easy-to-track software change.

Figure 1-1. Mercury Deployment Management overview



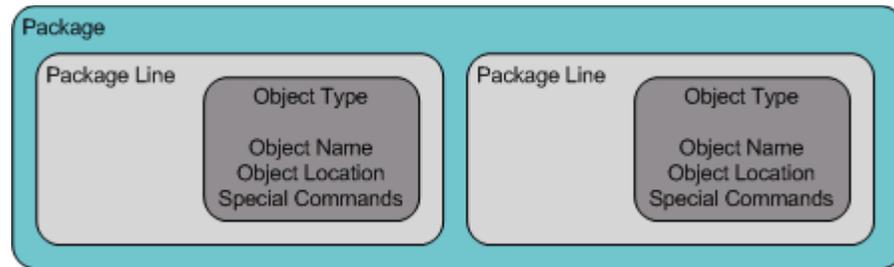
A package consists of a group of objects, each of which belongs to an object type. Each object type is associated with a set of parameters that defines the information the package requires, and any additional information it can accept.

Note

For an overview of packages, see [Chapter 2, Understanding Packages, on page 25](#).

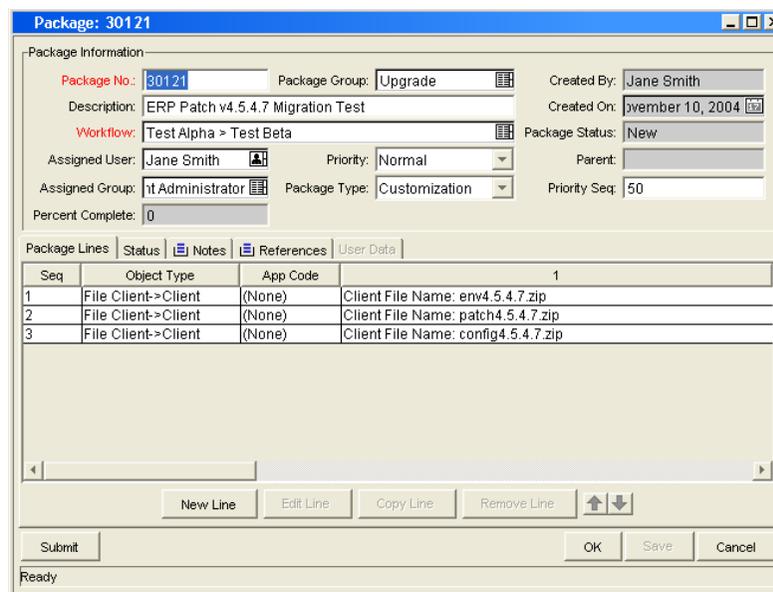
For example, the File Migration object is a standard object type. The File Migration object type copies files (objects) from one environment to another.

Figure 1-2. A package



Although each package line can be acted upon separately, the group of package lines and objects represent a logical unit that is moved and tracked as a unit. The way a package is processed depends on the specific workflow applied to it. [Figure 1-3](#) shows a sample package in the Package window of the Package Workbench.

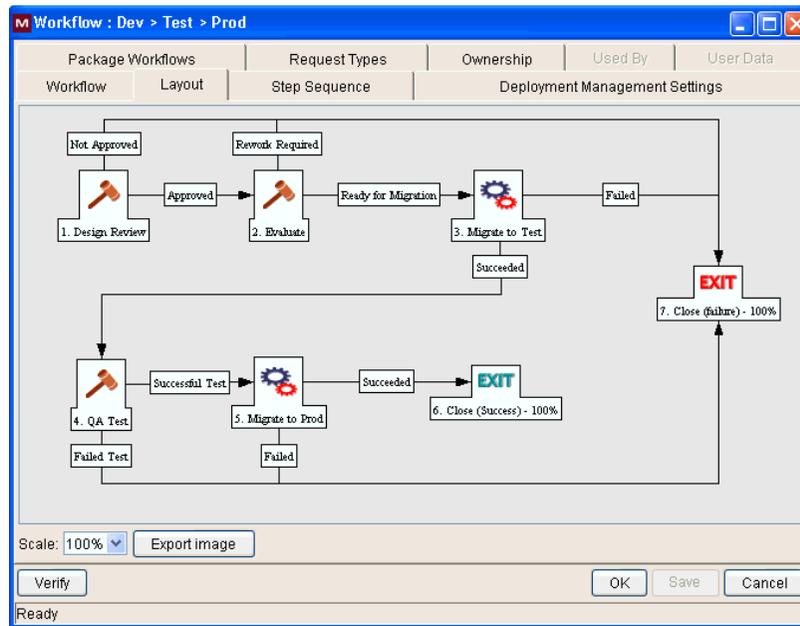
Figure 1-3. Sample package in the Package window



A workflow represents the business process associated with the package. Workflows consist of a logical series of steps that define the path that the objects in a package follow. Some of those workflow steps might represent decision-making processes; for example, code reviews and management approvals. Usually, at least one workflow step represents, or actually performs, the migration of the objects from one environment to another environment.

In [Figure 1-4](#) on page 15, the workflow step Migrate to Test represents the migration of package objects from one environment to another. The workflow steps Design Review and Evaluate represent business package-related decisions.

Figure 1-4. Package workflow



Deployment Management Terms and Concepts

Standard Interface, Workbench, and Deployment Management

Mercury Deployment Management provides two different interfaces for working with packages. In the standard interface, Deployment Management provides the package detail page. In the Mercury IT Governance Workbench, Deployment Management provides the Package Workbench. Although these two interfaces have common functionality, a package detail page does not have all of the functionality of the Package Workbench.

The key difference between the Package Workbench and the detail page for a package is that you can use the Workbench to add package lines and submit a package. You cannot add package lines and submit a package from a detail page.

The Package Workbench is designed to allow engineers or developers to create and submit packages. Creating and submitting packages requires specialized knowledge. In some cases, it requires passwords to production servers. Some of this information is not available to the resources making decisions about a package. Additionally, not everyone who makes decisions about a package requires access to the Workbench.



Note

For instructions on how to create and submit packages to deploy software and application changes, see [Chapter 3, *Creating Packages*, on page 45](#).

From a package detail page, and with the required permissions, you can process a package through its workflow. You can add notes and references to the package and make decisions about package status. However, you cannot change or delete a package from the detail page.



Note

For information about the licenses and access grants required to work with packages in Deployment Management, see the *Security Model Guide and Reference*. For information on how to move packages through their associated workflows, see [Chapter 4, *Processing Packages*, on page 79](#). For instructions on how to work with (copy, cancel, merge, and so on) packages, see [Chapter 5, *Managing Packages*, on page 113](#).

Packages and Object Types

Packages are the fundamental work units of Mercury Deployment Management. A package consists of the objects that are processed through a business workflow.

While each package line can be acted upon separately, the group of package lines and objects represent a logical unit that should be moved and tracked together. The processing of a package and package lines can vary greatly depending upon the workflow specified for that package. [Figure 1-5](#) shows a sample package in the Package window.

Figure 1-5. Sample package in the Package window

The screenshot shows a window titled "Package: 30121". The "Package Information" section includes the following fields:

- Package No.: 30121
- Package Group: Upgrade
- Created By: Jane Smith
- Description: ERP Patch v4.5.4.7 Migration Test
- Created On: November 10, 2004
- Workflow: Test Alpha > Test Beta
- Package Status: New
- Assigned User: Jane Smith
- Priority: Normal
- Parent: (empty)
- Assigned Group: Administrator
- Package Type: Customization
- Priority Seq: 50
- Percent Complete: 0

Below the information is a table with columns: Seq, Object Type, App Code, and Client File Name. The table contains three rows:

Seq	Object Type	App Code	Client File Name
1	File Client->Client	(None)	env4.5.4.7.zip
2	File Client->Client	(None)	patch4.5.4.7.zip
3	File Client->Client	(None)	config4.5.4.7.zip

At the bottom of the window are buttons for "New Line", "Edit Line", "Copy Line", "Remove Line", "Submit", "OK", "Save", and "Cancel".

Each object belongs to an object type. An object type can also have associated commands that determine its behavior as it moves from one environment to another. Commands are used to define what a migration means for each type of object. For example, a File Object Type might only require that a file be copied, while a Database Script Object Type might require the copying of a file as well as executing it against the destination database.

Object types are user-configurable. Mercury Deployment Management and its extensions also provide libraries of predefined object types, falling under one of the following categories:

- Standard Objects.** Standard objects are predefined object types that are shipped with Mercury Deployment Management or one of the Mercury Deployment Management Extensions. These object types encapsulate the basic functionality that manages crucial Deployment Management actions; for example, migrating and executing file system-level objects and operating system commands, or applying patches to an Oracle applications instance. A simple standard object type is the File Migration object, which copies a file from one environment to another. A more complex standard object type is the SQL Script, which not only copies the file but also executes the script against the destination database.
- Custom Objects.** Customers must often customize standard object types, or produce entirely new object types, in order to handle the requirements of their software environments and deployment management processes. A custom objects category is provided to help distinguish these customized

object types from those that come with Mercury Deployment Management or its extensions. Such object types, designed by the customer or by consultants, are often used for integration with third-party tools or in-house products.

- **Mercury IT Governance Center Migrator Objects.** Mercury IT Governance Center migrator objects are object types that contain functionality to export and import configuration information. You can use these for several purposes, including:
 - To transfer configuration information between distinct Mercury IT Governance Center instances
 - To extract information from a Mercury IT Governance Center database schema into an XML file
 - To load information from an XML file into a database schema.

This means that you can migrate configuration information using standard deployment management test practices, processing a package through a workflow.

Workflows and Workflow Steps

A workflow consists of a logical series of steps that define the path to be followed by objects in a package. Workflow configuration and routing can be customized, and the workflow engine can handle virtually any business practice. This allows a department to generate workflows to automate existing processes, rather than forcing users to adopt a new set of processes to perform their work.

Workflow steps can range in usage from functional approvals to actual migrations. For example, migration steps automatically move specified objects from source environments to destination environments.

Workflow steps are events that are linked together to form a complete workflow. Mercury Deployment Management uses the following four types of workflow steps:

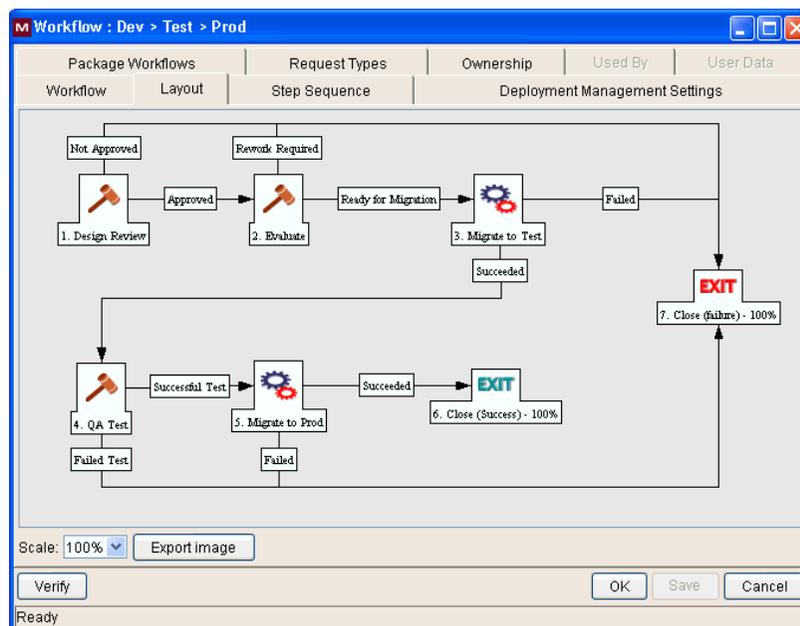
- **Decision workflow steps.** In decision workflow steps, you or group of users must indicate a result or outcome; for example, work is approved or a review is completed.
- **Execution workflow steps.** In execution workflow steps, the system performs an action, and then updates the step with its result. The action can

be as simple as calculating the value for a token, or as complex as copying files, running programs, or updating Web pages.

- **Condition workflow steps.** Condition workflow steps are logic steps used for complex workflow processing; for example, allowing a workflow to proceed only after all prerequisite steps are completed.
- **Subworkflow workflow steps.** Subworkflow workflow steps are entire workflows that have been enabled for use as subworkflows. Subworkflows are useful time savers in modeling business processes that include predefined procedures.

Figure 1-6 illustrates a sample package workflow.

Figure 1-6. Package workflow



Releases

In Mercury Deployment Management, you can configure releases to group packages and related requests that must be deployed together. For example, the software company XYZ Corporation has a product update release scheduled for five months from now. To ensure successful product delivery, they decide to use Deployment Management to create a release that lets them track all changes to their original code.

As developers complete their packages, the packages are included in the release and processed together. Because every required change is grouped in

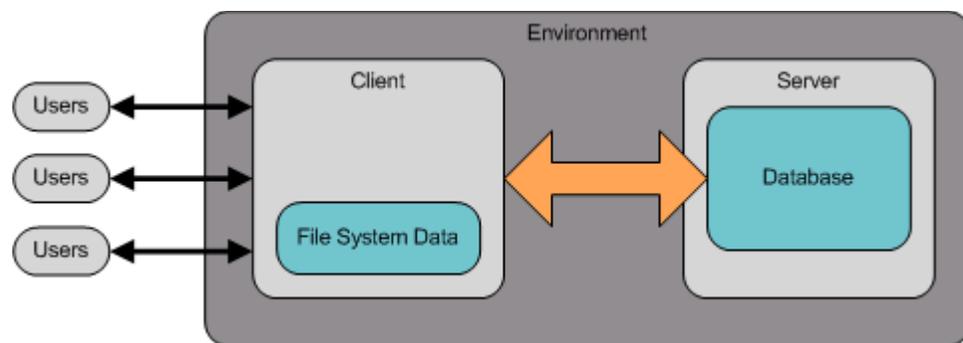
the release, the company can quickly and easily assess the product status and progress toward delivery.

For detailed information about releases, and how to create and configure them, see the *Mercury Deployment Management Configuration Guide*.

Deployment Management Environments

A Deployment Management environment is composed of a unique combination of server, client, database, and file system data that represent one logical group.

Figure 1-7. Deployment Management environment



The environment server represents the main host machine for the environment. This machine may be of any platform type; for example, UNIX or Windows. Typically, the server is a UNIX machine that also houses the database for this environment.

The environment client represents a remote client machine that also serves to identify the specified environment. The client is typically defined if users are doing multiplatform development in a client/server environment, with some development done on UNIX, and some on Windows Server. The client can correspond to a file server that stores client code accessed by users. Many programs, such as transactional forms, have both client and server components; for example, the user interface code and the database objects, respectively. [Figure 1-8 on page 21](#) shows an example of an environment information displayed in the Environment window.

Figure 1-8. Sample environment

Integrating Deployment Management

Integration with Version Control

Mercury Deployment Management provides complete integration with all standard version control systems, including RCS, SCCS, PVCS, ClearCase, CCC/Harvest, and Visual SourceSafe. The Deployment Management outbound API provides all check-in, check-out, and promotion functions during deployment.

File system objects can be deployed to new installations and updated in version control simultaneously. This synchronizes your version control repository, physical installations, and Deployment Management workflows.

Integration with Mercury Demand Management

Mercury Deployment Management serves as a system to manage the physical deployment of changes to your application environment, but of equal importance is the gathering and analysis of issues (requests) related to system stability. To facilitate this request gathering, Deployment Management is integrated with Mercury Demand Management™.

Like Deployment Management, Demand Management is a workflow-based system that features complete audit trail tracking of requests from inception to close. The tight integration between Demand Management and Deployment Management allows packages to be automatically generated from requests.

The requests are then automatically updated upon completion of the package. This closed-loop integration ensures a coherent and consistent view of issue status and removes the need for redundant data across multiple applications.

Integration with Third-Party Applications through Deployment Management Extensions

Mercury Deployment Management Extensions are application-specific modules that automate the deployment and post-deployment steps of objects, files, and patches. Extensions are fully integrated with Mercury Deployment Management to provide a total solution for managing various enterprise applications. The following extensions are available:

- Mercury Deployment Management Extension for Oracle® E-Business Suite™
- Mercury Deployment Management Extension for Oracle® Technology™
- Mercury Deployment Management Extension for E-Commerce Technologies™
- Mercury Deployment Management Extension for SAP® Solutions™
- Mercury Deployment Management Extension for PeopleSoft® Enterprise™

Related Documents

You can find related information in the following documents:

- *Mercury Deployment Management Configuration Guide*
- *What's New and What's Changed*
- *Mercury Demand Management User's Guide*
- *Commands, Tokens, and Validations Guide and Reference*
- *Reports Guide and Reference*
- *Security Model Guide and Reference*
- *Mercury-Supplied Entities Guide* includes descriptions of all Mercury Deployment Management portlets, request types, and workflows.



Chapter
2

Understanding Packages

In This Chapter:

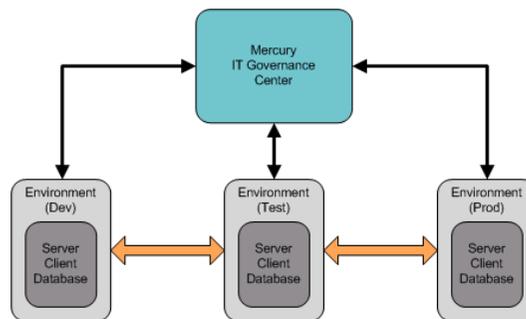
- *About Packages*
 - *Packages in the Standard Interface*
 - *Package Detail Page, Header Section*
 - *Package Detail Page, Status Section*
 - *Package Detail Page, Notes Section*
 - *Package Detail Page, References Section*
 - *Packages in the Workbench*
 - *Package Window, Package Information Section*
 - *Package Window, Package Lines Tab*
 - *Package Window, Status Tab*
 - *Package Window, Notes Tab*
 - *Package Window, References Tab*
 - *Reference Relationships*
 - *Predecessor Relationships*
 - *Successor Relationships*
-

About Packages

Deployment Management automates the migration and deployment of software code, configurations, and content. These objects are grouped into packages and are routed along business processes (workflows) using the Workbench and the standard interface. Workflows automate the movement of each package through their required steps.

Suppose, for example, that you want to copy an operating system patch from the development environment, Dev, to the test environment, Test (*Figure 2-1*). After the operating system patch is thoroughly tested, you can copy the patch to production environment, Prod.

Figure 2-1. Package processing in Deployment Management



A package is the fundamental work unit of Deployment Management. Each package follows a workflow (business process). As with any business process, decisions are made and actions taken that affect the package. Each decision or action becomes a workflow step.

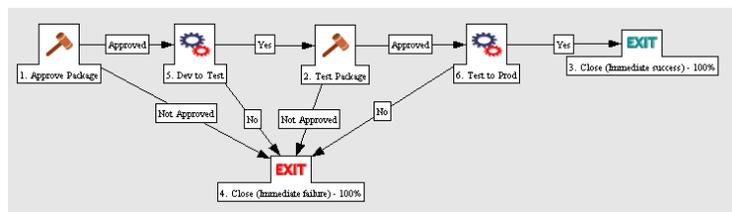
If a decision on a package is required, you must manually update a workflow step to reflect that decision. However, you can configure some actions (for example, migrating a file from one environment to another) to occur automatically. The basic components required to configure a workflow step to automatically migrate a file from one environment to another environment are as follows:

- A package, which defines the file name, location, and type, and the workflow to apply.
- A workflow, which determines the conditions that trigger file migration (workflow step), and the source and destination environments for the file migration.

When you create a package, a single workflow is associated with that package. The workflow defines the steps the package must take. *Figure 2-2* illustrates a simple workflow. In this example, the workflow steps are:

- Approve Package
- Dev to Test (migration)
- Test Package
- Test to Prod (migration)
- Exit

Figure 2-2. Package workflow and workflow steps



You must perform the Approve Package and Test Package workflow steps manually during the course of the workflow. You configure the Dev to Test and Test to Prod workflow steps to execute automatically.

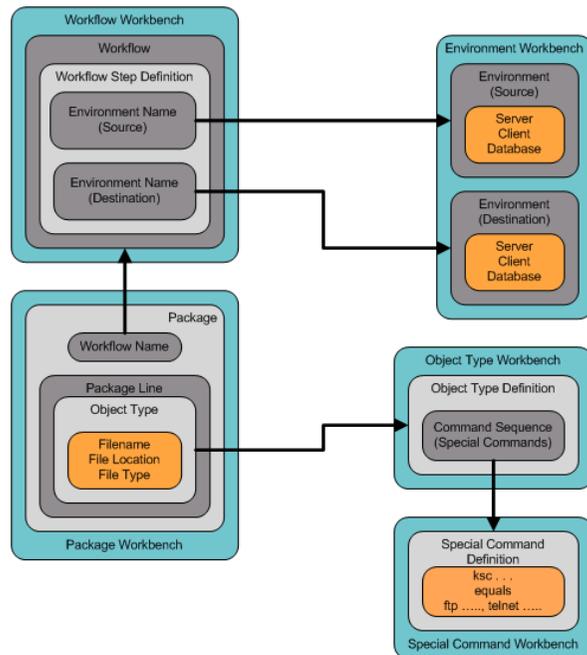
■ ■ Note

To assign a workflow to a package, you do not need to know how to configure a workflow step, or even a workflow. However, you do need to know how to access the information in a workflow, workflow step, and environment.

In *Figure 2-3* on page 28, the package displayed in the Package Workbench is associated with a workflow in the Workflow Workbench. The workflow consists of workflow steps. The migration workflow steps (for example, Dev to Test and Test to Prod) are configured with source and destination environments. You access and configure environments using the Environment Workbench.

The package illustrated in *Figure 2-3* is configured with a single package line, which is, in turn, configured with an object type. A package line can contain only one object type. The object type contains the information required to migrate a file (an object), including the file name, location, and type (ASCII or binary). Depending on the object type definition, additional information may be required.

Figure 2-3. How a package comes together



Object types are defined in the Object Type Workbench. In the Object Type Workbench, you can open and access the command sequence used to migrate the object across environments. These command sequences include special commands defined in the Special Command Workbench.

To create a package, you must know the following:

- The workflow (defined in the Workflow Workbench)
- The environments (specified in the workflow step and defined in the Environment Workbench)
- The object type (defined in the Object Type Workbench)

For information about the licenses and access grants required to work with packages in Mercury Deployment Management, see the *Security Model Guide and Reference*.

Packages in the Standard Interface

You can use the package detail page in the standard interface to view information for a package, add notes and references to the package, and process the package through its workflow. When you add a new package as a reference to a request, you do not access the Package window in the Workbench. Instead, you access a New Package page in the standard interface.

Creating a new package using the New Package page does not create a complete package that you can submit. Instead, the New Package page opens and saves a package in the standard interface. Before you can submit the package, you must use the Package window in the Workbench to add package lines to it. *Figure 2-4 on page 30* shows the detail page for a package that was created from the standard interface as a reference to a request, and contains no lines yet.

Figure 2-4. Package detail page

MERCURY
SIGN OUT

IT Governance Center Dashboard > Package Search > Search Results > Package #30121

Package 30121

Save
Reset

Header

Package No.: <input type="text" value="30121"/>	Package Group: <input type="text" value="Upgrade"/>	Created By: Jane Smith	
Description: <input type="text" value="ERP Patch v4.5.4.7 Migration Test"/>		Created On: November 10, 2004	
Workflow: <input type="text" value="Test Alpha > Test Beta"/>		Status: New	

Assigned User: <input type="text" value="Jane Smith"/>	Priority: <input type="text" value="Normal"/>	Parent:	
Assigned Group: <input type="text" value="ITG Change Management"/>	Package Type: <input type="text" value="Customization"/>	Priority Seq: <input type="text" value="50"/>	

Percent Complete: 0

Status

View Lines: Pending Lines Only All Lines

View Steps: Steps Waiting my Action All Steps Refresh

View All Lines	View Eligible Steps for Test Alpha > Test Beta
Seq Object Name Object Type	⊙ ⊙ ⊙ ⊙
This package does not have any Lines to show.	
Check All Clear All Line Details	Check All Clear All Workflow Action ⏪ ⏩ ⏴ ⏵

Notes

Add Notes

Notes to be added on save:

Existing Notes

Jane Smith (jasmith)	Run env script before adding the patch.
November 11, 2004 11:23:20 AM PST	

References

Requests

Req #	Assigned User	Description	Request Type	Status	% Complete	Relationship	Relationship Details
30330	Jane Smith	ERP Patch v4.5.4.7 Mig...	DEM - Application Enhancement	New	0%	Related to this Package	Informational: Request 30330 is related to Package 30121

Reference Additions

New Reference: Add Highlighted items are actively controlling this Package

References to be added on Save:

Open
Remove

Save
Reset

Package Detail Page, Header Section

On the detail page for a package, the **Header** section displays general information such as who created the package, the date it was created, and its associated workflow. *Figure 2-5* shows a typical **Header** section.

Figure 2-5. Package detail page, Header section

Header			
Package No.:	<input type="text" value="30121"/>	Package Group:	<input type="text" value="Upgrade"/>
Description:	<input type="text" value="ERP Patch v4.5.4.7 Migration Test"/>		Created By:
Workflow:	<input type="text" value="Test Alpha > Test Beta"/>		Created On:
Assigned User:	<input type="text" value="Jane Smith"/>	Priority:	<input type="text" value="Normal"/>
Assigned Group:	<input type="text" value="ITG Change Management"/>	Package Type:	<input type="text" value="Customization"/>
Percent Complete:	<input type="text" value="0"/>		Parent:
			Priority Seq: <input type="text" value="50"/>
			Status: <input type="text" value="New"/>

Package Detail Page, Status Section

The **Status** section on a Package detail page displays the current package status. After a package is first created in the standard interface, the **Status** section does not exist. After the package is submitted, the **Status** section displays the package lines and all of the steps in the associated workflow. *Figure 2-6* shows the **Status** section on a package detail page.

Figure 2-6. Package detail page, Status section

Status			
View Lines:		View Steps:	
<input type="radio"/> Pending Lines Only		<input type="radio"/> Steps Waiting my Action	
<input checked="" type="radio"/> All Lines		<input checked="" type="radio"/> All Steps <input type="button" value="Refresh"/>	
View All Lines		View All Steps for DEV > TEST > PROD	
Seq	Object Name	Object Type	
<input type="checkbox"/>	1 TestFile	RCS Server File	<input type="radio"/> 1: Design Review <input type="radio"/> 2: Evaluate <input type="radio"/> 3: Migrate to Test <input type="radio"/> 4: QA Test
<input type="checkbox"/>	3 KeyConcepts.pdf	PVCS File Migration	Eligible
<input type="button" value="Check All"/> <input type="button" value="Clear All"/> <input type="button" value="Line Details"/> <input type="button" value="View Log"/>		<input type="button" value="Check All"/> <input type="button" value="Clear All"/> <input type="button" value="Workflow Action"/> <input type="button" value="Previous"/> <input type="button" value="Next"/>	

Package Detail Page, Notes Section

The **Notes** section on a package detail page contains fields that you can use to enter package information that you want to store. The **Notes** section is included in the package creation process and continues with package until it is closed.

Figure 2-7 shows a typical **Notes** section. All **Notes** sections include the **Add Notes** subsection. The **Existing Notes** section is created after the first note is created for the request.

Figure 2-7. Package detail page, Notes section

Notes

Add Notes

Notes to be added on save:

Existing Notes

Jane Smith (jasmith)
November 11, 2004 11:23:20 AM PST

Run env script before adding the patch.

Package Detail Page, References Section

A package's detail page **References** section contains additional information about the package. For example, you might add a URL to a package or delete an outdated document from a package. The **References** section is included in the package creation process and continues with the package until the package is closed. References can be added or deleted from the package.

Use the **Reference Additions** section to add a reference to the package. Saved references are organized by reference type; for example, all saved reference packages are saved in the **Packages** section. For some references, a functional dependency between the package the referenced entity can be created. For more information about references and their functional dependencies, see [Reference Relationships](#) on page 40.

Figure 2-8 on page 33 shows the **References** section on the Package detail page.

Figure 2-8. Package detail page, References section

References								
Requests								
Req #	Assigned User	Description	Request Type	Status	% Complete	Relationship	Relationship Details	
30330	Jane Smith	ERP Patch v4.5.4.7 Mig...	DEM - Application Enhancement	New	0%	Related to this Package	Informational: Request 30330 is related to Package 30121	

Reference Additions

New Reference: Attachment ■ Highlighted items are actively controlling this Package

References to be added on Save:

Packages in the Workbench

You can use the Package Workbench to view existing packages. From a Package window, you can add notes and references to the package, and process the package through its workflow. From the Package window, you can create and submit a package, process it through its workflow, cancel it, or delete it.

Figure 2-9 shows the detail page for a package.

Figure 2-9. Package window

Package: 30121

Package Information

Package No: 30121 Package Group: Upgrade Created By: Jane Smith
 Description: ERP Patch v4.5.4.7 Migration Test Created On: November 10, 2004
 Workflow: Test Alpha > Test Beta Package Status: New
 Assigned User: Jane Smith Priority: Normal Parent:
 Assigned Group: It Administrator Package Type: Customization Priority Seq: 50
 Percent Complete: 0

Package Lines | Status | Notes | References | User Data

Seq	Object Type	App Code	
1	File Client->Client	(None)	Client File Name: env4.5.4.7.zip
2	File Client->Client	(None)	Client File Name: patch4.5.4.7.zip
3	File Client->Client	(None)	Client File Name: config4.5.4.7.zip

Ready

Package Window, Package Information Section

The **Package Information** section of the Package window displays general information about the package; for example, who created the package, the date, and the associated workflow. The **Package Information** section is included in the package creation process and stays with the package. Typically, when creating a package, you must complete some fields in this section. *Figure 2-10* shows a typical general information section. The information in a package's detail page Header section is found in the Package window's general information section.

Figure 2-10. Package window, Package Information section

The screenshot shows a software window titled "Package: 30121" with a "Package Information" section. The fields are as follows:

- Package No.: 30121
- Package Group: Upgrade
- Created By: Jane Smith
- Description: ERP Patch v4.5.4.7 Migration Test
- Created On: November 10, 2004
- Workflow: Test Alpha > Test Beta
- Package Status: New
- Assigned User: Jane Smith
- Priority: Normal
- Parent: (empty)
- Assigned Group: Administrator
- Package Type: Customization
- Priority Seq: 50
- Percent Complete: 0

Below the fields is a table with the following data:

Seq	Object Type	App Code	
1	File Client->Client	(None)	Client File Name: env4.5.4.7.zip
2	File Client->Client	(None)	Client File Name: patch4.5.4.7.zip
3	File Client->Client	(None)	Client File Name: config4.5.4.7.zip

At the bottom of the window are buttons for "New Line", "Edit Line", "Copy Line", "Remove Line", "Submit", "OK", "Save", and "Cancel". The status bar at the very bottom says "Ready".

Package Window, Package Lines Tab

The **Package Lines** tab lists the package lines associated with a package. Package lines define the object migrating from one environment to another environment. Each package line defines one object (file). Typically, objects (files) are defined by their object type (for example, **File Client > Client** or **File Migration**), filename, location, and file type. You cannot define multiple objects on a single package line. *Figure 2-11* shows the **Package Line** tab with package lines displayed.

Figure 2-11. Package window, Package Lines tab

Package: 30121

Package Information

Package No.: 30121 Package Group: Upgrade Created By: Jane Smith
 Description: ERP Patch v4.5.4.7 Migration Test Created On: November 10, 2004
 Workflow: Test Alpha > Test Beta Package Status: New
 Assigned User: Jane Smith Priority: Normal Parent:
 Assigned Group: it Administrator Package Type: Customization Priority Seq: 50
 Percent Complete: 0

Seq	Object Type	App Code	Client File Name
1	File Client->Client	(None)	Client File Name: enw4.5.4.7.zip
2	File Client->Client	(None)	Client File Name: patch4.5.4.7.zip
3	File Client->Client	(None)	Client File Name: config4.5.4.7.zip

New Line Edit Line Copy Line Remove Line

Submit OK Save Cancel

Ready

Package Window, Status Tab

The **Status** tab in the Package window displays the current package status. When a package is first created, the **Status** tab displays the package lines. Once the package is saved or submitted, the **Status** tab displays the package lines and all of the workflow steps in the associated workflow.

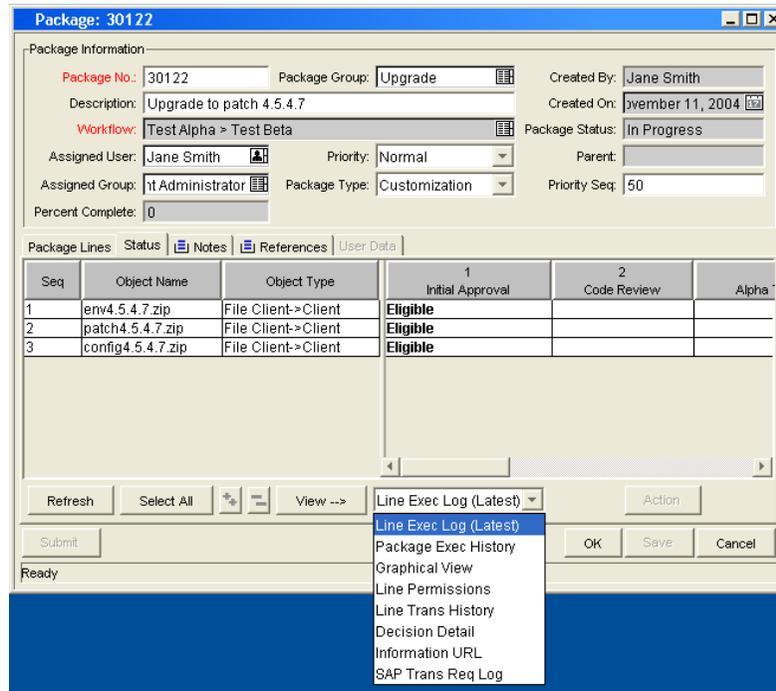
Use the **Status** tab to see:

- All the possible actions that can be performed on the package
- All the actions performed to date
- Pending actions that must be performed to resolve the package

After a package is submitted, you can use the **Status** tab to process it through the workflow. Each package line includes all of the steps in the workflow. If a

workflow step is eligible for action, the **Initial Approval** field for that step displays the value **Eligible** (Figure 2-12).

Figure 2-12. Package window, Status tab



After a package is submitted, you can expand a package line to display any of the subworkflow steps defined in the workflow. To view the subworkflow steps, above the subworkflow step, click **Expand**.

To expand all subworkflows on the **Status** tab, click **Expand All**. To hide expanded subworkflow steps, above the subworkflow step, click **Collapse**. To hide all subworkflow steps, click **Collapse All**. Clicking **Expand** expands the package line horizontally to display all of the steps within the subworkflow.

The header text for subworkflow steps is displayed in a different color than the header for workflow steps. The step numbering includes additional decimal places to indicate the level of the workflow. For example, if step 3 of a top-level workflow is a subworkflow, its steps are numbered as 3.1, 3.2, 3.3, and so on. Similarly, if the second step in that subworkflow is also a subworkflow step, its steps are numbered as 3.2.1, 3.2.2, 3.2.3, and so on.

From the **Status** tab, you can access several log files, a graphical view of the workflow, transaction histories, and detail files related to the package. Use these files, views, and histories to help you as you process your package.

Accessing the Line Execution Log

You can select a package line and get the details of the execution of that particular line by viewing the line execution log. The line viewed is a subset of the overall execution batch log.

If the package line has been configured to migrate to multiple environments using an environment group, the execution log returns an execution summary and link for each destination environment. To see the detailed execution log for an environment, click the associated link.

Accessing the Package Execution History

The Package Execution History displays the execution history of each package line. It displays the workflow step name, execution date, and outcome (**Succeeded** or **Failed**) of each execution step.

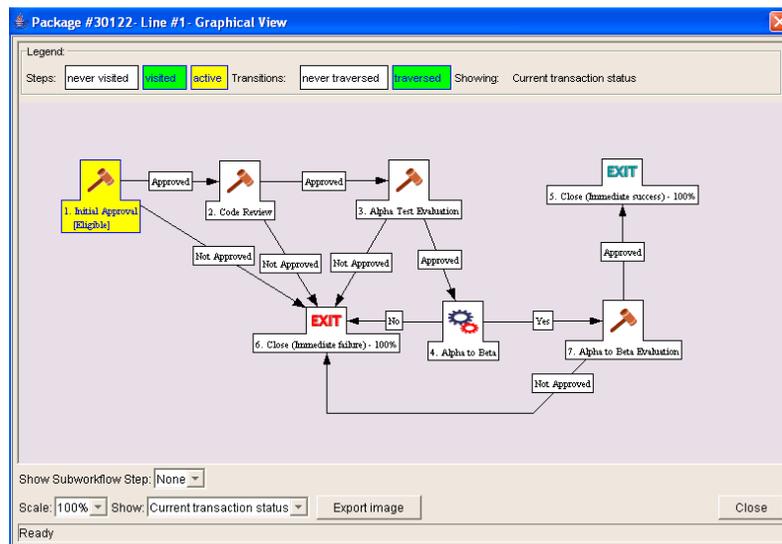
If the package line has been configured to migrate to multiple environments using an environment group, the execution log returns an execution summary and link for each destination environment. Click the link to get the detailed execution logs for each environment.

Graphical View

To see the workflow, select the Graphical View window (*Figure 2-13 on page 38*). Use the legend at the top of the window to determine which workflow steps have been visited and which steps are active. This image displays all of the workflow steps and subworkflows used in the workflow definition.

To obtain a graphical view of a particular subworkflow, select the subworkflow name from the **Show Subworkflow Step** list at the bottom of the window. This opens a new window that displays the graphical view of that subworkflow.

Figure 2-13. Graphical View window for a package



Viewing Line Permissions

Workflow permissions are established during the initial workflow configuration. The Line Permissions view lists all users in security groups that have been enabled for the related workflow step. If you have questions regarding the permissions set for a particular workflow step, see your application administrator.

Viewing Line Transaction History

You can view the transaction history of each package line. This provides detailed information on how the package line has proceeded through a Deployment Management workflow. Information, for example, as the date of the transaction event, the user linked to the workflow step, the workflow step name and number, status, and results are displayed.

To display information on the subworkflow steps that have been traversed more than once by a single package line, select **Full transaction history**.

Selecting **Current transaction status** displays only information on the most recent transaction up through that step.

Viewing the Decision Detail

The Decision Detail displays the decision history of each workflow step. It displays the date, resource, and decision of the decision workflow step.

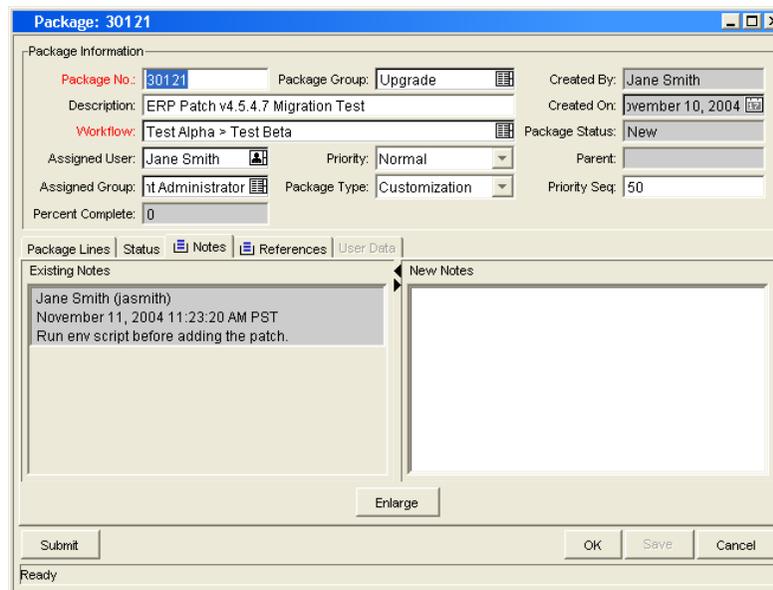
Viewing the Workflow Step Information URL

Workflow steps can be configured to include a URL that provides information regarding the workflow step. If the information URL is configured for a workflow step, you can view the URL in the Workflow Step Information URL window.

Package Window, Notes Tab

The **Notes** tab in the Package window (*Figure 2-14*) contains fields where notes on and information about the package can be entered and stored. The **Notes** section is included in the package creation process and continues with the package until it is closed. Typically, when creating a package, you do not need to add a note to the package.

Figure 2-14. Package window, Notes tab



Package Window, References Tab

A package window's **References** tab (*Figure 2-15*) displays additional information about the package. For example, you might add a URL to a package or delete an outdated document from a package. The **References** tab is included in the package creation process and continues with the package until the package is closed. You can add references to or delete them from the package.

Use the **Reference Addition** section to add a reference to the package. Saved references are organized by reference type; for example, all saved reference packages are saved in the Packages section. For some references, a functional dependency between the package and the referenced entity can be created. For more information about references and their functional dependencies, see the next section.

Figure 2-15. Package window, References tab

Package: 30121

Package Information

Package No.: 30121 Package Group: Upgrade Created By: Jane Smith

Description: ERP Patch v4.5.4.7 Migration Test Created On: November 10, 2004

Workflow: Test Alpha > Test Beta Package Status: New

Assigned User: Jane Smith Priority: Normal Parent:

Assigned Group: it Administrator Package Type: Customization Priority Seq: 50

Percent Complete: 0

Package Lines | Status | Notes | References | User Data

Type	Name	Details	Status	% Complete	Description
Request	30330	DEM - Application Enhancement	New	0%	ERP Patch v4.5.4.7 Migratio

Items in **Bold** are actively controlling this Package

New Reference: Attachment Add Remove Open Reference

Submit OK Save Cancel

Ready

Reference Relationships

For some references, you can create a functional dependency between the package and the referenced entity. For example, you can specify that a request is a predecessor to the package. This means the package cannot continue until the request is closed.

- Predecessor Relationships.** Predecessor relationships dictate that action is not allowed on one entity until the referenced entity closes. For example, a package cannot complete any workflow action until the referenced task is closed (**Completed**, **Bypassed**, or **Cancelled**). The package status is **Pending Task**. After the task closes, the package can be acted upon once more. This does not stop other request fields from being edited.
- Successor Relationships.** Successor relationships dictate that action is not allowed on the referenced entity until the entity closes. For example, a

referenced task cannot change status until the original package is closed. The task status is **Pending Request**. When the package closes, the task can again be changed. This does not prevent other task fields from being edited.

Table 2-1 lists the references, their definitions, and possible dependency relationships.

Table 2-1. References and relationships (page 1 of 3)

Reference	Reference Description	Possible Relationships	Relationship Description
Attachment	Attach a file from a local machine to the current package. The attached file is copied to the server.	Standard attachment interaction	(Informational) The attachment is related to this package.
Packages (Existing)	Reference an existing package.	Duplicate package	(Informational) The referenced package is a duplicate of the package.
		Run before this package in a release	(Informational) In a release, the selected package must be run before this package.
		Run after this package in a release	(Informational) In a release, the selected package must be run after this package.
		Predecessor	(Blocking) Action is not allowed on the package until the referenced package closes.
		Successor	(Blocking) Action is not allowed on the referenced package until the package closes.

Table 2-1. References and relationships (page 2 of 3)

Reference	Reference Description	Possible Relationships	Relationship Description
Packages (New)	New packages can also be created from a package. If configured as part of the workflow, you can spawn a package from a workflow step. When this happens, a reference to that package is automatically generated, establishing a two-way link between the packages.	Run before this package in a release	(Informational) In a release, the selected package must be run before this package.
		Run after this package in a release	(Informational) In a release, the selected package must be run after this package.
		Predecessor	(Blocking) Action is not allowed on the package until the referenced package closes.
		Successor	(Blocking) Action is not allowed on the referenced package until the package closes.
Programs	Reference a program in Mercury Program Management™.	Related to This program	(Informational) Selected program is related to this package.
Projects	Reference a project in Mercury Project Management™.	Related to This package	(Informational) Selected project is related to this package.
Releases	Reference a release in Deployment Management.	Contains this package	(Informational) The package is contained in the selected release.
Request (Existing)	Reference an existing request.	Parent of this package	(Informational) The referenced request is the parent of the package.
		Related to this package	(Informational) Referenced request is related to this package.
		Predecessor	(Blocking) Action not allowed on this package until the referenced request closes.
		Successor	(Blocking) Action not allowed on the referenced request until this package closes.

Table 2-1. References and relationships (page 3 of 3)

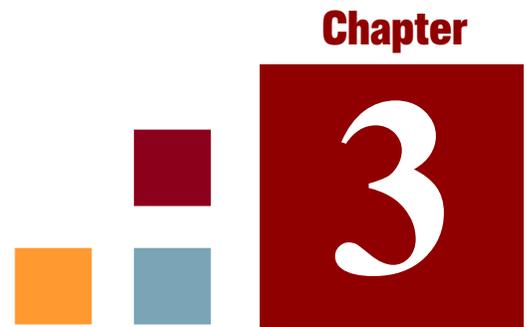
Reference	Reference Description	Possible Relationships	Relationship Description
Requests (New)	New packages can be created from existing packages. Also, if configured as part of the workflow, you can spawn a package from a package. When this happens, a reference to that package is automatically generated, establishing a two-way link between the packages.	Parent of this package	(Informational) The referenced request is the parent of the package.
		Related to this package	(Informational) Referenced request is related to this package.
		Predecessor	(Blocking) Action not allowed on this package until the referenced request closes.
		Successor	(Blocking) Action not allowed on the referenced request until this package closes.
Tasks	Reference a task in a Mercury Project Management.	Related to this package	(Informational) The referenced task is related to this package.
URL	Reference a Web address. Documents at the URL must be in MIME format.	Standard URL interaction	(Informational) The URL is related to this package.

Predecessor Relationships

Predecessor relationships dictate that action cannot be performed on one entity until the referenced entity closes. For example, a package cannot complete any workflow action until the referenced task is closed (**Completed**, **Bypassed**, or **Cancelled**). The package status is **Pending Task**. After the referenced task closes, the package can be acted on again. This does not prevent the editing of other request fields.

Successor Relationships

Successor relationships dictate that action is not allowed on a referenced entity until the entity closes. For example, a referenced task cannot change its status until the original package is closed. The task status is **Pending Request**. After the package closes, the task can be acted on. This does not stop other task fields from being edited.



Chapter
3
Creating Packages

In This Chapter:

- *Overview of Creating Packages*
 - *Before You Create Packages*
 - *Creating Packages*
 - *Creating Packages Using the Workbench*
 - *Creating Packages by Copying Existing Packages*
 - *Creating Packages as References from Packages*
 - *Creating Packages as References from Requests*
 - *Entering General Information for Packages*
 - *Adding Lines to Packages*
 - *Adding Notes to Packages*
 - *Adding References to Packages from the Package Workbench*
 - *Adding Attachments as Package References*
 - *Adding Existing Packages as Package References*
 - *Adding New Packages as Package References*
 - *Attaching Programs as Package References*
 - *Attaching Projects as Package References*
 - *Adding Releases as References*
 - *Adding Existing Requests as Package References*
 - *Adding New Requests as Package References*
 - *Adding Tasks as Package References*
 - *Adding URLs as Package References*
 - *Submitting Packages*
-

Overview of Creating Packages

This chapter provides instructions on how to create and submit packages to deploy software and application changes. Although you typically use the Workbench to create packages, you can also create them by copying an existing package, or create them as referenced entities.

After you create a package, you must submit it to start the workflow (business process). For information on how to submit a package, see [Submitting Packages](#) on page 77.

Before You Create Packages

Before you create a package, make sure you understand the following:

- The workflow (defined in the Workflow Workbench)
- The environments (specified in the workflow step and defined in the Environment Workbench)
- The object type (defined in the Object Type Workbench)

Selecting Workflows

Each package is processed through a predefined workflow. You must select a workflow to apply to each package you create. It is important that you route your packages through workflows that accurately model the appropriate business process.

To view the graphical layout of a Deployment Management workflow:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Configuration > Workflows**.

The Workflow Workbench window opens.

4. To display all of the workflows, in the Workflow Workbench, click **List**.



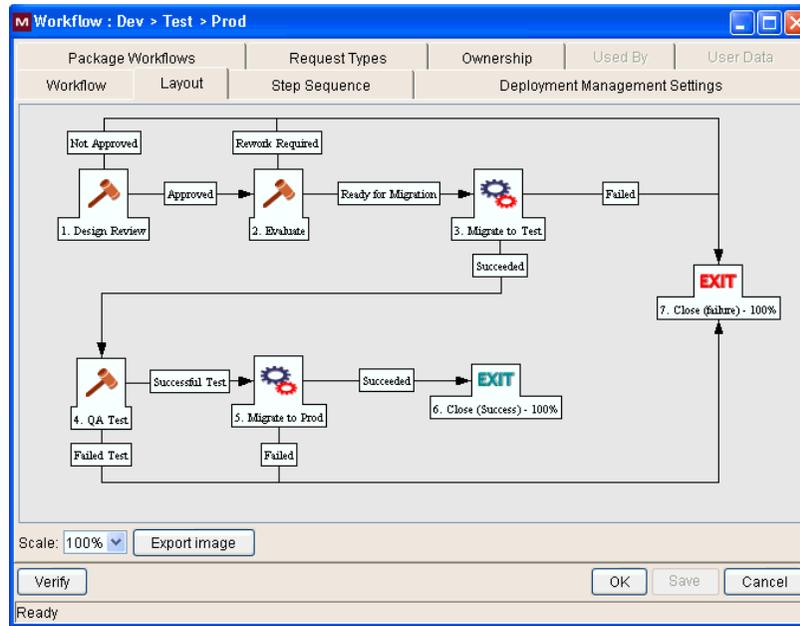
Note

To restrict your search, you would specify one or more of the query parameters.

The **Results** tab lists all workflow records.

5. Select a workflow for which the **Workflow Scope** column displays **Packages**, and then click **Open**.

The Workflow <Workflow Name> window opens to the **Layout** tab, which displays a graphical representation of the workflow steps.

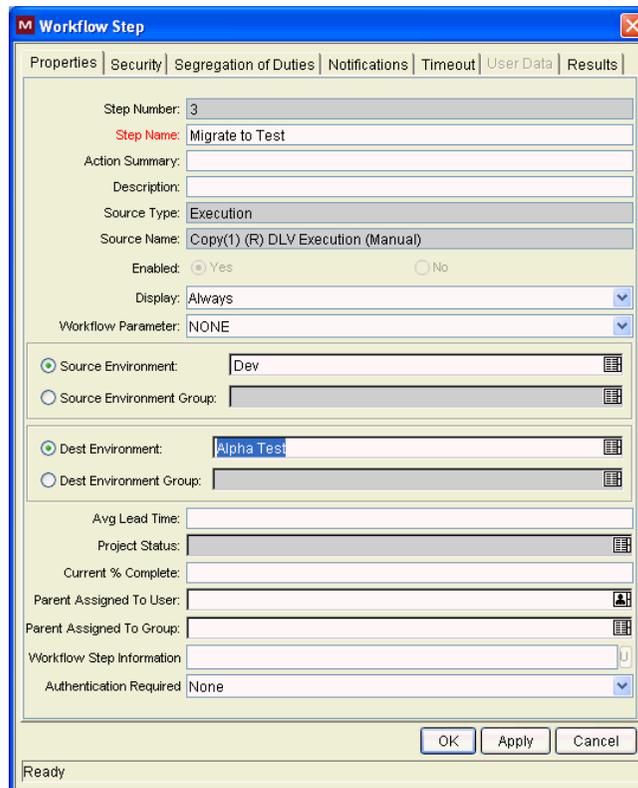


Check to ensure that the workflow follows the correct business process for your package. If it does not, open a different workflow.

6. Double-click a migration workflow step (for example, Migrate to Test or Migrate to Prod).

The Workflow Step window for that step opens to the **Properties** tab.

7. Verify that the source and destination environments listed on the tab are correct.



8. To view additional information in the Workflow Step window, click the following tabs:
 - **Security.** This tab is used to define the security groups allowed to act on the selected step. All enabled security groups are displayed in either the **Available Security Groups** field or the **Linked Security Groups** field. Each user who belongs to a linked security group can perform the actions defined for that particular workflow step.
 - **Segregation of Duties.** Use this tab to specify workflow steps to be segregated from the selected step in the workflow. This segregation prevents a given user from acting on both steps in the workflow. For example, you might use it to prevent the person who requests a change from then deploying that change to the production system without approval from a third party.

You can also use this tab to specify that the package creator cannot act on the selected workflow step.

- **Notifications.** Use this tab to attach email notifications to the workflow step. Email messages can be sent whenever a particular event (for example, the step completes with a specific result) occurs at the step.
 - **Timeout.** This tab displays timeout intervals set for this workflow step.
 - **User Data.** This tab displays any custom fields for each workflow step.
 - **Results.** This tab lists all of the possible valid results for the step.
9. Close the Workflow Step window.

Verifying Environments

Some workflow steps include specify source and destination environments. If you are not sure the specified environments are correct, you can open them in the Environment Workbench.

To access an environment:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench.**

The Workbench opens.

3. From the shortcut bar, select **Environments > Environments.**

The Environment Workbench opens.

4. To display all of the environments, click **List.**

To restrict the search, specify one or more query parameters.

The **Results** tab lists the environment records.

5. Select one or more of the listed environments, and then click **Open.**

The Environment <*Environment Name*> window opens.

6. Check the settings for the server, client, and database.
7. Close the Environment window.

Verifying Object Types

When creating a package, the available object types are displayed in an auto-complete list. To view the available object types before you start to create a package, you can open an object type in the Object Type Workbench. *Table 3-1* lists the Mercury-supplied object types.



Note

The Mercury Deployment Management Extensions provide additional object types.

Table 3-1. Object types (page 1 of 2)

Object Type	Description
File Client>Client	File copy from client to client.
File Migration	File copy from environment to environment.
ITG Data Source Migrator	Migrates an IT Governance data source migrator.
ITG Module Migrator	Migrates an Mercury IT Governance Center module from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG Object Type Migrator	Migrates an Mercury IT Governance Center object type from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG Overview Page Section Migrator	Migrates an Mercury IT Governance Center overview page section from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG Portlet Definition Migrator	Migrates an Mercury IT Governance Center portlet definition from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG Project Template Migrator	Migrates an Mercury IT Governance Center project template from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG Report Type Migrator	Migrates an Mercury IT Governance Center report type from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.

Table 3-1. *Object types (page 2 of 2)*

Object Type	Description
ITG Request Header Type Migrator	Migrates an Mercury IT Governance Center request header type from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG Request Type Migrator	Migrates an Mercury IT Governance Center request type from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG Special Command Migrator	Migrates an Mercury IT Governance Center special command from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG User Data Context Migrator	Migrates an Mercury IT Governance Center user data context from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG Validation Migrator	Migrates an Mercury IT Governance Center validation from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
ITG Workflow Migrator	Migrates an Mercury IT Governance Center workflow from one Mercury IT Governance Center environment to another Mercury IT Governance Center environment.
OS/390 JCL Migration	Migrates an OS/390 JCL from a source environment to the destination environment.
PVCS File Migration	Migrates an object from the PVCS archive to the destination environment.
RCS File Migration	This object manages the check out and distribution of code in RCS.
RCS Server File	Old file will be checked in on the destination environment. New file will be checked in and checked out Read-Only on the destination environment.
SourceSafe File Migration	Manages the check out and promotion of files from the SourceSafe version control archive.

To open an object type:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.
The Workbench opens.
3. From the shortcut bar, select **Deployment Mgmt > Object Types**.
The Object Type Workbench window opens to the **Query** tab.
4. To display all object types, click **List**.
To restrict the search, specify one or more query parameters.
5. Select one or more of the listed object types, and then click **Open**.
The Object Type *<Object Type Name>* window opens.
6. Click the **Commands** tab.
7. Check the commands that the object type calls.
8. Close the Object Type *<Object Type Name>* window.

Creating Packages

Although you typically use the Workbench to create packages, you can also create one by copying an existing package, or by creating a new package as a referenced entity. This section provides information about how to use all three methods.

Creating Packages Using the Workbench

Packages are used to deploy software and application changes. This section covers how to define the contents of a package, including adding package lines and specifying the appropriate deployment workflow. For each instance of a software migration submission, a new package needs to be created.

To create a package from the Workbench:

1. Log on to Mercury IT Governance Center.

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

The Workbench opens.

- From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

- Click **New Package**.

The Package window opens. You use this window to define the package contents.

The screenshot shows the 'Package: 30121' window. The 'Package Information' tab is active, displaying the following fields:

- Package No.: 30121
- Package Group: Upgrade
- Created By: Jane Smith
- Description: ERP Patch v4.5.4.7 Migration Test
- Created On: November 10, 2004
- Workflow: Test Alpha > Test Beta
- Package Status: New
- Assigned User: Jane Smith
- Priority: Normal
- Parent: (empty)
- Assigned Group: Administrator
- Package Type: Customization
- Priority Seq: 50
- Percent Complete: 0

The 'Package Lines' tab is also visible, showing a table with the following data:

Seq	Object Type	App Code	
1	File Client->Client	(None)	Client File Name: env4.5.4.7.zip
2	File Client->Client	(None)	Client File Name: patch4.5.4.7.zip
3	File Client->Client	(None)	Client File Name: config4.5.4.7.zip

At the bottom of the window, there are buttons for 'New Line', 'Edit Line', 'Copy Line', 'Remove Line', 'Submit', 'OK', 'Save', and 'Cancel'. The status bar at the bottom left shows 'Ready'.

- Under **Package Information**, complete the fields.

For information on how to complete the fields in section, see [Entering General Information for Packages](#) on page 59.

- On the **Package Lines** tab, add the package lines to the package.

Package lines capture information specifically related to each object that you are migrating from one environment to another environment. For information on how to add package lines, see [Adding Lines to Packages](#) on page 60.

- On the **Notes** tab, add any notes to the package.

For information on how to add notes, see [Adding Notes to Packages](#) on page 63.

8. On the **References** tab, you can add references to the package.

For information on how to add references, see *Adding References to Packages from the Package Workbench* on page 64.

9. To save the package and close the Package window, click **OK**. To save the package and leave the Package window open, click **Save**.

Creating Packages by Copying Existing Packages

You can create a package by copying an existing package (either new or saved). You can specify which sections of the existing package you want to include in the new package.

To create a package from an existing package:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

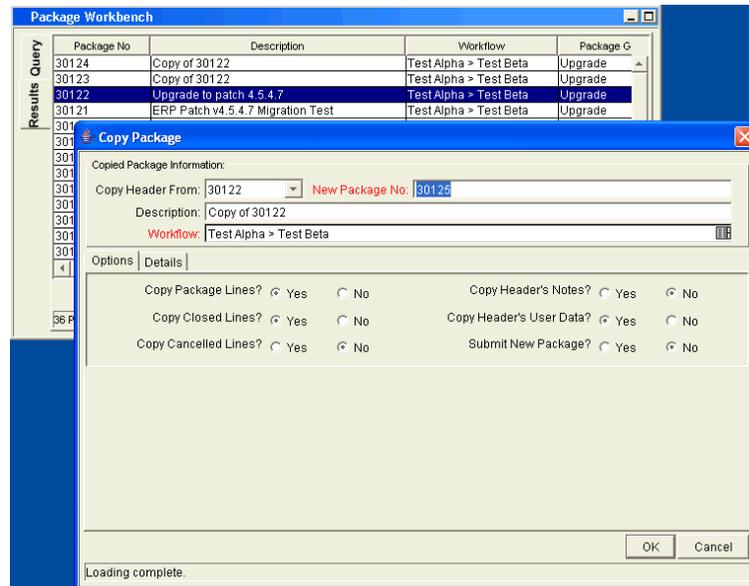
The Package Workbench opens.

4. On the **Query** tab, click **List**.

The Package Workbench **Results** tab lists all existing packages.

5. On the **Results** tab, select a package, and then click **Copy**.

The Copy Package window opens to the **Options** tab. *Table 3-2* on page 56 lists the options available on the tab.



6. Select the sections to include in the new package, and then click the **Details** tab.

The **Details** tab lists all of the package lines that were copied from the original package. By default, all of these lines are included in the new package.

7. To exclude a listed package line from the new package:
 - a. In the **Package Lines to Copy** table, select the line.
 - b. At the bottom of the tab, click **Exclude**.

The Package window opens and displays the information you selected from the copied package.

8. To view the parameters that describe a listed package line:
 - a. In the **Package Lines to Copy** table, select the line.
 - b. At the bottom of the tab, select the **Show Parameters** checkbox.

Additional columns in the **Package Lines to Copy** table display the package line parameters.

- c. To view all of the parameters, use the horizontal scrollbar.

9. Click **OK**.

Table 3-2. Copy package options

Option	Description
Copy Package Lines	Copies all of the package lines (with a status other than Closed or Cancel) in the existing package into the new package. The default is Yes .
Copy Closed Lines	Copies all of the closed package lines in the existing package into the new package. The default is Yes .
Copy Cancelled Lines	Copies all of the cancelled package lines in the existing package into the new package. The default is No .
Copy Header's Notes	Copies all of the notes attached to the existing package to the new package. The default is No .
Copy Header's User Data	If applicable, copy the user data from the existing package to the new package.
Submit New Package	After the package is copied, submit the package. The default is No . If you select No , your package is not submitted. To submit the package, you must open the copied package and click Submit .

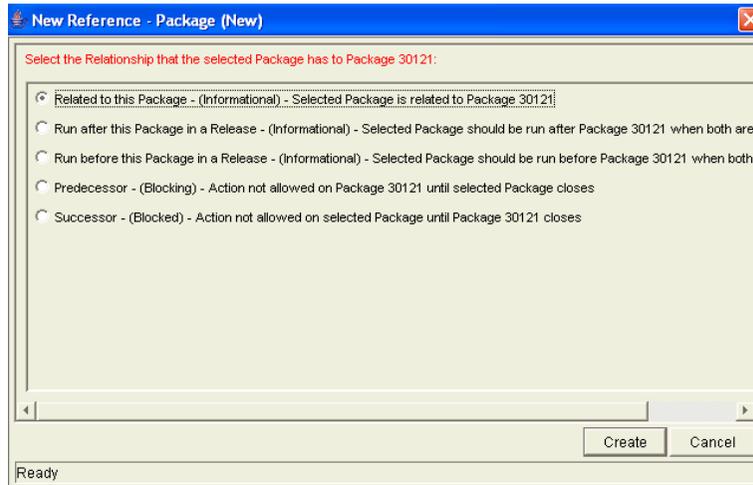
Creating Packages as References from Packages

You can create a new package as part of referencing a new package to an entity such as a request or another package.

To create a new package as a reference to an existing package:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.
The Workbench opens.
3. From the shortcut bar, select **Deployment Mgmt > Packages**.
The Package Workbench opens to the **Query** tab.
4. Use the available fields to provide your search criteria, and then click **List**.
The **Results** tab lists the packages that match your search criteria.
5. Double-click the row for the package you want to open.
The Package window opens.

6. Click the **References** tab.
7. From the **New Reference** list at the bottom on the tab, select **Package (New)**.
The New Reference - Package (New) window opens.



8. Under **Select the Relationship that the selected Package has to Package # <Package Number>**, select the option that describes the relationship that the new package is to have to the existing package.



Note

For descriptions of the possible relationships between referenced entities, see [Table 2-1 on page 41](#).

9. Click **Create**.
A new Package window opens.
10. Provide all required information, and any optional information you want to associate with the new package.
11. Click **OK** or **Save**.
12. In the Package window that displays the original package, click **OK**.

Creating Packages as References from Requests

You can create a package as a reference from a request. When creating a package from a request, you do not open the Workbench, even if you have the correct permissions. Instead, you work in the standard interface using the Create New Package and New Package pages. When you create a package as a reference from a request, you cannot add package lines, which means you cannot complete and submit the package. Adding package lines to a package and submitting a package requires using the Package Workbench.

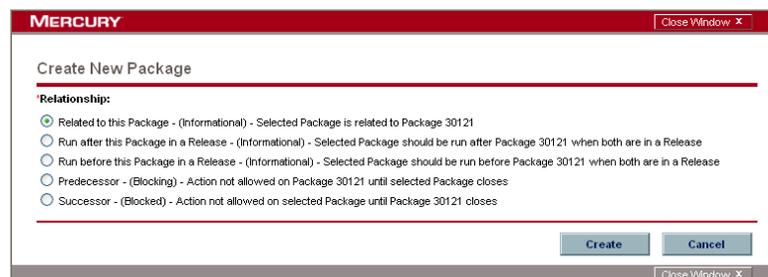
To create a package as a reference from a request:

1. Log on to the Mercury IT Governance Center.
2. Open a request.

You can open an existing request or create a new request from the menu bar. For information about how to open an existing request or create a request, see the *Mercury Demand Management User's Guide*. The detail page for the request opens.

3. In the request's detail page, scroll to the **References** section.
4. From the **New Reference** list, select **Package (New)**.
5. Click **Add**.

The Create New Package page opens.



6. Select the relationship, and then click **Create**.

The New Package page opens.

7. Select the workflow from the list.
8. Type any optional information you want to keep as part of the package record.

9. Click **Save**.
10. On the detail page for the request, click **Save**.

Entering General Information for Packages

To complete the general information section for packages:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. From the **Query** tab, click **New Package**.

The Package window opens. You use the Package window to define the package contents. The package number is assigned automatically.

5. Complete the required fields.
 - The **Package No.** field is required. The package number is automatically completed.
 - The **Workflow** field is required. Select the workflow from the auto-complete list.
6. In the Package window, click **OK** or **Save**.

The package line is added to the package.

Adding Lines to Packages

Package lines define the object migrating from one environment to another environment. Each package line defines just one object (often a file). Typically, objects are defined by their object type (for example, **File Client > Client** or **File Migration**), name, location, and, in the case of files, file type.

To add a line to package:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. Open a package or create a new package.

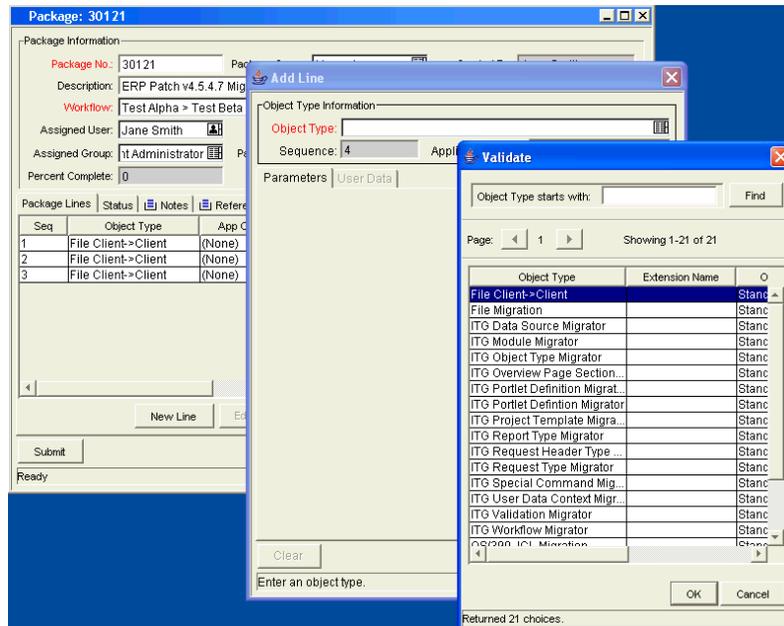
The Package window opens. You use this window to define the package contents.

5. Click **New Line**.

The Add Line window opens.

6. In the **Object Type** field, select the object type for the object (file) you plan to migrate using the auto-complete icon.

Selecting an object type dynamically updates the fields in the Add Line window.



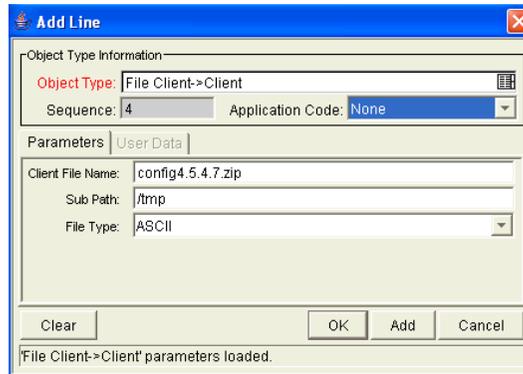
7. In the Add Line window, complete the associated fields, and then click **OK**.

In some cases, you can add multiple package lines from a single Add Line window. You can do this when you have multiple objects (files) to migrate that are in the same location.

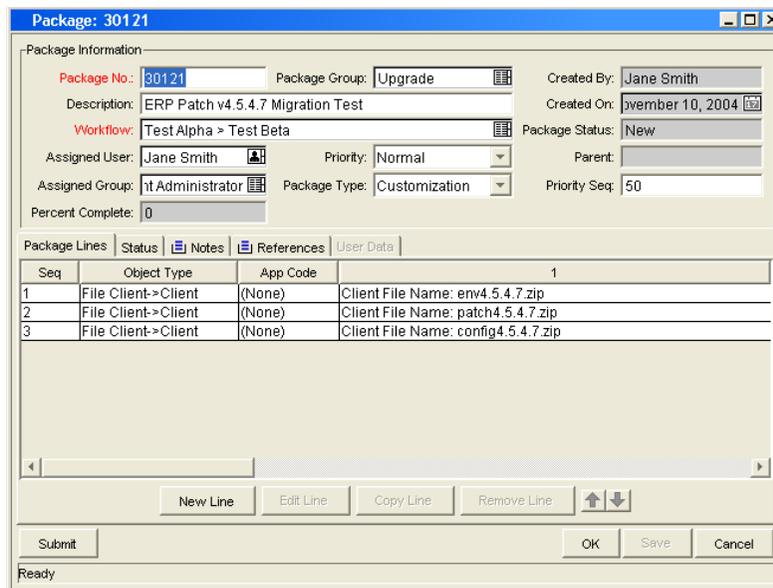
To add multiple lines from a single Add Line window, the following must be true:

- Package lines all follow the same workflow and use the same object type.
- The object type parameters include at least one auto-complete list field with an enabled **Multi** button.

By selecting multiple files, you add multiple package lines to the package. Remember, you cannot define multiple objects on a single package line.



The contents of the Add Line window are included in the Packages window as a package line. The Add Line window closes. In the Add Line window, clicking **Add** includes the package line in the Package window and the Add Line window remains open.



8. In the Package window, click **OK** or **Save**.

The package line is added to the package. To start the package on its associated workflow, click **Submit**. Submitting the package to the workflow generates entries under the **Status** tab for each of the workflow steps defined for the workflow. As each step in the workflow is ready to be processed, the step shows an Eligible status. You can view the progress of each package line by clicking the **Status** tab. For information about how to submit a package, see [Submitting Packages](#) on page 77.

Adding Notes to Packages

In the Package window, the **Notes** tab (*Figure 3-1*) displays additional package information. Notes are attached to the package, and not to individual package lines. If you have a note that is specific to a package line, you must include the package line reference as part of the note text.

Figure 3-1. Package window, Notes section

You can add notes to existing packages, packages you are creating, and to submitted packages that have not been closed. You cannot add a note to a closed package.

To add a note, open a package and click the **Notes** tab. In the **New Notes** field, type the text of the note, and then click **Save**. The note is added to the **Existing Notes** field.

Adding References to Packages from the Package Workbench

You can use the **References** section of the package detail page to add references of several types to a package. For example, you might want to add a document or URL to a package. You can add the following types of references to a package:

- Attachments
- Packages (new and existing)
- Projects
- Releases
- Requests (new and existing)
- Tasks
- URLs

For some references (such as requests and other packages), you can create a functional dependency to the original package. For example, you can specify that a request is a predecessor to the package. This means the package cannot continue until the request is closed. For a list of the references and their possible dependency relationships, see [Reference Relationships on page 40](#).

This section provides details on how to use the Package Workbench to add the different entity types as package references. For information on how to add package references from the standard interface, see [Adding References to Packages from the Standard Interface on page 86](#).

Adding Attachments as Package References

This section provides the steps you use to add an attachment as a package reference.

To add an attachment as a package reference:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. Open a package.

5. In the Package window, click the **References** tab.

The screenshot shows the 'Package: 30121' window. The 'Package Information' section includes fields for Package No. (30121), Package Group (Upgrade), Created By (Jane Smith), Description (ERP Patch v4.5.4.7 Migration Test), Created On (November 10, 2004), Workflow (Test Alpha > Test Beta), Package Status (New), Assigned User (Jane Smith), Priority (Normal), Parent, Assigned Group (it Administrator), Package Type (Customization), Priority Seq (50), and Percent Complete (0). The 'References' tab is selected, showing a table with one entry: Request 30330, DEM - Application Enhancement, New, 0%, ERP Patch v4.5.4.7 Migratio. Below the table, there are buttons for 'Add', 'Remove', and 'Open Reference'. The 'New Reference' dropdown is set to 'Attachment'.

Type	Name	Details	Status	% Complete	Description
Request	30330	DEM - Application Enhancement	New	0%	ERP Patch v4.5.4.7 Migratio

6. From the **New Reference** list, select **Attachment**.

The Add Document window opens.

The screenshot shows the 'Add Document' window. It has a 'File:' field with a 'Browse' button. Below that, there is an 'Author:' field with 'Jane Smith' entered. There is a 'Description:' field with a large text area below it. At the bottom, there are 'Add' and 'Cancel' buttons.

7. Complete the fields.

8. Click **Add**.

The **References** tab lists the attachment.

9. Click **OK** or **Save**.

Adding Existing Packages as Package References

To add an existing package as a reference to a package:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

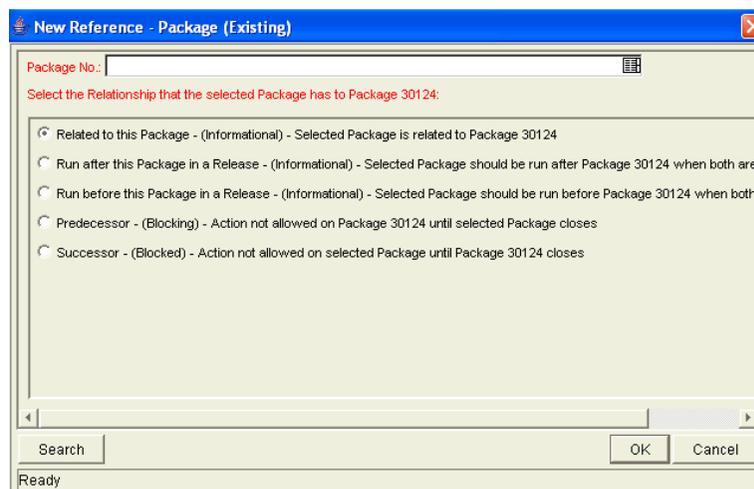
The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. Open a package.
5. In the Package window, click the **References** tab.
6. From the **New Reference** list, select **Package (Existing)**.
7. Click **Add**.

The New Reference - Package (Existing) window opens.



8. Under **Select the Relationship that the selected Package has to Package <Package Number>**, select the option that describes the relationship that the package(s) has (have) to the open package.

9. Select the existing package(s) from the **Package No.** auto-complete list.

The **Package No** field displays the number of the package you selected.

10. Under **Select the Relationship that the selected Package has to Package <Package Number>**, select the option that describes the relationship that the package(s) has (have) to the open package.

11. Click **OK**.

The **References** tab lists the selected package(s).

12. Click **OK** or **Save**.

Adding New Packages as Package References

You can reference a new package to a package.

To create a package to add as a package reference:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

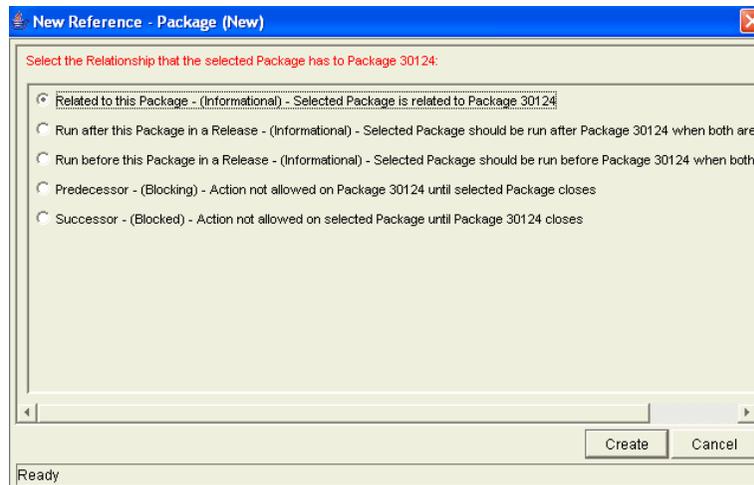
3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. Open the package to which you want to add a reference.
5. In the Package window, click the **References** tab.
6. From the **New Reference** list, select **Package (New)**.

7. Click **Add**.

The **New Reference - Package (New)** window opens.



8. Under **Select the Relationship that the selected Package has to Package <Package Number>**, select the option that describes the relationship that the new package is to have to the package you opened.
9. Click **Create**.

A new Package window opens.

10. In the **Workflow** field, type the name of a workflow to apply to the new package.
11. Type any other information you want to record, and modify any values that you want to change.
12. Click **Save** or **OK**.

The **References** tab lists the new package reference.

13. Save the package.

Attaching Programs as Package References

You can reference existing programs to a package.

To reference an existing program:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

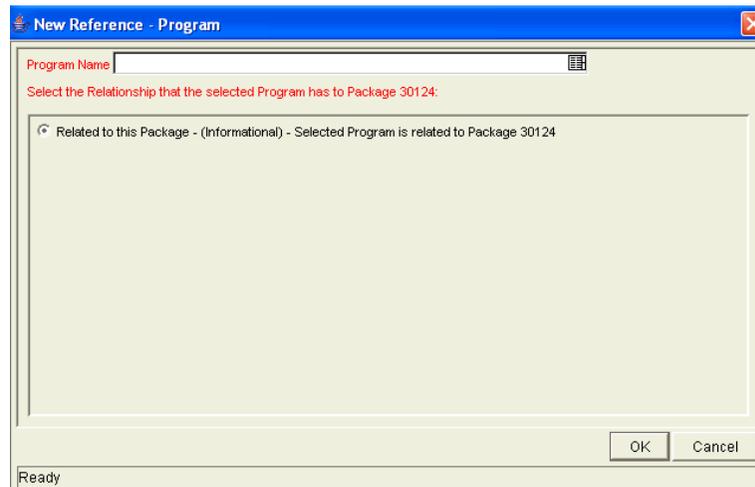
The Workbench opens.

- a. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

3. Open the package to which you want to add a reference.
4. In the Package window, click the **References** tab.
5. From the **New Reference** list, select **Program**.

The New Reference - Program window opens.



6. Select the program from the auto-complete list.

The **Program Name** field displays the program you selected.

7. Under **Select the Relationship that the selected Program has to Package <Package Number>**, select the option that describes how the program is related to the package.

8. Click **OK**.

The **References** tab displays the program name.

9. Save the package.

Attaching Projects as Package References

To attach a project as a package reference:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

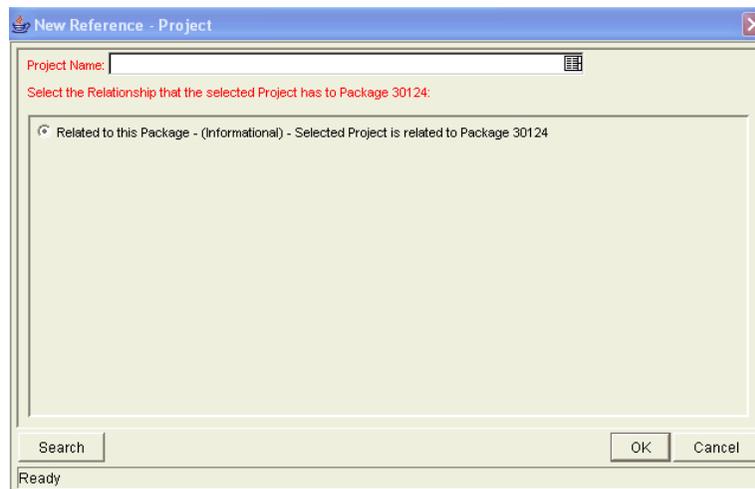
The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. Open the package to which you want to add a reference.
5. In the Package window, click the **References** tab.
6. From the **New Reference** list, select **Project**.
7. Click **Add**.

The New Reference - Project window opens.



8. In the **Project Name** field, select a project name from the auto-complete list.

9. Under **Select the Relationship that the selected Project has to Package <Package Number>**, select the option that describes how the project is related to the package
10. Click **OK**.
The **References** tab lists the selected project.
11. Save the package.

Adding Releases as References

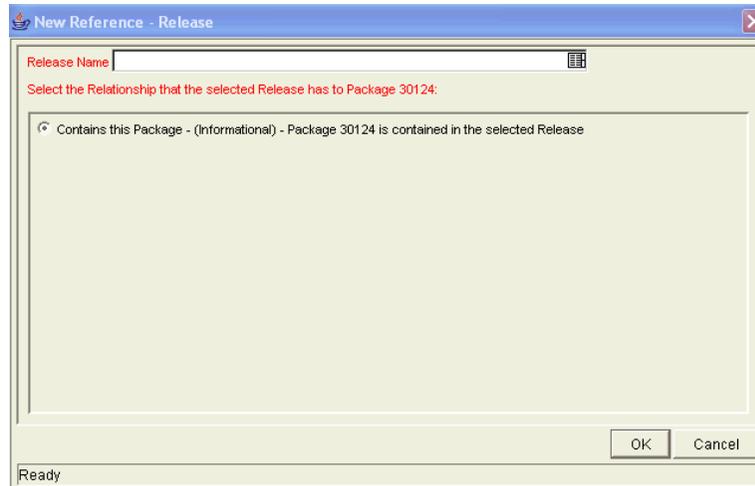
You can reference existing releases to requests.

To reference a release:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.
The Workbench opens.
3. From the shortcut bar, select **Deployment Mgmt > Packages**.
The Package Workbench opens.
4. Open the package to which you want to add a reference.
5. In the Package window, click the **References** tab.
6. From the **New Reference** list, select **Release**.

7. Click **Add**.

The **New Reference - Release** window opens.



8. To specify a release name in the **Release Name** field, use the auto-complete list.
9. Under **Select the Relationship that the selected Release has to Package <Package Number>**, select the option that describes how the release is related to the package.
10. Click **OK**.
The **References** tab lists the release.
11. Save the package.

Adding Existing Requests as Package References

You can reference existing requests to requests.

To reference an existing request:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

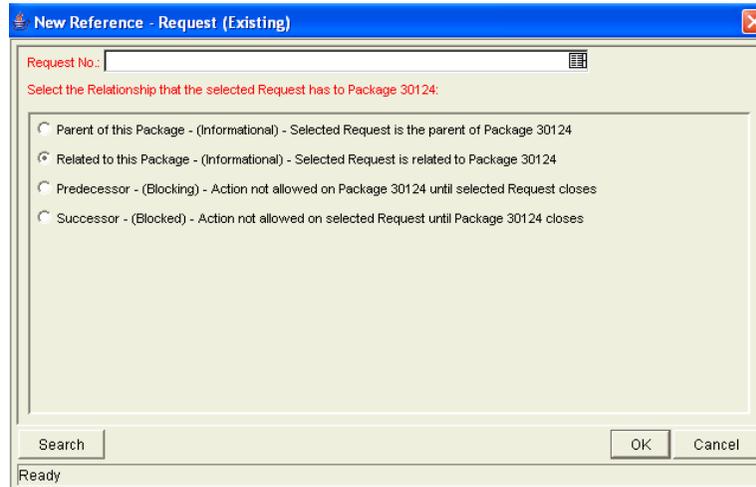
3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. Open the package to which you want to add a reference.

5. In the Package window, click the **References** tab.
6. From the **New Reference** list, select **Request (Existing)**.
7. Click **Add**.

The New Reference - Request (Existing) window opens.



8. To specify the request name in the **Request No** field, use the auto-complete list.
 9. Under **Select the Relationship that the selected Request has to Package <Package Number>**, select the option that describes how the request is related to the package.
 10. Click **OK**.
- The **References** tab lists the existing request.
11. Save the package.

Adding New Requests as Package References

You can create new requests and add them as package references.

To reference a new request:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

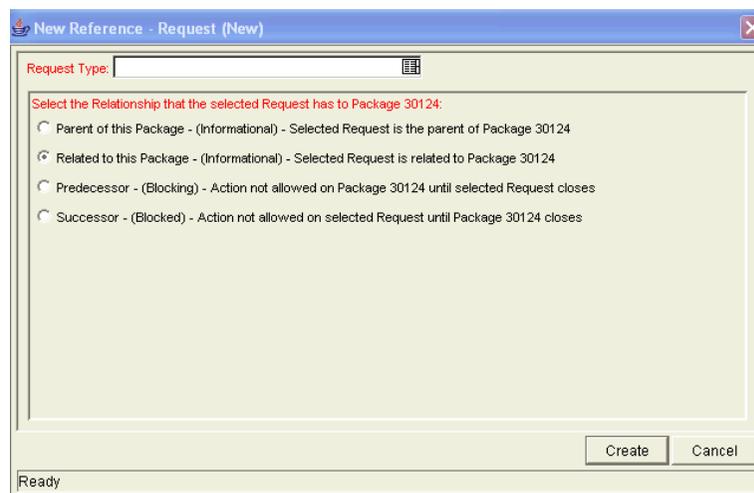
The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. Open the package to which you want to add a reference.
5. In the Package window, click the **References** tab.
6. From the **New Reference** list, select **Request (New)**.
7. Click **Add**.

The New Reference - Request (New) window opens.



8. To specify the type of request you want to create in the **Request Type** field, use the auto-complete list.
9. Under **Select the Relationship that the Request has to Package <Package Number>**, select the option that describes how the new request is related to the package.
10. Click **Create**.

The Create New Request page opens in a new browser window.

11. Complete all required fields (marked with a red asterisk), and, in the optional fields, type any additional information about the request you want to make available to users.
12. Click **Submit**.

In the Package window, the **References** tab lists the new request.

13. Save the package.

Adding Tasks as Package References

You can add existing tasks as references to packages.

To reference an existing task:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

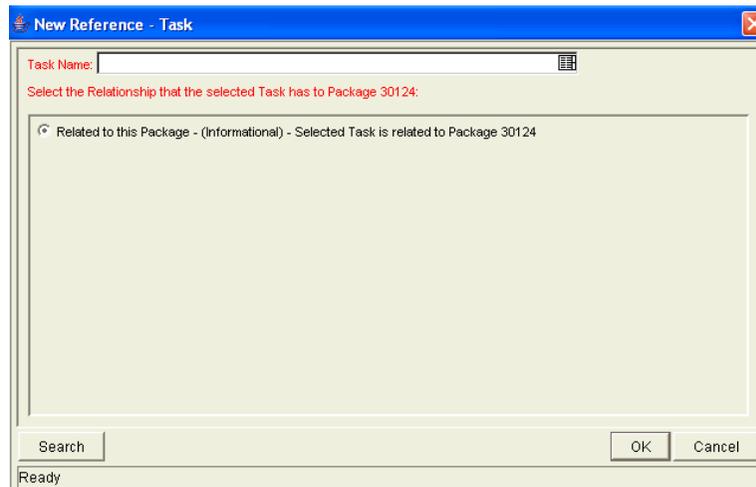
The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. Open the package to which you want to add a reference.
5. From the **New Reference** list, select **Task**.
6. Click **Add**.

The New Reference - Task window opens.



7. To specify a task in the **Task Name** field, use the auto-complete list.
8. Under **Select the Relationship that the selected task has to Package <Package Number>**, select the option that describes how the task is related to the package.

9. Click **OK**.

In the Package window, the **References** tab lists the task.

10. Save the package.

Adding URLs as Package References

You can reference a URL to a package.

To reference a URL:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

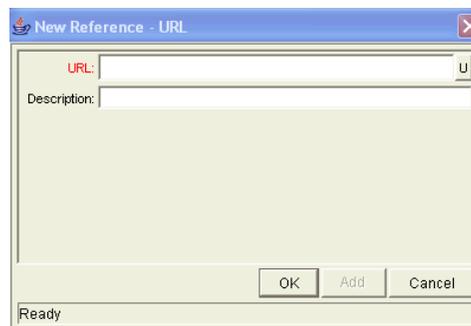
The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. Open the package to which you want to add a reference.
5. In the Package window, click the **References** tab.
6. From the **New Reference** list, select **URL**.
7. Click **Add**.

The New Reference URL window opens.



8. In the **URL** field, type the URL address.
9. To test the URL you specified, to the right of the **URL** field, click **U**.
10. In the **Description** field, you can type a description or an explanation of how the URL is related to the package.

11. Click **OK**.

In the Package window, the **References** tab displays the URL(s).

12. Save the package.

Submitting Packages

After you complete a package, you can submit it. A complete package must have the following:

- A package number
- An associated workflow
- At least one package line

In the Package window, if a package is complete and eligible for submission, the **Submit** button is enabled.

To submit a completed package, click **Submit**.

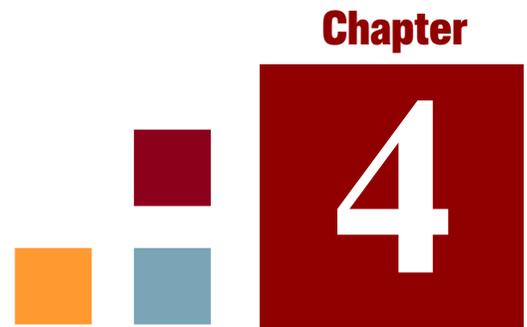
The screenshot shows the 'Package: 30121' window. The 'Package Information' section includes the following fields:

- Package No.: 30121
- Package Group: Upgrade
- Created By: Jane Smith
- Description: ERP Patch v4.5.4.7 Migration Test
- Created On: November 10, 2004
- Workflow: Test Alpha > Test Beta
- Package Status: New
- Assigned User: Jane Smith
- Priority: Normal
- Parent: (empty)
- Assigned Group: Administrator
- Package Type: Customization
- Priority Seq: 50
- Percent Complete: 0

The 'Package Lines' section contains a table with the following data:

Seq	Object Type	App Code	Client File Name
1	File Client->Client	(None)	env4.5.4.7.zip
2	File Client->Client	(None)	patch4.5.4.7.zip
3	File Client->Client	(None)	config4.5.4.7.zip

At the bottom of the window, there are buttons for 'New Line', 'Edit Line', 'Copy Line', 'Remove Line', 'Submit', 'OK', 'Save', and 'Cancel'. The 'Submit' button is highlighted, indicating it is enabled.



Chapter
4
Processing Packages

In This Chapter:

- *Overview of Processing Packages*
 - *Using the Standard Interface to Process Packages*
 - *Opening Packages from the Standard Interface*
 - *Approving Packages in the Standard Interface*
 - *Adding Notes to Packages in the Standard Interface*
 - *Adding References to Packages from the Standard Interface*
 - *Processing Packages Using the Workbench*
 - *Opening Packages from the Workbench*
 - *Checking Package Status*
 - *Acting on Eligible Steps*
 - *Acting on Decision Workflow Steps*
 - *Acting on Execution Workflow Steps*
 - *Adding Notes*
 - *Adding, Updating, and Deleting References*
 - *Adding a Package Line to an In-Progress Package*
-

Overview of Processing Packages

In Mercury Deployment Management, once a package has been submitted, you can process and manage it. Processing packages refers to working with the package as the package progresses through its workflow.

You can locate packages that require attention using one of the following methods:

- **Use the Package Workbench.** Use the **Query** tab in the Package Workbench to specify search criteria to find a package. After the package is found, highlight the package and click **Open**. The selected package opens in the Package window.
- **Use Portlets in the Standard Interface.** The quickest way to locate relevant packages is through the standard interface. When included on a Mercury IT Governance Dashboard™ page, the My Packages portlet displays all the packages you have created or are assigned to you. Click a package number to open the package in the standard interface.
- **Use Searches in the Standard Interface.** Using search in the standard interface is another way to find a package. In the standard interface, on the menu bar, select **Search > Packages**. The Package Search page opens. In the Package Search page, complete the search criteria and click **Search**. The Package Search Results page opens. In the Package Search Results page, click a package number to open the package in the standard interface.
- **Use Notifications.** As a package proceeds through its workflow, email notifications can be sent to alert you of pending actions. The notification might include a link to the package. Click the link to open the referenced package in the Workbench. If you are not currently logged on to the Mercury IT Governance Center, the Logon page opens.
- **Generate a Report on Open Packages (Standard Interface).** Mercury Deployment Management includes a predefined set of reports that generate HTML text and can be accessed by a Web browser. Among these reports is the Packages Pending Report, which reports on open packages with pending activity.

For information about generating and viewing Deployment Management reports, see the *Reports Guide and Reference*.

For detailed information on how to process packages in the standard interface, see *Using the Standard Interface to Process Packages*.

Using the Standard Interface to Process Packages

You can use the package detail page in the standard interface to process a package through its entire workflow. You can add notes and references to the package, and make decisions about package status. For information on how to process packages from the Workbench, see *Processing Packages Using the Workbench* on page 100.

Opening Packages from the Standard Interface

This section provides information on how to open packages from the standard interface using the Search Packages page, using portlets, or using the package links in email notifications.

Opening Packages from the Search Packages Page

To open a package using the Search Packages page:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Deployment Management > Search Packages**.

The Search Packages page opens.

3. Enter your search criteria, and then click **Search**.

The Package Search Results page lists the packages that match the criteria you specified.

4. In the **Pkg #** column, click the package number for the package you want to open.

The detail page for that package opens.

Opening Packages from the My Packages Portlet

To use the My Packages portlet to open a package that you have created or a package that is assigned to you:



If the My Packages portlet is not displayed on one of your Dashboard pages, you can add it. For information on how to add portlets to Dashboard pages, see *Configuring the Standard Interface*.

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Dashboard**, and then select the name of a Dashboard page that displays the My Packages portlet.

The My Packages portlet lists all of the packages that you have created.

3. On the My Packages portlet, in the **Pkg #** column, select the number for the package you want to open.

The detail page for that package opens.

Opening Packages Using Notification Links

In some cases, as a package moves through the steps in its workflow, email notifications are sent to alert you of pending actions for a package. If the notification includes a link to the package, click the link to open the referenced package in the Workbench. (If you are not logged on to Mercury IT Governance Center, the Logon page opens.)

Approving Packages in the Standard Interface

To approve a package using the standard interface:

1. Log on to the Mercury IT Governance Center.
2. Open the package that you want to approve.

For information on how to open the details page, see *Opening Packages from the Standard Interface* on page 81.

MERCURY SIGN OUT

IT Governance Center Search Results > Package #30121 > Package Search > Search Results > Package #30122

Package 30122 Save Reset

Header

Package No.: 30122 Package Group: Upgrade Created By: Jane Smith
 Description: Upgrade to patch 4.5.4.7 Created On: November 11, 2004
 Workflow: Test Alpha > Test Beta Status: In Progress

Assigned User: Jane Smith Priority: Normal Parent:
 Assigned Group: ITG Change Management Package Type: Customization Priority Seq: 50
 Percent Complete: 0

Status

View Lines: Pending Lines Only View Steps: Steps Waiting my Action
 All Lines All Steps Refresh

View All Lines			View Eligible Steps for Test Alpha > Test Beta		
Seq	Object Name	Object Type			
<input type="checkbox"/>	1	env4.5.4.7.zip	File Client->Client	<input type="radio"/> 1: Initial Approval	<input type="radio"/>
<input type="checkbox"/>	2	patch4.5.4.7.zip	File Client->Client	<input type="checkbox"/> Eligible	
<input type="checkbox"/>	3	config4.5.4.7.zip	File Client->Client	<input type="checkbox"/> Eligible	

Check All Clear All Line Details Check All Clear All Workflow Action <<< << >> >>>

Notes

Add Notes

Notes to be added on save:

Existing Notes

Jane Smith (jasmith) Run env script before adding the patch.
 November 11, 2004 11:23:20 AM PST

References

Requests

Req #	Assigned User	Description	Request Type	Status	% Complete	Relationship	Relationship Details
<input checked="" type="checkbox"/> 30330	Jane Smith	ERP Patch v4.5.4.7 Mig...	DEM - Application Enhancement	New	0%	Related to this Package	Informational: Request 30330 is related to Package 30122

Reference Additions

New Reference: Attachment Add ■ Highlighted items are actively controlling this Package

References to be added on Save:

Open Remove

Save Reset

The **Status** section displays the package line(s) for the workflow.

3. Under **Status**, next to **View Lines**, do one of the following:
 - To display only the package lines that are eligible for action, click **Pending Lines Only**, and then click **Refresh**.
 - To display all of the package lines for the workflow, click **All Lines**, and then click **Refresh**.
4. Under **Status**, next to **View Steps**, do one of the following:
 - To display only workflow steps that await action, click **Steps Waiting my Action**, and then click **Refresh**.
 - To display all workflow steps, click **All Steps**, and then click **Refresh**.

The screenshot shows a 'Status' interface with two main sections: 'View All Lines' and 'View Eligible Steps for Test Alpha > Test Beta'.

View All Lines: A table with columns 'Seq', 'Object Name', and 'Object Type'. It lists three items:

Seq	Object Name	Object Type
<input type="checkbox"/>	env4.5.4.7.zip	File Client->Client
<input type="checkbox"/>	patch4.5.4.7.zip	File Client->Client
<input type="checkbox"/>	config4.5.4.7.zip	File Client->Client

View Eligible Steps for Test Alpha > Test Beta: A section with a radio button selected for '1: Initial Approval' and three other radio buttons. Below it are three rows, each with an 'Eligible' checkbox.

Buttons at the bottom include 'Check All', 'Clear All', 'Line Details', 'Workflow Action', and directional arrows.

Under the workflow name are the numbered steps that must be completed and approved to move the workflow to completion. To navigate forward or backward through the workflow steps, use the directional arrows.

5. In the **View Eligible Steps for <Workflow Name>** section, select the option button that corresponds to the numbered workflow step that you want to approve (or not), and then select the **Eligible** checkbox for the specific line you want to approve (or not).

To select all of the steps, click **Check All**.

To clear all eligible checkboxes for the workflow step, click **Clear All**.

6. If one or more package lines are selected for action, click **Workflow Action**.
The Package Workflow Action page opens.

MERCURY SIGN OUT

IT Governance Center Search Results > Package #30121 > Package Search > Search Results > Package #30122

Package: Workflow Action

Package # 30122 **Workflow:** Test Alpha > Test Beta

Description: Upgrade to patch 4.5.4.7

Action Required

Please choose an outcome for the step:

Initial Approval

Approved

Not Approved

Notes:

Selected Lines:		
Seq	Object Name	Object Type
1	env4.5.4.7.zip	File Client->Client

7. Under **Action Required**, select the outcome of the step.

Because the outcome is configurable, it can be different for each workflow step, depending on the configuration at your site. In the above example, the possible actions are:

- If the workflow step is ready for approval, select **Approved**.
- If the workflow step is not ready for approval, select **Not Approved**.

8. Click **OK**.

The package details page opens. Because you approved (or disapproved) the selected workflow step(s), the workflow has moved on to the next step. This process continues until the package moves through all of its steps to resolution.

Adding Notes to Packages in the Standard Interface

The **Notes** section contains fields in which you can enter supplementary package information. To add a note, in the **Add Notes** subsection, type the information. To save the note, save the package.

View existing notes in the **Existing Notes** section. Existing notes are listed in chronological order with the most recent note listed first. You can filter existing notes, using **Show Only User Notes**, **Note Author**, and **Changed Fields**.

You cannot delete an existing note.

Figure 4-1. Notes section

The screenshot shows a web interface for managing notes. At the top, there is a header bar labeled 'Notes'. Below it is a section titled 'Add Notes' with a sub-label 'Notes to be added on save:' and a large empty text input field. Below the input field is another section titled 'Existing Notes'. This section contains a list of notes, with the first entry showing the author 'Jane Smith (jasmith)', the date 'November 11, 2004 11:23:20 AM PST', and the text 'Run env script before adding the patch.'

Adding References to Packages from the Standard Interface

In some cases, it might be useful to reference a Web-accessible file, a document, or file from a local machine to the package. You can add such references to a package from the standard interface.

You can add the following entities as package references:

- Attachments
- Packages (new or existing)
- Projects
- Releases
- Requests (new or existing)
- Tasks
- URLs

For some reference types (such as requests and other packages), you can create a functional dependency to the reference. For example, you can specify that a request is a predecessor to the package. This means that the package cannot

continue along its workflow until the referenced request closes. For more information about valid package references and the relationships they can have to packages, see [Reference Relationships on page 40](#).

This section provides information on how to add various types of references to a package from the standard interface. For information on how to add package references from the Workbench, see [Adding References to Packages from the Package Workbench on page 64](#).

Adding Attachments as References

To reference an attachment to a package:

To add an attachment as a package reference:

1. Log on to Mercury IT Governance Center.
2. Open (or create) a package.



Note

For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **Attachment**.
5. Click **Add**.

The Add Document window opens.

6. Next to the **File** field, click **Browse**, and then locate and selected the document file you want to attach as a reference.
7. In the **Author** field, you can replace the default value with the name of the attachment author.
8. In the **Description** field, you can type a description of the attachment.

9. Click **Add**.

On the Package page, the **References to be added on Save** field lists the referenced attachment.

10. Click **Save**.

The attachment is added as a request reference.

Adding Existing Packages as Package References

You can reference existing packages to a package.

To specify an existing package as a package reference:

1. Log on to Mercury IT Governance Center.
2. Open (or create) a package.



Note

For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **Package (Existing)**.

- Click **Add**.

The Add References: Package window opens.

Add Reference: Package

Select which relationship the selected Packages will have to Package #30122:

- Related to this Package - (Informational) - Selected Package is related to Package 30122
- Run after this Package in a Release - (Informational) - Selected Package should be run after Package 30122 when both are in a Release
- Run before this Package in a Release - (Informational) - Selected Package should be run before Package 30122 when both are in a Release
- Predecessor - (Blocking) - Action not allowed on Package 30122 until selected Package closes
- Successor - (Blocked) - Action not allowed on selected Package until Package 30122 closes

Package Search Results Showing 1 to 11 of 11

Pkg #	Workflow	Status	Priority	Assigned To	Pkg Lines	Description
<input type="checkbox"/> 30124	Test Alpha > Test Beta	New	Normal	jasmith	3	Copy of 30122
<input type="checkbox"/> 30123	Test Alpha > Test Beta	New	Normal	jasmith	3	Copy of 30122
<input type="checkbox"/> 30122	Test Alpha > Test Beta	In Progress	Normal	jasmith	3	Upgrade to patch 4.5.4.7
<input type="checkbox"/> 30121	Test Alpha > Test Beta	New	Normal	jasmith	3	ERP Patch v4.5.4.7 Migration Test
<input type="checkbox"/> 30105	Test Alpha > Test Beta	New	Low			Need information concerning HMO.
<input type="checkbox"/> 30080	Test Alpha > Test Beta	In Progress	Low		3	
<input type="checkbox"/> 30079	Test Alpha > Test Beta	New	Low	jasmith	3	Copy of 30073
<input type="checkbox"/> 30074	Test Alpha > Test Beta	In Progress	Low		2	
<input type="checkbox"/> 30073	Test Alpha > Test Beta	In Progress	Low	jasmith	3	Test package.
<input type="checkbox"/> 30072	Test Alpha > Test Beta	In Progress	Low		1	Test package
<input type="checkbox"/> 30071	Test Alpha > Test Beta	In Progress	Low		3	

Showing 1 to 11 of 11

Close Window X

- Enter your search criteria, and then click **Search**.

The **Package Search Results** section lists the packages that match your search criteria.

- Select the checkbox for the package that you want to add as a reference, and then click **Add**.

On the Package page, the **References to be added on Save** field lists the referenced package.

- Click **Save**.

Adding New Packages as Package References

You can reference a new package to a package.

To add a new package as a package reference:

- Log on to Mercury IT Governance Center.
- Open a package.



For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **Package (New)**.
5. Click **Add**.

The Create New Package window opens.

6. Under **Relationship**, select the option that describes the relationship that the new package has to the open package.
7. Under **Select which Relationship the new Package will have to Package #**, select the option that describes the relationship that the new package has to the open package.



For information about the different types of reference relationships, see [Reference Relationships on page 40](#).

8. Click **Create**.

The New Package window opens.

The screenshot shows the 'New Package 30126' window in the Mercury system. The window has a red header bar with the Mercury logo and a 'Close Window' button. The main content area is divided into several sections:

- Header:** Contains 'Package No.' (30126), 'Package Group' (empty), 'Created By' (Jane Smith), 'Description' (Upgrade to patch 4.5.4.7), 'Created On' (November 12, 2004), and 'Workflow' (empty).
- Assigned User:** Jane Smith (with a user icon).
- Priority:** Normal (dropdown menu).
- Parent:** (empty field).
- Assigned Group:** ITG Change Management (with a group icon).
- Package Type:** Customization (dropdown menu).
- Priority Seq:** 50.
- Percent Complete:** 0.
- Notes:** A section titled 'Notes to be added on save:' with an empty text area.
- References:** A section titled 'Reference Additions' with a 'New Reference:' dropdown set to 'Attachment' and an 'Add' button.
- References to be added on Save:** An empty text area with 'Open' and 'Remove' buttons below it.

At the bottom of the window, there are 'Save' and 'Cancel' buttons, and a 'Close Window' button in the footer.

9. In the **Workflow** field, type the name of a workflow to apply to the new package.
10. Type any optional information you want to save, and modify any values that you want to change.
11. Click **Save**.

On the Package page, the **References to be added on Save** field lists the new referenced package.

12. Click **Save**.

Attaching Programs as Package References

You can reference existing programs to a package.

To specify a program as a package reference:

1. Log on to Mercury IT Governance Center.
2. Open (or create) a package.



Note

For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **Program**.
5. Click **Add**.

The Reference Program window opens.

6. To specify the name of the program to reference in the **Program** field, select the name from the auto-complete list.
7. Click **Add**.

On the Package page, the **References to be added on Save** field lists the program you selected.

8. Click **Save**.

Attaching Projects as Package References

To specify a project as a package reference:

1. Log on to Mercury IT Governance Center.
2. Open (or create) a package.



Note

For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **Project**.
5. Click **Add**.

The Add Reference: Project window opens.

6. Enter your search criteria in one or more of the available fields, and then click **Search**.

The Add Reference: Project window lists the projects that match the search criteria you specified.

7. In the **Select Project to View** section, select the checkbox for the project that you want to add as a package reference.

8. Click **Add**.

On the Package page, the **References to be added on Save** field lists the referenced project.

9. Click **Save**.

Adding Releases as Package References

You can add existing releases as package references.

To reference an existing release to a package:

1. Log on to Mercury IT Governance Center.
2. Open (or create) a package.



Note

For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **Release**.
5. Click **Add**.

The Reference Release window opens.

A screenshot of a web application window titled "Reference Release". The window has a red header bar with the word "MERCURY" on the left and a "Close Window X" button on the right. Below the header, the title "Reference Release" is displayed. There is a "Release:" label followed by a text input field and a small icon. Below that is a "Relationship:" label followed by a radio button and the text "Contains this Package - (Informational) - Package 30122 is contained in the selected Release". At the bottom right of the form area, there are two buttons: "Add" and "Cancel". At the very bottom of the window, there is another "Close Window X" button.

6. To specify the name of the release to reference in the **Release** field, select it from the auto-complete list.
7. Click **Add**.

On the Package page, the **References to be added on Save** field lists the referenced release.

8. Click **Save**.

Adding Existing Requests as Package References

You can reference existing requests to packages.

To reference an existing request:

1. Log on to Mercury IT Governance Center.
2. Open (or create) a package.



For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **Request (Existing)**.
5. Click **Add**.

The Add Reference: Request window opens.

Add Reference: Request

*Select which relationship the selected Requests will have to Package #30008:

- Parent of this Package - (Informational) - Selected Request is the parent of Package 30008
- Related to this Package - (Informational) - Selected Request is related to Package 30008
- Predecessor - (Blocking) - Action not allowed on Package 30008 until selected Request closes
- Successor - (Blocked) - Action not allowed on selected Request until Package 30008 closes

Request Search Results							
Req #	Request Type	Description	Status	Assigned To	Priority	Created By	
<input type="checkbox"/> 30253	DEM - Application Bug	inventory counts do not transaction details	New	Finn Gill	Normal	Admin User	
<input type="checkbox"/> 30122	DEM - Application Bug	Cycle count screen throws error	New	Finn Gill	Normal	Admin User	
Added 30121	DEM - Application Bug	inventory counts do not transaction details	New	Finn Gill	Critical	Admin User	

Showing 1 - 3 of 3

6. Under **Select which Relationship the selected Requests will have to Package #**, select the option that describes the relationship that the new request has to the open package.



For information about the different types of reference relationships, see [Reference Relationships on page 40](#).

7. Select the checkbox for the request that you want to add as a package reference, and then click **Add**.

On the Package page, the **References to be added on Save** field lists the referenced request.

8. Click **Save**.

Adding New Requests as Package References

To reference a new request:

1. Log on to Mercury IT Governance Center.
2. Open (or create) a package.



Note

For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **Request (New)**.
5. Click **Add**.

The Create New Request window opens.

A screenshot of a web application window titled 'MERCURY' with a 'Close Window' button in the top right corner. The main content area is titled 'Create New Request'. It features a 'Request Type' dropdown menu currently set to 'Enhancement', with a sub-label 'Enhancement: Enhancement Request type'. Below this is a 'Relationship' section with four radio button options: 'Parent of this Package - (Informational) - Selected Request is the parent of Package 30122', 'Related to this Package - (Informational) - Selected Request is related to Package 30122' (which is selected), 'Predecessor - (Blocking) - Action not allowed on Package 30122 until selected Request closes', and 'Successor - (Blocked) - Action not allowed on selected Request until Package 30122 closes'. At the bottom right of the form are 'Create' and 'Cancel' buttons. A second 'Close Window' button is located at the bottom right of the window frame.

6. From the **Request Type** list, select the type of request you want to create as a reference to the open package.
7. Under **Relationship**, select the option that describes the relationship that the new request has to the open package.



Note

For a list of reference types, their definitions, and possible dependency relationships, see [Reference Relationships on page 40](#).

8. Click **Create**.

The Create New *<Request Type>* window opens.

9. Provide data in all of the required fields (marked with a red asterisk), and any optional data about the new request that you want to record.
10. Click **Submit**.

On the Package page, the **References to be added on Save** field lists the new request.

11. Click **Save**.

Adding Tasks as Package References

You can reference existing tasks to packages.

To reference an existing task:

1. Log on to Mercury IT Governance Center.
2. Open (or create) a package.



Note

For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **Task**.
5. Click **Add**.

The Add Reference: Task window opens.

6. Enter your search criteria in any of the available fields, and then click **Search**.

The Add Reference: Task window lists the tasks that match the search criteria you specified.

MERCURY Close Window X

Add Reference: Task

Select which relationship the selected Tasks will have to Package #30008:

Related to this Package - (Informational) - Selected Task is related to Package 30008

Select Task to View									Showing 1 to 2 of 2 Prev Next
Task Name	Project	Project Path	Scheduled Start	Scheduled Finish	Task Status	Resources	Exception Indicator	% Complete	
<input type="checkbox"/> Agree requirements and priorities with managers	Finance Specialized Web Site Development		Nov 7, 2006	Nov 7, 2006	pending-predecessor	Brandi Greer, William Klein, Nancy Giddings	0	0% <input type="text"/>	
<input type="checkbox"/> Agree requirements and priorities with managers	Ops: BackOffice Acceleration		Jul 26, 2006	Jul 26, 2006	pending-predecessor	Brandi Greer, William Klein, Nancy Giddings	8	0% <input type="text"/>	
<input type="checkbox"/> Check all									Showing 1 to 2 of 2 Prev Next

Search for Tasks to View

Task Name starts with: Resource:

Include finished Tasks? Yes No Task Status:

Scheduled Start Date From: To:

Scheduled Finish Date From: To:

Show only tasks with exceptions? Yes No

Show only milestones? Yes No

Search in projects that meet the following criteria

Project Name: Project Manager:

Sort By: Ascending Descending **Results Displayed Per Page:**

Close Window X

7. In the **Select Task to View** section, select the checkbox for the task you want to add as a reference to the open package.

8. Click **Add**.

On the Package page, the **References to be added on Save** field lists the task you selected.

9. Click **Save**.

Adding URLs as Package References

You can reference an existing URL to a package.

To add a URL as a package reference:

1. Log on to Mercury IT Governance Center.
2. Open (or create) a package.



Note

For information on how to open a package, see [Opening Packages from the Standard Interface on page 81](#).

3. Scroll to, and then expand, the **References** section.
4. From the **New Reference** list, select **URL**.
5. Click **Add**.

The Reference URL window opens.

6. In the **URL** field, type the URL address.
7. To test the URL you specified in your default browser window, to the right of the **URL** field, click **U**.
8. In the **Description** field, you can type a description, or an explanation of how the URL is related to the package.
9. Click **OK**.

On the Package page, the **References to be added on Save** field lists the URL you specified.

10. Click **Save**.

Processing Packages Using the Workbench

You can process submitted packages from the Workbench using the Package window. You can use the Package window to process a submitted package through its workflow. You can add notes and references to the package, make decisions on package status, and execute actions for the package. You can also change or delete a package from the Package window.

Opening Packages from the Workbench

To open a package from the Workbench:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Packages Workbench opens.

4. To display all of the existing packages, click **List**.

To restrict the search, enter information in one or more of the search fields.

5. Select a listed package, and then click **Open**.

Checking Package Status

Each package line must follow the business process defined by its assigned workflow. You can view all of the workflow and subworkflow steps for each line on the **Status** tab in the Package window.

To view the status of a submitted package:

1. Open a submitted package.

For information on how to open a package, see *Opening Packages from the Workbench* on page 100.

The Package window opens.

2. Click the **Status** tab.
3. Review the status of the package.

You can now review the status of each package line. The result of each workflow step is recorded in the package line row, so that you can quickly get an idea of which lines have closed with success or closed with a failure.

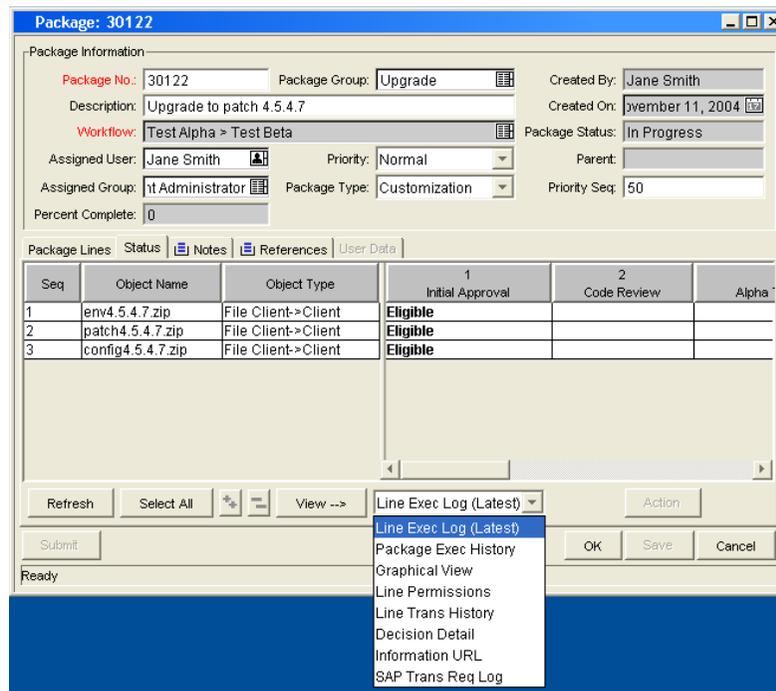
Although all the workflow steps are listed, a package line does not necessarily go through every step before it is resolved. For example, you might have a rework step that is only used if rework is required. For more information about the package, select one of the logs, views, or reports related to the package. For more information about the **Status** tab, see *Package Window, Status Tab* on page 35.

Acting on Eligible Steps

After a package is submitted, each package line has one or more workflow steps set to **Eligible**, as shown in *Figure 4-2* on page 101. This indicates that the execution or decision steps must be performed first in the resolution process for the package. If you have permission to act on the eligible step, the text in the **Status** column is displayed in bold text.

If the eligible step is an *execution* workflow step, you must perform an action. If the eligible step is a *decision* workflow step, you must specify a decision. After you perform the action or make the decision, Mercury Deployment Management displays the outcome.

Figure 4-2. Status tab eligible steps



To process a workflow step:

1. Open a submitted package.

For information on how to open a package, see *Opening Packages from the Workbench* on page 100.

The Package window opens.

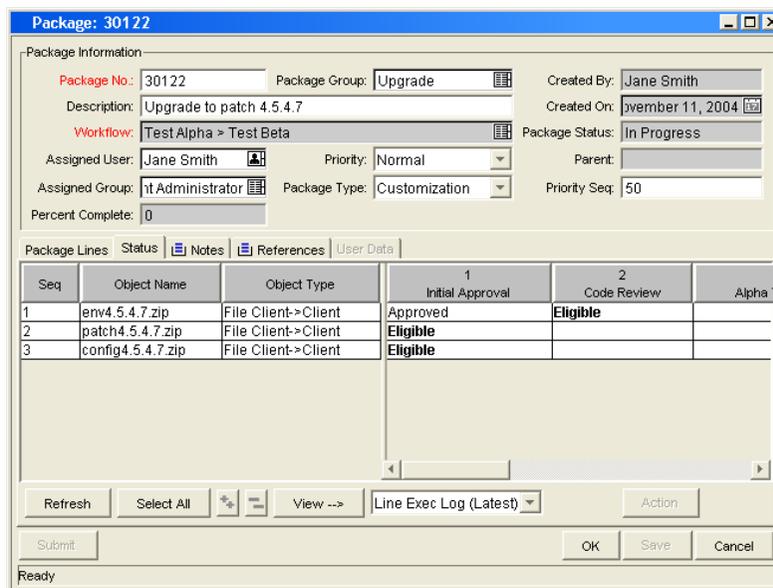
2. Click the **Status** tab.
3. Select one or more package lines.

You can select a single step for a single package line, or multiple steps for multiple lines. To select all of the steps on all of the open package lines that are eligible for action, click **Select All**.

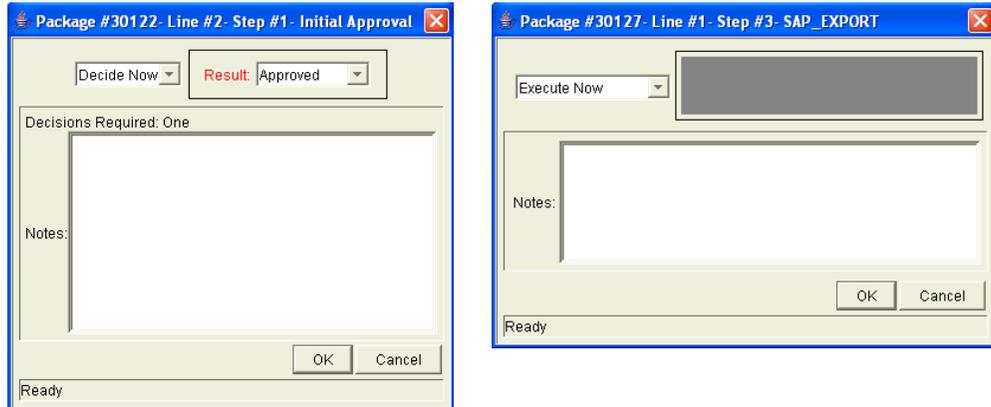
You cannot act on multiple workflow steps at one time. If you select cells for multiple workflow steps, the **Action** button remains disabled. If you have the security privileges to perform the given workflow step, the button is enabled.

After you select a workflow step for which you have the security permission to act on the workflow step, the label on the button at the bottom-right of the **Status** tab changes from **Action** to the workflow step name.

4. To perform an action or decision on a selected step, click **Action**.



A decision window or the execution window opens.



Each step is configured with a selection of valid results.

5. If a decision window is open, from the **Result** list, select a decision result. If an execution window is open, in the list, select **Execute Now**, **Schedule Execution**, or **Bypass Execution**.
6. In the **Notes** field, you can type any comments you want to save.



Note

For information about the decision window, see [Acting on Decision Workflow Steps](#). For information about the execution window, see [Acting on Execution Workflow Steps on page 106](#).

7. Click **OK**.

Whether the step is a decision or an execution, the result of the step is processed by the workflow engine and can lead to a number of changes. Based on the workflow definition (which can contain multiple workflow steps and subworkflows) the given workflow step result can cause additional workflow steps to become eligible, indicating the next actions to be performed in the package resolution process.

When subsequent steps become eligible, those steps are completed sequentially, making eligible the next workflow step(s) in line. This process continues until the package reaches its final state and is resolved, usually at a close step.

Acting on Decision Workflow Steps

Decisions workflow steps are steps for which you (or a group) must indicate the outcome. For example, Approve Migration is an example of a decision workflow step. You can either act on the decision step yourself or delegate the decision.

To act on a decision workflow step:

1. In the Package window, on the **Status** tab, select an eligible decision workflow step.

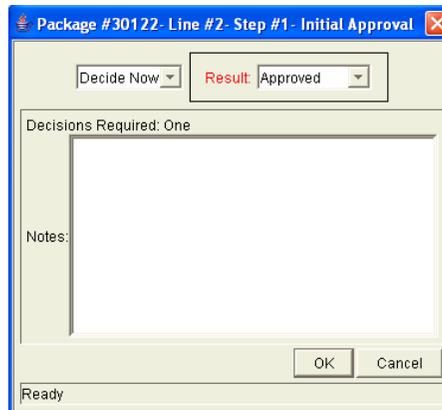
For information on how to access the **Status** tab and select an eligible decision workflow step, see [Acting on Eligible Steps on page 101](#).

The **Action** button is enabled. Notice that the label changes, depending on the step selected.



2. Click **Action**.

The Package Decision window opens.

3. In the list, select **Decide Now** or **Delegate Decision**.

- If you select **Decide Now**, you must make a decision about the decision workflow step; for example, **Approve**. To make a decision regarding the decision workflow step:
 - a. From the **Result** list, select the result.
 - b. In the **Notes** field, you can type the text of a note.
 - c. Click **OK**.

The decision window closes. The decision results are added to the **Status** tab.

- If you select **Delegate Decision**, you must then assign the decision to someone. To delegate a decision:
 - a. To select a resource, use the **Delegate To** auto-complete.
 - b. In the **Notes** field, type note text.
 - c. Click **OK**.

The decision window closes, and the results are listed on the **Status** tab.

4. Click **OK**.

The decision is entered into the system. If the workflow step was configured so that more than one person must decide on the result, the package line may not move to the next step until all required decisions are made. If a decision

workflow step requires that more than one person decide on the result, you can view the current decision results for the workflow step using the Decision Detail view.

Acting on Execution Workflow Steps

Execution workflow steps are workflow steps where the Deployment Management system performs an action and then updates the step with its result. These actions can be as simple as calculating the value for a token or as complex as copying files, running programs, or updating Web pages.

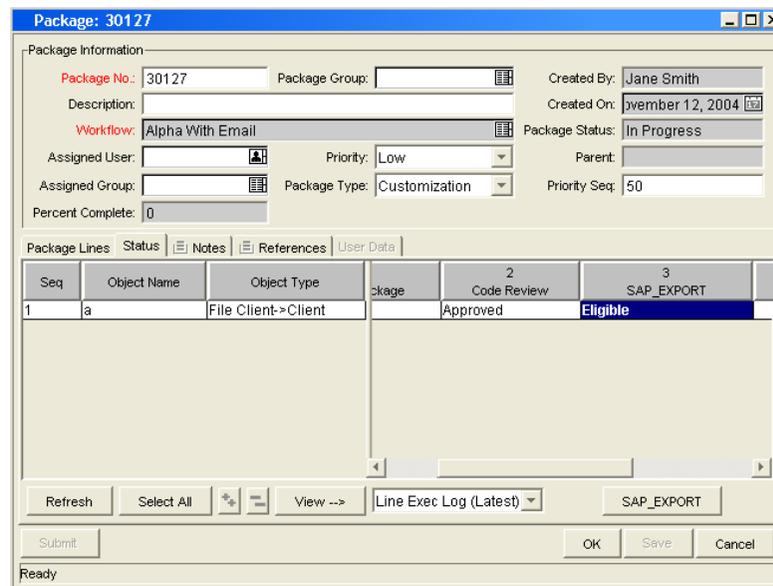
To act on an execution workflow step:

1. In the Package window, on the **Status** tab, select the eligible decision workflow step.



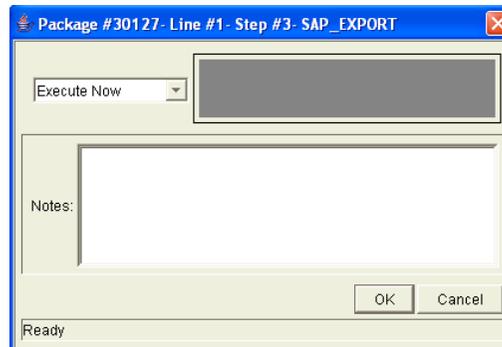
For information on how to access the Status tab and select an eligible decision workflow step, see [Acting on Eligible Steps on page 101](#).

The **Action** button is enabled. Notice that the button label changes with the step selected.



2. On the **Status** tab, click **Action**.

The execution window opens.



3. In the list, select **Execute Now**, **Schedule Execution**, or **Bypass Execution**.
 - If you select **Execute Now** and click **OK**, you execute the workflow step. The Execution window closes. The result of the workflow step is displayed on the package line at the workflow step. This process could be a software migration, execution of a PL/SQL function, the closing of the package line, or other activity. The execution engine executes the step and sets the result accordingly.
 - If you select **Schedule Execution**, the **Execution Date** and **Execution Time** fields are visible. Select a date and time, and then click **OK**. The Execution window closes. The execution schedule is displayed on the package line at the workflow step. This process could be a software migration, execution of a PL/SQL function, the closing of the package line, or other activity. The execution engine executes the step and sets the results accordingly.
 - If you select **Bypass Execution**, the **Execution Result** field is visible. Select the results, and then click **OK**. The execution window closes and the selected result is displayed on the package line at the workflow step.
4. Click **OK**.

Adding Notes

In the Package window, the **Notes** tab (*Figure 4-3*) displays additional package information.

Figure 4-3. Notes tab in the Package window

Package: 30121

Package Information

Package No.: 30121 Package Group: Upgrade Created By: Jane Smith

Description: ERP Patch v4.5.4.7 Migration Test Created On: November 10, 2004

Workflow: Test Alpha > Test Beta Package Status: New

Assigned User: Jane Smith Priority: Normal Parent:

Assigned Group: it Administrator Package Type: Customization Priority Seq: 50

Percent Complete: 0

Package Lines | Status | **Notes** | References | User Data

Existing Notes

Jane Smith (jasmith)
November 11, 2004 11:23:20 AM PST
Run env script before adding the patch.

New Notes

Enlarge

Submit OK Save Cancel

Ready

To add a note, type the text in the **New Notes** field, and then click **Save**. The note is added to the **Existing Notes** field.

Adding, Updating, and Deleting References

You can add and update package references from the **References** tab of the Package window in the Workbench. The following sections provide information about how to add, update, and delete references.

Adding References

The Package window **References** section contains additional package information. For example, you might add a document or URL as a reference to a package. You can add the following entities as package references:

- Attachments
- Packages (New and Existing)
- Projects
- Releases
- Requests (New and Existing)
- Tasks
- URLs

For some references (such as requests and other packages), a functional dependency to the original package can be created. For example, you can specify that a request is a predecessor to the package. This means the package will not continue until the request is closed. For a list of the references and their possible dependency relationships, see *Reference Relationships* on page 40.

For information on how to add a reference, see *Adding References to Packages from the Package Workbench* on page 64 or *Adding References to Packages from the Standard Interface* on page 86.

Updating Reference Relationships

To update an existing reference relationship from the Workbench:

1. Open a package.

For information on how to open a package, see *Opening Packages from the Workbench* on page 100.

2. Click the **References** tab.
3. Click the **Relationship** field for the referenced entity, and then select an item from the displayed list.

The screenshot shows a window titled "Package: 30121" with the following details:

- Package Information:**
 - Package No.: 30121
 - Package Group: Upgrade
 - Created By: Jane Smith
 - Description: ERP Patch v4.5.4.7 Migration Test
 - Created On: November 10, 2004
 - Workflow: Test Alpha > Test Beta
 - Package Status: New
 - Assigned User: Jane Smith
 - Priority: Normal
 - Parent: (empty)
 - Assigned Group: Management Administrator
 - Package Type: Customization
 - Priority Seq: 50
 - Percent Complete: 0
- References Tab:**

Details	Status	% Complete	Description	Relationship
DEM - Application Enhancement	New	0%	ERP Patch v4.5.4.7 Migration Test	Related to this Package

The 'Relationship' dropdown menu is open, showing the following options: Related to this Package, Parent of this Package, Related to this Package, Predecessor, and Successor.

4. Click **OK** or **Save**.

The change to the package is saved.

Deleting References

To delete a reference from the Workbench:

1. Open a package.

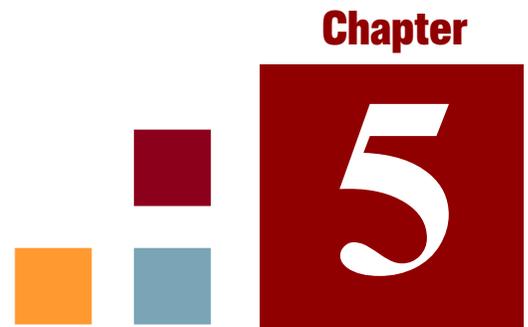
For information on how to open a package, see *Opening Packages from the Workbench* on page 100.

2. Click the **References** tab.
3. Select the reference, and then click **Remove**.

Adding a Package Line to an In-Progress Package

Occasionally, you may want to add a package line to a package that is already submitted. You can add package lines to a package that has a status of In Progress. For information on how to do this, see [Adding Lines to Packages](#) on page 60.

A new package line is not submitted until you save the package. After you save the package, the new package line is submitted to the first step in the workflow, and the user can process it separately until it catches up with the other lines in the package. At submission, the new package line is not automatically assigned the same status as the other lines.



Chapter
5

Managing Packages

In This Chapter:

- *Managing Packages in Mercury Deployment Management*
 - *Merging Packages*
 - *Creating New Package Groups*
 - *Editing Package Groups*
 - *Deleting Packages*
 - *Cancelling Package Lines*
-

Managing Packages in Mercury Deployment Management

After a package is submitted in Mercury Deployment Management, it can be processed and managed. Package management involves copying, merging, grouping, editing, and deleting packages, modifying package groups, and cancelling package lines. The following sections address all of these tasks, except for copying packages. The procedure for copying packages is provided in *Creating Packages by Copying Existing Packages* on page 54.

Merging Packages

You can generate a new package by merging two or more existing packages. This is useful if you want to generate a single package consisting of certain package lines contained in separate packages.

To merge two packages:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

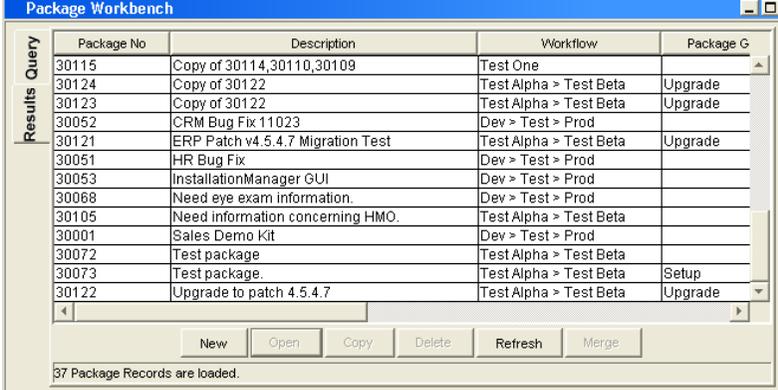
The Package Workbench opens.

4. On the **Query** tab, click **List**.

The **Results** tab lists all of the existing packages.

- Select the packages to merge.

To select nonadjacent rows, use the **ctrl** key. To select adjacent rows, use the **shift** key.

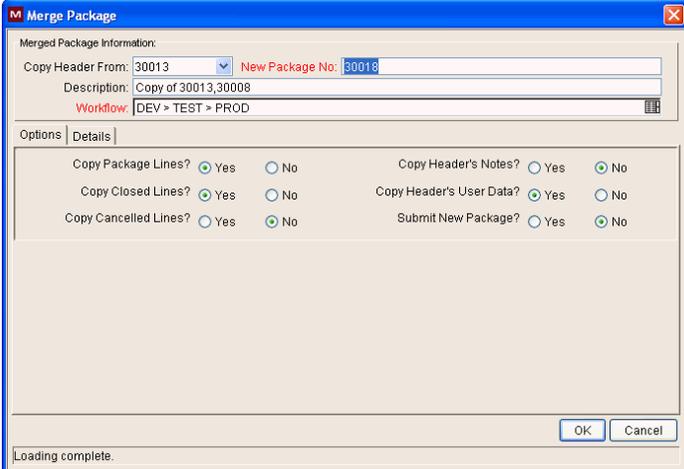


Package No	Description	Workflow	Package G
30115	Copy of 30114,30110,30109	Test One	
30124	Copy of 30122	Test Alpha > Test Beta	Upgrade
30123	Copy of 30122	Test Alpha > Test Beta	Upgrade
30062	CRM Bug Fix 11023	Dev > Test > Prod	
30121	ERP Patch v4.5.4.7 Migration Test	Test Alpha > Test Beta	Upgrade
30051	HR Bug Fix	Dev > Test > Prod	
30053	InstallationManager GUI	Dev > Test > Prod	
30068	Need eye exam information.	Dev > Test > Prod	
30105	Need information concerning HMO.	Test Alpha > Test Beta	
30001	Sales Demo Kit	Dev > Test > Prod	
30072	Test package	Test Alpha > Test Beta	
30073	Test package.	Test Alpha > Test Beta	Setup
30122	Upgrade to patch 4.5.4.7	Test Alpha > Test Beta	Upgrade

37 Package Records are loaded.

- On the **Results** tab, click **Merge**.

The Merge Package window opens to the **Options** tab.



Merge Package

Merged Package Information:

Copy Header From: 30013 New Package No: 30018

Description: Copy of 30013,30008

Workflow: DEV > TEST > PROD

Options | Details

Copy Package Lines? Yes No

Copy Header's Notes? Yes No

Copy Closed Lines? Yes No

Copy Header's User Data? Yes No

Copy Cancelled Lines? Yes No

Submit New Package? Yes No

OK Cancel

Loading complete.

- In the **New Package Number** field, you can either leave the generated package number, or type a different unique package number.
- In the **Workflow** field, you can leave the default workflow name, or select a different one in the auto-complete list.
- On the **Options** tab, select **Yes** or **No** for the listed options.

10. Click **OK**.

The Package Workbench prompts you to indicate whether you want to edit the new (merged) package.

11. To edit the package, click **Yes**. To save and close the package, click **No**.

Creating New Package Groups

It is often useful to link a new package to a package group for tracking and reporting purposes. After you link packages to package groups, you can use the package group to search for packages or in building custom portlets. You create new package groups from the Package Workbench.

To create a package group:

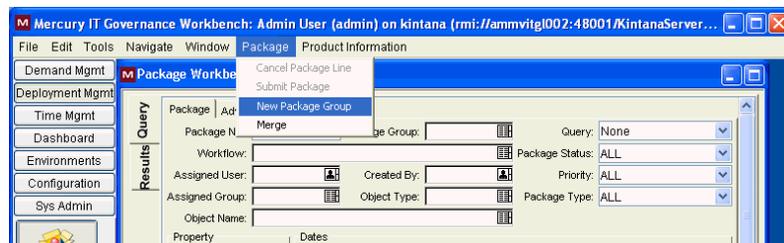
1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. From the **Package** menu, select **New Package Group**.



The Validation: KNTA - Package and Request Groups window opens and displays the existing groups for packages and requests.

Seq	Code	Meaning	Description	Enabled	Default
1	CUSTOMIZATION	Customization	Customization	Y	N
2	SETUP	Setup	Setup	Y	N
3	UPGRADE	Upgrade	Upgrade	Y	N

5. Click **New**.

The Add Validation Value window opens to the **Value Information** tab.

6. In the **Code** field, type the token name for the new **Package Group** list item.
7. In the **Meaning** field, type the name to display for this item in the **Package Group** list.
8. In the **Desc** field, you can type a short description of the package group.
9. To disable the new list value, clear the **Enable?** checkbox.

10. To make this value the default selection in the **Package Group** list, select the **Default** checkbox.
11. Click **OK**.

Editing Package Groups

You can edit existing package groups from within the Workbench. You can only change the KNTA Package and Request Groups Validation list if you have the required access grants and security privileges. For information about the licenses and access grants required to work with packages in Mercury Deployment Management, see the *Security Model Guide and Reference*.

Before you add or edit any package groups, consult your application administrator. Package groups are typically generated to adhere to specific business application standards.

To edit a package group:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

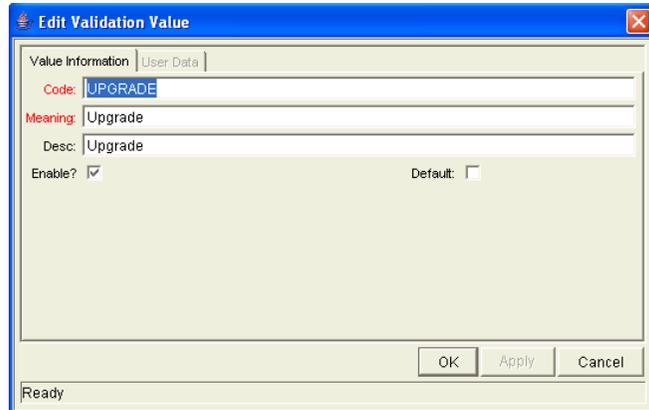
The Package Workbench opens.

4. Select **Package > New Package Group**.

The Validation: KNTA - Package and Request Groups window opens and lists existing package groups.

5. Select the package group you want to change, and then click **Edit**.

The Edit Validation Value window opens.



6. Edit the field values, as necessary, and then click **OK**.

The new validations are added to the Validation: KNTA - Package and Request Groups window.

7. Click **OK**.

Deleting Packages

You can delete packages from the Packages Workbench.

To delete a package:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

4. To see a list of all the packages, on the **Query** tab, click **List**.

The **Results** tab displays the packages list.

5. Select the packages you want to delete, and then click **Delete**.



To select nonadjacent rows, press and hold the **Ctrl** key. To select adjacent rows, press and hold the **Shift** key.

The Packages Workbench prompts you to confirm that you want to delete the package.

6. Click **Yes**.

Cancelling Package Lines

In some instances, you might want to cancel package lines because, for example, the files are out of date or are no longer required. You use the Package Workbench to cancel package line.

To cancel a package line:

1. Log on to Mercury IT Governance Center.
2. From the menu bar, select **Administration > Open Workbench**.

The Workbench opens.

3. From the shortcut bar, select **Deployment Mgmt > Packages**.

The Package Workbench opens.

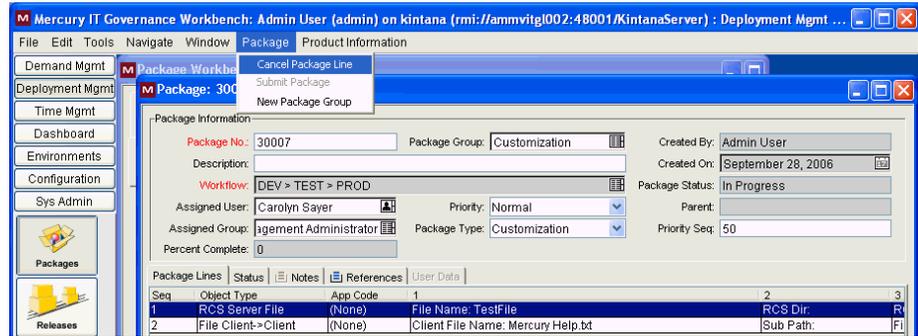
4. To see a list of all of the packages, on the **Query** tab, click **List**.

The **Results** tab displays the list.

5. Select a package that contains lines you want to cancel, and then click **Open**.

The Package window opens.

6. On the **Package Line** tab, select the package line to cancel.



7. From the menu bar, select **Package > Cancel Package Line**.

A

accessing

- package decision detail in Workbench **38**
- package execution history in Workbench **37**
- package graphical view in Workbench **37**
- package line execution log in Workbench **37**
- package line permissions in Workbench **38**
- package line transaction history in Workbench **38**
- package workflow step information URL in Workbench **39**

adding

- attachments to packages **87**
- existing packages to packages **88**
- existing packages to packages in Workbench **66**
- existing requests to packages **95**
- existing requests to packages in Workbench **72**
- new packages to packages **89**
- new packages to packages in Workbench **67**
- new requests to packages **96**
- new requests to packages in Workbench **73**
- notes to packages in the standard interface **86**

- notes to packages in Workbench **63, 108**
- package line to in-progress package in Workbench **111**
- package lines in Workbench **60**
- programs to packages **92**
- programs to packages in Workbench **69**
- projects to packages **93**
- projects to packages in Workbench **70**
- references to packages **86**
- references to packages in Workbench **64, 109**
- releases to packages **94**
- releases to packages in Workbench **71**
- tasks to packages **97**
- tasks to packages in Workbench **75**
- URLs to packages **99**
- URLs to packages in Workbench **76**

approving

- packages **82**

attachments

- adding to packages from the standard interface **87**

C

cancelling

- packages lines in Workbench **120**

condition workflow steps **19**

configuring
 package groups in Workbench 118
 package lines in Workbench 60
 packages general information in Workbench 59

copying
 creating packages in Workbench 54

creating
 package groups in Workbench 116
 packages as references in Workbench 56, 58
 packages in Workbench 52
 packages overview 46
 packages using copy in Workbench 54

custom object types definition 17

D

decision workflow steps 18

deleting
 package references in Workbench 110
 packages in Workbench 119

Demand Management
 integrating with Deployment Management 21

Deployment Management 21
 Extension for E-Commerce Technologies 22
 Extension for Oracle E-Business Suite 22
 Extension for Oracle Technology 22
 Extension for PeopleSoft Enterprise 22
 Extension for SAP® Solutions 22
 integrating with Version Control 21
 integration with third-party applications 22
 overview 13
 using the standard interface 15
 using the Workbench 15

E

Environment Workbench 49

environments
 defined for packages 46

definition 20
 verifying in the Environment Workbench 49

execution workflow steps 18

Extensions
 for E-Commerce Technologies 22
 for Oracle E-Business Suite 22
 for Oracle Technology 22
 for PeopleSoft Enterprise 22
 for SAP® Solutions 22

I

integrating
 Deployment Management 21

integrating with Demand Management 21

M

managing
 packages in Workbench 114

merging packages in Workbench 114

migrators, defined 18

My Packages portlet
 opening packages from 82

N

notes
 adding to packages from the standard interface 86
 adding to packages in Workbench 63, 108

notifications
 links to packages 82

O

object types
 custom objects definition 17
 defining in Workbench 50
 definition 16
 migrator definition 18
 standard objects definition 17
 where defined 28

-
- opening
 - packages **81**
 - packages from My Packages portlet **82**
 - packages from the Search Packages page **81**
 - packages in Workbench **100**
 - P**
 - Package Information section
 - Package window **34**
 - Package window
 - Package Information section **34**
 - Package Workbench **52**
 - packages
 - accessing decision detail in Workbench **38**
 - accessing graphical view in Workbench **37**
 - accessing line execution log in Workbench **37**
 - accessing line permissions in Workbench **38**
 - accessing line transaction history in Workbench **38**
 - accessing package execution history in Workbench **37**
 - accessing workflow step information URL in Workbench **39**
 - adding attachments from the standard interface **87**
 - adding existing packages in Workbench **66**
 - adding existing requests **95**
 - adding existing requests in Workbench **72**
 - adding line to in-progress package in Workbench **111**
 - adding new packages in Workbench **67**
 - adding new requests **96**
 - adding new requests in Workbench **73**
 - adding notes from the standard interface **86**
 - adding notes in Workbench **63, 108**
 - adding package lines in Workbench **60**
 - adding packages **89**
 - adding programs **92**
 - adding programs in Workbench **69**
 - adding projects **93**
 - adding projects in Workbench **70**
 - adding references **86**
 - adding references in Workbench **64**
 - adding references to Workbench **109**
 - adding releases **94**
 - adding releases in Workbench **71**
 - adding tasks **97**
 - adding tasks in Workbench **75**
 - adding to packages **88**
 - adding URLs **99**
 - adding URLs in Workbench **76**
 - approving **82**
 - cancelling package lines in Workbench **120**
 - checking status in Workbench **100**
 - configuring general information in Workbench **59**
 - creating by copying in Workbench **54**
 - creating by referencing in Workbench **56, 58**
 - creating in Workbench **52**
 - creating overview **46**
 - creating package groups in Workbench **116**
 - definition **16**
 - deleting in Workbench **119**
 - deleting references in Workbench **110**
 - editing package groups in Workbench **118**
 - in the standard interface **29**
 - in the Workbench **33**
 - managing in Workbench **114**
 - merging in Workbench **114**
 - opening **81**
 - opening from email notification links **82**
 - opening in Workbench **100**
 - overview **26**
 - package detail page header section **31**
 - package detail page Notes section **32**
 - package detail page Reference section **32**
 - package detail page Status section **31**
 - processing **81**
 - processing in Workbench **100**
 - processing packages through decisions in Workbench **104**
-

- processing packages through executions in Workbench **106**
- processing steps in Workbench **101**
- processing through workflow steps in Workbench **101**
- reference relationships **40**
- submitting **77**
- submitting in Workbench **77**
- updating references in Workbench **109**
- Workbench package window Notes tab **39**
- Workbench package window Package Lines tab **35**
- Workbench package window References tab **39**
- Workbench Package window Status tab **35**
- workflows defining environments **46**
- predecessor reference relationships **40, 43**
- processing
 - packages **81**
 - packages in Workbench **100, 101**
 - packages through decisions in Workbench **104**
 - packages through executions in Workbench **106**
 - packages through workflow steps in Workbench **101**
- programs
 - adding to packages **92**
 - adding to packages in Workbench **69**
- projects
 - adding to packages **93**
 - adding to packages in Workbench **70**

R

- references
 - added to packages in Workbench **109**
 - adding attachments to packages from the standard interface **87**
 - adding existing packages to packages **88**
 - adding existing packages to packages in Workbench **66**
 - adding existing references to packages **95**

- adding existing requests to packages in Workbench **72**
- adding new packages to packages **89**
- adding new packages to packages in Workbench **67**
- adding new requests to packages **96**
- adding new requests to packages in Workbench **73**
- adding programs to packages **92**
- adding programs to packages in Workbench **69**
- adding projects to packages **93**
- adding projects to packages in Workbench **70**
- adding releases to packages **94**
- adding releases to packages in Workbench **71**
- adding tasks to packages **97**
- adding tasks to packages in Workbench **75**
- adding to packages **86**
- adding to packages in Workbench **64**
- adding URLs to packages **99**
- adding URLs to packages in Workbench **76**
- creating packages in Workbench **56, 58**
- deleting in packages in Workbench **110**
- package relationships **40**
- predecessor relationships **40, 43**
- relationships list **41**
- successor relationships **40, 43**
- updated in packages in Workbench **109**

- releases
 - adding to packages **94**
 - adding to packages in Workbench **71**
- requests
 - adding to packages **95, 96**
 - adding to packages in Workbench **72, 73**

S

- Search Packages page
 - opening packages **81**
- Segregation of Duties tab **48**
- standard interface

- packages in **29**
- standard object types definition **17**
- submitting
 - packages **77**
 - packages in Workbench **77**
- subworkflow workflow steps **19**
- successor reference relationships **40, 43**

T

- tasks
 - adding to packages **97**
 - adding to packages in Workbench **75**

U

- updating
 - package references in Workbench **109**
- URLs
 - adding to packages **99**
 - adding to packages in Workbench **76**

V

- Version Control
 - integrating with Deployment Management **21**
- viewing
 - package status in Workbench **100**

W

- workflow steps
 - condition **19**
 - decision **18**
 - execution **18**
 - segregation of duties **48**
 - subworkflow **19**
- workflows
 - definition **18**
 - packages and environments **46**
 - processing packages in Workbench **101**
 - processing packages through decisions in Workbench **104**

- processing packages through executions in Workbench **106**
- workflow step definition **18**
- workflow step types **18**

