

Peregrine

Get-Resources 4.0

Installation Guide

For Windows, AIX, Linux, and Solaris

Copyright © 2002 Peregrine Systems, Inc. or its subsidiaries. All rights reserved.

Information contained in this document is proprietary to Peregrine Systems, Incorporated, and may be used or disclosed only with written permission from Peregrine Systems, Inc. This book, or any part thereof, may not be reproduced without the prior written permission of Peregrine Systems, Inc. This document refers to numerous products by their trade names. In most, if not all, cases these designations are claimed as Trademarks or Registered Trademarks by their respective companies.

Peregrine Systems® and ServiceCenter® are registered trademarks and Get-Resources™ is a trademark of Peregrine Systems, Inc. or its subsidiaries.

This product includes software developed by the Apache Software Foundation (<http://www.apache.org/>) and by Advantys (<http://www.advantys.com>). This product also contains software developed by the following companies or individuals: Sun Microsystems, Inc., Jean-Marc Lugin, Netscape Communications Corporation, and Original Reusable Objects, Inc.

This document and the related software described in this manual are supplied under license or nondisclosure agreement and may be used or copied only in accordance with the terms of the agreement. The information in this document is subject to change without notice and does not represent a commitment on the part of Peregrine Systems, Inc. Contact Peregrine Systems, Inc., Customer Support to verify the date of the latest version of this document.

The names of companies and individuals used in the sample database and in examples in the manuals are fictitious and are intended to illustrate the use of the software. Any resemblance to actual companies or individuals, whether past or present, is purely coincidental.

If you need technical support for this product, or would like to request documentation for a product for which you are licensed, contact Peregrine Systems, Inc. Customer Support by e-mail at support@peregrine.com.

If you have comments or suggestions about this documentation, contact Peregrine Systems, Inc. Technical Publications by e-mail at doc_comments@peregrine.com.

This edition applies to version 4.0 of the licensed program.

Peregrine Systems, Inc.
Worldwide Corporate Headquarters
3611 Valley Centre Drive San Diego, CA 92130
Tel 800.638.5231 or 858.481.5000
Fax 858.481.1751
www.peregrine.com



Contents

	About this Guide	7
	Book audience	7
	Related documentation	8
	Associated applications	8
	Terminology	8
	Typographical conventions	9
	Special elements	9
	Organization of the guide	10
	Contacting customer support	10
	Peregrine CenterPoint web site	10
	Documentation web site	11
Chapter 1	Introduction	13
	Installation requirements.	14
	Types of installations	15
	Back-end systems	16
Chapter 2	Installation for Windows Platforms	17
	Choosing an installation environment	18
	Development Environment.	18
	Production Environment	20
	Migrating Get-Resources from previous versions.	21
	Recreating customizations in Get-Resources 4.0	21
	Configuring alternate application servers	24
	Tomcat 4.1.12 connecting to IIS 5.0	25

WebSphere 4.0.2	33
Installing WebSphere Portal Server	38
Recommended WebSphere Portal Server configuration	38
Alternate WebSphere Portal Server configuration	40
Configuring WebSphere Translation Server for Get-Resources	47
WebLogic 6.1 SP3 or SP4.	50
JRun 3.1.	54
Typical installation option	62
Typical installation components.	62
Typical installation procedures	63
Custom installation option	68
Custom installation components	68
Custom installation procedures	69
Uninstalling Get-Resources	77
Testing your installation	78
Chapter 3 Installation for Unix Platforms	79
Choosing an installation environment	80
Development Environment.	80
Production Environment	82
Migrating Get-Resources from previous versions.	83
Recreating customizations in Get-Resources 4.0	83
Configuring an existing back-end database for Get-Resources 4.0	86
Configuring alternate application servers	87
Tomcat 4.1.12	87
WebSphere 4.0.2	90
Installing WebSphere Portal Server	95
Recommended WebSphere Portal Server configuration	95
Alternate WebSphere Portal Server configuration	97
Configuring WebSphere Translation Server for Get-Resources	104
WebLogic 6.1 SP3 or SP4.	107
JRun 3.1.	109
Typical Installation Option	118
Typical Installation Components	118
Typical Installation Procedures	120

	Custom Installation Option.	126
	Custom Installation Components	126
	Custom Installation Procedures	127
	Uninstall—AIX, Linux, or Solaris	138
Chapter 4	Load Balancing	139
	Load balancing application servers	140
	Creating multiple instances of Tomcat for Apache	142
	Copying the Tomcat directory	143
	Editing the workers.properties file	143
	Editing the mod_jk.conf-auto file	145
	Editing the httpd.conf file	145
	Editing the server.xml files for Apache	146
	Editing the jk2.properties files for Apache	147
	Installing Tomcat instances as services for Apache	148
	Testing load balancing on Apache	148
	Creating multiple instances of Tomcat for IIS	150
	Copying the Tomcat directory	151
	Configuring the ISAPI Plugin for IIS.	151
	Creating and configuring a jakarta virtual directory in IIS	152
	Configuring the isapi_redirector2.dll as an ISAPI filter	152
	Creating and configuring an oaa virtual directory in IIS.	153
	Editing the workers2.properties file for IIS	153
	Editing the server.xml files for IIS	155
	Editing the jk2.properties files for IIS	156
	Installing Tomcat instances as services for IIS	157
	Testing load balancing on IIS	157
Chapter 5	Back-end System Configuration	159
	Configuring AssetCenter for Get-Resources	160
	Overview of AssetCenter as a back-end system.	160
	Using the Get-Resources AssetCenter demo database.	160
	Using an existing AssetCenter database.	161
	Configuring ServiceCenter for Get-Resources	167
	Overview of ServiceCenter as a back-end system.	167
	Using ServiceCenter 4.x with Get-Resources	167

	Using ServiceCenter 5.x with Get-Resources	168
Chapter 6	Get-Resources Configuration	169
	Logging in to Peregrine Portal Administration	170
	Configuring connections to the back-end system.	173
	Instructions for AssetCenter	173
	Instructions for ServiceCenter	178
	Index	185



About this Guide

Get-Resources™, part of the Get-It™ suite of Employee Self-Service products from Peregrine Systems®, provides a way for businesses to make the procurement of resources available to employees through the corporate intranet.

Get-Resources integrates with AssetCenter Procurement or ServiceCenter Request Management to enable employees to create requests for resources and to streamline the approval workflow of those requests throughout the organization.

This book provides step-by-step instructions for installing Get-Resources. This guide enables you to:

- Install the Peregrine OAA Platform and Get-Resources.
- Configure AssetCenter or ServiceCenter for use as a back-end system with Get-Resources.
- Utilize the Peregrine Portal Administration module to connect to the back-end system and configure Get-Resources settings.

Book audience

This guide is for Get-Resources administrators who configure and maintain the application. To use this guide effectively, you need to have knowledge of the following:

- XML and ECMA Script (or JScript/JavaScript)
- Operating guides, reference manuals, and other documentation for your PC hardware and operating system
- AssetCenter administration and functionality
- ServiceCenter administration and functionality

Related documentation

Refer to the following documentation for additional information:

- *Get-Resources Administration Guide* describes the Peregrine OAA platform and Get-Resources administration.
- *Get-Resources Release Notes* covers any late breaking documentation or known issues with Get-Resources. These documents are constantly updated and posted to the Customer Support web site. See [Contacting customer support](#) on page 10 for details on accessing the Customer Support website.

Associated applications

This guide does not contain information about products that may be used with Get-Resources, such as Peregrine OAA, AssetCenter, ServiceCenter, or Password Management. Refer to the appropriate product documentation for information about installing, configuring, and using these associated applications.

Note: A back-end system, either AssetCenter or ServiceCenter, must be installed and configured before you can install and configure Get-Resources. Peregrine OAA installs with Get-Resources, and only the installation of Peregrine OAA for Get-Resources is included in this guide.

Terminology

The terminology used in this guide and in the Get-Resources interface is based on AssetCenter 3.6 and 4.x, and ServiceCenter 4.x and 5.x.

Typographical conventions

This guide uses typeface conventions to indicate special terms and actions. These conventions and their meanings are:

Convention	Meaning
Bold	Information that you must type exactly as shown appears in bold . The names of buttons, menus, and menu options also appear in bold .
<i>Italics</i>	Variables and values that you must provide appear in <i>italics</i> . New terms also appear in <i>italics</i> .
Monospace	Code or script examples, output, and system messages appear in a monospace font. <pre>var msgTicket = new Message("Problem"); ... msgTicket.set("_event", "epmc");</pre> <p>An ellipsis (...) is used to indicate that portions of a script have been omitted because they are not needed for the current topic. Samples of code are not entire files, but they are representative of the information discussed in a particular section.</p>
Sans Serif	Filenames, such as login.asp , appear in a sans serif font.

Special elements

This book uses special elements to help you locate information. These special elements and their uses are in the following table:

Element	Usage
Important:	Information that is required to complete a task
Note:	Information that is of general interest
Tip:	Information that can make a task easier or faster
Warning:	Information that is needed when there is a risk of losing data

Organization of the guide

The following table shows you where in this guide to find the information you need.

This section	Provides information about
<i>Chapter 1, Introduction</i>	Installation requirements and description of different types of installations for Get-Resources.
<i>Chapter 2, Installation for Windows Platforms</i>	Detailed instructions for installing Get-Resources in a Windows environment.
<i>Chapter 3, Installation for Unix Platforms</i>	Detailed instructions for installing Get-Resources in a Unix environment.
<i>Chapter 4, Load Balancing</i>	Information on load balancing Get-Resources for optimum system memory management.
<i>Chapter 5, Back-end System Configuration</i>	Detailed instructions for configuring ServiceCenter or AssetCenter as a back-end system with Get-Resources.
<i>Chapter 6, Get-Resources Configuration</i>	Final configuration settings within the Peregrine Portal Administration module to complete the installation of Get-Resources.

Contacting customer support

For further information and assistance with this release, contact Peregrine Systems' Customer Support.

Peregrine CenterPoint web site

Current details of local support offices are available through the following main contacts or through the Peregrine CenterPoint Web site at:

<http://support.peregrine.com>

You need your current login user name and password to access this Web page.

To contact Peregrine customer support

- 1 Log in to the Web site with your login user name and password.
- 2 Click Go beside the CenterPoint support area.
- 3 From **Contents** on the left, select **Whom Do I Call?** to display the **Peregrine Worldwide Contact Information**.

Documentation web site

A complete listing of the available documentation is on Peregrine's CenterPoint Web site at:

<http://support.peregrine.com>

Important: Release Notes for this product are continually updated after the release of the product. Visit the Peregrine Customer Support Web site to ensure that you have the most current version of the Release Notes.

1 Introduction

CHAPTER

This chapter covers the following topics for Get-Resources:

- *Installation requirements* on page 14
- *Types of installations* on page 15
- *Back-end systems* on page 16

Installation requirements

This section outlines the recommended minimum configuration for proper installation and configuration of Get-Resources. Before beginning installation, ensure that you have the following.

Requirement	for Windows	for Unix
Java run-time environment	Java 2 SDK Standard Edition v1.3.1_05. <i>Available on the Get-Resources Installation CD.</i>	Java 2 SDK Standard Edition v1.3.1_05. <i>Available on the Get-Resources Installation CD.</i>
Application server	Any one of the following: ■ Tomcat 4.1.12 <i>Available on the Get-Resources Installation CD.</i> ■ WebSphere 4.0.2 or later ■ WebLogic 6.1 SP3 or later ■ JRun 3.1	Any one of the following: ■ Tomcat 4.1.12 <i>Available on the Get-Resources Installation CD.</i> ■ WebSphere 4.0.2 or later ■ WebLogic 6.1 SP3 or later ■ JRun 3.1
Back-end database(s)	ServiceCenter 4.0.x or later AssetCenter 3.6 or later	ServiceCenter 4.0.x or later AssetCenter 3.6 or later
Operating systems	■ Windows 2000 Server SP2	■ AIX 5.1 ■ Red Hat Linux 7.3 ■ Solaris 2.7 or Solaris 2.8
Web Server	One of the following: ■ Apache 2.0.43 <i>Available on the Get-Resources installation CD.</i> ■ Microsoft IIS Server 5.0 ■ IBM HTTP Server 1.3.19 <i>Available on the WebSphere installation CD and from the IBM support Website</i>	One of the following: ■ Apache 2.0.43 <i>Available on the Get-Resources installation CD.</i> ■ IBM HTTP Server 1.3.19 <i>Available on the WebSphere installation CD and from the IBM support Website</i>
System Processor	Pentium, 400 MHz or faster	Pentium, 400 MHz or faster

Requirement	for Windows	for Unix
RAM	512 MB or more.	512 MB or more
Hard disk space	100 MB for Get-Resources.	100 MB for Get-Resources.

Note: The AssetCenter API must be installed on the same system as your Web application. However, the AssetCenter database can be on another system. This means that when you install AssetCenter on the Get-Resources server, you must either use a full installation or a custom installation where you select the “AssetCenter API” option.

Types of installations

The Get-Resources installer offers two basic types of installation:

- Typical installation
- Custom installation

A *typical installation* installs a fixed configuration of Get-Resources all on one server. Typical installations are intended to set up development environments (see below).

A *custom installation* allows you to choose the exact components installed on a given server. Custom installations are intended for users who will be using alternate application servers, web servers, or to set up a production environment.

The Get-Resources custom installation can be optimized for two types of environments:

- Development environment
- Production environment

A *development environment* installation places all needed software and data on one server. It is intended for Get-Resources implementers to review application functionality and test customizations before deploying to a production environment. By default, the Get-Resources installer uses the development environment installation.

A *production environment* installation is optimized for performance and scalability. Each software function, such as an application server and Web server, are installed on different servers. In addition, there may be multiple instances of any one part of the basic Get-Resources environment. Given the amount of flexibility involved in such an installation, users must manually set up a production environment.

Back-end systems

In order to use Get-Resources you must have a properly configured back-end system. Get-Resources utilizes the back-end system to:

- Authenticate users and define access rights
- Process application workflows and store data
- Store personalization settings for the web application

Get-Resources uses ServiceCenter or AssetCenter as a back-end system. Refer to *Installation requirements* on page 14 for a complete list of the AssetCenter and ServiceCenter versions compatible with Get-Resources.

2 Installation for Windows Platforms

CHAPTER

This chapter covers the following topics:

- *Choosing an installation environment* on page 18
- *Migrating Get-Resources from previous versions* on page 21
- *Configuring alternate application servers* on page 24
- *Typical installation option* on page 62
- *Custom installation components* on page 68
- *Uninstalling Get-Resources* on page 77
- *Testing your installation* on page 78

Choosing an installation environment

You can install Get-Resources in one of two installation environments:

- Development environment
- Production environment

The Get-Resources development environment is intended for you to evaluate product features and customize your installation prior to deployment in a production environment. In a development environment, you install all software required for Get-Resources on one computer system.

You have two choices of development environment:

- Typical installation
 - Apache 2.0 Web server
 - Get-Resources deployed on Tomcat 4.1.12 application server
- Custom installation
 - Choice of Web server
 - Choice of application server where you deploy Get-Resources

The Get-Resources production environment is intended to maximize server performance and scalability, and to deploy any customizations you have made. In a production environment, you install the various components of Get-Resources on different servers to maximize performance.

You have two choices of production environment:

- Typical installation
 - Apache 2.0 Web server
 - Get-Resources deployed on multiple instances of Tomcat 4.1.12 application server
- Custom installation
 - Choice of Web server
 - Choice of application server where you deploy Get-Resources

Development Environment

The following procedures describe how to install Get-Resources in a development environment.

To install Get-Resources in a typical development environment:

- Step 1** Acquire all necessary hardware and software.
- Step 2** Install the back-end database required for Get-Resources.
- Step 3** Run the Get-Resources installer and select the Typical installation option.
See *Typical installation option* on page 62.
- Step 4** Configure the back-end databases and create Get-Resources users.

To install Get-Resources in a custom development environment:

- Step 1** Acquire all necessary hardware and software.
- Step 2** Install the back-end database required for Get-Resources.
- Step 3** Install alternate application and Web servers.
- Step 4** Configure the alternate application server for Get-Resources. See *Configuring alternate application servers* on page 24.
- Step 5** Run the Get-Resources installer and select the Custom installation option.
See *Custom installation option* on page 68.
- Step 6** Configure the back-end databases and create Get-Resources users.

Production Environment

The following procedures describe how to install Get-Resources in a production environment.

To install Get-Resources in a typical production environment:

- Step 1** Acquire all necessary hardware and software.
- Step 2** Install the back-end database required for Get-Resources on a separate server.
- Step 3** Run the Get-Resources installer and select the Typical installation option. See *Typical installation option* on page 62.
- Step 4** Configure multiple instances of Tomcat for load balancing on the Apache Web server.
- Step 5** Configure the back-end databases and create Get-Resources users.

To install Get-Resources in a custom development environment:

- Step 1** Acquire all necessary hardware and software.
- Step 2** Install the back-end database required for Get-Resources.
- Step 3** Install the alternate application server and Web server on separate servers.
- Step 4** Configure the alternate application server for Get-Resources. See *Configuring alternate application servers* on page 24.
- Step 5** Run the Get-Resources installer and select the Custom installation option. See *Custom installation option* on page 68.
- Step 6** Configure the Web servers and application servers for load balancing.
- Step 7** Configure the back-end databases and create Get-Resources users.

Migrating Get-Resources from previous versions

To migrate older versions of Get-It or Get-Resources to Get-Resources 4.0 requires both a manual data migration process and the recreation of any interface customizations you have made. The following steps describe the migration process.

Important: You should backup all Get-Resources and back-end system data prior to performing any steps for migration.

To migrate previous versions to Get-Resources 4.0:

- Step 1** Review the customizations of previous version and determine which customizations need to be recreated in Get-Resources 4.0. See *Recreating customizations in Get-Resources 4.0* on page 21.
- Step 2** Install Get-Resources 4.0 on a new system. See *Choosing an installation environment* on page 18.
- Step 3** Apply any required configuration changes to the back-end database you want to migrate to Get-Resources 4.0. See *Chapter 5, Back-end System Configuration*

Recreating customizations in Get-Resources 4.0

You cannot directly migrate customizations implemented in previous versions to Get-Resources 4.0. Instead, you must recreate your changes using the new features and methods available in Get-Resources 4.0.

The following sections describe how to recreate your customizations from previous versions.

No customizations

If you have made no customizations to Get-Resources, you can simply install Get-Resources 4.0 on a new system and migrate your data from your existing back-end database.

Customized JSP files

In previous versions, customers had to directly modify JSP files in order to add or remove certain functionality. The following table describes how to recreate some of the more common JSP file modifications.

JSP file modification	New method to use
Remove the user self-registration option from login page	Enable or disable the user registration option from the Administration Settings page
Remove the change password option from the login page	Enable or disable the change password option from the Administration Settings page

Personalized pages

Get-Resources 4.0 offers many more pages that you can personalize directly from the Web interface. If you personalized pages in a previous version, you must recreate your personalized pages in Get-Resources 4.0 using DocExplorer.

You can use personalization to:

- Add or remove fields from a page
- Save a personalized search results or details on your portal page

Customized skins, stylesheets, and themes

Get-Resources 4.0 has combined all interface images and stylesheets into themes. Users can no longer select separate skins and stylesheets. The new themes consist of skins (which themselves are composed of image files, frame definitions, and layer files), cascading stylesheet definitions, and XSL templates.

Although you may copy over older custom themes to Get-Resources 4.0, you may experience rendering errors due to the new images, CSS definitions, frame definitions, and layers. It is recommended that you recreate any custom themes using the Get-Resources 4.0 version of the classic theme as your template.

Alternate login pages and authentication methods

If you used a custom login page or an alternate authentication method in a previous version, you can re-use or recreate these customizations using the updated instruction. You can find information about alternate security methods in the *Get-Resources Administration Guide*.

Customizations made with a previous tailoring kit

Many customizations that required a tailoring kit in previous versions can now be done directly from the Get-Resources Web interface. The following table describes how to recreate some of the more common tailoring kit changes.

Tailoring kit modification	New method to use
Added or removed fields form a form	Add or remove fields from Personalization
Added a new language or locale to the Get-Resources interface	Create and edit language strings files directly. You may also purchase officially supported language packs from Peregrine Systems
Made changes to the common, portal, or Peregrine Studio packages	These packages are no longer available for tailoring, however most common interface settings can now be customized from the Administration Settings page.
Made changes to schemas or ECMA server-side scripts	Review new functionality and determine if you still need the customized scripts and schemas. If you do need the customizations, you will to recreate them in the current version of the Get-Resources tailoring kit.

Configuring alternate application servers

You must install a Java-enabled application server to support your Peregrine Web applications. Peregrine OAA supports the following alternate application servers:

- *Tomcat 4.1.12 connecting to IIS 5.0*
- *WebSphere 4.0.2*
- *WebLogic 6.1 SP3 or SP4*
- *JRun 3.1*

The Get-Resources typical installation option installs Tomcat 4.1.12 and connects it to an Apache 2.0 web server. You can also install Tomcat 4.1.12 using the custom installation option.

Important: If you want to use an application server other than Tomcat 4.1.12, then you must configure your application and Web servers *prior* to running the Get-Resources installer.

See the following sections for instructions configuring alternate application servers for Get-Resources.

Tomcat 4.1.12 connecting to IIS 5.0

You can use the Get-Resources installer to install the Tomcat application server. If you use the typical installation option, the Get-Resources installer configures Tomcat for the Apache Web server. In order to configure the Tomcat for the IIS Web server, you must perform a custom installation and configure IIS using the following instructions.

To configure Tomcat to connect to an IIS 5.0 Web server:

- Step 1** Run the Get-Resources installer and select the Custom installation option. See *Custom installation option* on page 68.
- Step 2** Configure the ISAPI Plugin for IIS. See *Configuring the ISAPI Plugin for IIS* on page 26.
- Step 3** Configure IIS to use `isapi_redirector2.dll` as an ISAPI Filter. See *Configuring the isapi_redirector2.dll as an ISAPI filter* on page 26.
- Step 4** Create and configure a jakarta virtual directory in IIS. See *Configuring a jakarta virtual directory in IIS* on page 27.
- Step 5** Create and configure an oaa virtual directory in IIS. See *Configuring an oaa virtual directory in IIS* on page 27.
- Step 6** Edit the `workers2.properties` file of the first or master Tomcat instance to set the values for each additional Tomcat instance. See *Editing the workers2.properties file for IIS* on page 28.
- Step 7** Edit the `server.xml` file. See *Editing the server.xml file for IIS* on page 29.
- Step 8** Edit the `jk2.properties` file. See *Editing the jk2.properties file for IIS* on page 30.
- Step 9** Install Tomcat as a service using `installservice.bat` (Optional). This file can be found in the `Tomcat\bin` directory. See *Installing Tomcat as a service* on page 31.
- Step 10** Pre-compile JSP files for your production environment. See *Precompiling JSP files for a Tomcat production environment* on page 31.

Configuring the ISAPI Plugin for IIS

The ISAPI plugin for IIS establishes a connection between Tomcat and the IIS Web server. Before configuring IIS to use this connector, you must update the registry file entry for the connector to ensure that it has the proper paths listed for the Tomcat application server.

The Get-Resources installer automatically places a copy of the ISAPI plugin for IIS in the following folder:

`c:\Program Files\Peregrine\Common\Tomcat4\bin`

Use the following procedures to configure the plugin for your intranet environment.

To configure the ISAPI plugin for IIS:

- 1 Open the file `jk2.reg` in a text editor. This file is located at:
`C:\Program Files\Peregrine\Common\Tomcat4\conf`
- 2 Verify that the “serverRoot” and “workersFile” values list the proper installation path to Tomcat. By default these values are:

```
“ServerRoot”=“C:\\Program Files\\Peregrine\\Common\\Tomcat4”
“workersFile”=“C:\\Program Files\\Peregrine\\Common\\Tomcat4\\conf\\
workers2.properties”
```
- 3 Save and close the `jk2.reg` file.
- 4 Double-click on the `jk2.reg` file from Windows Explorer.
Windows adds the registry settings to the Windows registry.

Configuring the `isapi_redirector2.dll` as an ISAPI filter

To establish a connection between Tomcat and IIS, you will need to install `isapi_redirector2.dll` as an ISAPI filter.

To install `isapi_redirect2.dll` as an ISAPI filter:

- 1 Open the Internet Services management console.
- 2 Right-click the **Default Web Site** node and then click **Properties**.
- 3 Click the **ISAPI Filters** tab.
- 4 Click **Add**.
- 5 Enter the following information:

- a **Filter Name:** jakarta. The filter name must match the name you defined the `jk2.reg` registry file. By default the filter name is jakarta.
 - b **Executable:** `isapi_redirector2.dll`. By default this file is located at:
`C:\Program Files\Peregrine\Common\Tomcat4\bin\isapi_redirector2.dll`
- 6 Click OK.
 - 7 Close the Internet Services management console.

Configuring a jakarta virtual directory in IIS

The ISAPI plugin for IIS requires a specific IIS virtual directory in order to run. Use the following guidelines to create the IIS virtual directory. For specific instructions about configuring IIS, refer to your Windows Help.

Requirements for a jakarta virtual directory

Requirement	Setting
Create virtual directory	jakarta
Directory access rights	execute
Map to physical path	<Tomcat>\bin\isapi_redirector2.dll

For <Tomcat>, enter the path to your tomcat installation. By default the path is:

`C:\Program Files\Peregrine\Common\Tomcat4`

Configuring an oaa virtual directory in IIS

To run Get-Resources from IIS, you need to create a virtual directory that maps to your Tomcat deployment folder.

Requirements for an oaa virtual directory

Requirement	Setting
Create virtual directory	<oaa>
Directory access rights	anonymous
Map to physical path	<Tomcat>\webapps\oaa
Set Execute Permissions to	Scripts and Executables
Remove “allow anonymous access” to	default.asp, login.asp, e_login_main_start.asp
Set security access to only allow “System” and “Authenticated Users” to	default.asp, login.asp, e_login_main_start.asp

For <oaa>, enter the name of the virtual directory you want to use for Get-Resources. Whatever name you enter here you will need to replicate in your application server configuration.

For <Tomcat>, enter the path to your tomcat installation. By default the path is:

C:\Program Files\Peregrine\Common\Tomcat4

Editing the workers2.properties file for IIS

For each server on which Tomcat is installed, there is only one **workers2.properties** file. Tomcat installs the **workers.properties** file in the **conf** directory of your primary Tomcat instance. This file will be shared by all other Tomcat instances on that particular server.

The **workers2.properties** file specifies the worker threads that the Web server connector will create in order to communicate with your Tomcat instances.

To edit the worker2.properties file:

- 1 Open the **workers2.properties** file (located in the **conf** directory of your Tomcat installation) in any text editor.
- 2 Create a **channel.socket** entry.

Example:

```
[channel.socket:<server>:<port>]
```

```
info=Description of Tomcat instance
debug=0
tomcatId=<server>:<port>
lb_factor=1
disabled=0
```

For *<server>*, enter the server name of where Tomcat is installed.

For *<port>*, enter the communications port on which Tomcat is listening.

- 3 Verify that the uri settings lists the proper IIS virtual directory. By default the virtual directory should be **oaa**.

If you have defined a different virtual directory other than **oaa** to run Get-Resources, you will need to change the uri values listed here.

Example:

```
[uri:/oaa/servlet/*]
info=Prefix mapping

[uri:/oaa/*.jsp]
info=Extension mapping
```

- 4 Save the file.

Editing the server.xml file for IIS

You must edit this file to connect Tomcat to IIS as well as to find the Get-Resources Web application files.

To edit the server.xml file:

- 1 Open the file server.xml in any text editor. By default this file is located at:
C:\Program Files\Peregrine\Common\Tomcat4\conf
- 2 Update the port number attribute of the *<Server>* element to a unique value that will not conflict with other port numbers on your network.

Example:

```
<Server port="8005" shutdown="SHUTDOWN" debug="0">
```

- 3 Update the port number attribute of the Coyote Connector *<Connector>* element to a unique value that will not conflict with other port numbers on your network.

Example:

```
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
port="8009" minProcessors="5" maxProcessors="75" enableLookups="true"
redirectPort="8443" acceptCount="10" debug="0"
connectionTimeout="20000" useURIVValidationHack="false"
protocolHandlerClassName="org.apache.jk.server.JkCoyoteHandler" />
```

- 4 Create a `<Context>` element entry from Tomcat to the Get-Resources deployment directory.

Add the entry just above the “examples” Context entry.

Example:

```
<Context path="/oaa"
docBase="<Tomcat>/webapps/oaa"
crossContext="false"
debug="0"
reloadable="false" >
</Context>
```

For the `docBase` attribute, set `<Tomcat>` to the absolute path of the first or master Tomcat instance.

- 5 Update the `<Engine>` element with the server name and communications port used by Tomcat.

List the server information in the `jvmRoute` attribute.

Example:

```
<Engine jvmRoute="localhost:8009" name="Standalone"
defaultHost="local host" debug="0">
```

- 6 Update the `<Host>` element with the `webapps` directory used by the first or master Tomcat instance.

List the server information in the `appBase` attribute.

Example:

```
<Host name="localhost" debug="0" appBase="<Tomcat>/webapps"
unpackWARs="true" autoDeploy="true">
```

For the `appBase` attribute, set `<Tomcat>` to the absolute path of the first or master Tomcat instance.

- 7 Save the file `server.xml`.

Editing the `jk2.properties` file for IIS

You will need to modify the `jk2.properties` file to set the `jk2` communication port.

To edit the `jk2.properties` file:

- 1 Open the `jk2.properties` file in any text editor. By default this file is located at:
`C:\Program Files\Peregrine\Common\Tomcat4\conf`
- 2 Insert a line for the `channelSocket` port. The port number must match the port number defined in `workers2.properties` file.

Example:

```
channelSocket.port=8009
```

- 3 Save the file.

Installing Tomcat as a service

After you have edited the Tomcat files, you can install Tomcat as Windows services using `installservice.bat`.

To install Tomcat as a service:

- 1 Open a DOS command prompt and change directories to your Tomcat bin directory.
- 2 Enter the following command to create each Tomcat instance:

```
installservice <service name> <tomcat_home> <jvm_dll_path>
```

Where *<service name>* is the name you wish to give the Tomcat service, *<tomcat_home>* is the Tomcat install directory of the instance for which you are creating the service, and *<jvm_dll_path>* is the Java SDK install directory.

The second and third parameters are optional if you have already set the `CATALINA_HOME` and `JAVA_HOME` environment variables.

Example:

```
installservice Tomcat8009 C:\Program Files\Peregrine\Common\Tomcat4  
C:\Program Files\Peregrine\Common\jdk1.3.1_05
```

- 3 Repeat step 1 through step 2 for each Tomcat service you wish to create.

Precompiling JSP files for a Tomcat production environment

If you plan to use Tomcat 4.1.12 in a production environment or on a system with multiple processors, you will need to pre-compile the JSP files deployed for Get-Resources. This extra configuration step is the result of a known issue with Tomcat. See the following URL for more details about this issue:

http://nagoya.apache.org/bugzilla/show_bug.cgi?id=14077

To pre-compile JSP files for Tomcat 4.1.12:

- 1 Stop the Tomcat application server.
- 2 Open a command prompt.
- 3 Enter one of the following commands based upon your operating system:

Operating system	Command required
------------------	------------------

Windows	set JASPER_HOME=<Tomcat directory>
---------	------------------------------------

For <Tomcat directory> enter the absolute path to your Tomcat installation.

- 4 Change directories to the Tomcat bin folder.

Operating system	Command required
------------------	------------------

Windows	cd %jasper_home%\bin
---------	----------------------

- 5 Run the precompile batch file.

Operating system	Command required
------------------	------------------

Windows	precompile <Web app name> <Tomcat instance name>
---------	--

For <Web app name> enter the name of the Get-Resources deployment folder. You may omit this name if you are using the default folder named **oaa**.

For <Tomcat instance name> enter the name of the Tomcat instance you have installed. You may omit this name if you are using the default instance named **Standalone**.

The batch file displays the progress of the conversion. When it is complete, the command prompt returns.

- 6 Start the Tomcat application server.

WebSphere 4.0.2

Use the following procedures to configure WebSphere to run Get-Resources on Windows.

To configure WebSphere 4.0.2:

- Step 1** Install WebSphere 4.0.2. Your version of WebSphere 4.0.2 includes the IBM HTTP Server.
- Step 2** Deploy the Portal WAR file to WebSphere to create the necessary folder structure for Get-Resources. See *Deploying the Portal WAR file to WebSphere* on page 33.
- Step 3** Set the JVM Java heap size for each WebSphere instance running Get-Resources. See *Setting the Java heap size* on page 35.
- Step 4** Create the virtual directory you want to use for Get-Resources in your Web server. See *Configuring a virtual directory for IBM HTTP Server* on page 37.
- Step 5** Run the Get-Resources installer.

If you plan on setting up a WebSphere Portal Server or a WebSphere Translation Server, see *Installing WebSphere Portal Server* on page 38 or *Configuring WebSphere Translation Server for Get-Resources* on page 47.

Deploying the Portal WAR file to WebSphere

The Portal WAR file creates the folder structure necessary to deploy Get-Resources in your application server. After you have deployed this file to WebSphere you will be ready to run the Get-Resources installer.

To deploy the Portal WAR file to WebSphere:

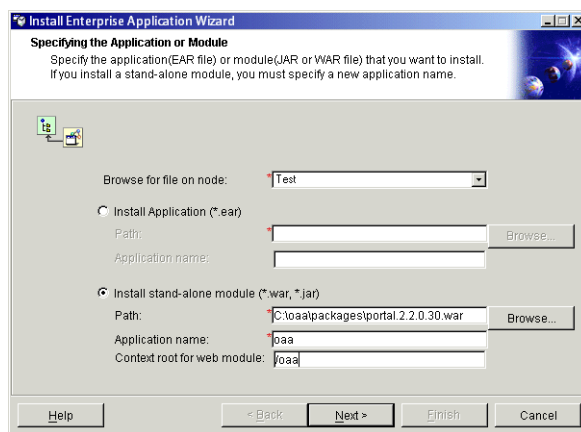
- 1** Verify that the WebSphere Admin Server has been started.
- 2** Open the WebSphere Advanced Administrator's Console (**Start > Programs > IBM WebSphere > Application Server > Administrator's Console**).
- 3** On the menu at the left side of the console, right-click on **Enterprise Applications** and select **Install Enterprise Application**.
- 4** On the screen displayed, do the following:
 - a** Select **Install stand-alone module**.
 - b** In the **Path** field, browse to the path to the portal<version #>.war file. The default is <CD Rom Drive>:\portal<version #>.war.

For <version #>, Select the most recent version available (4.0.0.44 or greater).

- c In the **Application Name** field, type oaa.
- d In the **Context Root** field, type the name of Get-Resources virtual Web server directory you wish to use. Example: /oaa.

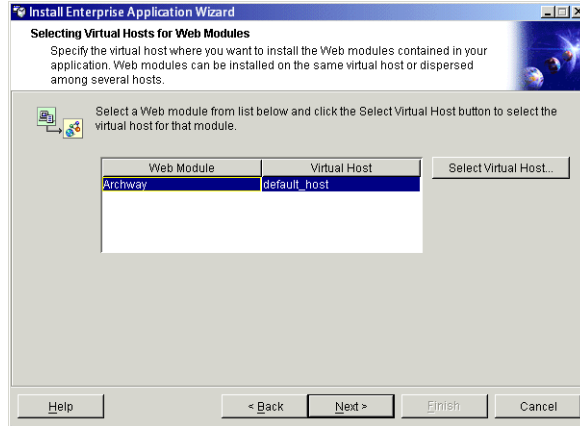
Important: You must create a Web server virtual directory matching the context root you enter here.

The following screen shows the completed form.

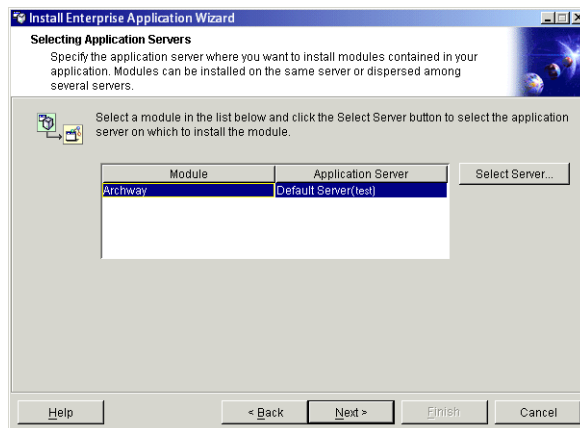


- 5 Click **Next**.
- 6 Click **Next** on the following dialog boxes. These screens will not be used.
 - Mapping Users to Roles
 - Mapping EJB Run As Roles to Users
 - Binding Enterprise Beans to JNDI Names
 - Mapping EJB References to Enterprise Beans
 - Mapping Resource References to Resources
 - Specifying the Default Datasource
 - Specifying Data Sources for Individual CMP Beans

- 7 In the Selecting Virtual Hosts for Web Modules, select the WebSphere server instance you want to use, and then click Next.



- 8 In the Selecting Application Servers dialog box, select the WebSphere server instance you want to use, and then click Next.



- 9 On the dialog box displayed, click Finish.

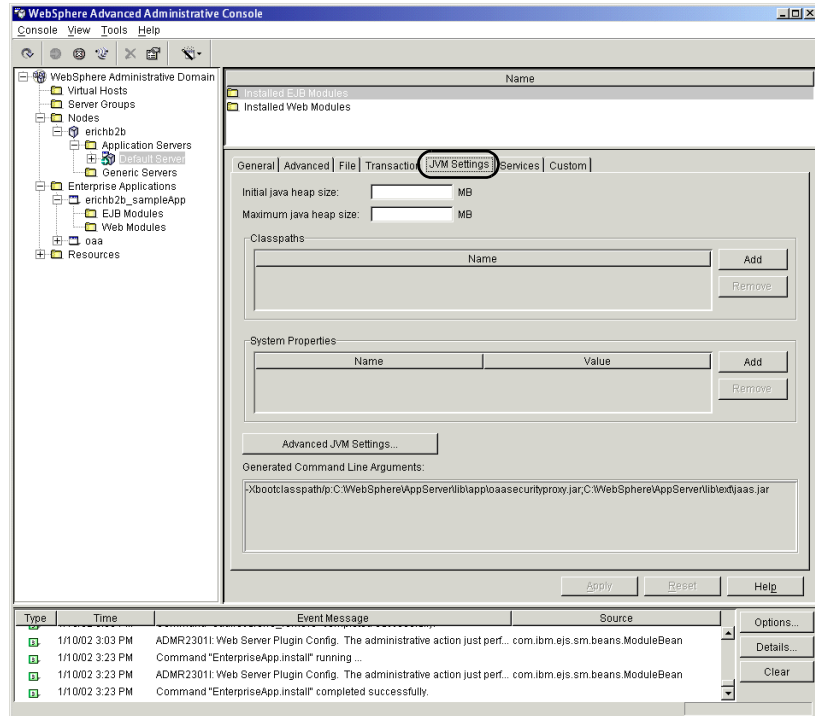
Setting the Java heap size

You can configure how much memory is available for your application server instances. The following instructions assume you are only using one WebSphere instance. You will need to adjust the heap size accordingly if you are load balancing across several WebSphere instances.

To set the Java heap size:

- 1 Verify that the WebSphere Admin Server has been started.
- 2 Open the WebSphere Advanced Administrator's Console (**Start > Programs > IBM WebSphere > Application Server > Administrator's Console**).
- 3 Click **Nodes > <System Name> > Application Servers > <Application server name>**.

The server settings page opens.



- 4 Click the JVM Settings tab.
- 5 Set the following JVM settings:
 - a **Initial java heap size.** Type 60.
 - b **Maximum java heap size.** Type the value you want for heap memory. This setting should be at least 225 MB, but not more than 512 MB.

Note: Make sure that the setting for maximum heap size is less than the free RAM available to the application server(s). Exceeding the amount of available RAM causes the JVM processes to swap to disk, reducing overall performance. A setting of 256 MB should be sufficient for most systems.

Configuring a virtual directory for IBM HTTP Server

You must configure a virtual directory for Get-Resources in your Web server. The following instructions assume that you are using WebSphere's built-in Web server – IBM HTTP Server. See your Web server documentation to determine how to create a virtual directory if you are using another Web server.

To configure a virtual directory for IBM HTTP Server:

- 1 Stop the IBM HTTP Server.
- 2 Open the file `httpd.conf` in any text editor. By default this file is located at:
`C:\IBM HTTP Server\conf`
- 3 Add the following line to the end of the file:

```
Alias /oaa/ "C:/WebSphere/AppServer/installedApps/oaa.ear/portal.  
<version>.war/"
```

For `<version>`, enter the version number of the WAR file you installed.

Important: The name you define for the virtual directory here must match the context root you defined in WebSphere.

- 4 Save the file.
- 5 Start the IBM HTTP Server.

Installing WebSphere Portal Server

You can configure Get-Resources to display in a WebSphere Portal Server in one of two configurations:

- All Get-Resources and WebSphere components running on a single system. See *Recommended WebSphere Portal Server configuration* on page 38.
- Get-Resources components running on one system and WebSphere components running on another. See *Alternate WebSphere Portal Server configuration* on page 40.

Important: In either configuration, you must first install WebSphere Portal Server. See your WebSphere Portal Server documentation for details.

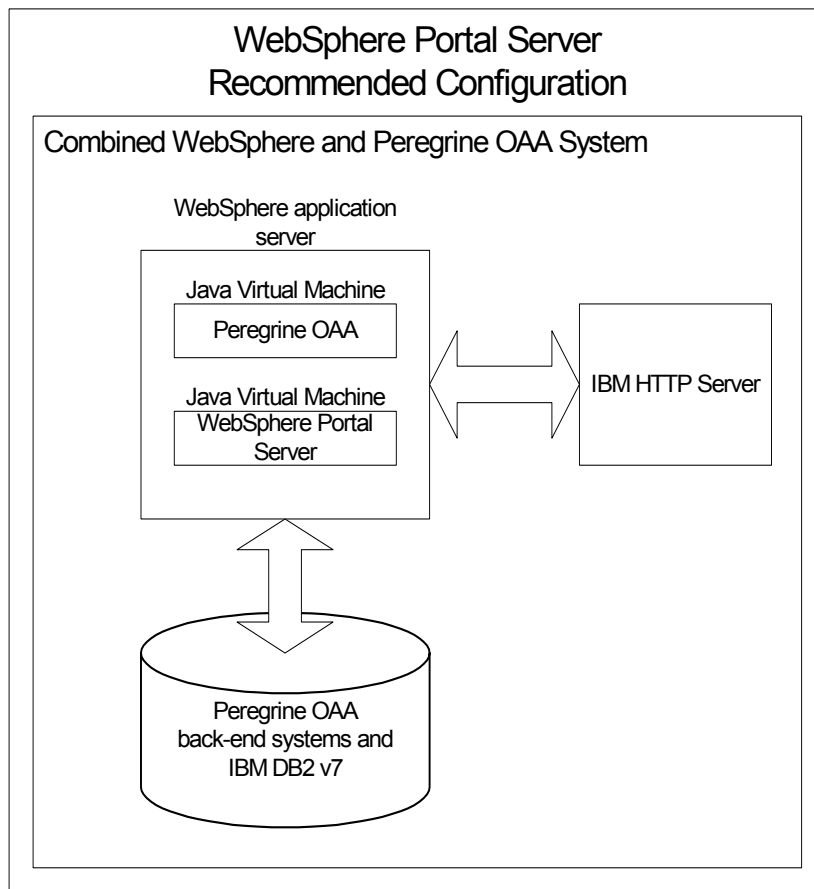
Recommended WebSphere Portal Server configuration

Use the following steps to configure Get-Resources for the recommended WebSphere Portal Server configuration:

- Step 1** Review the WebSphere Portal Server installation requirements. See *WebSphere Portal Server installation requirements* on page 41.
- Step 2** Generate a Get-Resources WAR file containing the portal components WebSphere Portal Server can display. See *Generating a Get-Resources WAR file* on page 42.
- Step 3** Login to the Get-Resources server and stop the WebSphere application server.
- Step 4** Modify the `archway.xml` to change the HTTP authentication method used from Basic to Alternate. See *Modifying the archway.xml file* on page 43.
- Step 5** Modify the `web.xml` to enable the AuthController servlet. See *Modifying the web.xml file* on page 43.
- Step 6** Start the WebSphere application server.
- Step 7** Deploy the Get-Resources WAR file to WebSphere Portal Server. See *Deploying the Get-Resources WAR file to WebSphere Portal Server* on page 44.
- Step 8** Create places and pages in WebSphere Portal Server to display Get-Resources portlets. See *Configuring WebSphere Portal Server places and pages* on page 45.

Step 9 Enable edit rights for Get-Resources portlets. See *Enabling edit rights for Get-Resources portlets* on page 45.

When complete, your installation will have the following configuration:

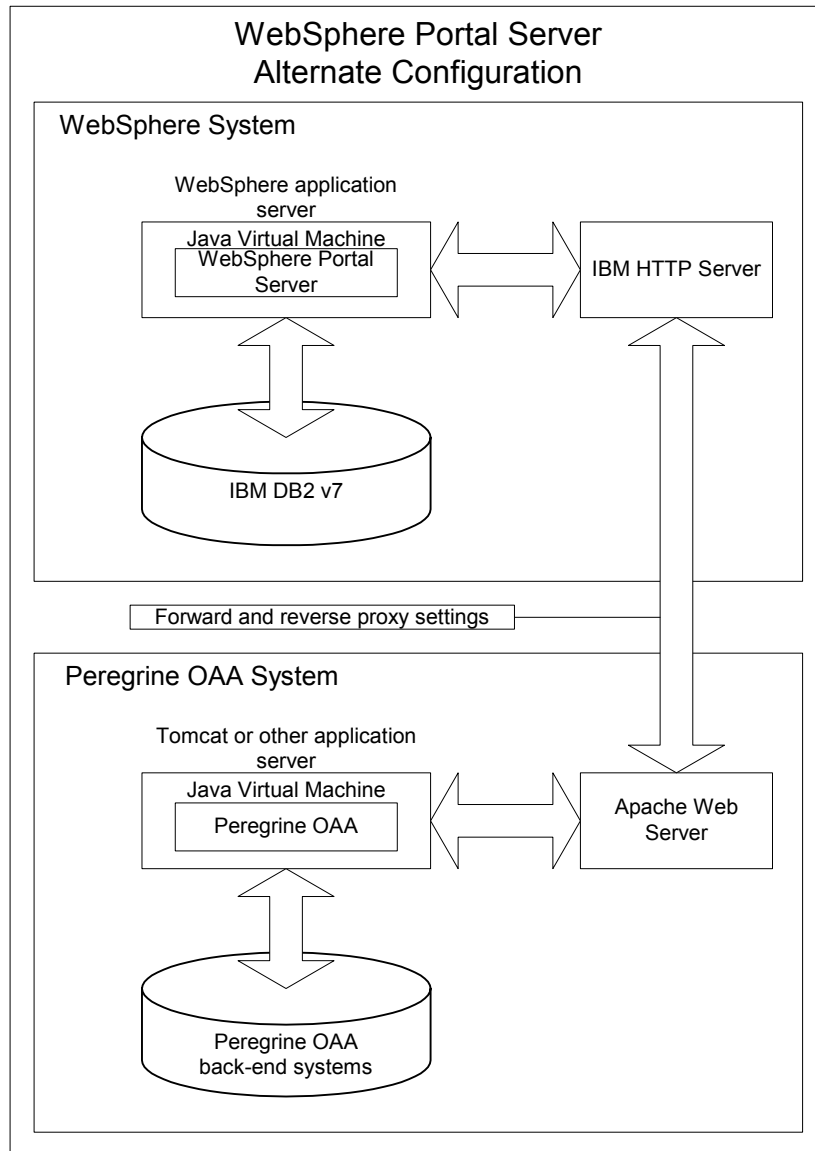


Alternate WebSphere Portal Server configuration

Use the following steps to configure Get-Resources for the alternate WebSphere Portal Server configuration:

- Step 1** Review the WebSphere Portal Server installation requirements. See *WebSphere Portal Server installation requirements* on page 41.
- Step 2** Generate a Get-Resources WAR file containing the portal components WebSphere Portal Server can display. See *Generating a Get-Resources WAR file* on page 42.
- Step 3** Login to the Get-Resources server and stop the WebSphere application server.
- Step 4** Modify `archway.xml` to change the HTTP authentication method used from Basic to Alternate. See *Modifying the archway.xml file* on page 43.
- Step 5** Modify `web.xml` to enable the AuthController servlet. See *Modifying the web.xml file* on page 43.
- Step 6** Modify `setDomain.js` to call the SetDomain function. See *Modifying the setDomain.js file* on page 44.
- Step 7** Start the WebSphere application server.
- Step 8** Deploy the Get-Resources WAR file to WebSphere Portal Server. See *Deploying the Get-Resources WAR file to WebSphere Portal Server* on page 44.
- Step 9** Create places and pages in WebSphere Portal Server to display Get-Resources portlets. See *Configuring WebSphere Portal Server places and pages* on page 45.
- Step 10** Enable edit rights for Get-Resources portlets. See *Enabling edit rights for Get-Resources portlets* on page 45.
- Step 11** Modify IBM HTTP Server's `httpd.conf` file to add forward and reverse proxy URLs. See *Modifying httpd.conf for IBM HTTP Server* on page 46.

When complete, your installation will have the following configuration:



WebSphere Portal Server installation requirements

The recommended configuration of the WebSphere Portal Server requires the following items to be installed on the same server:

- WebSphere application server 4.0.2

- IBM HTTP Server 1.3.19
- IBM DB2 v7 database server
- WebSphere Portal Server
- A custom installation of Get-Resources with WebSphere selected as the application server

The alternate configuration of the WebSphere Portal Server requires the following items be installed on a minimum of two servers:

- Server 1
 - WebSphere application server 4.0.2
 - IBM HTTP Server 1.3.19
 - IBM DB2 v7 database server
 - WebSphere Portal Server
- Server 2
 - Get-Resources compatible application server
 - Web server
 - Back-end database for Get-Resources
 - An installation of Get-Resources

Generating a Get-Resources WAR file

In order to display Get-Resources in WebSphere Portal Server, you must first export the Get-Resources portal components as a WAR file. You can then import this WAR file into WebSphere Portal Server, and choose the portal components you want to display as WebSphere Portal Server portlets.

To generate a Get-Resources WAR file:

- 1 Login to the Get-Resources administration page (`admin.jsp`).
- 2 Click **IBM WebSphere Portal Integration**.
- 3 Enter the following configuration information:
 - a **Source Path**. Enter the full path to the `WebSphere.war` in the Get-Resources **package** folder. By default this folder is:
`<WebSphere>/oaa/packages`
 - b **Destination Path**. Enter the full path and file name you want to use for the generated Get-Resources WAR file.

- c **Base URL.** Enter the full URL to the Get-Resources deployment directory.
By default this URL is:

`http://<server>:<port>/oaa/servlet/basicauth`

- 4 Click **Generate WAR file**.

Get-Resources generates a new WAR file with the name and path specified in the Destination Path of step 3.

Modifying the archway.xml file

In order to login via WebSphere Portal Server, you configure Get-Resources to use an alternate HTTP authentication method.

To modify the archway.xml file:

- 1 Using a text editor, open the archway.xml file located at:

`<application server>\webapps\oaa\WEB-INF\default.`

- 2 Edit the line containing:

```
<httpauthclass ...>HttpBasicAuthenticationManager</httpauthclass>
```

- 3 Change the value `HttpBasicAuthenticationManager` to

`HttpAlternateAuthenticationManager`.

- 4 Save the file.

Modifying the web.xml file

You will need to enable the AuthController servlet to establish a proxy for HTTP basic authentication.

To modify the web.xml file:

- 1 Using a text editor, open the web.xml file located at:

`<application server>\webapps\oaa\WEB-INF.`

- 2 Add the following lines at the end of the last `<servlet>` definition:

```
<servlet>
  <servlet-name>AuthController</servlet-name>
  <display-name>AuthController</display-name>
  <description>A controller (decorator) servlet that can be used to
enable configurable auth protection of any resource.</description>

  <servlet-class>com.peregrine.oaa.archway.AuthControllerServlet
</servlet-class>
  <load-on-startup>2</load-on-startup>
</servlet>
```

```

<servlet-mapping>
  <servlet-name>AuthController</servlet-name>
  <url-pattern>/servlet/basicauth/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
  <servlet-name>AuthController</servlet-name>
  <url-pattern>/servlet/auth/*</url-pattern>
</servlet-mapping>

```

- 3 Save the file.

Modifying the setDomain.js file

To use the alternate configuration of WebSphere Portal Server, you must enable the setDomain function.

Note: If you are setting up WebSphere Portal Server in the recommended configuration, you may skip these instructions.

To modify the setDomain.js file:

- 1 Login to the Get-Resources server.
- 2 Stop your application server.
- 3 Using a text editor, open the setDomain.js file located at:

<application server>\webapps\oaa\js.

- 4 Add the following line to the end of the file:

```
setDomain();
```

- 5 Save the file.

Deploying the Get-Resources WAR file to WebSphere Portal Server

After you deploy the Get-Resources WAR file to WebSphere Portal Server, you can then configure the portlets you want to display, the display settings, and the access rights to each portlet.

See your WebSphere Portal Server documentation for detailed instructions.

To deploy the Get-Resources WAR file:

- 1 Login to the WebSphere Portal as wpsadmin or another user with administrative rights.
- 2 Select **Portal Administration** from the Places menu.

- 3 Click **Portlets > Install Portlets**.
- 4 Click **Browse** and navigate to the Destination path you entered when you created the Get-Resources WAR file.
- 5 Click **Next** to load the Get-Resources WAR file.
WebSphere Portal Server displays a list of portlets to be installed.
- 6 Click **Install**.
WebSphere Portal Server installs the portlets and displays the message “Portlets successfully installed.”

Configuring WebSphere Portal Server places and pages

You can deploy Get-Resources portlets in any place or page that meet the following requirements.

Places

Your WebSphere Portal Server places must have the following characteristics:

- Supported markups must include HTML

Pages

Your WebSphere Portal Server pages must have the following characteristics:

- Supported markups must include HTML
- The page must be set to “allow all portlets that a user can access”
- All Get-Resources portlets that you display in a page must grant “all authenticated users” the minimum edit permission.

Enabling edit rights for Get-Resources portlets

WebSphere Portal Server users will need edit rights to the Get-Resources portlets in order to add and customize them to their portal page.

To enable edit rights for Get-Resources portlets:

- 1 Login to the WebSphere Portal as **wpsadmin** or another user with administrative rights.
- 2 Select **Portal Administration** from the Places menu.
- 3 Click **Security > Access Control List**.
- 4 Select the **Special groups** option and select **All authenticated users** from the select box.

- 5 From the Select the objects for the permissions select box, select **portlet applications**.
 - 6 Select the Search on option, and then enter Peregrine in the Name contains field.
 - 7 Click Go.
- WebSphere Portal Server displays a list of portlets with Peregrine in the name.
- 8 In the Edit column, click **Select All** at the bottom of the table.
 - 9 Click **Save**.

Users can now view and customize Get-Resources portlets from the WebSphere Portal Server interface.

Modifying httpd.conf for IBM HTTP Server

In order to use the alternate configuration of WebSphere Portal Server, you will need to modify the `httpd.conf` file used by the IBM HTTP Server to add the forward and reverse proxy URLs to your remote instance of Get-Resources.

Note: If you are setting up WebSphere Portal Server in the recommended configuration, you may skip these instructions.

To modify httpd.conf for IBM HTTP Server:

- 1 Login to the Get-Resources server.
- 2 Stop your IBM HTTP Server.
- 3 Using a text editor, open the `httpd.conf` file located at:
C:\IBM HTTP Server\conf
- 4 Add the following lines to the end of the file:


```
ProxyPass /<aaa root>/ http://<server>:<port>/
<aaa root>/servlet/basicauth/
ProxyPassReverse /<aaa root>/ http://<server>:<port>/
<aaa root>/servlet/basicauth/
```

For `<aaa root>`, enter the name of the oaa virtual directory used by IBM HTTP Server. By default, this virtual directory is `oaa`.

For `<server>:<port>`, enter the server name and communications port number where Get-Resources is installed.

- 5 Save the file.

Configuring WebSphere Translation Server for Get-Resources

You can configure Get-Resources to use a WebSphere Translation Server to provide real-time translations of on-screen data.

To configure WebSphere Translation Server for Get-Resources:

- Step 1** Copy the file `wt.s.jar` to the Get-Resources deployment folder. See *Copying wt.s.jar to the Get-Resources deployment folder* on page 47.
- Step 2** Configure Get-Resources to use the WebSphere Translation Server. See *Configuring Get-Resources to use the WebSphere Translation Server* on page 47.

Copying wt.s.jar to the Get-Resources deployment folder

The following instructions describe where to find and copy the file `wt.s.jar`.

To copy wt.s.jar to the Get-Resources deployment folder:

- 1 Stop your application server.
- 2 Browse to the location of your WebSphere Translation Server installation.
- 3 Copy the file `wt.s.jar` from this folder.
- 4 Paste the file `wt.s.jar` into the Get-Resources deployment folder located at:
`<Application server install>\WEB-INF\lib`
- 5 Restart your application server.

Configuring Get-Resources to use the WebSphere Translation Server

The following instructions describe how to configure Get-Resources to use the WebSphere Translation Server.

To configure Get-Resources to use the WebSphere Translation Server:

- 1 Login to the Get-Resources admin page (`admin.jsp`).
- 2 Click **Settings** > **Common** tab.

The Admin Settings page opens.

The screenshot shows the Admin Settings page with the following configuration details:

Section	Field	Value	Description
Common Backend	Common Backend	portalDB	Adapter target name used to support common user operations.
	List of target aliases:	webapplication;mail	Specifies a list of semicolon delimited target aliases used by web applications in this package.
	Admin name:	Admin	The admin login name. This login provides access to administrative functionality without authenticating against back-end adapters.
	Admin password:		The admin login password.
Language Translation	Translation Server Factory Class:	com.peregrine.util.WTSLanguageTranslatorFactory	The Java factory class which generates the proper class associated with the Translation Server.
	Language from which to translate:	English	The language from which to translate or the base language in which all text is currently displayed as.
	Translation Server IP Address:	10.3.128.181:1097	The IP address of the Translation Server. This address may or may not contain a port number depending on the Translation Server requirements. Click for default.
	Application path:	WEB-INF/apps/	Directory location of the Peregrine Portal Web Applications.
	Event queue:	portalDB	Enter the name of the adapter that should be used by the Peregrine Portal event queue engine. For example: <ul style="list-style-type: none"> To use ServiceCenter's repository, enter "sc" To use AssetCenter's repository, enter "ac"

- 3 Enter the following configuration settings:
 - a **Translation Server Factory Class:** Enter the Java factory class for the Translation server. The default Java factory class is:
`com.peregrine.util.WTSLanguageTranslatorFactory`
 - b **Language from which to translate:** Enter the source language that you want translated. The default value is English.
 - c **Translation Server IP Address:** Enter the IP address and communications port to the Translation Server. For example: 10.3.128.181:1097.
 - d **Application path:** Enter the relative path to the application server applications directory. The default value is:
`WEB-INF/apps/`
 - e **Event queue:** Enter the adapter name to you want to use for the event queue engine. The default value is: **PortalDB**
- 4 Click **Save**.
The Control Panel opens.
- 5 Click **Reset Server**.

Translating on-screen data with a Translation Server

If you plan to store Get-Resources data in a mixture of languages, you can configure Get-Resources to send data to a Translation Server for real time translation. This interface will only translate data retrieved from the back-end database or manually typed into form inputs. If you need a translated user interface, you can purchase a Get-Resources language pack directly from Peregrine Systems.

To translate on-screen data with a Translation Server:

- 1 Enable the translation server from the **Administration > Settings** page as described in *Configuring Get-Resources to use the WebSphere Translation Server* on page 47.

The translate button appears in the upper right tool bar.

The Translation button.



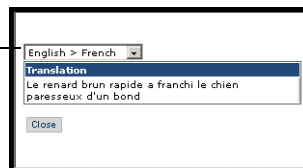
- 2 Click on the source data or form input you want to translate.

Click on the text you want to translate.

- 3 Click the translate button.

The Translation window opens.

Select the target language from the select box.



- 4 Select the target language to which you want to translate from the drop down select box.

The translation of your selection displays in the Translation box.

WebLogic 6.1 SP3 or SP4

The following procedures configure WebLogic to run Get-Resources on Windows.

To configure WebLogic 6.3 SP3 or SP4:

- Step 1** Stop both WebLogic and your Web server
- Step 2** Edit the `startWebLogic.cmd` file to set the system password, memory settings, and start mode. See *Editing startWebLogic.cmd* on page 50.
- Step 3** Edit the `Server.Policy` file to set the debug to true. See *Editing Server.Policy* on page 51.
- Step 4** Run the Get-Resources installer.
- Step 5** Configure IIS to use `iisforward.dll` as an ISAPI filter and create an extension. See *Configuring the iisforward.dll as an ISAPI filter and an extension* on page 51.
- Step 6** Configure IIS to use `iisproxy.dll` as an ISAPI filter and create an extension. See *Configuring the iisproxy.dll as an ISAPI filter and an extension* on page 52.
- Step 7** Create a virtual directory for Get-Resources in your Web server. See *Creating a virtual directory for Get-Resources* on page 53.
- Step 8** Restart WebLogic and your Web server.

Editing startWebLogic.cmd

To edit startWebLogic.cmd:

- 1 Open the file `startWebLogic.cmd` file in any text editor. By default the file is located at:

```
c:\bea\wlserver6.1\config\<mydomain>\
```

- 2 Scroll to the following section of the script:

```
echo *****
echo * To start WebLogic Server, use the password      *
echo * assigned to the system user. The system        *
echo * username and password must also be used to    *
echo * access the WebLogic Server console from a web  *
echo * browser.                                       *
echo *****
@rem Set WLS_PW equal to your system password for no password prompt
server startup.
set WLS_PW=password
```

- 3 In the last line, change the word “password” to your WebLogic system password.
- 4 Search for the -mx parameter setting in the file. Change this setting to at least 225 MB, but not more than 512 MB.

Note: Make sure that the setting for maximum heap size is less than the free RAM available to the application server(s). Exceeding the amount of available RAM causes the JVM processes to swap to disk, reducing overall performance. A setting of 256 MB should be sufficient for most systems.

- 5 Set the STARTMODE variable to STARTMODE=false.

The first time you start WebLogic after the installation, you will need to start it in development mode for it to find the Web applications that have been deployed.

- 6 Add the following line before the end of the “goto finish” entry:

```
“-Djava.security.auth.login.config==<Weblogic>\lib\server.policy”
weblogic.server
```

For <Weblogic>, enter the installation path for Weblogic. By default this is:

c:\bea\wlserver6.1

- 7 Save the file.

Editing Server.Policy

To edit Server.Policy:

- 1 Open the file Server.Policy file in any text editor. By default the file is located at:

c:\bea\wlserver6.1\lib\

- 2 Add the following lines to the end of the file:

```
ServerLoginModule
{
    weblogic.security.internal.ServerLoginModule required debug=true;
};
```

- 3 Save the file.

Configuring the iisforward.dll as an ISAPI filter and an extension

To establish a connection between WebLogic and IIS, you will need to install the file iisforward.dll as an ISAPI filter.

To install issforward.dll as an ISAPI filter and an extension:

- 1 Open Internet Services management console.
- 2 Right-click the **Default Web Site** node and then click **Properties**.
- 3 Click **Edit** from the Master Properties pane.
- 4 Click the **ISAPI Filters** tab.
- 5 Click **Add**.
- 6 Enter the following information:
 - a **Filter Name:** iisforward.
 - b **Executable:** issforward.dll. By default this file is located at:
c:\bea\wlserver6.1\bin\issforward.dll
- 7 Click **OK**.
- 8 Click the **Home Directory** tab.
- 9 Click **Configuration**.
The Application Configuration page opens on the App Mappings tab.
- 10 Click **Add**.
- 11 Enter the following information:
 - a **Executable:** issforward.dll. By default this file is located at:
c:\bea\wlserver6.1\bin\issforward.dll
 - b **Extension:** .wlforward.
- 12 Close the Internet Services management console.

Configuring the iisproxy.dll as an ISAPI filter and an extension

To establish a connection between WebLogic and IIS, you will need to install the file iisproxy.dll as an ISAPI filter.

To install iisproxy.dll as an ISAPI filter and an extension:

- 1 Open Internet Services management console.
- 2 Right-click the **Default Web Site** node and then click **Properties**.
- 3 Click the **ISAPI Filters** tab.
- 4 Click **Add**.
- 5 Enter the following information:
 - a **Filter Name:** iisproxy.
 - b **Executable:** iisproxy.dll. By default this file is located at:

c:\bea\wlserver6.1\bin\iisproxy.dll

- 6 Click OK.
- 7 Click the Home Directory tab.
- 8 Click Configuration.
The Application Configuration page opens on the App Mappings tab.
- 9 Click Add.
- 10 Enter the following information:
 - a Executable: iisproxy.dll. By default this file is located at:
c:\bea\wlserver6.1\bin\iisproxy.dll
 - b Extension: .jsp.
- 11 Close the Internet Services management console.

Creating a virtual directory for Get-Resources

To run Get-Resources, you need to create a virtual directory in your Web server that maps to your WebLogic deployment folder. The typical installation creates a virtual directory called oaa, but you may specify a different virtual directory name.

Requirements for Get-Resources virtual directory

Requirement	Setting
Create virtual directory	<aaa>
Directory access rights	anonymous
Map to physical path	<WebLogic>\applications\aaa
Set Execute Permissions to	Scripts and Executables
Remove “allow anonymous access” to	default.asp, login.asp, e_login_main_start.asp
Set security access to only allow “System” and “Authenticated Users” to	default.asp, login.asp, e_login_main_start.asp

For <aaa>, enter the name of the virtual directory you want to use for Get-Resources. Whatever name you enter here you will need to replicate in your application server configuration.

For `<WebLogic>`, enter the path to your WebLogic installation. By default the path is:

`c:\bea\wlserver6.1\config\<mydomain>`

JRun 3.1

The following procedures configure JRun to run Get-Resources on Windows.

To configure JRun 3.1:

- Step 1** Install a Java run-time environment. The Get-Resources installer includes the Java 2 SDK Standard Edition v1.3.1_05. However, you can also use JRE 1.3.1 if you already have it installed. See *Custom installation components* on page 68.
- Step 2** Install JRun from the Allaire Web site to the root of your hard drive (for example, C:\).
- Step 3** Apply the latest JRun update. See *Applying the latest JRun update* on page 54.
- Step 4** Deploy the Portal WAR file to JRun to create the necessary folder structure for Get-Resources. See *Deploying the Portal WAR file to JRun* on page 55.
- Step 5** Run the Get-Resources installer.
- Step 6** Move `js.jar` to the Java development kit `ext` folder. See *Moving js.jar to the Java development kit* on page 57.
- Step 7** Run the JRun Connector Wizard to establish a connection between JRun and your Web server. See *Running the JRun Connector Wizard* on page 57.
- Step 8** Configure your JRun Java settings. See *Configuring Java settings* on page 58.
- Step 9** Create a virtual directory for Get-Resources in your Web server. See *Creating a virtual directory for Get-Resources* on page 60.
- Step 10** Restart JRun and your Web server.

Applying the latest JRun update

Before you install Get-Resources, you must apply the latest JRun 3.1 update.

To install the latest JRun update:

- 1 Browse to the following URL:
http://www.macromedia.com/support/jrun/updates/3/updates_31.html
- 2 Click the link for the JRun edition (Enterprise, Advanced, or Professional) and operating system of your server.
- 3 Follow the installation instructions provided.

Deploying the Portal WAR file to JRun

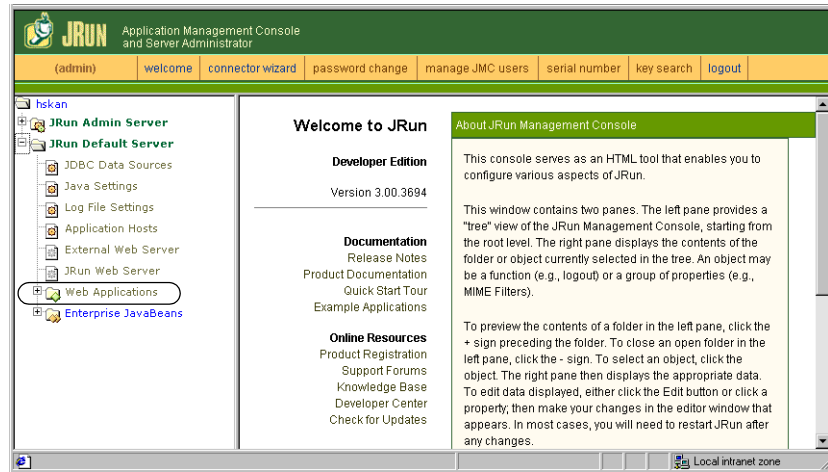
The Portal WAR file creates the folder structure necessary to deploy Get-Resources in your application server. After you have deployed this file to JRun you will be ready to run the Get-Resources installer.

To deploy the Get-Resources Portal WAR file to JRun:

- 1 Open the JRun Management Console and log in.



2 Select JRun Default Server > Web Applications.



The Edit / Create / Deploy and Remove Applications page opens.



3 Click the Deploy an Application link.

4 In the page that opens, fill out the fields as follows:

- Servlet War File or Directory:

Browse to <CD Rom Drive>:\oaa\packages\portal<version #>.war.

For <version>, select the most current version.

Select this file, and then click **Accept**.

- JRun Server Name:

Select **JRun Default Server**.

- Application Name:

Type **oaa**.

- Application URL:

Type **/oaa**.

- Application Deploy Directory:

JRun generates this directory. Make a note of this path. You will need this information later in the procedure. Example:

c:\JRun\servers\default\oaa

5 Click Deploy.

A message that OAA has been successfully deployed appears.

Moving js.jar to the Java development kit

JRun requires an updated version of js.jar in the Java development kit.

To move the js.jar to the Java development kit:

- 1 Stop any running JRun services.
- 2 Locate js.jar. By default this file is installed at:
<JRun installation>\servers\default\oaa\WEB-INF\lib
- 3 Cut and paste the file to your Java Development Kit ext folder. For example:
C:\Program Files\Peregrine\Common\jdk1.3.1_05\jre\lib\ext
- 4 Restart JRun.

Running the JRun Connector Wizard

The JRun Connector Wizard establishes a connection between JRun and your Web server.

To run the JRun Connector Wizard:

- 1 Login to the JRun Management Console.
- 2 Click Connector Wizard.
- 3 Select the JRun Default Server as the JRun Server Name.

- 4 Select your Web server from the drop down list box.
- 5 If your Web server uses a different IP address than your JRun server, enter the IP address of your JRun server in JRun Server IP Address.
- 6 Confirm that the JRun Server Connector Port is not in conflict with another communications port used on this server.
- 7 Enter the path to the Scripts Directory. For IIS 5.0 this value is:
C:\Inetpub\Scripts
- 8 Click Done.

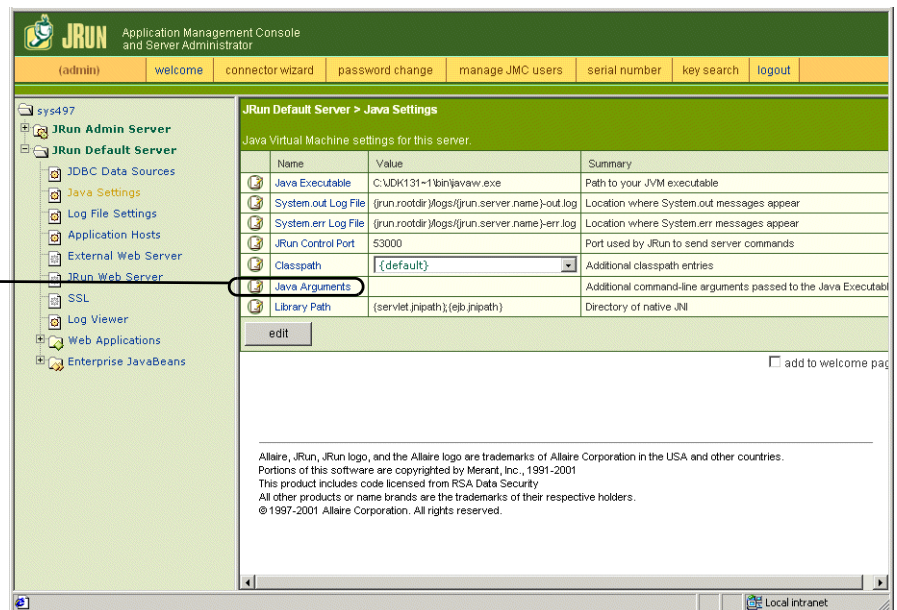
Configuring Java settings

After you have installed Get-Resources, you must configure the Java settings that JRun will use to run the Web application.

To configure Java settings:

- 1 Login to the JRun Management Console.
- 2 Click JRun Default Server > Java Settings.
The Java Settings page opens.
- 3 Click Java Arguments.

Click **Java Arguments**.



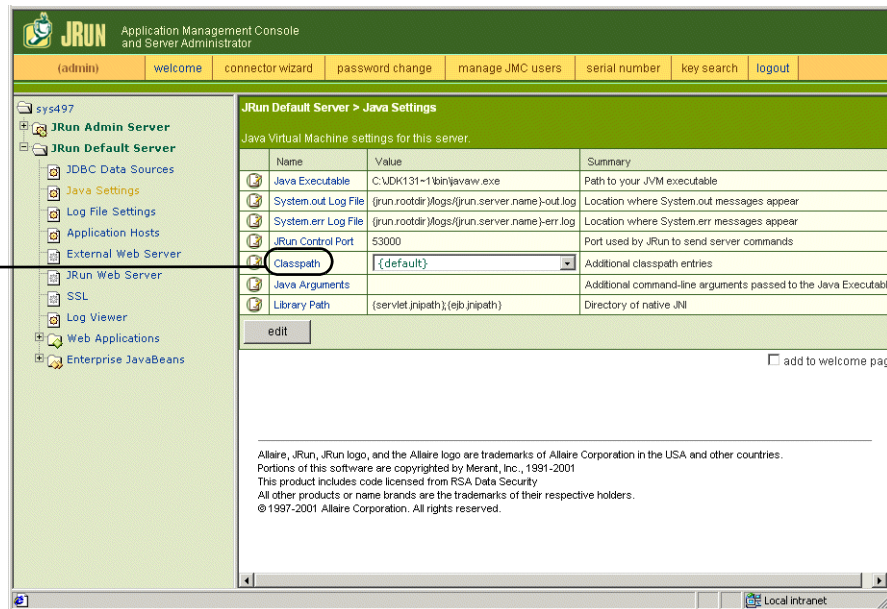
The Edit Window opens.

- 4 Enter an -Xmx value to define the maximum amount of heap memory allocated for your system. It is recommended that you set this value to at least 225 MB, but not more than 512 MB.

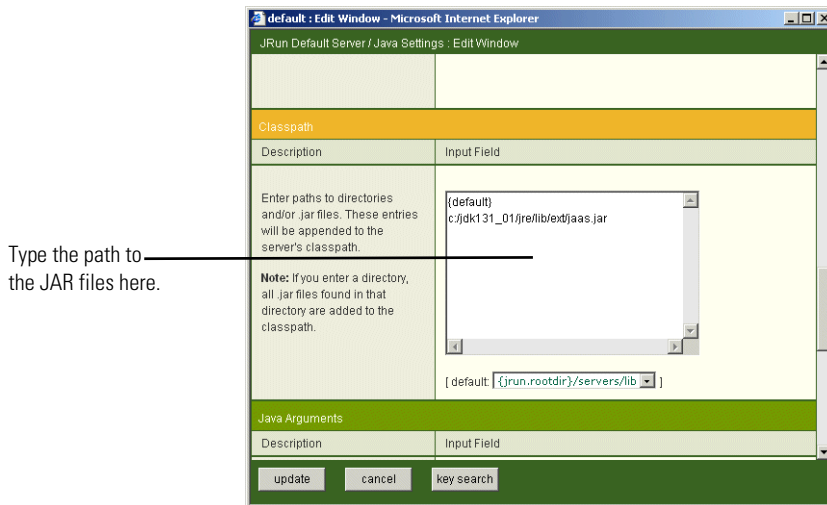
Note: Make sure that the setting for maximum heap size is less than the free RAM available to the application server(s). Exceeding the amount of available RAM causes the JVM processes to swap to disk, reducing overall performance. A setting of 256 MB should be sufficient for most systems. Applications using Persistence may require a higher setting.

- 5 On the Java Settings page, click **Classpath**.

Click
ClassPath.



The Edit Window opens.



- 6 Enter the following classpaths:
 - Java Development Kit ext folder. For example:
C:\Program Files\Peregrine\Common\jdk1.3.1_05\jre\lib\ext
- 7 On the Java Settings page, click **Java Executable**.
- 8 Verify that path to your Java Development kit matches the path listed in the Classpath setting. For example:
C:\Program Files\Peregrine\Common\jdk1.3.1_05\bin\javaw.exe
- 9 Click **update**.
- 10 Log out of the JRun Management Console.

Creating a virtual directory for Get-Resources

To run Get-Resources, you need to create a virtual directory in your Web server that maps to your JRun deployment folder. The typical installation creates a virtual directory called **oaa**, but you may specify a different virtual directory name.

Requirements for Get-Resources virtual directory

Requirement	Setting
Create virtual directory	<aaa>
Directory access rights	anonymous
Map to physical path	<JRun installation>\aaa
Set Execute Permissions to	Scripts and Executables
Remove “allow anonymous access” to	default.asp, login.asp, e_login_main_start.asp
Set security access to only allow “System” and “Authenticated Users” to	default.asp, login.asp, e_login_main_start.asp

For <aaa>, enter the name of the virtual directory you want to use for Get-Resources. Whatever name you enter here you will need to replicate in your application server configuration.

For <JRun>, enter the path to your JRun installation. The recommended installation path is:

C:\JRun\servers\default

Typical installation option

A typical installation of Get-Resources installs the most commonly used components of the product and saves application files and data in default destination directories. Most users choose Typical installation.

Typical installation components

Following is a brief description of the components that are automatically installed with a Typical installation of Get-Resources:

Applications and File Locations

Get-Resources Component	Default Installation Directory
Apache Web Server	C:\Program Files\Peregrine\Common\Apache2
Tomcat Application Server	C:\Program Files\Peregrine\Common\Tomcat4
Java Development Kit	C:\Program Files\Peregrine\Common\jdk1.3.1_05
OAA Platform and Get-Resources	C:\Program Files\Peregrine\oaa

Services

The installation program will also create and start the following services on your Windows server:

- Apache Web Service
- Peregrine Tomcat Service

Important: If you are already running another web or application server on the target Windows system, you will need to stop the service(s) for the application(s) before beginning the Get-Resources setup program. If you do not stop these services before beginning the install process, the setup program will complete successfully, however, it may not create the Apache and Peregrine Tomcat services needed to run Get-Resources.

Communications ports

Get-Resources uses the following communications ports in a typical installation. After installation, you can configure Get-Resources to use one or more of the alternate communications ports if your local network already uses these communications ports.

Default port	Component used by	Alternate port
80	Apache Web Server	8081
8005	Tomcat application server administration	8015
8009	Tomcat application server worker file	8019
8011	Tomcat application server worker file for load balancing (optional).	8021
8013	Tomcat application server worker file for load balancing (optional).	8023
8015	Tomcat application server worker file for load balancing (optional).	8025

Note: To change settings for these components or to use or install different components, use the Custom installation option for Get-Resources.

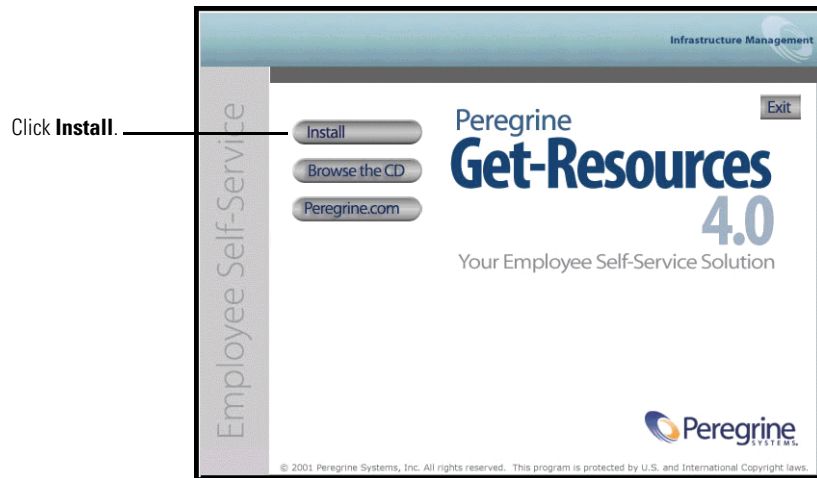
Typical installation procedures

This section explains how to install Get-Resources with a Tomcat application server and an Apache web server on a Windows operating system.

To perform a typical installation of Get-Resources on Windows:

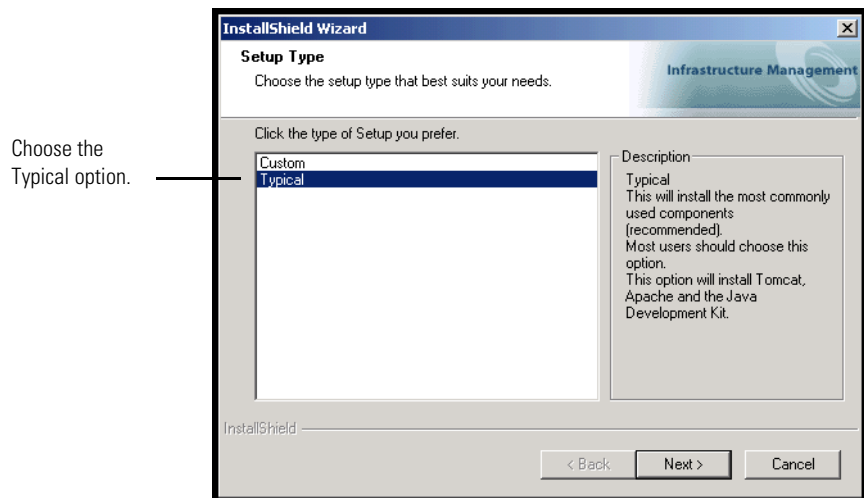
- 1 Insert the Get-Resources installation CD into your computer's CD ROM drive. Your computer should automatically launch the installation program. If the installation program does not automatically start, using the Windows **Start > Run** command, browse to the CD ROM drive and open **Setup.exe**.

- 2 In the main installation program screen, click **Install**.



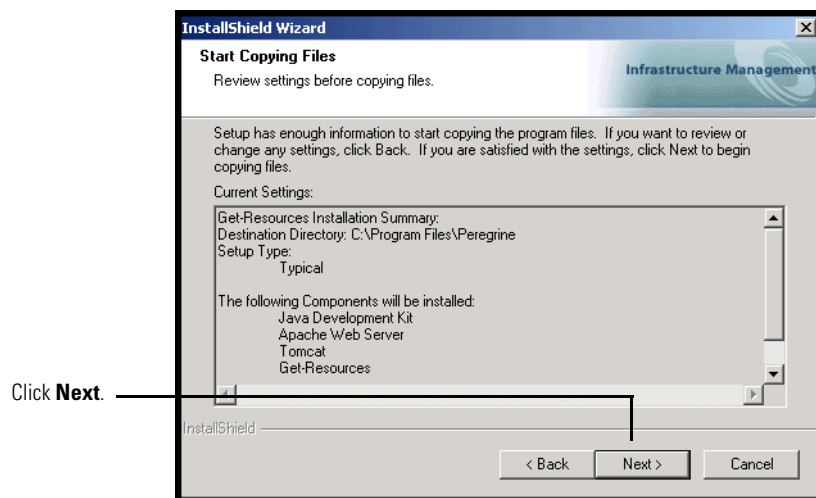
Status messages indicate that the Setup program is preparing the InstallShield Wizard.

- 3 In the Setup Type screen, select **Typical**, and then click **Next**.

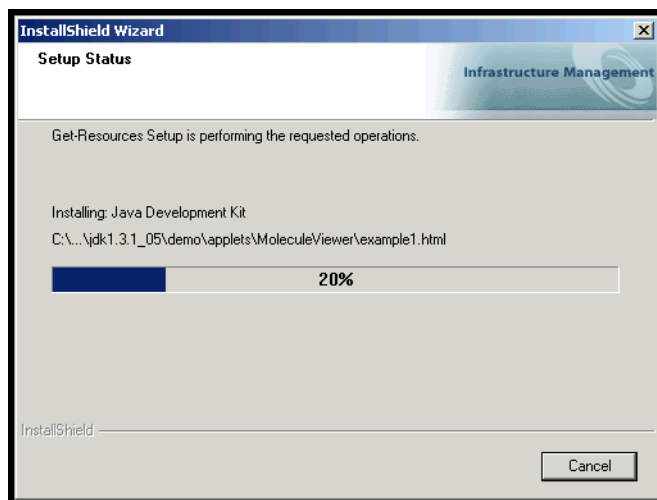


The installer displays status messages to validate the location of the Host system name.

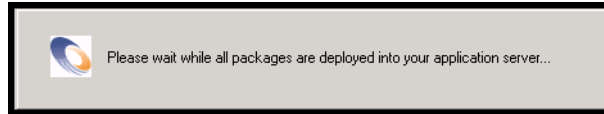
- 4 In the Start Copying Files screen, click Next.



The Setup Status screen that appears shows that Get-Resources is installing the files to your system.



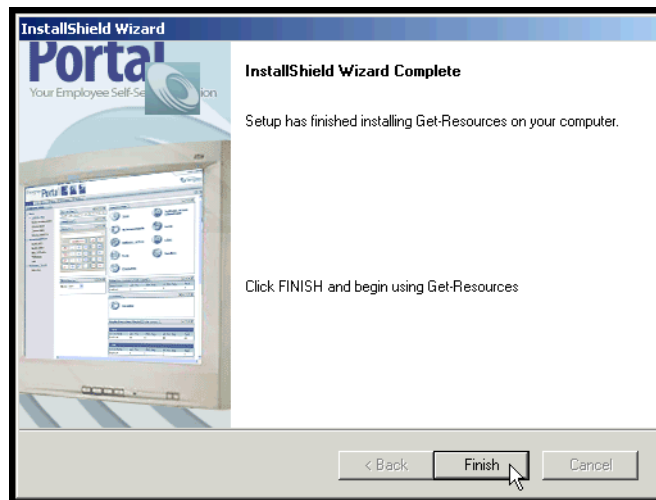
A status message indicates that the Get-Resources packages are being deployed to your server. This phase can take several minutes.



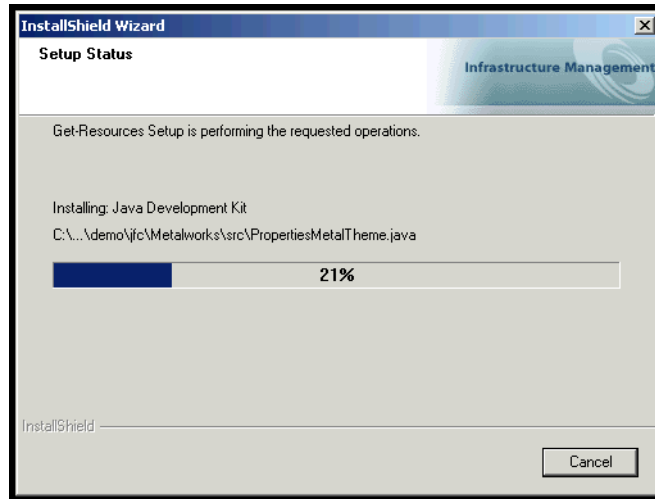
While this status message is displayed, the setup program is performing the following:

- Copying the documents
- Creating and starting the “Apache” and “PeregrineTomcat” services

5 In the InstallShield Wizard Complete screen, click **Finish**.



The setup program completes the final steps of the installation.



- 6 After the InstallShield Wizard screen closes, access Windows Services to verify that the “Apache” and “Peregrine Tomcat” services were created. Status for both services should display “Started”.
If one or both of these services did not startup, refer to [Troubleshooting](#) to troubleshoot possible install issues.
This completes the procedures required for a Typical installation of Get-Resources on a Windows operating system server.

Custom installation option

The following section describes how to perform a custom installation of Get-Resources on a Windows operating system server, including overview steps for a Development and Production environment.

Custom installation components

Following is a brief description of the components that are available for a custom installation of Get-Resources:

Application options

Get-Resources Component	Options
Web Server	<ul style="list-style-type: none"> ■ Apache 2.0.43 ■ IBM HTTP Server 1.3.19 ■ Microsoft IIS 5.0
Application Server	<ul style="list-style-type: none"> ■ Tomcat 4.1.12 ■ WebSphere 4.02 ■ WebLogic 6.1 SP3 ■ JRun 3.1
Java Development Kit	Java 2 SDK
OAA Platform and Get-Resources	None

Communications ports

The communications ports used by a custom installation of Get-Resources depend upon the application components that you select. Refer to your Web and application server documentation to determine what communications port they require. After installation, you can configure Get-Resources to use alternate communications ports if your local network already uses particular communications ports.

Get-Resources on servers running Oracle 9.2.0.1

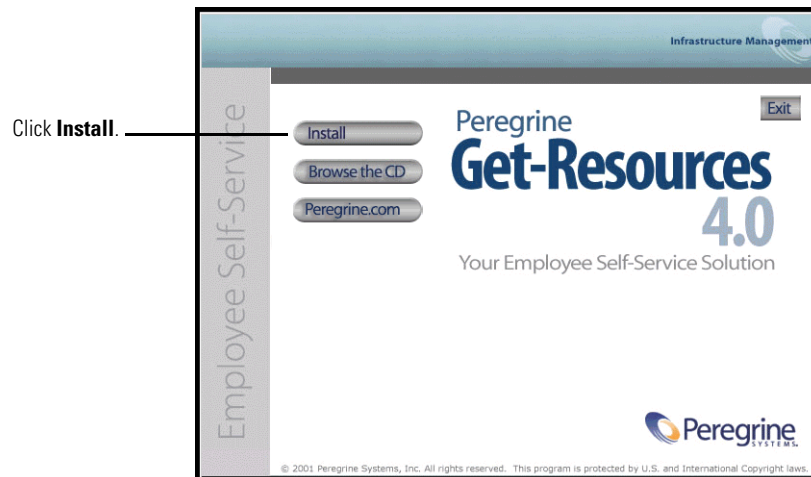
If you are running Get-Resources on a server using Oracle 9.2.0.1 you may experience a port conflict over communications ports 8009 and 8080. Consult your Web and application server documentation to see if they use either of these two ports.

If you are using Tomcat as your application server, then by default, there will be a port conflict over port 8009. It is recommended that you change Tomcat to use a different communications port on servers running Oracle 9.2.0.1.

Custom installation procedures

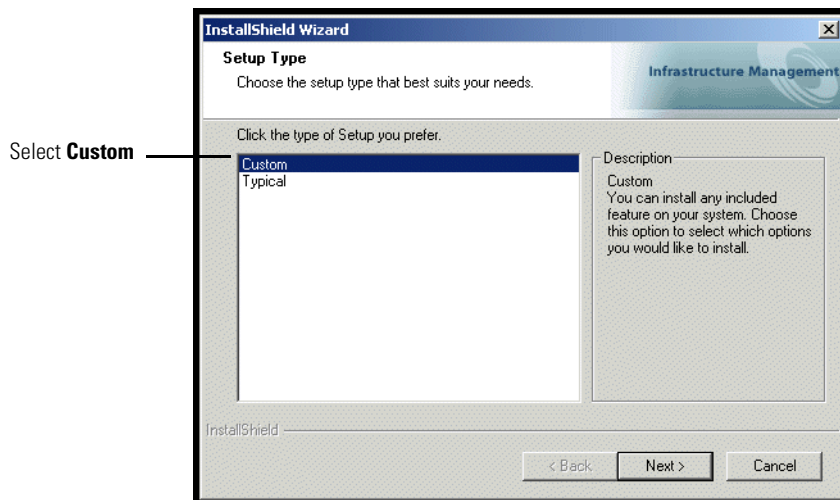
To perform a custom installation of Get-Resources on Windows:

- 1 Insert the Get-Resources installation CD into your computer's CD ROM drive. Your computer should automatically launch the installation program.
If the installation program does not automatically start, using the Windows **Start > Run** command, browse to the CD ROM drive and open **Setup.exe**.
- 2 In the main installation program screen, click **Install**.



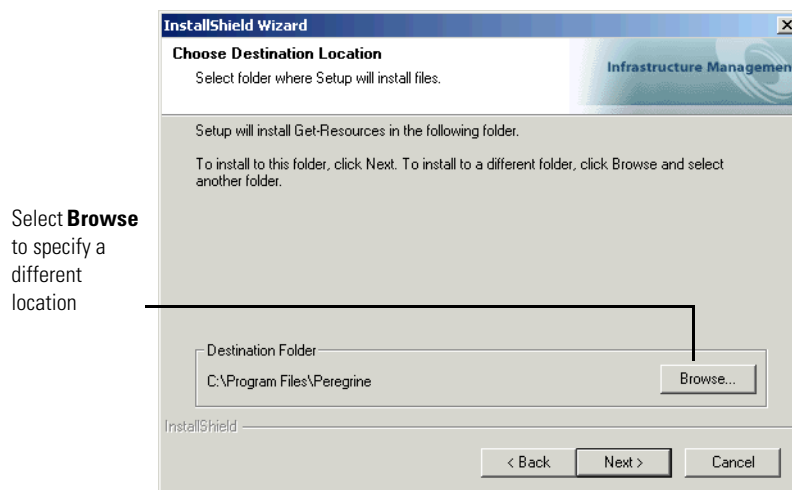
Status messages indicate that the Setup program is preparing the InstallShield Wizard.

- 3 In the Setup Type screen, select **Custom**, and then click Next.

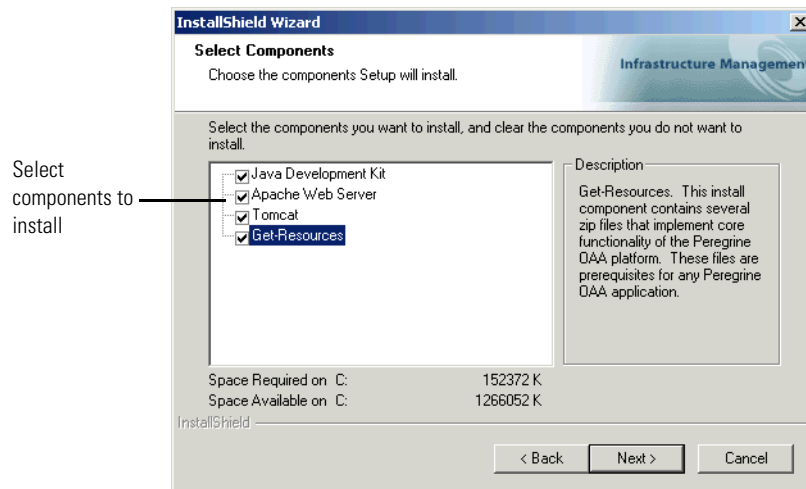


- 4 In the Choose Destination Location screen, browse to the location where you want Get-Resources installed. Click Next to continue.

The default location is C:\Program Files\Peregrine.



- 5 In the Select Components screen, choose the components that you want installed on this machine and click **Next**.



For a development environment, either select all components or clear the components that you will install manually or for which you have alternate software. For example, clear the Tomcat option to install an alternate application server.

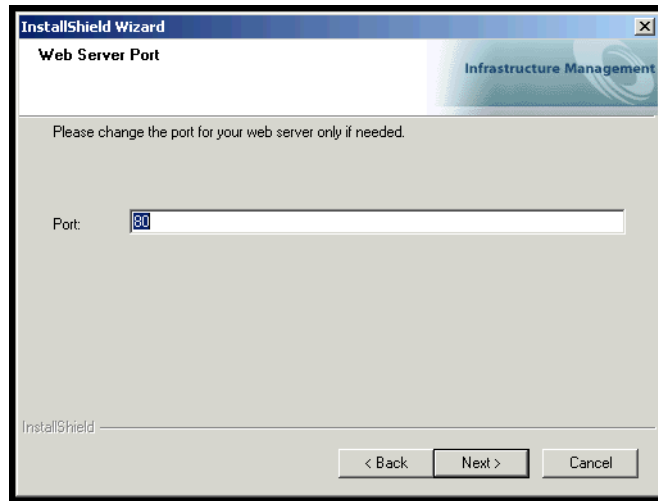
For a production environment, select the components you want to run from this machine.

- **Java Development Kit.** Installs the Sun Microsystems Java Development Kit (Java 2 SDK). This component needs to be installed on every machine running an application server.
- **Apache Web Server.** Installs the Apache Web server. Get-Resources requires a Web server in order to serve pages. See [Custom installation components](#) on page 68 for a list of available Web servers. If you also select the Tomcat application server, then the installer will automatically configure a connection between Apache and Tomcat.
- **Tomcat.** Installs the Tomcat application server. Get-Resources requires at least one application server to process Java applications. You may also install multiple instances of your application server for load balancing. See [Custom installation components](#) on page 68 for a list of available application servers. If you also select the Apache Web server, then the installer will automatically configure a connection between Apache and Tomcat.

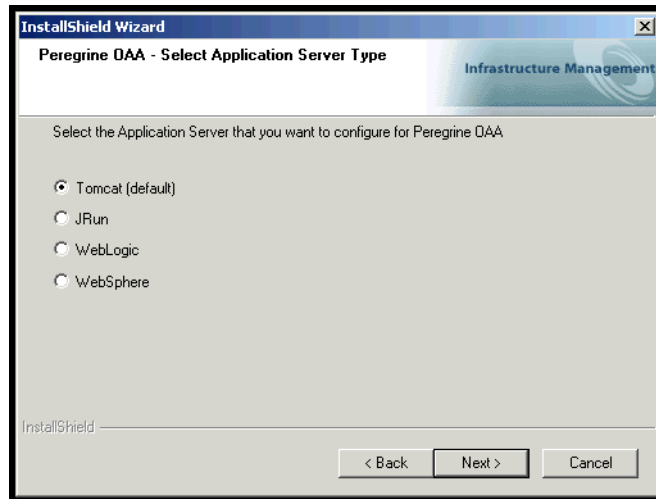
- **Get-Resources.** Installs the programs files necessary for Get-Resources. These files need to be on the same machine as the primary application server.

Status messages indicate the validation and location of the Host system name.

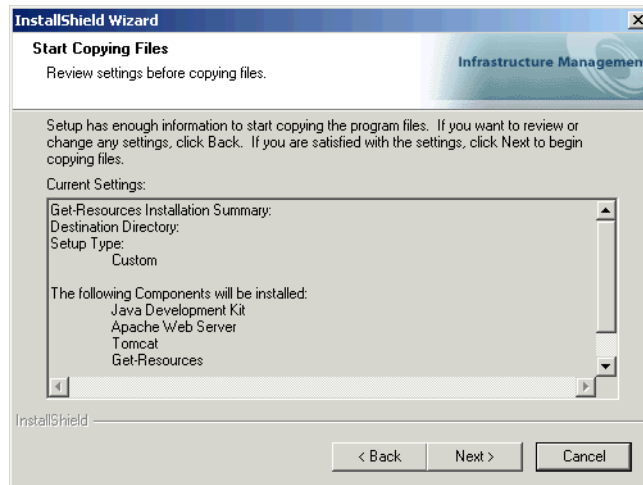
- 6 In the Web Server Port screen, select the communication port you want to use for Web server. The default communications port number for Get-Resources is port 80. To continue, click **Next**.



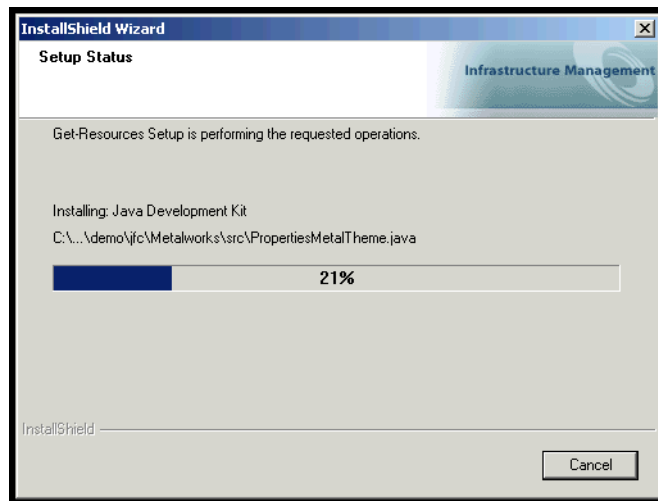
- 7 In the Application Server Type screen, select the application server that you would like to use with Get-Resources. The default application server is Tomcat. To continue, click **Next**.



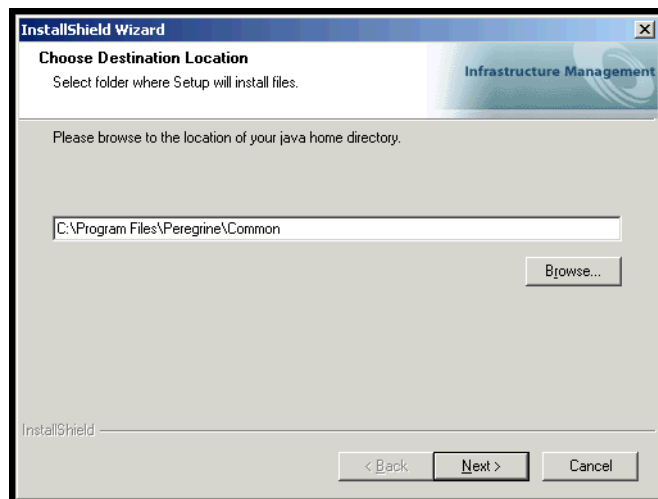
- 8 In the Start Copying Files screen, verify the Custom installation components. To review or change settings, click **Back**. To continue with the installation, click **Next**.



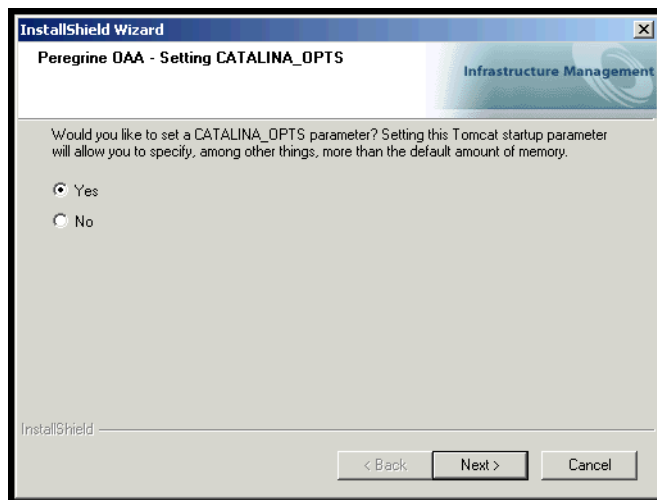
The Setup Status screen is displayed while the installation program performs the requested operations.



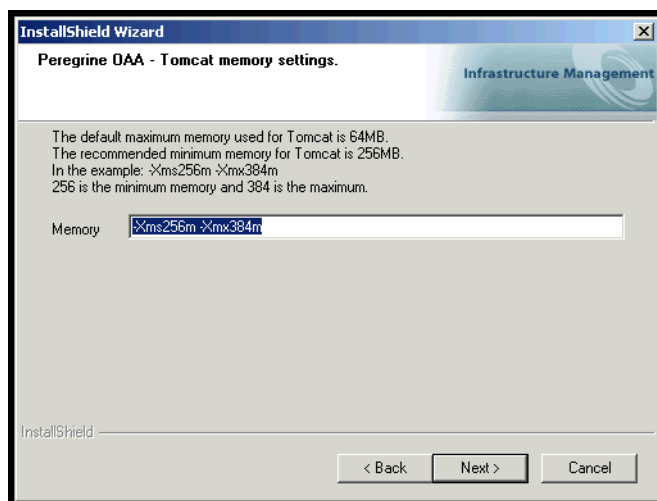
- 9 In the Choose Destination Location screen, browse to the location where you would like the Java application files installed. To continue, click Next.



- 10 In the Setting CATALINA_OPTS screen, select whether or not you would like to configure this parameter for Tomcat memory settings.

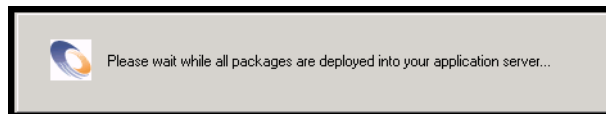


- To configure this parameter, select Yes, then click Next. The installation program will display a Tomcat Memory Settings screen. Modify the minimum and maximum memory range used for Tomcat as necessary, then click Next to continue with the installation.



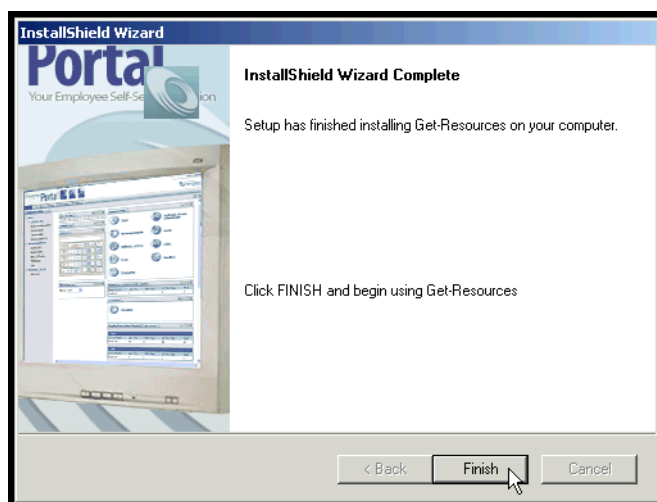
- To bypass configuration of this parameter, select No, then click Next.

A status message indicates that the Get-Resources packages are being deployed to your server.



This phase can take several minutes.

- 11 In the InstallShield Wizard Complete screen, click **Finish** to end the Get-Resources installation program.



- 12 After the InstallShield Wizard screen closes, access Windows Services to verify startup of your web and application server services.
If one or both of these services did not startup, refer to [Troubleshooting](#) to troubleshoot possible installation issues.
This completes the procedures required for a Custom installation of Get-Resources on a Windows operating system server.

Uninstalling Get-Resources

Follow these procedures to uninstall Get-Resources from your Windows system:

- 1 Access the Windows Add/Remove Programs utility.
- 2 Select **Peregrine Portal 4.0** and click **Change/Remove**.
A status message indicates that the setup program is preparing the InstallShield wizard to guide you through the process.
- 3 The Close Programs screen opens if any Get-Resources services or applications are running. Click **Next** to continue.
- 4 The verification message box opens, “Are you sure you want to stop shared applications and/or services?” Click **Yes** to continue.
Status messages indicate the termination of the services for Apache and Tomcat.
- 5 The Confirm Uninstall dialog box opens, “Warning, you are about to uninstall all Peregrine Portal 4.0 applications and data.” Click **Yes** to remove Get-Resources.

Important: Back up any data you want to save before continuing.

- 6 The Shared Files screen opens if there are any shared files to be removed during setup.
If WebSphere is installed on this computer, setup will prompt you confirm the removal of six JAR files. Click **No** or **No to All** to retain these JAR files.

Warning: Do not remove the shared JAR files as the WebSphere Advanced Administrative Console requires these files to function.

If there are no shared files to remove, then a status message indicates that the uninstall program is removing files from your computer.

- 7 The Maintenance Complete screen opens. Click **Finish** to complete the uninstall of Get-Resources.

Testing your installation

Use the following steps to confirm that you have properly installed Get-Resources on Windows.

To test your Get-Resources installation:

- 1 Verify that your application and Web servers are started.
- 2 Open a Web browser and type the following in the Address field:

`http://<server name>:<port>/oaa/admin.jsp`

For *<server name>*, enter the server name where the Get-Resources Web server resides.

For *<port>*, enter one of the following communications port numbers:

Application Server used	Port Number
WebSphere	9080
WebLogic	7001
JRun	80, can be omitted from URL
Tomcat	80, can be omitted from URL

If everything is configured properly, the Administrator login page opens.

If the Get-Resources administration login page does not open, see [Troubleshooting](#) for more information.

3 Installation for Unix Platforms

CHAPTER

This chapter covers the following topics:

- *Choosing an installation environment* on page 80
- *Migrating Get-Resources from previous versions* on page 83
- *Configuring alternate application servers* on page 87
- *Typical Installation Option* on page 118
- *Custom Installation Option* on page 126
- *Uninstall—AIX, Linux, or Solaris* on page 138

Choosing an installation environment

You can install Get-Resources in one of two installation environments:

- Development environment
- Production environment

The Get-Resources development environment is intended for you to evaluate product features and customize your installation prior to deployment in a production environment. In a development environment, you install all software required for Get-Resources on one computer system.

You have two choices of development environment:

- Typical installation
 - Apache 2.0 Web server
 - Get-Resources deployed on Tomcat 4.1.12 application server
- Custom installation
 - Choice of Web server
 - Choice of application server where you deploy Get-Resources

The Get-Resources production environment is intended to maximize server performance and scalability, and to deploy any customizations you have made. In a production environment, you install the various components of Get-Resources on different servers to maximize performance.

You have two choices of production environment:

- Typical installation
 - Apache 2.0 Web server
 - Get-Resources deployed on multiple instances of Tomcat 4.1.12 application server
- Custom installation
 - Choice of Web server
 - Choice of application server where you deploy Get-Resources

Development Environment

The following procedures describe how to install Get-Resources in a development environment.

To install Get-Resources in a typical development environment:

- Step 1** Acquire all necessary hardware and software.
- Step 2** Install the back-end database required for Get-Resources.
- Step 3** Run the Get-Resources installer and select the Typical installation option.
See *Typical Installation Option* on page 118.
- Step 4** Configure the back-end databases and create Get-Resources users.

To install Get-Resources in a custom development environment:

- Step 1** Acquire all necessary hardware and software.
- Step 2** Install the back-end database required for Get-Resources.
- Step 3** Install alternate application and Web servers.
- Step 4** Configure the alternate application server for Get-Resources. See *Configuring alternate application servers* on page 87.
- Step 5** Run the Get-Resources installer and select the Custom installation option.
See *Custom Installation Option* on page 126.
- Step 6** Configure the back-end databases and create Get-Resources users.

Production Environment

The following procedures describe how to install Get-Resources in a production environment.

To install Get-Resources in a typical production environment:

- Step 1** Acquire all necessary hardware and software.
- Step 2** Install the back-end database required for Get-Resources on a separate server.
- Step 3** Run the Get-Resources installer and select the Typical installation option. See *Typical Installation Option* on page 118.
- Step 4** Configure multiple instances of Tomcat for load balancing on the Apache Web server.
- Step 5** Configure the back-end databases and create Get-Resources users.

To install Get-Resources in a custom development environment:

- Step 1** Acquire all necessary hardware and software.
- Step 2** Install the back-end database required for Get-Resources.
- Step 3** Install the alternate application server and Web server on separate servers.
- Step 4** Configure the alternate application server for Get-Resources. See *Configuring alternate application servers* on page 87.
- Step 5** Run the Get-Resources installer and select the Custom installation option. See *Custom Installation Option* on page 126.
- Step 6** Configure the Web servers and application servers for load balancing.
- Step 7** Configure the back-end databases and create Get-Resources users.

Migrating Get-Resources from previous versions

To migrate older versions of Get-It or Get-Resources to Get-Resources 4.0 requires both a manual data migration process and the recreation of any interface customizations you have made. The following steps describe the migration process.

To migrate previous versions to Get-Resources 4.0:

- Step 1** Review the customizations of previous version and determine which customizations need to be recreated in Get-Resources 4.0. See *Recreating customizations in Get-Resources 4.0* on page 83.
- Step 2** Install Get-Resources 4.0 on a new system. See *Choosing an installation environment* on page 80.
- Step 3** Apply any required configuration changes to the back-end database you want to migrate to Get-Resources 4.0. See *Configuring an existing back-end database for Get-Resources 4.0* on page 86.

Recreating customizations in Get-Resources 4.0

You cannot directly migrate customizations implemented in previous versions to Get-Resources 4.0. Instead, you must recreate your changes using the new features and methods available in Get-Resources 4.0.

The following sections describe how to recreate your customizations from previous versions.

No customizations

If you have made no customizations to Get-Resources, you can simply install Get-Resources 4.0 on a new system and migrate your data from your existing back-end database.

Customized JSP files

In previous versions, customers had to directly modify JSP files in order to add or remove certain functionality. The following table describes how to recreate some of the more common JSP file modifications.

JSP file modification	New method to use
Remove the user self-registration option from login page	Enable or disable the user registration option from the Administration Settings page
Remove the change password option from the login page	Enable or disable the change password option from the Administration Settings page

Personalized pages

Get-Resources 4.0 offers many more pages that you can personalize directly from the Web interface. If you personalized pages in a previous version, you must recreate your personalized pages in Get-Resources 4.0 using DocExplorer.

You can use personalization to:

- Add or remove fields from a page
- Save a personalized search results or details on your portal page

Customized skins, stylesheets, and themes

Get-Resources 4.0 has combined all interface images and stylesheets into themes. Users can no longer select separate skins and stylesheets. The new themes consist of skins (which themselves are composed of image files, frame definitions, and layer files), cascading stylesheet definitions, and XSL templates.

Although you may copy over older custom themes to Get-Resources 4.0, you may experience rendering errors due to the new images, CSS definitions, frame definitions, and layers. It is recommended that you recreate any custom themes using the Get-Resources 4.0 version of the classic theme as your template.

Alternate login pages and authentication methods

If you used a custom login page or an alternate authentication method in a previous version, you can re-use or recreate these customizations using the updated instruction. You can find information about alternate security methods in the *Get-Resources Administration Guide*.

Customizations made with a previous tailoring kit

Many customizations that required a tailoring kit in previous versions can now be done directly from the Get-Resources Web interface. The following table describes how to recreate some of the more common tailoring kit changes.

Tailoring kit modification	New method to use
Added or removed fields form a form	Add or remove fields from Personalization
Added a new language or locale to the Get-Resources interface	Create and edit language strings files directly. You may also purchase officially supported language packs from Peregrine Systems
Made changes to the common, portal, or Peregrine Studio packages	These packages are no longer available for tailoring, however most common interface settings can now be customized from the Administration Settings page.
Made changes to schemas or ECMA server-side scripts	Review new functionality and determine if you still need the customized scripts and schemas. If you do need the customizations, you will to recreate them in the current version of the Get-Resources tailoring kit.

Configuring an existing back-end database for Get-Resources 4.0

The following table lists the options available for data migration.

Get-Resources 2.5 to Get-Resources 4.0

Back-end version	Migration required
AssetCenter 3.6	Apply Get-Resources 4.0 scripts to existing AssetCenter 3.6
AssetCenter 4.x	Apply Get-Resources 4.0 scripts to existing AssetCenter 4.x
ServiceCenter 3.0	Upgrade to ServiceCenter 4.x or 5.0.x
ServiceCenter 4.x	Apply Get-Resources 4.0 unloads to existing ServiceCenter 4.x
ServiceCenter 5.0.x	Apply Get-Resources 4.0 unload files to existing ServiceCenter 5.0.x

Configuring alternate application servers

You must install a Java-enabled application server to support your Peregrine Web applications. Peregrine OAA supports the following alternate application servers:

- *Tomcat 4.1.12*
- *WebSphere 4.0.2*
- *WebLogic 6.1 SP3 or SP4*
- *JRun 3.1*

The Get-Resources typical installation option installs Tomcat 4.1.12 and connects it to an Apache 2.0 web server. You can also install Tomcat 4.1.12 using the custom installation option.

Important: If you want to use an application server other than Tomcat 4.1.12, then you must configure your application and Web servers *prior* to running the Get-Resources installer.

See the following sections for instructions configuring alternate application servers for Get-Resources.

Tomcat 4.1.12

If you use select both the Apache Web server and the Tomcat application server for your installation of Get-Resources, then the Get-Resources installer will automatically configure a connection between Apache and Tomcat. In addition, if you select a typical installation, then the installer will automatically pre-compile the Get-Resources JSP files. If you are running a custom installation, you can manually pre-compile the Get-Resources JSP files when you are ready to deploy into a production environment.

Precompiling JSP files for a Tomcat production environment

If you plan to use Tomcat 4.1.12 in a production environment or on a system with multiple processors, you will need to pre-compile the JSP files deployed for Get-Resources. This extra configuration step is the result of a known issue with Tomcat. See the following URL for more details about this issue:

http://nagoya.apache.org/bugzilla/show_bug.cgi?id=14077

To pre-compile JSP files for Tomcat 4.1.12:

- 1 Stop the Tomcat application server.
- 2 Open a command prompt.
- 3 Change directories to `usr/local/peregrine/samples`.
- 4 Copy the following files to the locations specified:

Copy this file	to this location
<code>excludes.txt</code>	<code><Tomcat>/webapps</code>
<code>builds.xml</code>	<code><Tomcat>/webapps</code>
<code>precompile.sh</code>	<code><Tomcat>/bin</code>

For `<Tomcat>`, enter the path to your Tomcat installation. For example:
`/usr/local/peregrine/common/Tomcat4`

- 5 Enter one of the following commands based upon your operating system:

Operating system	Command required
UNIX ksh	<code>export JASPER_HOME=<Tomcat directory></code>
UNIX csh	<code>setenv JASPER_HOME <Tomcat directory></code>

For `<Tomcat directory>` enter the absolute path to your Tomcat installation.

- 6 Change directories to the Tomcat `bin` folder.

Operating system	Command required
UNIX, Solaris, AIX	<code>cd \$JASPER_HOME/bin/samples/</code>

- 7 Run the `precompile` batch file.

Operating system	Command required
UNIX, Solaris, AIX	<code>./precompile.sh <Web app name> <Tomcat instance name></code>

For `<Web app name>` enter the name of the Get-Resources deployment folder. You may omit this name if you are using the default folder named `oaa`.

For *<Tomcat instance name>* enter the name of the Tomcat instance you have installed. You may omit this name if you are using the default instance named **Standalone**.

The batch file displays the progress of the conversion. When it is complete, the command prompt returns.

- 8 Start the Tomcat application server.

WebSphere 4.0.2

Use the following procedures to configure WebSphere to run Get-Resources on AIX, Linux, and Solaris.

To configure WebSphere 4.02:

- Step 1** Install WebSphere 4.02. Your version of WebSphere 4.0.2 includes the IBM HTTP Server.
- Step 2** Deploy the Portal WAR file to WebSphere to create the necessary folder structure for Get-Resources. See *Deploying the Portal WAR file to WebSphere* on page 90.
- Step 3** Set the JVM Java heap size for each WebSphere instance running Get-Resources. See *Setting the Java heap size* on page 92.
- Step 4** Create the virtual directory you want to use for Get-Resources in your Web server. See *Configuring a virtual directory for IBM HTTP Server* on page 94.
- Step 5** Run the Get-Resources installer.

If you plan on setting up a WebSphere Portal Server or a WebSphere Translation Server, see *Installing WebSphere Portal Server* on page 95 or *Configuring WebSphere Translation Server for Get-Resources* on page 104.

Deploying the Portal WAR file to WebSphere

The Portal WAR file creates the folder structure necessary to deploy Get-Resources in your application server. After you have deployed this file to WebSphere you will be ready to run the Get-Resources installer.

To deploy the Portal WAR file to WebSphere:

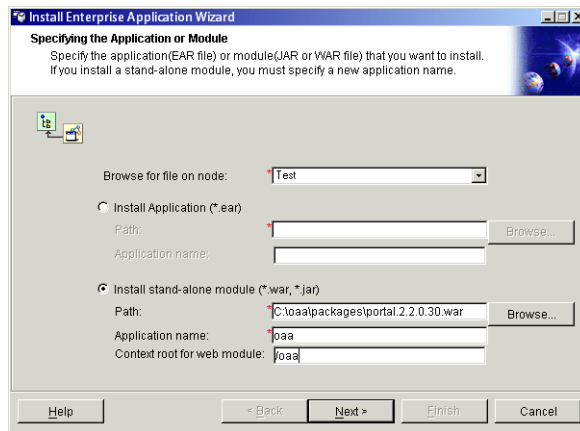
- 1 Verify that the WebSphere Admin Server has been started.
- 2 Open the WebSphere Advanced Administrator's Console (/WebSphere/AppServer/bin/adminclient.sh).
- 3 On the menu at the left side of the console, right-click on **Enterprise Applications** and select **Install Enterprise Application**.
- 4 On the screen displayed, do the following:
 - a Select **Install stand-alone module**.
 - b In the **Path** field, browse to the path to the portal<version #>.war file. The default is <CD Rom Drive>/portal<version #>.war.

For <version #>, Select the most recent version available (4.0.0.44 or greater).

- c In the **Application Name** field, type oaa.
- d In the **Context Root** field, type the name of Get-Resources virtual Web server directory you wish to use. Example: /oaa.

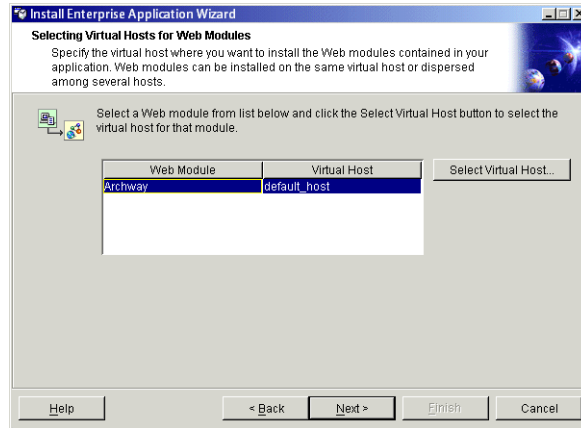
Important: You must create a Web server virtual directory matching the context root you enter here.

The following screen shows the completed form.

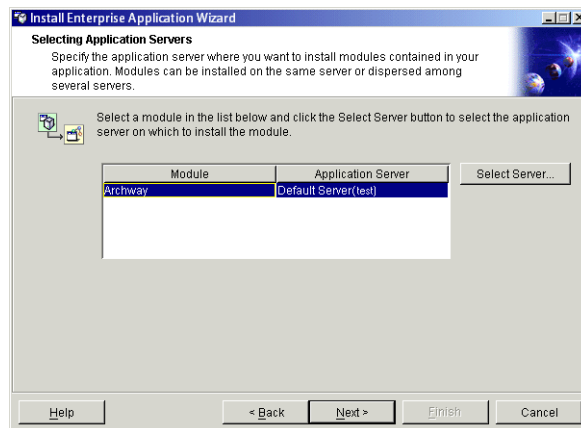


- 5 Click **Next**.
- 6 Click **Next** on the following dialog boxes. These screens will not be used.
 - Mapping Users to Roles
 - Mapping EJB Run As Roles to Users
 - Binding Enterprise Beans to JNDI Names
 - Mapping EJB References to Enterprise Beans
 - Mapping Resource References to Resources
 - Specifying the Default Datasource
 - Specifying Data Sources for Individual CMP Beans

- 7 In the Selecting Virtual Hosts for Web Modules, select the WebSphere server instance you want to use, and then click **Next**.



- 8 In the Selecting Application Servers dialog box, select the WebSphere server instance you want to use, and then click **Next**.



- 9 On the dialog box displayed, click **Finish**.

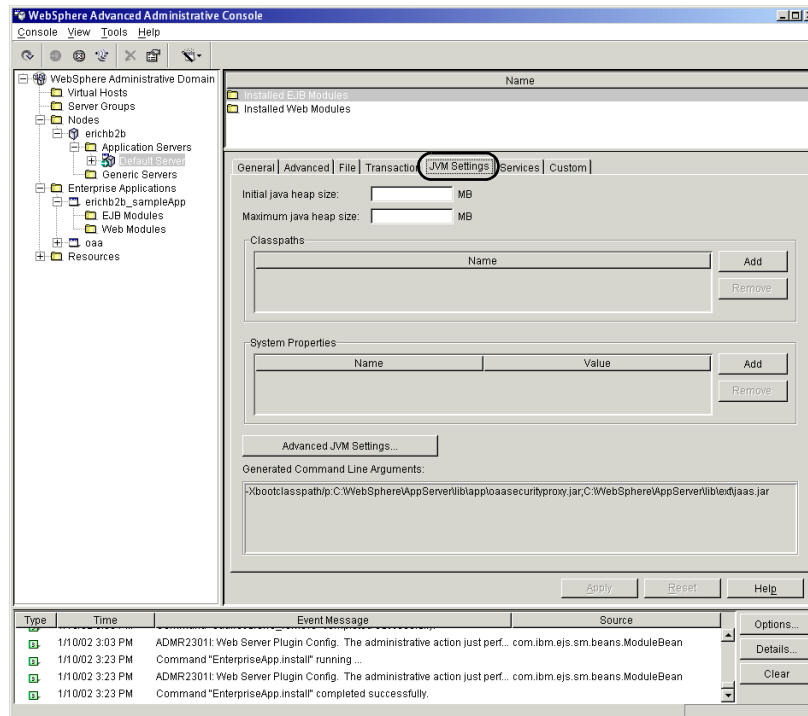
Setting the Java heap size

You can configure how much memory is available for your application server instances. The following instructions assume you are only using one WebSphere instance. You will need to adjust the heap size accordingly if you are load balancing across several WebSphere instances.

To set the Java heap size:

- 1 Verify that the WebSphere Admin Server has been started.
- 2 Open the WebSphere Advanced Administrator's Console (Start > Programs > IBM WebSphere > Application Server > Administrator's Console).
- 3 Click Nodes > <System Name> > Application Servers > <Application server name>.

The server settings page opens.



- 4 Click the JVM Settings tab.
- 5 Set the following JVM settings:
 - a **Initial java heap size.** Type 60.
 - b **Maximum java heap size.** Type the value you want for heap memory. This setting should be at least 225 MB, but not more than 512 MB.

Note: Make sure that the setting for maximum heap size is less than the free RAM available to the application server(s). Exceeding the amount of available RAM causes the JVM processes to swap to disk, reducing overall performance. A setting of 256 MB should be sufficient for most systems.

Configuring a virtual directory for IBM HTTP Server

You must configure a virtual directory for Get-Resources in your Web server. The following instructions assume that you are using WebSphere's built-in Web server – IBM HTTP Server. See your Web server documentation to determine how to create a virtual directory if you are using another Web server.

To configure IBM HTTP Server for Get-Resources:

- 1 Stop the IBM HTTP Server.
- 2 Open the file `httpd.conf` in any text editor. By default this file is located at:
`<root>/usr/HTTPServer/conf`
- 3 Add the following line to the end of the file:

```
Alias /oaa/ "<root>/WebSphere/AppServer/installedApps/oaa.ear/portal.  
<version>.war/"
```

For `<root>`, enter the root directory of the system.

For `<version>`, enter the version number of the WAR file you installed.

Important: The name you define for the virtual directory here must match the context root you defined in WebSphere.

- 4 Save the file.
- 5 Start the IBM HTTP Server.

Installing WebSphere Portal Server

You can configure Get-Resources to display in a WebSphere Portal Server in one of two configurations:

- All Get-Resources and WebSphere components running on a single system. See *Recommended WebSphere Portal Server configuration* on page 95.
- Get-Resources components running on one system and WebSphere components running on another. See *Alternate WebSphere Portal Server configuration* on page 97.

Important: In either configuration, you must first install WebSphere Portal Server. See your WebSphere Portal Server documentation for details.

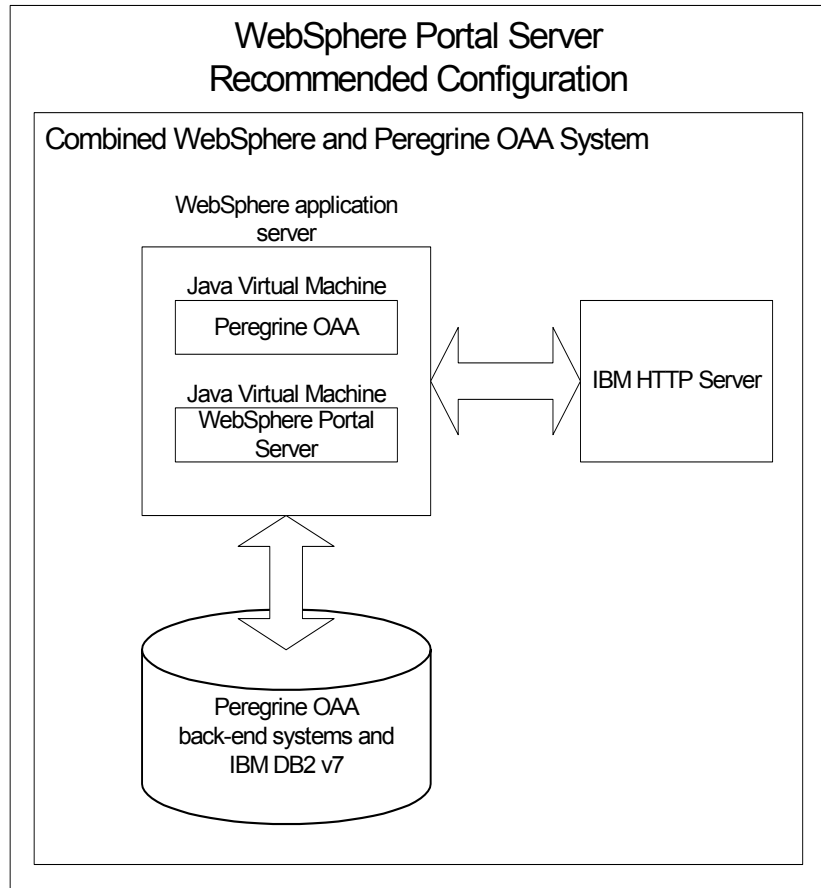
Recommended WebSphere Portal Server configuration

Use the following steps to configure Get-Resources for the recommended WebSphere Portal Server configuration:

- Step 1** Review the WebSphere Portal Server installation requirements. See *WebSphere Portal Server installation requirements* on page 98.
- Step 2** Generate a Get-Resources WAR file containing the portal components WebSphere Portal Server can display. See *Generating a Get-Resources WAR file* on page 99.
- Step 3** Login to the Get-Resources server and stop the WebSphere application server.
- Step 4** Modify the `archway.xml` to change the HTTP authentication method used from Basic to Alternate. See *Modifying the archway.xml file* on page 100.
- Step 5** Modify the `web.xml` to enable the AuthController servlet. See *Modifying the web.xml file* on page 100.
- Step 6** Start the WebSphere application server.
- Step 7** Deploy the Get-Resources WAR file to WebSphere Portal Server. See *Deploying the Get-Resources WAR file to WebSphere Portal Server* on page 101.

- Step 8** Create places and pages in WebSphere Portal Server to display Get-Resources portlets. See *Configuring WebSphere Portal Server places and pages* on page 102.
- Step 9** Enable edit rights for Get-Resources portlets. See *Enabling edit rights for Get-Resources portlets* on page 102.

When complete, your installation will have the following configuration:

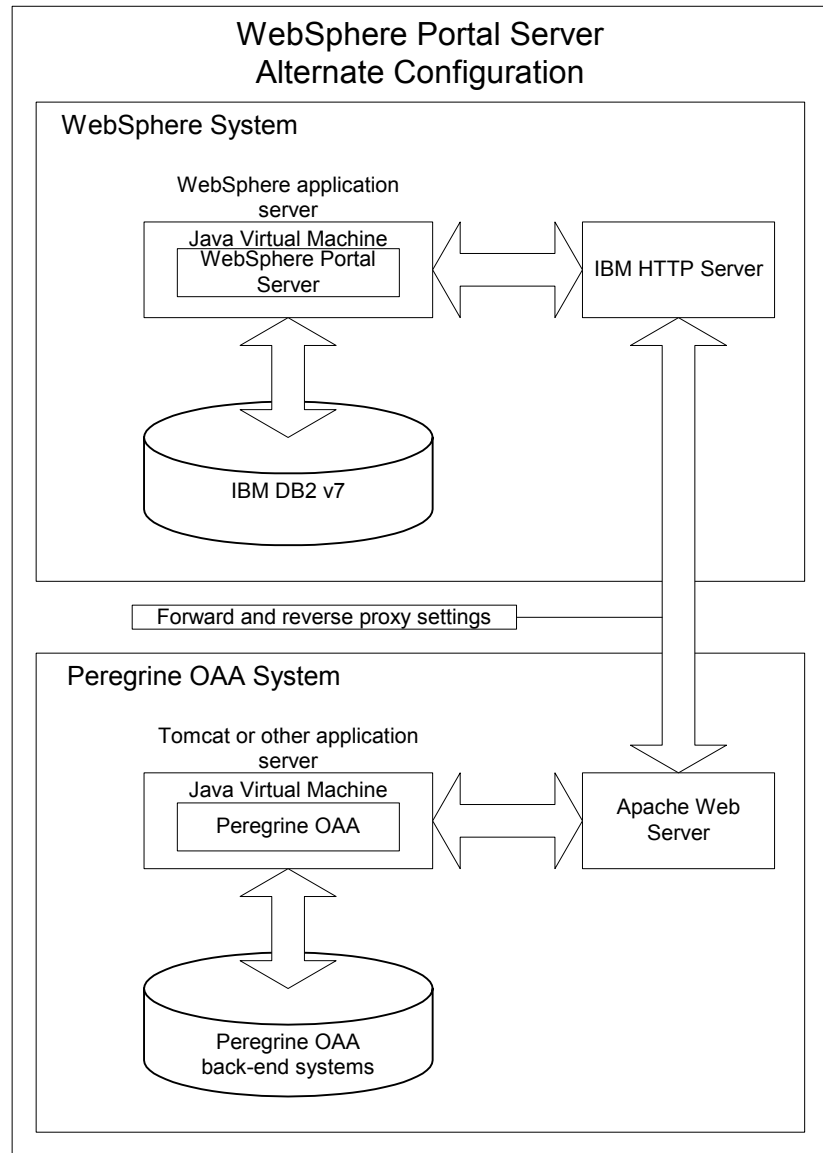


Alternate WebSphere Portal Server configuration

Use the following steps to configure Get-Resources for the alternate WebSphere Portal Server configuration:

- Step 1** Review the WebSphere Portal Server installation requirements. See *WebSphere Portal Server installation requirements* on page 98.
- Step 2** Generate a Get-Resources WAR file containing the portal components WebSphere Portal Server can display. See *Generating a Get-Resources WAR file* on page 99.
- Step 3** Login to the Get-Resources server and stop the WebSphere application server.
- Step 4** Modify `archway.xml` to change the HTTP authentication method used from Basic to Alternate. See *Modifying the archway.xml file* on page 100.
- Step 5** Modify `web.xml` to enable the AuthController servlet. See *Modifying the web.xml file* on page 100.
- Step 6** Modify `setDomain.js` to call the SetDomain function. See *Modifying the setDomain.js file* on page 101.
- Step 7** Start the WebSphere application server.
- Step 8** Deploy the Get-Resources WAR file to WebSphere Portal Server. See *Deploying the Get-Resources WAR file to WebSphere Portal Server* on page 101.
- Step 9** Create places and pages in WebSphere Portal Server to display Get-Resources portlets. See *Configuring WebSphere Portal Server places and pages* on page 102.
- Step 10** Enable edit rights for Get-Resources portlets. See *Enabling edit rights for Get-Resources portlets* on page 102.
- Step 11** Modify IBM HTTP Server's `httpd.conf` file to add forward and reverse proxy URLs. See *Modifying httpd.conf for IBM HTTP Server* on page 103.

When complete, your installation will have the following configuration:



WebSphere Portal Server installation requirements

The recommended configuration of the WebSphere Portal Server requires the following items to be installed on the same server:

- WebSphere application server 4.0.2

- IBM HTTP Server 1.3.19
- IBM DB2 v7 database server
- WebSphere Portal Server
- A custom installation of Get-Resources with WebSphere selected as the application server

The alternate configuration of the WebSphere Portal Server requires the following items be installed on a minimum of two servers:

- Server 1
 - WebSphere application server 4.0.2
 - IBM HTTP Server 1.3.19
 - IBM DB2 v7 database server
 - WebSphere Portal Server
- Server 2
 - Get-Resources compatible application server
 - Web server
 - Back-end database for Get-Resources
 - An installation of Get-Resources

Generating a Get-Resources WAR file

In order to display Get-Resources in WebSphere Portal Server, you must first export the Get-Resources portal components as a WAR file. You can then import this WAR file into WebSphere Portal Server, and choose the portal components you want to display as WebSphere Portal Server portlets.

To generate a Get-Resources WAR file:

- 1 Login to the Get-Resources administration page (`admin.jsp`).
- 2 Click **IBM WebSphere Portal Integration**.
- 3 Enter the following configuration information:
 - a **Source Path**. Enter the full path to the `WebSphere.war` in the Get-Resources **package** folder. By default this folder is:
`<WebSphere>/oaa/packages`
 - b **Destination Path**. Enter the full path and file name you want to use for the generated Get-Resources WAR file.

- c **Base URL.** Enter the full URL to the Get-Resources deployment directory. By default this URL is:

`http://<server>:<port>/oaa/servlet/basicauth`

- 4 Click **Generate WAR file.**

Get-Resources generates a new WAR file with the name and path specified in the Destination Path of step 3.

Modifying the archway.xml file

In order to login via WebSphere Portal Server, you configure Get-Resources to use an alternate HTTP authentication method.

To modify the archway.xml file:

- 1 Using a text editor, open the archway.xml file located at:
`<application server>/webapps/oaa/WEB-INF/default.`
- 2 Edit the line containing:
`<httpauthclass ...>HttpBasicAuthenticationManager</httpauthclass>`
- 3 Change the value `HttpBasicAuthenticationManager` to `HttpAlternateAuthenticationManager`.
- 4 Save the file.

Modifying the web.xml file

You will need to enable the AuthController servlet to establish a proxy for HTTP basic authentication.

To modify the web.xml file:

- 1 Using a text editor, open the web.xml file located at:
`<application server>/webapps/oaa/WEB-INF.`
- 2 Search for the line containing:
`<!-- Uncomment to add support for http basic authentication proxy`
- 3 Move the ending comment tag `-->` from the end of the servlet definition to the comment at the beginning of the servlet definition.

The new servlet definition should appear as follows:

```
<!-- Uncomment to add support for http basic authentication proxy-->
<servlet>
  <servlet-name>AuthController</servlet-name>
  <display-name>AuthController</display-name>
```

```

        <description>A controller (decorator) servlet that can be used to
        enable configurable auth protection of any resource.</description>

<servlet-class>com.peregrine.oaa.archway.AuthControllerServlet</servlet-
class>
    <load-on-startup>2</load-on-startup>
</servlet>

<servlet-mapping>
    <servlet-name>AuthController</servlet-name>
    <url-pattern>/servlet/basicauth/*</url-pattern>
</servlet-mapping>
<servlet-mapping>
    <servlet-name>AuthController</servlet-name>
    <url-pattern>/servlet/auth/*</url-pattern>
</servlet-mapping>

```

- 4 Save the file.

Modifying the setDomain.js file

To use the alternate configuration of WebSphere Portal Server, you must enable the setDomain function.

Note: If you are setting up WebSphere Portal Server in the recommended configuration, you may skip these instructions.

To modify the setDomain.js file:

- 1 Login to the Get-Resources server.
- 2 Stop your application server.
- 3 Using a text editor, open the setDomain.js file located at:

<application server>/webapps/oaaj.js.

- 4 Add the following line to the end of the file:

```
setDomain();
```

- 5 Save the file.

Deploying the Get-Resources WAR file to WebSphere Portal Server

After you deploy the Get-Resources WAR file to WebSphere Portal Server, you can then configure the portlets you want to display, the display settings, and the access rights to each portlet.

See your WebSphere Portal Server documentation for detailed instructions.

To deploy the Get-Resources WAR file:

- 1 Login to the WebSphere Portal as `wpsadmin` or another user with administrative rights.
- 2 Select **Portal Administration** from the Places menu.
- 3 Click **Portlets > Install Portlets**.
- 4 Click **Browse** and navigate to the Destination path you entered when you created the Get-Resources WAR file.
- 5 Click **Next** to load the Get-Resources WAR file.

WebSphere Portal Server displays a list of portlets to be installed.

- 6 Click **Install**.

WebSphere Portal Server installs the portlets and displays the message “Portlets successfully installed.”

Configuring WebSphere Portal Server places and pages

You can deploy Get-Resources portlets in any place or page that meet the following requirements.

Places

Your WebSphere Portal Server places must have the following characteristics:

- Supported markups must include HTML

Pages

Your WebSphere Portal Server pages must have the following characteristics:

- Supported markups must include HTML
- The page must be set to “allow all portlets that a user can access”
- All Get-Resources portlets that you display in a page must grant “all authenticated users” the minimum edit permission.

Enabling edit rights for Get-Resources portlets

WebSphere Portal Server users will need edit rights to the Get-Resources portlets in order to add and customize them to their portal page.

To enable edit rights for Get-Resources portlets:

- 1 Login to the WebSphere Portal as `wpsadmin` or another user with administrative rights.
- 2 Select **Portal Administration** from the Places menu.

- 3 Click **Security > Access Control List**.
- 4 Select the **Special groups** option and select **All authenticated users** from the select box.
- 5 From the Select the objects for the permissions select box, select **portlet applications**.
- 6 Select the Search on option, and then enter **Peregrine** in the Name contains field.
- 7 Click **Go**.
WebSphere Portal Server displays a list of portlets with Peregrine in the name.
- 8 In the Edit column, click **Select All** at the bottom of the table.
- 9 Click **Save**.
Users can now view and customize Get-Resources portlets from the WebSphere Portal Server interface.

Modifying httpd.conf for IBM HTTP Server

In order to use the alternate configuration of WebSphere Portal Server, you will need to modify the `httpd.conf` file used by the IBM HTTP Server to add the forward and reverse proxy URLs to your remote instance of Get-Resources.

Note: If you are setting up WebSphere Portal Server in the recommended configuration, you may skip these instructions.

To modify httpd.conf for IBM HTTP Server:

- 1 Login to the Get-Resources server.
- 2 Stop your IBM HTTP Server.
- 3 Using a text editor, open the `httpd.conf` file located at:
`<root>/usr/HTTPServer/conf`
- 4 Add the following lines to the end of the file:

```
ProxyPass /<aaa root>/ http://<server>:<port>/
<aaa root>/servlet/basicauth/
ProxyPassReverse /<aaa root>/ http://<server>:<port>/
<aaa root>/servlet/basicauth/
```

For `<aaa root>`, enter the name of the oaa virtual directory used by IBM HTTP Server. By default, this virtual directory is `oaa`.

For `<server>:<port>`, enter the server name and communications port number where Get-Resources is installed.

- 5 Save the file.

Configuring WebSphere Translation Server for Get-Resources

You can configure Get-Resources to use a WebSphere Translation Server to provide real-time translations of on-screen data.

To configure WebSphere Translation Server for Get-Resources:

- Step 1** Copy the file `wt.s.jar` to the Get-Resources deployment folder. See *Copying wt.s.jar to the Get-Resources deployment folder* on page 104.
- Step 2** Configure Get-Resources to use the WebSphere Translation Server. See *Configuring WebSphere Translation Server for Get-Resources* on page 104.

Copying wt.s.jar to the Get-Resources deployment folder

The following instructions describe where to find and copy the file `wt.s.jar`.

To copy wt.s.jar to the Get-Resources deployment folder:

- 1 Stop your application server.
- 2 Browse to the location of your WebSphere Translation Server installation.
- 3 Copy the file `wt.s.jar` from this folder.
- 4 Paste the file `wt.s.jar` into the Get-Resources deployment folder located at:
`<Application server install>/WEB-INF/lib`
- 5 Restart your application server.

Configuring Get-Resources to use the WebSphere Translation Server

The following instructions describe how to configure Get-Resources to use the WebSphere Translation Server.

To configure Get-Resources to use the WebSphere Translation Server:

- 1 Login to the Get-Resources admin page (`admin.jsp`).
- 2 Click **Settings** > **Common** tab.

The Admin Settings page opens.

The screenshot shows the Admin Settings page with the following tabs: Portal, Common, Portal DB, Themes, Web Application, Logging, and XSL. The 'Common' tab is selected, showing sub-tabs: Notification Services, NotificationDB, rome, and E-mail. The 'Common Backend' section includes fields for 'portalDB', 'List of target aliases' (weblication;mail), 'Admin name' (Admin), and 'Admin password'. The 'Language Translation' section includes 'Translation Server Factory Class' (com.peregrine.util.WTSLanguageTranslatorFactory), 'Language from which to translate' (English), 'Translation Server IP Address' (10.3.128.181:1097), 'Application path' (WEB-INF/apps/), and 'Event queue' (portalDB). Descriptive text and a link for default IP address are provided for the IP address field. A list of bullet points explains the repository settings for the event queue.

- 3 Enter the following configuration settings:
 - a **Translation Server Factory Class:** Enter the Java factory class for the Translation server. The default Java factory class is:
`com.peregrine.util.WTSLanguageTranslatorFactory`
 - b **Language from which to translate:** Enter the source language that you want translated. The default value is English.
 - c **Translation Server IP Address:** Enter the IP address and communications port to the Translation Server. For example: 10.3.128.181:1097.
 - d **Application path:** Enter the relative path to the application server applications directory. The default value is:
`WEB-INF/apps/`
 - e **Event queue:** Enter the adapter name to you want to use for the event queue engine. The default value is: PortalDB
- 4 Click **Save**.
The Control Panel opens.
- 5 Click **Reset Server**.

Translating on-screen data with a Translation Server

If you plan to store Get-Resources data in a mixture of languages, you can configure Get-Resources to send data to a Translation Server for real time translation. This interface will only translate data retrieved from the back-end database or manually typed into form inputs. If you need a translated user interface, you can purchase a Get-Resources language pack directly from Peregrine Systems.

To translate on-screen data with a Translation Server:

- 1 Enable the translation server from the **Administration > Settings** page as described in *Configuring WebSphere Translation Server for Get-Resources* on page 104.

The translate button appears in the upper right tool bar.

The Translation button.



- 2 Click on the source data or form input you want to translate.

Click on the text you want to translate.

- 3 Click the translate button.

The Translation window opens.

Select the target language from the select box.

- 4 Select the target language to which you want to translate from the drop down select box.

The translation of your selection displays in the Translation box.

WebLogic 6.1 SP3 or SP4

The following procedures configure WebLogic to run Get-Resources on UNIX.

To configure WebLogic 6.1 SP3 or SP4:

- Step 1** Stop both WebLogic and your Web server
- Step 2** Edit the `startWebLogic.cmd` file to set the system password, memory settings, and start mode. See *Editing startWebLogic.cmd* on page 107.
- Step 3** Run the Get-Resources installer.
- Step 4** Create a virtual directory for Get-Resources in your Web server. See *Creating a virtual directory for Get-Resources* on page 108.
- Step 5** Restart WebLogic and your Web server.

Editing startWebLogic.cmd

To edit startWebLogic.cmd:

- 1** Open the file `startWebLogic.cmd` file in any text editor. By default the file is located at:
`/bea/wlserver6.1/config/<mydomain>/`
- 2** Scroll to the following section of the script:


```
echo *****
echo * To start WebLogic Server, use the password      *
echo * assigned to the system user. The system        *
echo * username and password must also be used to    *
echo * access the WebLogic Server console from a web  *
echo * browser.                                       *
echo *****
@rem Set WLS_PW equal to your system password for no password prompt
server startup.
set WLS_PW=password
```
- 3** In the last line, change the word “password” to your WebLogic system password.
- 4** Search for the `-mx` parameter setting in the file. Change this setting to at least 225 MB, but not more than 512 MB.

Note: Make sure that the setting for maximum heap size is less than the free RAM available to the application server(s). Exceeding the amount of available RAM causes the JVM processes to swap to disk, reducing overall performance. A setting of 256 MB should be sufficient for most systems.

- 5 Set the STARTMODE variable to STARTMODE=false.

The first time you start WebLogic after the installation, you will need to start it in development mode for it to find the Web applications that have been deployed.

- 6 Save the file.

Creating a virtual directory for Get-Resources

To run Get-Resources, you need to create a virtual directory in your Web server that maps to your WebLogic deployment folder. The typical installation creates a virtual directory called oaa, but you may specify a different virtual directory name.

Requirements for Get-Resources virtual directory

Requirement	Setting
Create virtual directory	<oaa>
Directory access rights	anonymous
Map to physical path	<WebLogic>/applications/oaa
Remove “allow anonymous access” to	default.asp, login.asp, e_login_main_start.asp
Set security access to only allow “System” and “Authenticated Users” to	default.asp, login.asp, e_login_main_start.asp

For <oaa>, enter the name of the virtual directory you want to use for Get-Resources. Whatever name you enter here you will need to replicate in your application server configuration.

For <WebLogic>, enter the path to your WebLogic installation. By default the path is:

/bea/wlserver6.1/config/<mydomain>

JRun 3.1

The following procedures configure JRun to run Get-Resources on UNIX.

To configure JRun 3.1:

- Step 1** Install a Java run-time environment. The Get-Resources installer includes the Java 2 SDK Standard Edition v1.3.1_05. However, you can also use JRE 1.3.1 if you already have it installed. See *Custom Installation Components* on page 126.
- Step 2** Install JRun from the Allaire Web site to the root of your hard drive.
- Step 3** Apply the latest JRun update. See *Applying the latest JRun update* on page 109.
- Step 4** Deploy the Portal WAR file to JRun to create the necessary folder structure for Get-Resources. See *Deploying the Portal WAR file to JRun* on page 110.
- Step 5** Run the Get-Resources installer.
- Step 6** Move js.jar to the Java development kit ext folder. See *Moving js.jar to the Java development kit* on page 113.
- Step 7** Run the JRun Connector Wizard to establish a connection between JRun and your Web server. See *Running the JRun Connector Wizard* on page 113.
- Step 8** Configure your JRun Java settings. See *Configuring Java settings* on page 113.
- Step 9** Define any library path environmental variables to your back-end databases. See *Defining library path environment variables* on page 116.
- Step 10** Create a virtual directory for Get-Resources in your Web server. See *Creating a virtual directory for Get-Resources* on page 117.
- Step 11** Restart JRun and your Web server.

Applying the latest JRun update

Before you install Get-Resources, you must apply the latest Jrun 3.1 update.

To install the latest JRun update:

- 1 Browse to the following URL:
http://www.macromedia.com/support/jrun/updates/3/updates_31.html
- 2 Click the link for the JRun edition (Enterprise, Advanced, or Professional) and operating system of your server.

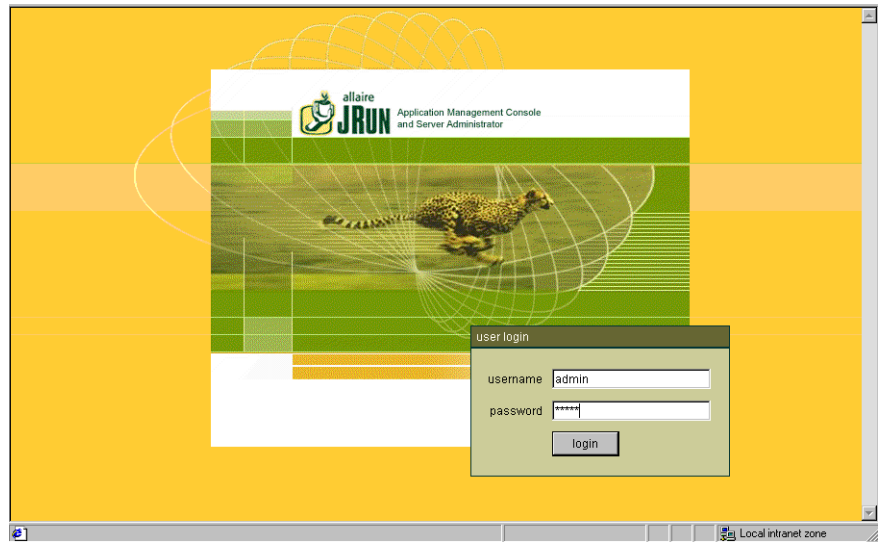
- 3 Follow the installation instructions provided.

Deploying the Portal WAR file to JRun

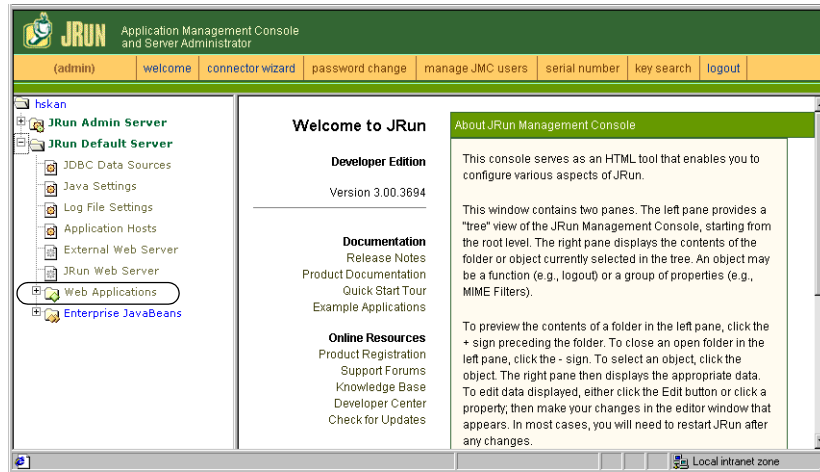
The Portal WAR file creates the folder structure necessary to deploy Get-Resources in your application server. After you have deployed this file to WebLogic you will be ready to run the Get-Resources installer.

To deploy the Get-Resources Portal WAR file to JRun:

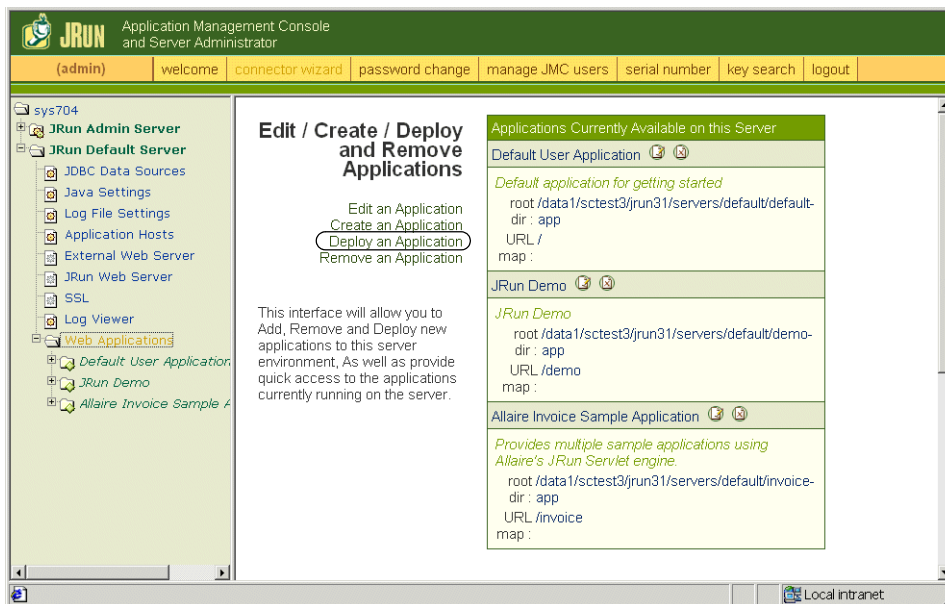
- 1 Open the JRun Management Console and log in.



2 Select JRun Default Server > Web Applications.



The Edit / Create / Deploy and Remove Applications page opens.



- 3 Click the Deploy an Application link.
- 4 In the page that opens, fill out the fields as follows:

- Servlet War File or Directory:

Browse to <CD Rom Drive>/portal<version #>.war.

For <version>, select the most current version.

Select this file, and then click **Accept**.

■ **JRun Server Name:**

Select **JRun Default Server**.

■ **Application Name:**

Type **oaa**.

■ **Application URL:**

Type **/oaa**.

■ **Application Deploy Directory:**

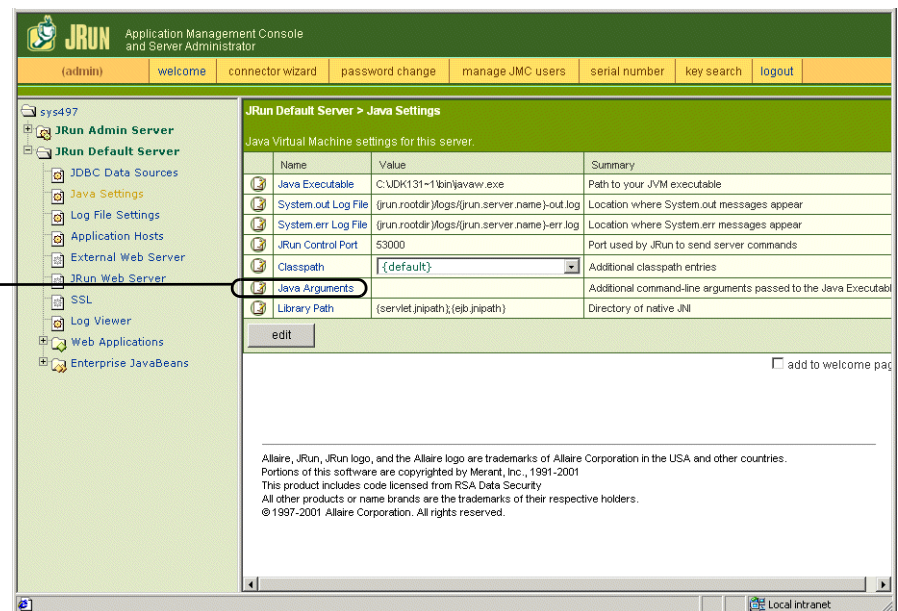
JRun generates this directory. Make a note of this path. You will need this information later in the procedure.

5 Click **deploy**.

A message that OAA has been successfully deployed appears.

6 On the Java Settings page, click **Java Arguments**.

Click **Java Arguments**.



7 In the edit window that opens, use the format example to enter an **-Xmx** value. This defines the maximum amount of heap memory allocated for your system. It is recommended that you set this value to at least 225 MB, but not more than 512 MB.

Note: Make sure that the setting for maximum heap size is less than the free RAM available to the application server(s). Exceeding the amount of available RAM causes the JVM processes to swap to disk, reducing overall performance. A setting of 256 MB should be sufficient for most systems. Applications using Persistence may require a higher setting.

Moving js.jar to the Java development kit

JRun requires an updated version of js.jar in the Java development kit.

To move the js.jar to the Java development kit:

- 1 Stop JRun.
- 2 Locate js.jar. By default this file is installed at:
`<JRun>/servers/default/oaawebinf/lib`
- 3 Cut and paste the file to the following path:
`/usr/local/peregrine/common/jdk1.3/jre/lib/ext`
- 4 Restart JRun.

Running the JRun Connector Wizard

The JRun Connector Wizard establishes a connection between JRun and your Web server.

To run the JRun Connector Wizard:

- 1 Login to the JRun Management Console.
- 2 Click Connector Wizard.
- 3 Select the JRun Default Server as the JRun Server Name.
- 4 Select your Web server from the drop down list box.
- 5 If your Web server uses a different IP address than your JRun server, enter the IP address of your JRun server in JRun Server IP Address.
- 6 Confirm that the JRun Server Connector Port is not in conflict with another communications port used on this server.
- 7 Enter the path to the Scripts Directory.
- 8 Click Done.

Configuring Java settings

After you have installed Get-Resources, you must configure the Java settings that JRun will use to run the Web application.

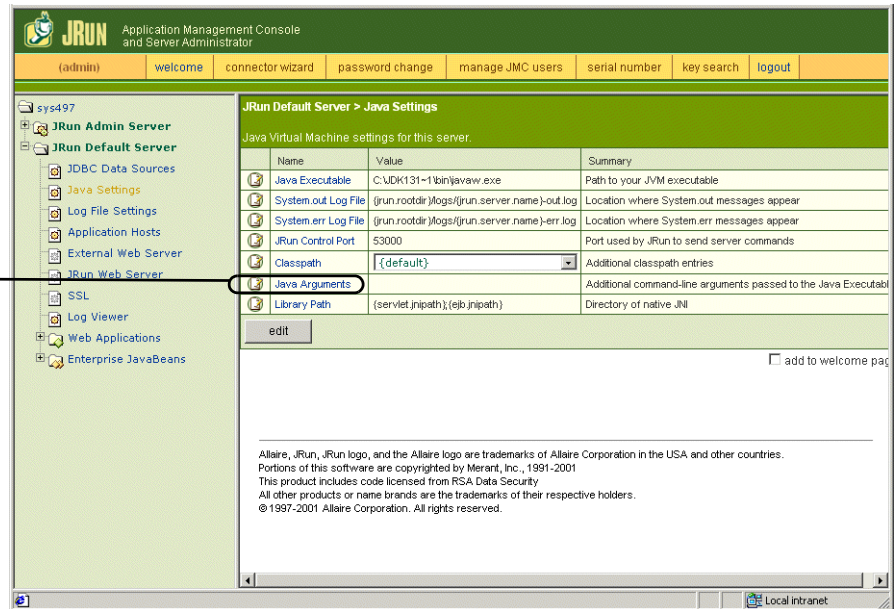
To configure Java settings:

- 1 Login to the JRun Management Console.
- 2 Click JRun Default Server > Java Settings.

The Java Settings page opens.

- 3 Click Java Arguments.

Click **Java Arguments**.



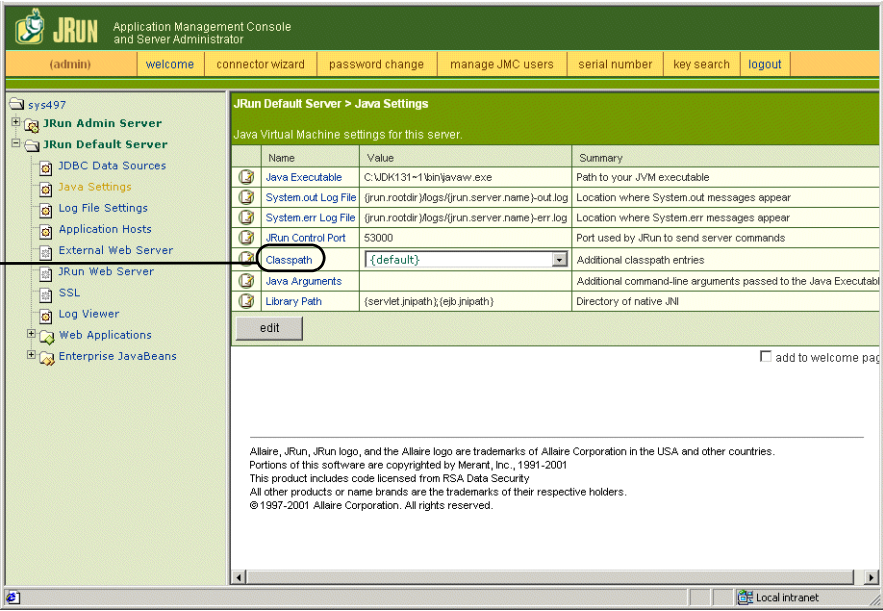
The Edit Window opens.

- 4 Enter an -Xmx value to define the maximum amount of heap memory allocated for your system. It is recommended that you set this value to at least 225 MB, but not more than 512 MB.

Note: Make sure that the setting for maximum heap size is less than the free RAM available to the application server(s). Exceeding the amount of available RAM causes the JVM processes to swap to disk, reducing overall performance. A setting of 256 MB should be sufficient for most systems. Applications using Persistence may require a higher setting.

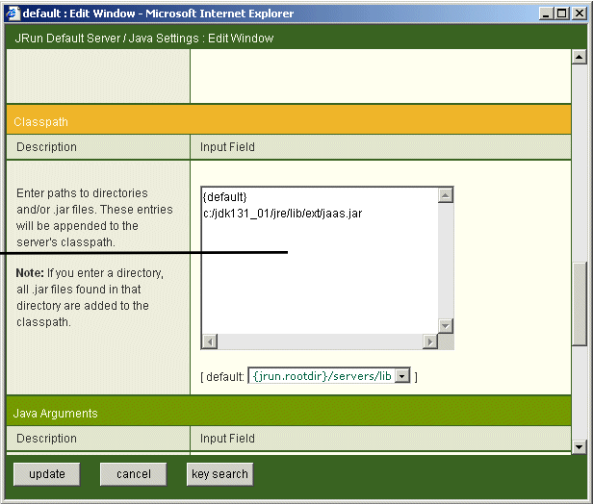
5 On the Java Settings page, click **Classpath**.

Click
ClassPath.



The Edit Window opens.

Type the path to
the JAR files here.



6 Enter the following classpaths:

- Java Development Kit ext folder. For example:
/usr/peregrine/Common/jdk1.3.1_05/jre/lib/ext

- 7 On the Java Settings page, click **Java Executable**.
- 8 Verify that path to your Java Development kit matches the path listed in the Classpath setting. For example:
`/usr/peregrine/Common/jdk1.3.1_05/bin/javaw.exe`
- 9 Click **update**.

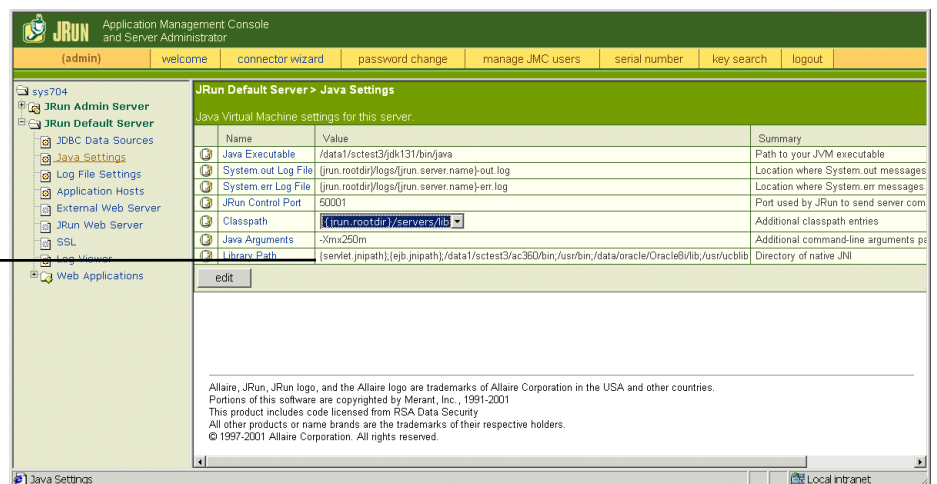
Defining library path environment variables

On UNIX-based systems, you must define the library paths to your back-end databases.

- 1 Open the JRun Management Console and log in.
- 2 On the menu at the left, select **JRun Default Server > Java Settings**.
- 3 Click **Library Path**.
- 4 Add the following library paths if needed:
 - If you are using AssetCenter, add:
 - AssetCenter `/bin` directory
 - Oracle `/lib` directory to the path.
 - If you are running Get-Resources on Solaris, add:
 - `/usr/bin`
 - `/usr/ucblib`
- 5 Click **update**.

JRun displays the updated library paths.

The updated path appears here.



6 Logout of the Management Console.

Creating a virtual directory for Get-Resources

To run Get-Resources, you need to create a virtual directory in your Web server that maps to your JRun deployment folder. The typical installation creates a virtual directory called `oaa`, but you may specify a different virtual directory name.

Requirements for Get-Resources virtual directory

Requirement	Setting
Create virtual directory	<oaa>
Directory access rights	anonymous
Map to physical path	<JRun>/oaa
Remove “allow anonymous access” to	default.asp, login.asp, e_login_main_start.asp
Set security access to only allow “System” and “Authenticated Users” to	default.asp, login.asp, e_login_main_start.asp

For <oaa>, enter the name of the virtual directory you want to use for Get-Resources. Whatever name you enter here you will need to replicate in your application server configuration.

For <JRun>, enter the path to your JRun installation. The recommended installation path is:

/JRun/servers/default

Typical Installation Option

A typical installation of Get-Resources installs the most commonly used components of the product and saves application files and data in default destination directories. Most users choose Typical installation.

Typical Installation Components

Following is a brief description of the components that are automatically installed with a Typical installation of Get-Resources:

Applications and File Locations

Get-Resources Component	Default Installation Directory
Apache Web Server	/usr/local/peregrine/common/Apache2
Tomcat Application Server	/usr/local/peregrine/common/Tomcat4
Java Development Kit	/usr/local/peregrine/common/jdk1.3.1
OAA Platform and Get-Resources	/usr/local/peregrine/oa

Communications Ports

Get-Resources uses the following communications ports in a typical installation. After installation, you can configure Get-Resources to use one or more of the alternate communications ports if your local network already uses these communications ports.

Default Port	Component used by	Alternate Port
80	Apache Web Server	8081
8005	Tomcat application server administration	8015
8009	Tomcat application server worker file	8019
8011	Tomcat application server worker file for load balancing (optional).	8021

Default Port	Component used by	Alternate Port
8013	Tomcat application server worker file for load balancing (optional).	8023
8015	Tomcat application server worker file for load balancing (optional).	8025

Note: To change settings for these components or to use or install different components, use the Custom installation option for Get-Resources.

This completes the installation. If you have not already done so, you will now need to configure your system to connect to the back-end database you are using. This is done on the Settings page of the Admin module.

Typical Installation Procedures

This section explains how to install Get-Resources with a Tomcat application server and an Apache web server on a AIX, Linux, or Solaris operating system.

To perform a typical installation of Get-Resources on UNIX:

- 1 Login to your server.

Important: On servers running AIX, you must login with an account that has root privileges.

- 2 Insert the Get-Resources installation CD into your computer's CD ROM drive. Your computer should automatically launch the installation program.

If the installation program does not automatically start, mount your CD ROM drive. For example:

```
mount /cdrom
```

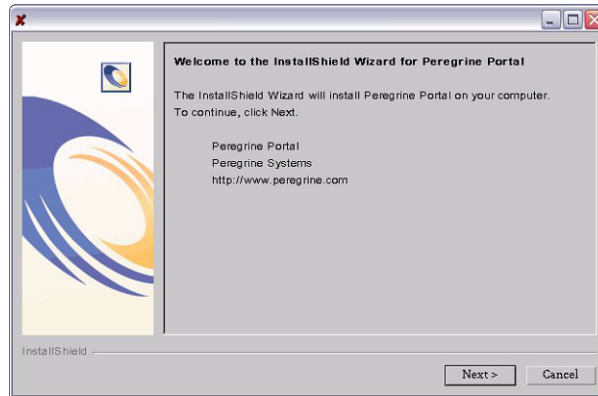
Change directories to your CD ROM. For example:

```
cd /cdrom
```

Enter the installer script specific for your operating system:

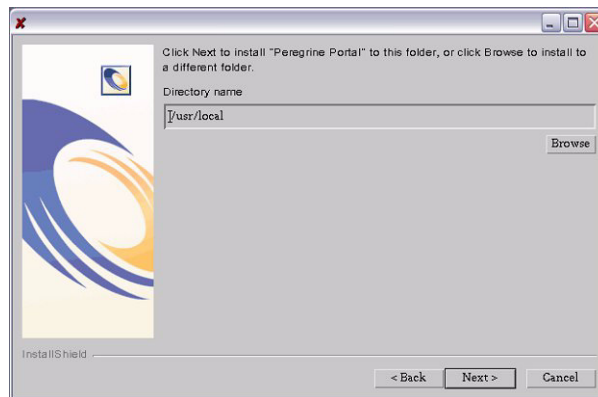
Operating system	Shell script to run
AIX 5.1	<code>./setupaix</code>
Red Hat Linux 7.3	<code>./setuplinux</code>
Solaris 2.7	<code>./setupsolaris</code>
Solaris 2.8	<code>./setupsolaris</code>

The installer welcome page opens.



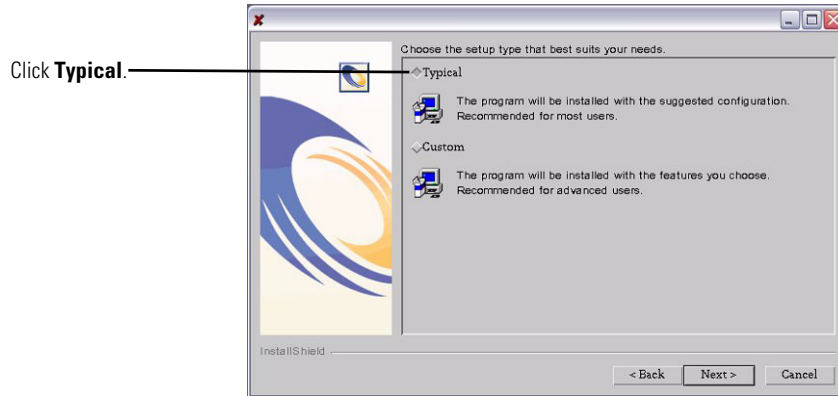
- 3 Click **Next** to continue to the next page of the wizard.

The installation location page opens.



- 4 Click **Browse** to change the default installation location of `/usr/local`.
- 5 Click **Next** to continue to the next page of the wizard.

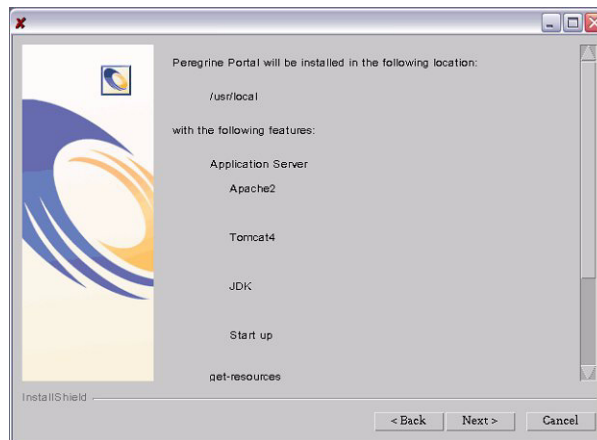
The setup type page opens.



6 Select **Typical**.

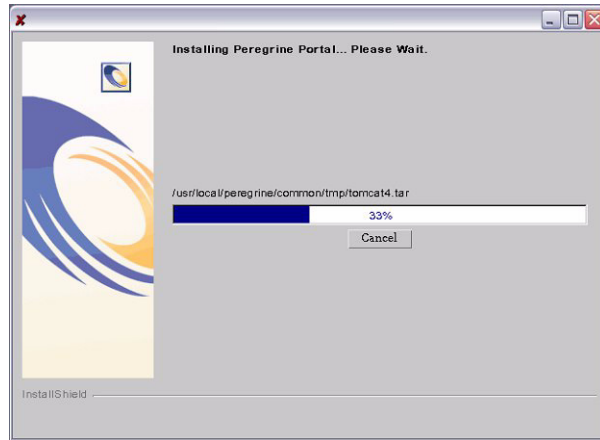
7 Click **Next** to continue to the next page of the wizard.

The review components page opens.

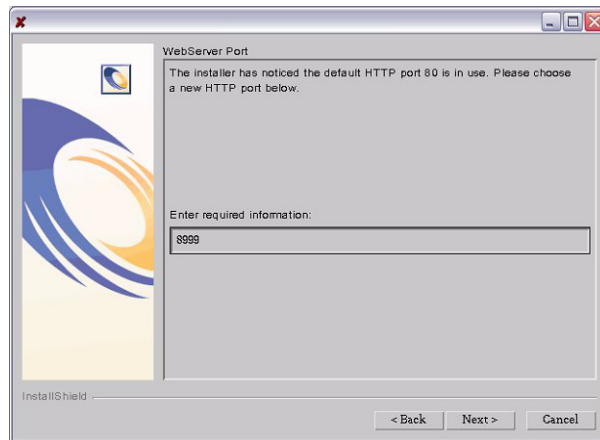


8 Click **Next** to continue to start installing Get-Resources components.

The installation progress page opens.

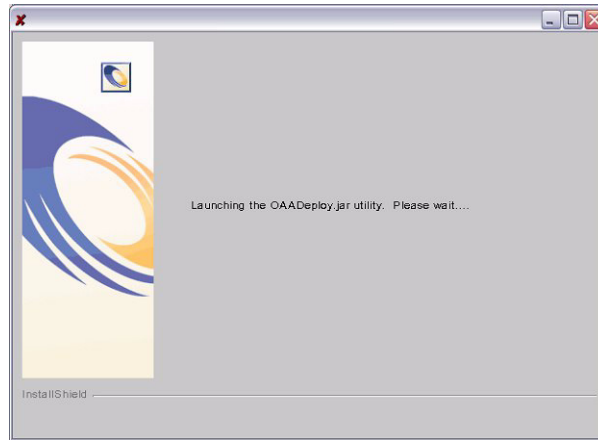


After installation is complete, the installer verifies the availability of port 80 for the Apache Web server. If the installer finds a port conflict on port 80 the Web server port page opens.

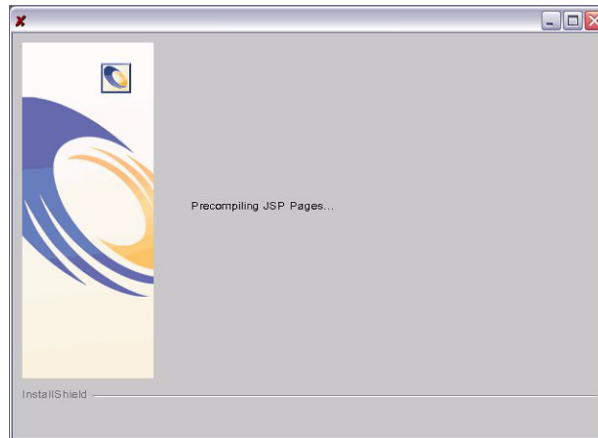


- 9 If required, enter the new Web server communications port.
- 10 Click **Next** to start deploying Get-Resources components.

The Get-Resources deployment utility page opens.

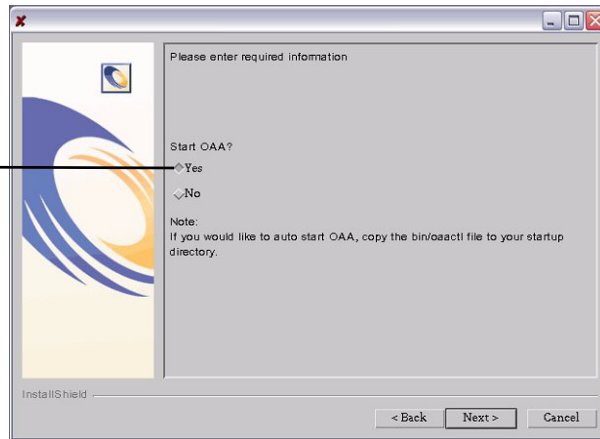


During deployment, the installer precompiles the Java server pages for use by Tomcat.



After the installer has successfully deployed Get-Resources, the start OAA page opens.

Select **Yes**
to start
Get-Resources
immediately.



- 11 Click **Yes** to start Get-Resources immediately or select **No** to manually start Get-Resources after installation is complete.

If you want Get-Resources start every time the server is started, then copy the file `oaactl` into your startup directory. By default this file is located at:

`/usr/local/peregrine/bin/`

You have now completed installing Get-Resources.

Custom Installation Option

The following section describes how to perform a custom installation of Get-Resources on a Unix operating system server, including overview steps for a Development and Production environment.

Custom Installation Components

Following is a brief description of the components that are available for a custom installation of Get-Resources:

Application options

Get-Resources Component	Options
Web Server	<ul style="list-style-type: none"> ■ Apache 2.0.43 ■ IBM HTTP Server 1.3.19 ■ Microsoft IIS 5.0
Application Server	<ul style="list-style-type: none"> ■ Tomcat 4.1.12 ■ WebSphere 4.02 ■ WebLogic 6.1 SP3 ■ JRun 3.1
Java Development Kit	Java 2 SDK
OAA Platform and Get-Resources	None

Communications Ports

The communications ports used by a custom installation of Get-Resources depend upon the application components that you select. Refer to your Web and application server documentation to determine what communications port they require. After installation, you can configure Get-Resources to use alternate communications ports if your local network already uses particular communications ports.

Get-Resources on servers running Oracle 9.2.0.1

If you are running Get-Resources on a server using Oracle 9.2.0.1 you may experience a port conflict over communications ports 8009 and 8080. Consult your Web and application server documentation to see if they use either of these two ports.

If you are using Tomcat as your application server, then by default, there will be a port conflict over port 8009. It is recommended that you change Tomcat to use a different communications port on servers running Oracle 9.2.0.1.

Custom Installation Procedures

To perform a custom installation of Get-Resources on UNIX:

- 1 Login to your server.

Important: On servers running AIX, you must login with an account that has root privileges.

- 2 Insert the Get-Resources installation CD into your computer’s CD ROM drive. Your computer should automatically launch the installation program. If the installation program does not automatically start, mount your CD ROM drive. For example:

```
mount /cdrom
```

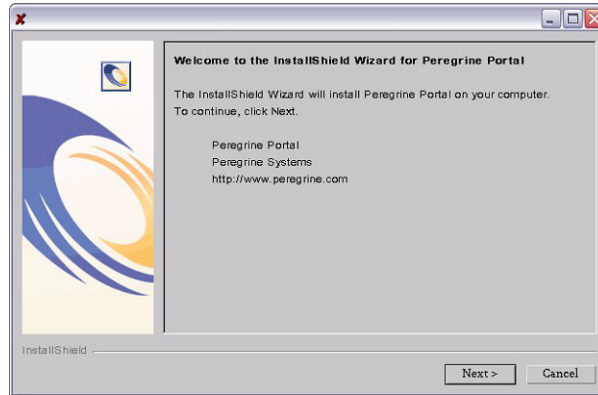
Change directories to your CD ROM. For example:

```
cd /cdrom
```

Enter the installer script specific for your operating system:

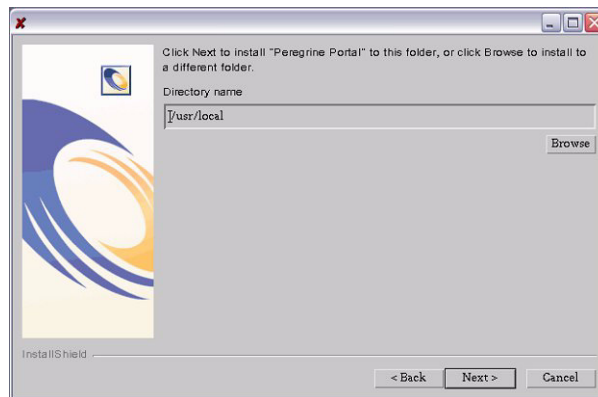
Operating system	Shell script to run
AIX 5.1	./setupaix
Red Hat Linux 7.3	./setuplinux
Solaris 2.7	./setupsolaris
Solaris 2.8	./setupsolaris

The installer welcome page opens.



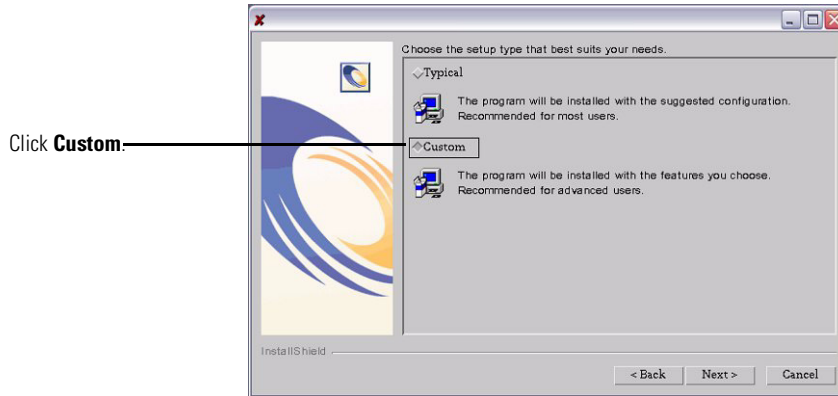
- 3 Click **Next** to continue to the next page of the wizard.

The installation location page opens.



- 4 Click **Browse** to change the default installation location of `/usr/local`.
- 5 Click **Next** to continue to the next page of the wizard.

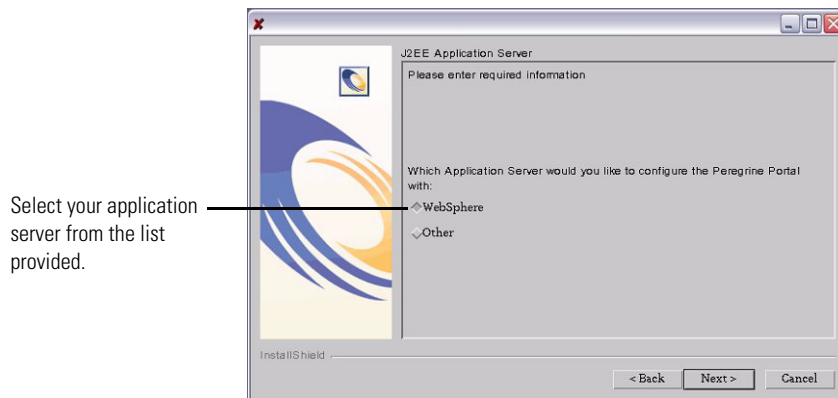
The setup type page opens.



6 Select **Custom**.

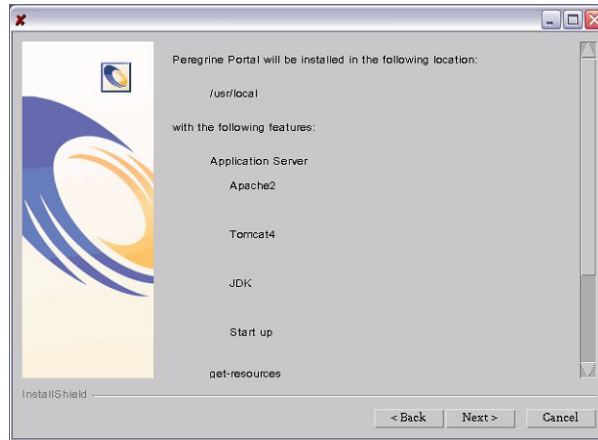
7 Click **Next** to continue to the next page of the wizard.

The select J2EE application server page opens.

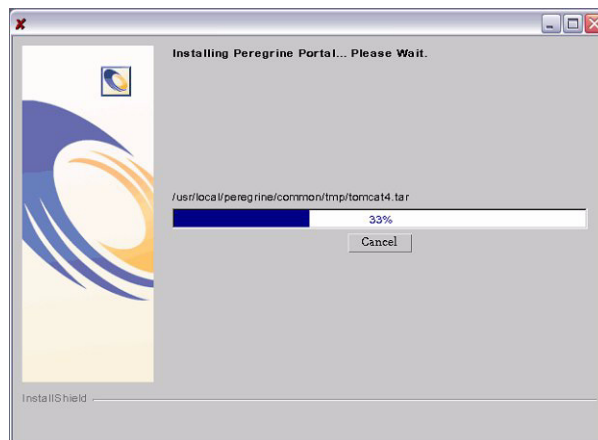


8 Click **WebSphere** to configure a WebSphere application server, or click **Other** to configure any other application server. Click **Next** to continue to the next page of the wizard.

The review components page opens.

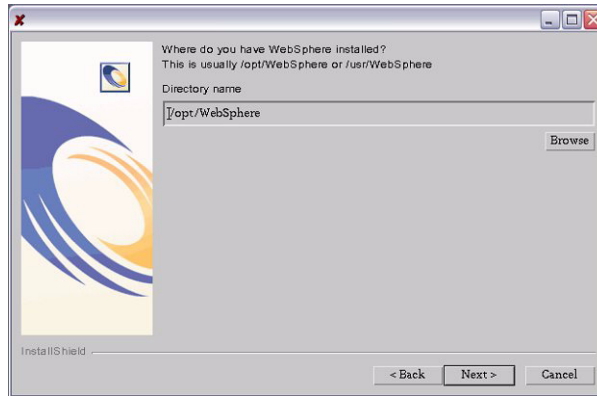


- 9 Click Next to continue to start installing Get-Resources components. The installation progress page opens.



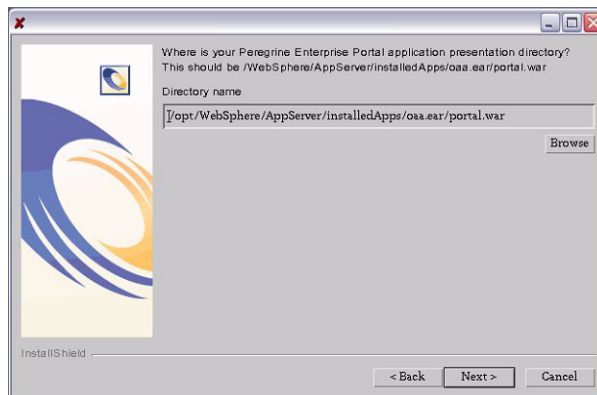
If you have selected to configure a WebSphere application server, you will see the screens in step a through step e.

The WebSphere installation location page opens.



- a Click **Browse** to locate the directory where you installed WebSphere. Click **Next** to continue.

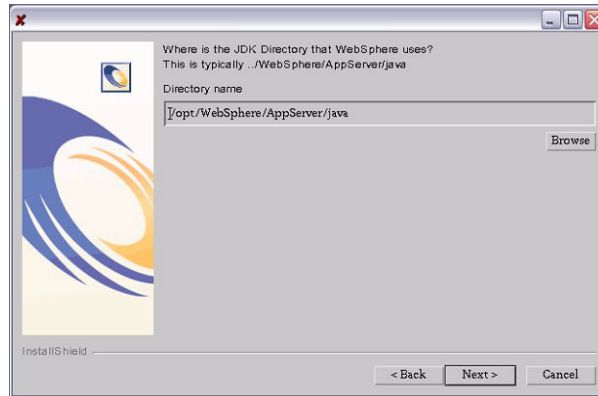
The presentation directory location page opens.



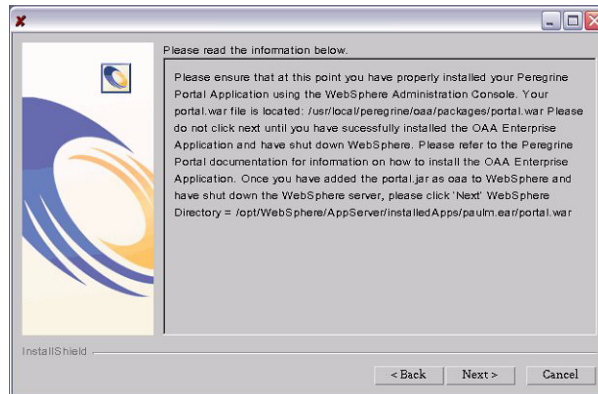
- b Click **Browse** to locate the directory where you deployed the `portal.war` file. Click **Next** to continue.

WebSphere automatically created this directory when you deployed the Get-Resources `portal.war` as an enterprise application. See [WebSphere 4.0.2](#) on page 90 for more information on deploying a WAR file.

The WebSphere JDK installation location page opens.

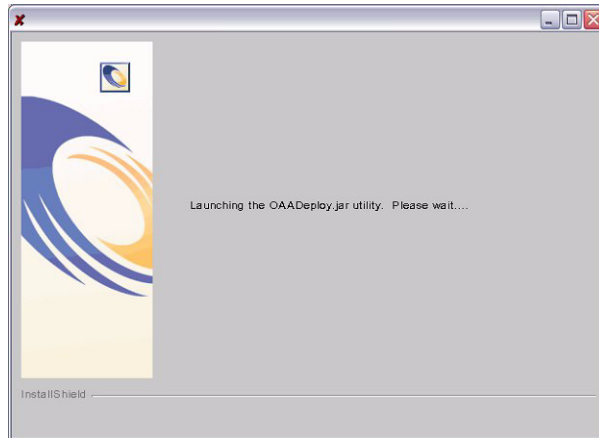


- c Click **Browse** to locate the directory where you installed the Java development kit used by WebSphere. Click **Next** to continue.
The deployment confirmation page opens.

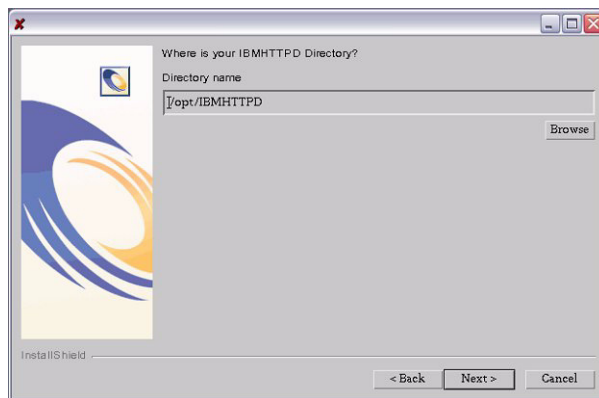


- d After you have successfully deployed **portal.war** in WebSphere, click **Next** to continue.

The Get-Resources deployment utility page opens.



The IBM HTTP Server location page opens.

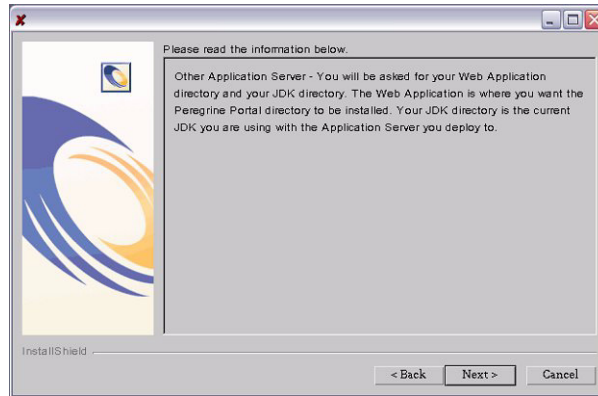


- e Click Browse to locate where you installed the IBM HTTP Server. Click **Next** to continue.

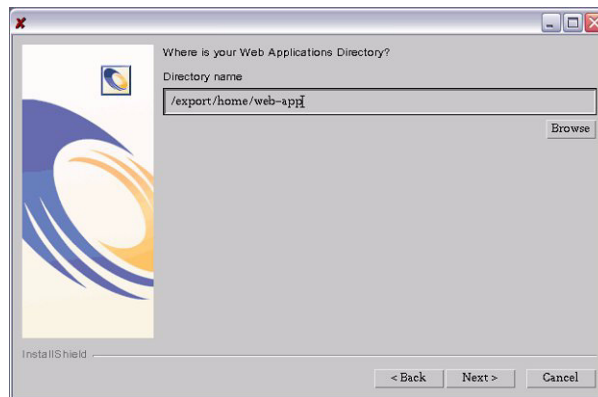
The Get-Resources installer will automatically configure a Web server virtual directory called **oaa**. If you want to define a different Web server virtual directory, see [WebSphere 4.0.2](#) on page 90 for a list requirements.

If you have selected to configure another application server, you will see the screens in step f through step h.

The other application server information page opens.



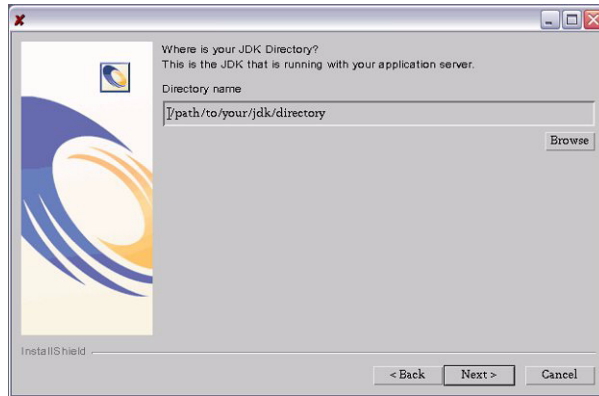
- f** Click **Next** to configure your application server.
The Web applications directory page opens.



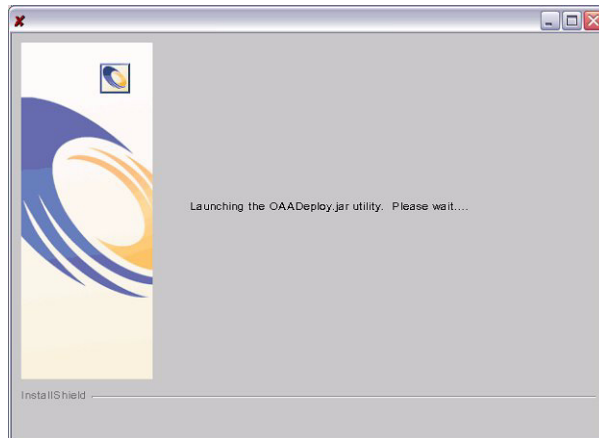
- g** Click **Browse** to locate the directory where you deployed the `portal.war` file. Click **Next** to continue.

For more information on deploying a WAR to application servers, see *Configuring alternate application servers* on page 87.

The application server JDK installation location page opens.



- h Click **Browse** to locate the directory where you installed the Java development kit used by your application server. Click **Next** to continue. The Get-Resources deployment utility page opens.



- 10 Click **Finish** to close the installer.

- 11 Enter the library path environment variables required by your operating system and application server.

Operating system	Environment variable	Add these path values
AIX	LIBPATH	<ul style="list-style-type: none"> ■ /<App server>/WEB-INF/lib/AIX ■ /<App server>/WEB-INF/lib/AIX/ServiceCenter4
Red Hat Linux 7.3	LD_LIBRARY_PATH	<ul style="list-style-type: none"> ■ /<App server>/WEB-INF/lib/Linux ■ /<App server>/WEB-INF/lib/Linux/ServiceCenter4
Solaris 2.7 Solaris 2.8	LD_LIBRARY_PATH	<ul style="list-style-type: none"> ■ /<App server>/WEB-INF/lib/SunOS ■ /<App server>/WEB-INF/lib/SunOS/ServiceCenter4 ■ /usr/ucblib

For <App server>, enter the path to your application server's context root including where you installed the oaa application. Use the following table to determine your application server's context root.

Application server	Context root
WebSphere	/WebSphere/AppServer/installedApps/<application>.ear/portal.4.0.0.<x>/oaa
WebLogic	/bea/wlserver6.1/config/<mydomain>/applications/oaa
JRun	/JRun/servers/default/oaa

For <application>, enter the folder created for Get-Resources.

For <x>, enter the version number of your portal WAR file.

For <my domain>, enter the WebLogic domain you created.

Configuring the WebSphere environment on Linux and Solaris

If you are running on a Linux or Solaris server, you can configure your WebSphere environment using the WebSphere Advanced Management Console.

To configure the WebSphere environment on Linux or Solaris:

- 1 Verify that the WebSphere Admin Server has been started.
- 2 Open the WebSphere Advanced Administrator's Console:
/WebSphere/AppServer/bin/adminclient.sh
- 3 On the left side of the console, click **Nodes** > <Server Name> > **Application Servers**.
- 4 Click on the application server where you installed Get-Resources.
- 5 Click the **General** tab.
- 6 Click **Environment**.
- 7 Click **Add**.
- 8 Enter the following environment details:
 - a **Name**. Type in LD_LIBRARY_PATH.
 - b **Value**. Type in the paths values for your operating system. This value consists of the context root and the relative path to your LIB directories.
- 9 Click **OK**.
- 10 Click **Apply**.
- 11 Close the WebSphere Advanced Administrator's Console.

Configuring the WebSphere startupServer.sh on AIX

If you are running on an AIX server, you must configure your WebSphere environment by editing the startupServer.sh script.

To configure the WebSphere environment on AIX:

- 1 Open startupServer.sh in any text editor.
- 2 Add an entry for LIBPATH and set it to the path values for AIX.

Example:

```
#!/bin/sh
LIBPATH=/usr/lib:/WebSphere/AppServer/installedApps/answer.ear/portal
.4.0.0.55.war/WEB-INF/lib/AIX:/WebSphere/AppServer/installedApps/answ
er.ear/portal.4.0.0.55.war/WEB-INF/lib/AIX/ServiceCenter4
export LIBPATH
```

- 3 Save the file.

Uninstall—AIX, Linux, or Solaris

Use the following instructions to uninstall Get-Resources.

To uninstall Get-Resources from AIX, Linux, or Solaris:

- 1 Open a command prompt.
- 2 Change directories to:
`<root>/usr/peregrine/_uninst`
- 3 Enter the following command to uninstall Get-Resources:
`./uninstall.bin`
- 4 Follow the on-screen instructions to complete the uninstall.

4 Load Balancing

CHAPTER

This chapter covers the following topics:

- *Load balancing application servers* on page 140
- *Creating multiple instances of Tomcat for Apache* on page 142
- *Creating multiple instances of Tomcat for IIS* on page 150

Load balancing application servers

A server running a Web application such as Peregrine's Get-Services or Get-Resources consumes approximately 256 MB of memory per application server instance. You should not set the maximum heap size of the JVM in excess of the free RAM available to the application server(s). Exceeding the amount of available RAM causes the JVM processes to swap to disk, reducing overall performance.

Unlike other Adapters, the AssetCenter and ServiceCenter Adapters each create a single connection to the respective back end. Therefore, the memory consumed on the AssetCenter database server is the same as that consumed by a single client connection. The memory consumed on the ServiceCenter server is also the same as that of a single ServiceCenter client process.

Note that memory usage does not increase significantly per session, because the architecture is based on the sharing of a set of resources and database connections among all sessions handled by the same application server instance. The small amount of memory consumed for session-specific information is released as the users log off or as their sessions expire. Note that server sessions do not expire unless the browser is closed or the user navigates to a different domain.

Because ServiceCenter and AssetCenter adapters maintain a single connection to the back end, adding extra application server instances brings the added benefit of concurrent access to the back-end data store.

The need for extra application server instances and therefore JVMs is directly related to three variables:

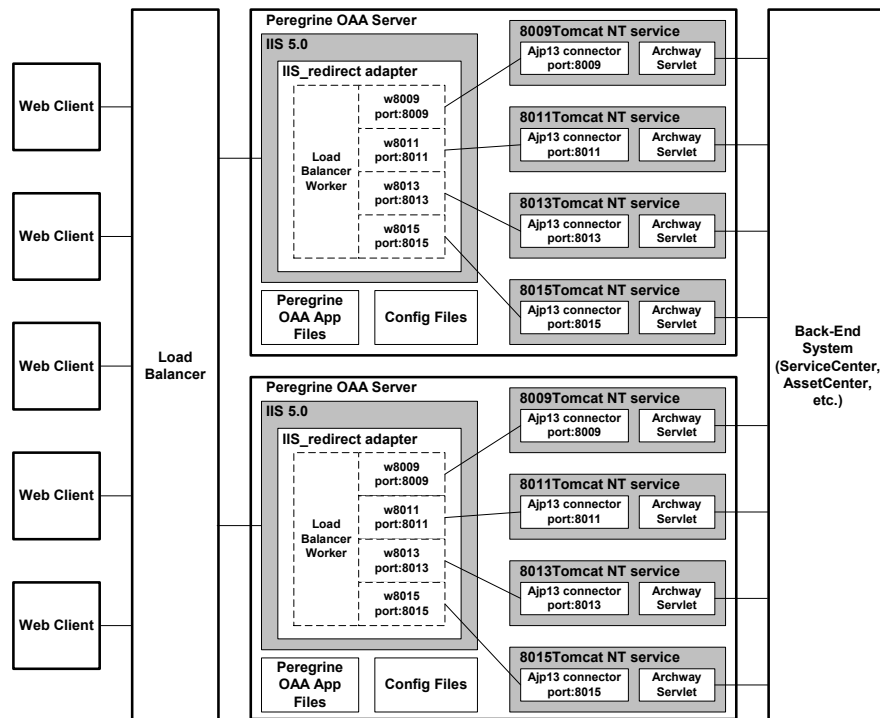
- The number of concurrent users.
- The processing power of the machine hosting the Get-Resources Web server.
- The number of processors on the machine.

Each deployment may make different demands of the software and hardware, but, in any case, optimal back-end throughput for ServiceCenter and AssetCenter is achieved with the maximum number of application server instances that the server can handle without degraded performance due to lack of CPU headroom, file system swapping, and context switching.

Cache synchronization with Symmetric MultiProcessing (SMP) servers can, in most cases, be ignored as a performance tuning factor except in the case of the extremely large-scale systems.

To serve as a control guideline, low-end processors, such as a Pentium 450, should be capable of producing acceptable load handling for around 100 concurrent sessions on a single application server process. A dual Pentium 1000 with 2 gigabytes of RAM (a common data center configuration) should be capable of handling 400+ concurrent sessions using multiple application server instances. When using adapters capable of pooling, for example, the JDBCAdapter or BizDocAdapter, performance beyond the 400-concurrent-user benchmark can be achieved.

The following diagram illustrates the architecture of multiple JVMs:



Creating multiple instances of Tomcat for Apache

Multiple instances of Tomcat are installed as services. Although this is not required, it improves performance, makes the instances easier to manage, and provides extra functionality, including restarting the service if it fails or if the machine on which the instances are installed needs to be restarted.

For systems using IIS, see *Creating multiple instances of Tomcat for IIS* on page 150.

To creating multiple Tomcat instance for Apache:

- Step 1** Create copies of the Tomcat directory. See *Copying the Tomcat directory* on page 143.
- Step 2** Delete the \webapps\oaa directory from the newly copied instances of Tomcat. See *Copying the Tomcat directory* on page 143.
- Step 3** Edit the `workers.properties` file of the first or master Tomcat instance to set the values for each additional Tomcat instance. See *Editing the workers.properties file* on page 143.
- Step 4** Edit the `mod_jk.conf-auto` file of the first or master Tomcat instance to establish a connection between Tomcat and Apache. See *Editing the mod_jk.conf-auto file* on page 145.
- Step 5** Edit the `httpd.conf` file to define the Tomcat workers available for Apache. See *Editing the httpd.conf file* on page 145.
- Step 6** Edit the `server.xml` files for each Tomcat instance. See *Editing the server.xml files for Apache* on page 146.
- Step 7** Edit the `jk2.properties` files for each Tomcat instance. See *Editing the jk2.properties files for Apache* on page 147.
- Step 8** Install multiple instances of Tomcat as a service using `installservice.bat`. This file can be found in the `Tomcat\bin` directory. See *Installing Tomcat instances as services for Apache* on page 148.
- Step 9** Testing the configuration. See *Testing load balancing on Apache* on page 148.

Copying the Tomcat directory

You must create a separate folder for each instance of Tomcat you want to use for load balancing.

To copy the Tomcat directory:

- 1 Open Windows Explorer and copy the Tomcat install folder. By default, this folder is located at:

`C:\Program Files\Peregrine\Common\Tomcat4`

- 2 Paste a copy into the same root path. By default this would be:

`C:\Program Files\Peregrine\Common`

- 3 Rename the new folder to a unique name.

Tip: Include the port number to be used by the Tomcat instance in the folder name. For example, if you are going to use 4 instances of Tomcat listening on ports 8009, 8011, 8013, and 8015, then you could create 4 copies of the Tomcat folder called `\Tomcat4_8009`, `\Tomcat4_8011`, `\Tomcat4_8013`, and `\Tomcat4_8015`.

- 4 Delete the `\webapps\oaa` subdirectory from the newly copied instance of Tomcat.

The additional instances will use the same document root as the first or primary Tomcat instance.

- 5 Repeat step 1 through step 4 for each instance of Tomcat you want to use.

Editing the `workers.properties` file

For each server on which Tomcat instances are installed, there is only one `workers.properties` file. Tomcat installs the `workers.properties` file in the `conf` directory of your primary Tomcat instance. This file will be shared by all other Tomcat instances on that particular server.

The `workers.properties` file specifies the worker threads that the Web server connector will create in order to communicate with the Tomcat instances. Each Tomcat instance must communicate on a different port. The host should be set to the name of the server running the Tomcat instances or `localhost` if they are running on the same server as Apache.

Cache size is the maximum number of user sessions that Apache should direct to the Tomcat instance at one time.

Lbfactor is a number greater than or equal to 1 that Apache uses to load balance the workers. If all the workers are running on servers that have equal performance strengths, the *lbfactor* numbers should be equal. Workers with a lower *lbfactor* will be assigned fewer user sessions by the load balancer worker in Apache.

To edit the `workers.properties` file:

- 1 Open the `workers.properties` file in any text editor.

This file is located in the `/conf` directory of your Tomcat installation.

- 2 Edit the following lines as shown. The paths for `workers.tomcat_home` and `workers.java.home` are the locations of your Tomcat installation and Java SDK installations.

Example:

```
workers.tomcat_home="c:\Program Files\Peregrine\common\Tomcat4"
workers.java.home="c:\Program Files\Peregrine\common\jdk1.3.1_05"
ps=\
worker.list=loadbalancer, w8009, w8011, w8013, w8015
worker.loadbalancer.type=lb
worker.loadbalancer.balanced_workers=w8009, w8011, w8013, w8015
```

Note: You can define the worker names any way you want as long as you continue the same naming convention throughout the procedure.

- 3 Add the following lines for each Tomcat instance you have installed, incrementing the port number for the values shown in step 2:

```
worker.w8009.port=8009
worker.w8009.host=localhost
worker.w8009.type=ajp13
worker.w8009.cachesize=40
worker.w8009.lbfactor=10
```

- 4 Comment out the following lines. These default workers will not be used.

```
worker.ajp12.port=8007
worker.ajp12.host=localhost
worker.ajp12.type=ajp12
worker.ajp12.lbfactor=1

worker.ajp13.port=8009
worker.ajp13.host=localhost
worker.ajp13.type=ajp13
worker.ajp13.lbfactor=1
worker.ajp13.cachesize=10
```

- 5 Save the file.

Editing the mod_jk.conf-auto file

The `mod_jk.conf-auto` file defines where the Worker files are available in Apache. This file is shared by all Tomcat instances on the server. It is important that you do this procedure after you have successfully deployed the necessary Get-Resources files, otherwise the Get-Resources mount points, file locations, and directories will not be included in the `mod_jk.conf-auto` file, and you will have to manually add them.

To edit the `mod_jk.conf-auto` file:

- 1 Make a copy of the `mod_jk.conf-auto` file and rename the copy to `mod_jk.conf-local`.

The `mod_jk.conf-auto` file is located in the Tomcat `conf` directory.

- 2 Open the `mod_jk.conf-local` file in any text editor.
- 3 Change `JKWorkersFile` to point to the `worker.properties` file of the primary Tomcat instance.

Example:

```
JKWorkersFile "C:\Apache\Tomcat\conf\worker.properties"
```

- 4 Change all `JKMounts` to use *loadbalancer* instead of *default worker ajp12*.

Usage: `JKMount<file(s) or directory> <worker name>`

Example:

```
JKMount/aaa/servlet/* loadbalancer
JKMount/aaa/*.jsp loadbalancer
```

- 5 Save the file.

Editing the httpd.conf file

The `httpd.conf` file must include `mod_jk.conf-local`.

To edit the `httpd.conf` file:

- 1 Open the `httpd.conf` file in any text editor.
- 2 Add the following line:

```
include "<Tomcat>/conf/mod_jk.conf-local"
```

For `<Tomcat>`, enter the path to your tomcat installation. By default the path is:

C:\Program Files\Peregrine\Common\Tomcat4

- 3 Save the file.

Editing the server.xml files for Apache

The `server.xml` file contains the information Tomcat needs to connect to the Web server as well as to find the Peregrine OAA Platform Web application files.

To edit the server.xml files:

- 1 Each Tomcat instance has a `server.xml` file located in the `\conf` directory. Open this file in any text editor.
- 2 Update the port number attribute of the `<Server>` element to a unique value that will not conflict with other port numbers used by Tomcat. It is recommended that the port numbers 8005-8008 be used when configuring four Tomcat instances.

Example:

```
<Server port="8005" shutdown="SHUTDOWN" debug="0">
```

- 3 Update the port number used by the Coyote Connector to a unique, non-conflicting value. If you are configuring four Tomcat instance, the values 8009, 8011, 8013, and 8015 are recommended.

Example:

```
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
port="8009" minProcessors="5" maxProcessors="75" enableLookups="true"
redirectPort="8443" acceptCount="10" debug="0"
connectionTimeout="20000" useURValidationHack="false"
protocolHandlerClassName="org.apache.jk.server.JkCoyoteHandler" />
```

- 4 Update the `jvmRoute` attribute of the `<Engine>` element with the the server name and communications port used by each Tomcat instance.

Example:

```
<Engine jvmRoute="localhost:8009" name="Standalone"
defaultHost="localhost" debug="0">
```

The port number should follow the convention used elsewhere in the configuration (8009, 8011, etc.). These entries must be the same as the Tomcat ID entries you added to the `workers2.properties` file.

- 5 Update the `appBase` attribute of the `<Host>` element with the absolute path to the `webapps` directory of the primary Tomcat instance.

Example:

```
<Host name="localhost" debug="0"
appBase="C:\Apache\Tomcat_w8009\webapps" unpackWARs="true"
autoDeploy="true">
```

- 6 Comment out a `<Connector>` tag with the `className="org.apache.coyote.tomcat4.CoyoteConnector"` using port 8080.

Tomcat uses this port to communicate with a browser for direct HTTP requests. Since Apache will be serving the static data, Tomcat does not need to listen on this connector. It will also prevent a user from directly accessing Tomcat instances.

Example:

```
<!--
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
port="8080" minProcessors="5" maxProcessors="75" enableLookups="true"
redirectPort="8443" acceptCount="10" debug="0"
connectionTimeout="20000" useURIVValidationHack="false" />
-->
```

- 7 Change the OAA context so that it is not reloadable.

This prevents Tomcat from reloading the servlet without restarting the service. This improves performance and helps keep the JSP code that the Tomcat instances are serving in sync during an update. All other contexts should be set to `reload=false`.

Example:

```
<Context path="/oaa"
docBase="oaa"
crossContext="false"
debug="0"
reloadable="false" >
</Context>
```

- 8 Save the file.
- 9 Repeat step 2 through step 7 for each copy of the `server.xml` file you made.

Editing the `jk2.properties` files for Apache

You will need to modify each Tomcat instance's `jk2.properties`. This file sets the `jk2` communication port.

To edit the `jk2.properties` files:

- 1 Open the `jk2.properties` file for a Tomcat instance in a text editor.

This file is located in the Tomcat `conf` directory.

- 2 Insert a line for the `channelSocket` port. The port number must match the port number defined in `workers.properties` file for this Tomcat instance.

Example:

```
channelSocket.port=8009
```

- 3 Save the file.
- 4 Repeat step 1 through step 3 for each Tomcat instance.

Installing Tomcat instances as services for Apache

After you have edited the Tomcat files, you can install each instances of Tomcat as Windows services using `installservice.bat`.

To install Tomcat instances as services on Apache:

- 1 Open a DOS command prompt and change directories to your Tomcat `bin` directory.
- 2 Enter the following command to create each Tomcat instance:

```
installservice <service name> <tomcat_home> <jvm_dll_path>
```

Where `<service name>` is the name you wish to give the Tomcat service, `<tomcat_home>` is the Tomcat install directory of the instance for which you are creating the service, and `<jvm_dll_path>` is the Java SDK install directory.

The second and third parameters are optional if you have already set the `CATALINA_HOME` and `JAVA_HOME` environment variables.

Example:

```
installservice Tomcat8009 C:\Apache\Tomcat8009 C:\jdk130_05
```

- 3 Repeat step 1 through step 2 for each Tomcat service you wish to create.

Testing load balancing on Apache

After you have created additional Tomcat instances, you can test if load balancing is occurring using the following steps.

To test load balancing:

- 1 Start all Tomcat instance services.

If you installed Tomcat as a service you can open the Windows Control Panel and start each instance from the Services dialog box.

- 2 Open a browser and log in to Get-Resources.
- 3 Perform an action in Get-Resources. For example, perform a search.
- 4 Logout of Get-Resources.
- 5 Close your browser to clear the connection cache.
- 6 Repeat step 1 through step 5 one time for each Tomcat instance installed. For example, if you have 4 Tomcat instances, then you will need to login and logout a total of 4 times.

The load balancing mechanism uses a Round-Robin algorithm. If load balancing is working successfully, each login attempt should use a different Tomcat instance.

- 7 Download the `archway.log` file.

You can download the `archway.log` file from the **Administration > Server Log** page.

- 8 Open the `archway.log` file in a text editor.
- 9 Verify that connection details list a different Tomcat instance for each connection.

If each connection uses a different Tomcat instance, then the system is load balancing properly.

If each connection uses the same Tomcat instance, the system is not load balancing and needs troubleshooting.

Creating multiple instances of Tomcat for IIS

Multiple instances of Tomcat are installed as services. Although this is not required, it improves performance, makes the instances easier to manage, and provides extra functionality, including restarting the service if it fails or if the machine on which the instances are installed needs to be restarted.

To creating multiple Tomcat instance for IIS:

- Step 1** Create copies of the Tomcat directory. See *Copying the Tomcat directory* on page 151.
- Step 2** Delete the \webapps\oaa directory from the newly copied instances of Tomcat. See *Copying the Tomcat directory* on page 143.
- Step 3** Configure the ISAPI Plugin for IIS. See *Configuring the ISAPI Plugin for IIS* on page 151.
- Step 4** Create and configure a jakarta virtual directory in IIS. See *Creating and configuring a jakarta virtual directory in IIS* on page 152.
- Step 5** Configure IIS to use isapi_redirector2.dll as an ISAPI Filter. See *Configuring the isapi_redirector2.dll as an ISAPI filter* on page 152.
- Step 6** Create and configure an oaa virtual directory in IIS. See *Creating and configuring an oaa virtual directory in IIS* on page 153.
- Step 7** Edit the workers2.properties file of the first or master Tomcat instance to set the values for each additional Tomcat instance. See *Editing the workers2.properties file for IIS* on page 153.
- Step 8** Edit the server.xml files for each Tomcat instance. See *Editing the server.xml files for IIS* on page 155.
- Step 9** Edit the jk2.properties files for each Tomcat instance. See *Editing the jk2.properties files for IIS* on page 156.
- Step 10** Install multiple instances of Tomcat as a service using installservice.bat. This file can be found in the Tomcat\bin directory. See *Installing Tomcat instances as services for IIS* on page 157.
- Step 11** Testing the configuration. See *Testing load balancing on IIS* on page 157.

Copying the Tomcat directory

You must create a separate folder for each instance of Tomcat you want to use for load balancing.

To copy the Tomcat directory:

- 1 Open Windows Explorer and copy the Tomcat install folder. By default, this folder is located at:

`C:\Program Files\Peregrine\Common\Tomcat4`

- 2 Paste a copy into the same root path. By default this would be:

`C:\Program Files\Peregrine\Common`

- 3 Rename the new folder to a unique name.

Tip: Include the port number to be used by the Tomcat instance in the folder name. For example, if you are going to use 4 instances of Tomcat listening on ports 8009, 8011, 8013, and 8015, then you could create 4 copies of the Tomcat folder called `\Tomcat4_8009`, `\Tomcat4_8011`, `\Tomcat4_8013`, and `\Tomcat4_8015`.

- 4 Delete the `\webapps\oaa` subdirectory from the newly copied instance of Tomcat.

The additional instances will use the same document root as the first or primary Tomcat instance.

- 5 Repeat step 1 through step 4 for each instance of Tomcat you want to use.

Configuring the ISAPI Plugin for IIS

The Get-Resources installer automatically places a copy of the ISAPI plugin for IIS in the following folder:

`c:\Program Files\Peregrine\Common\Tomcat4\bin`

Use the following procedures to configure the plugin for your intranet environment.

To configure the ISAPI plugin for IIS:

- 1 Open the file `jk2.reg` in a text editor. This file is located at:
`C:\Program Files\Peregrine\Common\Tomcat4\conf`
- 2 Verify that the “serverRoot” and “workersFile” values list the proper installation path to Tomcat. By default these values are:

```
"ServerRoot"="C:\\Program Files\\Peregrine\\Common\\Tomcat4"
"workersFile"="C:\\Program Files\\Peregrine\\Common\\Tomcat4\\conf\\
workers2.properties"
```

- 3 Save and close the `jk2.reg` file.
- 4 Double-click on the `jk2.reg` file from Windows Explorer.
Windows adds the registry settings to the Windows registry.

Creating and configuring a jakarta virtual directory in IIS

The ISAPI plugin for IIS requires a specific IIS virtual directory in order to run. Use the following guidelines to create the IIS virtual directory. For specific instructions about IIS, refer to Windows Help.

Requirements for jakarta virtual directory

Requirement	Setting
Create virtual directory	jakarta
Directory access rights	execute
Map to physical path	<Tomcat>\bin\isapi_redirector2.dll

For <Tomcat>, enter the path to your tomcat installation. By default the path is:

C:\Program Files\Peregrine\Common\Tomcat4

Configuring the isapi_redirector2.dll as an ISAPI filter

To establish a connection between Tomcat and IIS, you will need to install the file `isapi_redirector2.dll` as an ISAPI filter.

To install isapi_redirect2.dll as an ISAPI filter:

- 1 Open Internet Services management console.
- 2 Right-click the **Default Web Site** node and then click **Properties**.
- 3 Click the **ISAPI Filters** tab.
- 4 Click **Add**.
- 5 Enter the following information:

- a **Filter Name:** jakarta. The filter name must match the name you defined the `jk2.reg` registry file. By default the filter name is jakarta.
 - b **Executable:** `isapi_redirector2.dll`. By default this file is located at:
`C:\Program Files\Peregrine\Common\Tomcat4\bin\isapi_redirector2.dll`
- 6 Click OK.
 - 7 Close the Internet Services management console.

Creating and configuring an oaa virtual directory in IIS

To run Get-Resources from IIS, you need to create a virtual directory that maps to your Tomcat deployment folder.

Requirements for oaa virtual directory

Requirement	Setting
Create virtual directory	oaa
Directory access rights	anonymous
Map to physical path	<Tomcat>\webapps\oaa
Remove “allow anonymous access” to	default.asp, login.asp, e_login_main_start.asp
Set security access to only allow “System” and “Authenticated Users” to	default.asp, login.asp, e_login_main_start.asp

For <Tomcat>, enter the path to your tomcat installation. By default the path is:

`C:\Program Files\Peregrine\Common\Tomcat4`

Editing the workers2.properties file for IIS

For each server on which Tomcat instances are installed, there is only one `workers2.properties` file. Tomcat installs the `workers.properties` file in the `conf` directory of your primary Tomcat instance. This file will be shared by all other Tomcat instances on that particular server.

The `workers2.properties` file specifies the worker threads that the Web server connector will create in order to communicate with the Tomcat instances. Each Tomcat instance must communicate on a different port. The host should be set to the name of the server running the Tomcat instances or `localhost` if they are running on the same server as Apache.

To edit the `worker2.properties` file:

- 1 Open the `workers2.properties` file (located in the `config` directory of your Tomcat installation) in any text editor.
- 2 Create a `channel.socket` entry for each Tomcat instance (also known as a worker).

Example:

```
[channel.socket:<server>:<port>]
info=Description of Tomcat instance
debug=0
tomcatId=<server>:<port>
lb_factor=1
disabled=0
```

For `<server>`, enter the server name of where the Tomcat instance is located.

For `<port>`, enter the communications port on which the Tomcat instance is listening.

`lb_factor` is a number greater than or equal to 1 that IIS uses to load balance the workers. If all the workers are running on servers that have equal performance strengths, you should set the `lb_factor` numbers to equal values (typically 1). If you want to assign fewer user sessions to a given Worker, then assign it a lower `lb_factor` number relative to the other Workers.

- 3 Verify that the `uri` settings lists the proper IIS virtual directory. By default the virtual directory should be `oaa`.

If you have defined a different virtual directory other than `oaa` to run Get-Resources, you will need to change the `uri` values here.

Example:

```
[uri:/oaa/servlet/*]
info=Prefix mapping

[uri:/oaa/*.jsp]
info=Extension mapping
```

- 4 Save the file.

Editing the server.xml files for IIS

You will need a separate `server.xml` file for each Tomcat instance that will be running concurrently. This file contains the information Tomcat needs to connect to the Web server as well as to find the Peregrine OAA Platform Web application files.

To edit the server.xml files:

- 1 Each Tomcat instance has a `server.xml` file. Open it in any text editor.
- 2 Update the port number attribute of the `<Server>` element to a unique value that will not conflict with other port numbers used by Tomcat.

It is recommended that the port numbers 8005-8008 be used when configuring four Tomcat instances.

Example:

```
<Server port="8005" shutdown="SHUTDOWN" debug="0">
```

- 3 Update the port number attribute of the Coyote Connector `<Connector>` element to a unique value that will not conflict with other port numbers used by Tomcat.

It is recommended that the port numbers 8009, 8011, 8013, and 8015 be used when configuring the Coyote Connector.

Example:

```
<Connector className="org.apache.coyote.tomcat4.CoyoteConnector"
port="8009" minProcessors="5" maxProcessors="75" enableLookups="true"
redirectPort="8443" acceptCount="10" debug="0"
connectionTimeout="20000" useURIVValidationHack="false"
protocolHandlerClassName="org.apache.jk.server.JkCoyoteHandler" />
```

- 4 Create a `<Context>` element entry from the first or master Tomcat instance to the Get-Resources deployment directory.

Add the entry just above the “examples” Context entry.

Example:

```
<Context path="/oaa"
docBase="<First Tomcat install>/webapps/oaa"
crossContext="false"
debug="0"
reloadable="false" >
</Context>
```

For the `docBase` attribute, set `<First Tomcat install>` to the absolute path of the first or master Tomcat instance.

- 5 Update the `<Engine>` element with the server name and communications port used by each Tomcat instance.

List the server information in the `jvmRoute` attribute.

Example:

```
<Engine jvmRoute="localhost:8009" name="Standalone"
defaultHost="local host" debug="0">
```

- 6 Update the `<Host>` element with the **webapps** directory used by the first or master Tomcat instance.

List the server information in the `appBase` attribute.

Example:

```
<Host name="localhost" debug="0" appBase="<First Tomcat
install>/webapps" unpackWARs="true" autoDeploy="true">
```

For the `appBase` attribute, set *<First Tomcat install>* to the absolute path of the first or master Tomcat instance.

- 7 Save the file **server.xml**.
- 8 Repeat step 2 through step 7 for each copy of the **server.xml** file you made.

Editing the **jk2.properties** files for IIS

You will need to modify each Tomcat instance's **jk2.properties**. This file sets the **jk2** communication port.

To edit the `jk2.properties` files:

- 1 Open the **jk2.properties** file for a Tomcat instance in a text editor.
- 2 Insert a line for the `channelSocket` port. The port number must match the port number defined in **workers2.properties** file for this Tomcat instance.

Example:

```
channelSocket.port=8009
```

- 3 Save the file.
- 4 Repeat step 1 through step 3 for each Tomcat instance.

Installing Tomcat instances as services for IIS

After you have edited the Tomcat files, you can install each instances of Tomcat as Windows services using `installservice.bat`.

To install Tomcat instances as services on IIS:

- 1 Open a DOS command prompt and change directories to your Tomcat bin directory.
- 2 Enter the following command to create each Tomcat instance:
`installservice <service name> <tomcat_home> <jvm_dll_path>`

Where *<service name>* is the name you wish to give the Tomcat service, *<tomcat_home>* is the Tomcat install directory of the instance for which you are creating the service, and *<jvm_dll_path>* is the Java SDK install directory.

The second and third parameters are optional if you have already set the CATALINA_HOME and JAVA_HOME environment variables.

Example:

```
installservice Tomcat8009 C:\Program Files\Peregrine\Common\Tomcat4
C:\Program Files\Peregrine\Common\jdk1.3.1_05
```

- 3 Repeat step 1 through step 2 for each Tomcat service you wish to create.

Testing load balancing on IIS

After you have created additional Tomcat instances, you can test if load balancing is occurring using the following steps.

To test load balancing:

- 1 Start all Tomcat instance services.
 If you installed Tomcat as a service you can open the Windows Control Panel and start each instance from the Services dialog box.
- 2 Open a browser and log in to Get-Resources.
- 3 Perform an action in Get-Resources. For example, perform a search.
- 4 Logout of Get-Resources.
- 5 Close your browser to clear the connection cache.
- 6 Repeat step 1 through step 5 one time for each Tomcat instance installed. For example, if you have 4 Tomcat instances, then you will need to login and logout a total of 4 times.

The load balancing mechanism uses a Round-Robin algorithm. If load balancing is working successfully, each login attempt should use a different Tomcat instance.

7 Download the `archway.log` file.

You can download the `archway.log` file from the **Administration > Server Log** page.

8 Open the `archway.log` file in a text editor.

9 Verify that connection details list a different Tomcat instance for each connection.

If each connection uses a different Tomcat instance, then the system is load balancing properly.

If each connection uses the same Tomcat instance, the system is not load balancing and needs troubleshooting.

5 Back-end System Configuration

CHAPTER

Get-Resources can be configured for use with AssetCenter or ServiceCenter as a back-end system.

If you use AssetCenter as your back-end system, Get-Resources will access the AssetCenter Procurement module. A sample AssetCenter database is included with the installation of Get-Resources.

If you use ServiceCenter for your back-end system, Get-Resources will access the ServiceCenter Request Management module.

Get-Resources will also utilize the back-end system to authenticate user access rights and store personalization changes.

Important: To continue the installation for Get-Resources, an administrator must perform the procedures in this chapter within the back-end system, either AssetCenter or ServiceCenter. For detailed information on how to perform any of the steps listed in the following sections, refer to the respective administration guides for AssetCenter or ServiceCenter.

This chapter includes the following sections:

- *Configuring AssetCenter for Get-Resources* on page 160
- *Configuring ServiceCenter for Get-Resources* on page 167

Configuring AssetCenter for Get-Resources

Overview of AssetCenter as a back-end system

As requests are opened and processed through Get-Resources, data is stored and updated in the AssetCenter database.

Users are authenticated using AssetCenter profiles, with special user rights provided for use with Get-Resources.

Using the Get-Resources AssetCenter demo database

The AssetCenter Demo database included with the installation already contains the necessary updated tables, workflows and user rights that will enable AssetCenter to perform as a back-end system for Get-Resources.

Important: Although the Get-Resources installation includes an AssetCenter demo database, AssetCenter must be installed separately. If not already installed, during either a full or custom AssetCenter installation, select the AssetCenter API option. Refer to your AssetCenter documentation for more detailed instructions for this procedure.

Follow these steps to configure the AssetCenter Demo database installed with Get-Resources as a back-end system:

- Step 1** Create a new connection (User DSN only) to the demo database installed with Get-Resources for the version of AssetCenter that you are running.

The default path to the demo database is:

C:\Program Files\Peregrine\Common\<appsrv>\webapps\oaa\WEB-INF\etc\en
where <appsrv> is the name of the application server you are using with Get-Resources (i.e. Tomcat 4.1).

Back-end System	Default Folder	Database Filename
AssetCenter 3.60	ac3	am360.db
AssetCenter 4.10	ac4	am410.db

Back-end System	Default Folder	Database Filename
AssetCenter 4.11	ac4	am411.db
AssetCenter 4.2x ¹	ac4	am411.db

¹When using the **am411.db** demo database with AssetCenter 4.2x you may receive a warning message stating that the database version is compatible with the software version, but the database is not quite up to date. Refer to the *AssetCenter Migration Guide* for instructions on updating the database.

- Step 2** If your application server is configured to run as a service, set up a System DSN.
- Step 3** Test the connection by logging in to the database from AssetCenter.
- Step 4** Refer to *Configuring connections to the back-end system* on page 173 for instructions on how to configure settings within Get-Resources for connectivity to the AssetCenter back-end system.

Using an existing AssetCenter database

If you intend to use an existing AssetCenter database as a back-end system, you will need to import scripts which will update the necessary tables, workflows and user rights that AssetCenter needs to work with Get-Resources.

The type of scripts that you need to import will vary depending on whether or not your existing AssetCenter database has been configured to work with a previous version of Get-Resources.

This section includes instructions on how to import the scripts necessary for your version of AssetCenter in addition to a list of script locations and a description of script functions.

Configuring an AssetCenter database that has never been used with Get-Resources

Follow these steps to import the scripts appropriate for your version of AssetCenter:

- Step 1** Login to AssetCenter and connect to the existing database.
- Step 2** Import the scripts provided with the Get-Resources installation.

By default, these files are located in:

C:\Program Files\Peregrine\Common\<appsrv>\webapps\oaa\WEB-INF\etc\<lang>

...ac3 folder for AssetCenter 3.6

...ac4 folder for AssetCenter 4.x

where <appsrv> is the name of the application server you are using with Get-Resources (i.e. Tomcat 4.1) and <lang> is the ID for the localized script (i.e. en for English).

Important: The scripts must be imported in the order outlined here, according to the version of AssetCenter that you are running.

AssetCenter 3.x:

- From the **getit import** folder, import the following in the order listed:
 - feat.scr
 - calcflds.scr
- From the **getit b2b import** folder, import the following in the order listed. These must be imported even if you are not using B2B functionality:
 - calcflds.scr
 - feat.scr
- From the **getit import** folder, import the following scripts in the order listed:
 - profile.scr
 - workflow.scr
 - empldept.scr

AssetCenter 4.x:

- From the **essential** folder, import the following:
 - essential.scr
- From the **sample** folder, import the following:
 - profile.scr
 - workflow.scr
- From the **b2b** sub-folder, import the following:
 - profile.scr

- workflow.scr
- From the demo folder, import the following:
 - empldept.scr

Configuring an AssetCenter database that has been used with a previous version of Get-Resources

Login to AssetCenter and import the scripts necessary for your AssetCenter version from the following path:

C:\Program Files\Peregrine\Common\<appsrv>\webapps\oaa\WEB-INF\etc\<lang>\

where <appsrv> is the name of the application server you are using with Get-Resources (i.e. Tomcat 4.1) and <lang> is the ID for the localized script (i.e. en for English).

Version	Configured for	Path and script filename
AssetCenter 3.6	Get-Resources 2.0.1 or 2.5	ac3/getit import/profile.scr
AssetCenter 4.x	Get-Resources 2.5	ac4/sample/profile.scr

Script Locations

The following table lists the default folder and filename for each script:

Back-end System	Default Folder	Script
AssetCenter 3.6	<i>getit import</i>	feat.scr
		calcflds.scr
		profile.scr
		workflow.scr
		empldept.scr
	<i>getit b2b import</i>	feat.scr
		calcflds.scr
		profile.scr
		workflow.scr

Back-end System	Default Folder	Script
AssetCenter 4.x	<i>essential</i>	essential.scr (includes calcflds.scr, feat.scr, and the same scripts for B2B). This file also imports the itemized list values used by Get-Resources for the request status.
	<i>sample</i>	profile.scr
		workflow.scr
	<i>b2b</i>	profile.scr
		workflow.scr
	<i>demo</i>	empldept.scr

Script Descriptions

The following table lists the contents and functions of each script:

Script	Description
feat.scr	Features for table amEmployee; contains DelegatedApproverId and DelegatedApprExpiration scripts. These features delegate request approvals to a person given by DelegatedApproverId, until the date specified by DelegatedApprExpiration. Features for table amProduct include ManufacturerURL, a text-based feature containing the manufacturer's URL. Others include Architecture, Processor, ProcSpeed, RAM, and HDCapacity, used by the Description calculated field to enhance the product description, none of which are used directly by Get-Resources
calcflds.scr	Calculated field Description for the Product table. Gives a technical description for the product.

Script	Description
profile.scr	<p>Imports user rights which correspond to Get-Resources capability words to control access to the screens.</p> <p>These include:</p> <ul style="list-style-type: none"> getit.requester (access to the request) getit.advancedrequester getit.approver getit.receiver getit.pcardmanager getit.pcarduser getit.pcardbuyer getit.reserve <p>The final user right, getit.receiver, imports three predefined user profiles.</p> <p>The user profiles and their user rights are:</p> <p>getit.admin = getit.admin + getit.requester + getit.advancedrequester + getit.approver + getit.receiver + getit.pcardmanager + getit.pcarduser + getit.buyer + getit.b2badmin + getit.shopdirect + getit.reserve</p> <p>getit.default = getit.requester</p> <p>getit.full = getit.requester + getit.advancedrequester + getit.approver + getit.receiver + getit.pcardmanager + getit.pcarduser + getit.buyer + getit.b2badmin + getit.shopdirect + getit.reserve</p> <p>This script is provided as an example of how profiles can be defined. You can modify this as needed.</p>

Script	Description
workflow.scr	<p>Imports necessary workflows to make Get-Resources work out-of-the-box. Listed in order of Reference, Name, and Description.</p> <p>BUNDLPO; Bundle Ordering; Workflow necessary to work with the bundles</p> <p>REQROUTE; Routing Request; Sample workflow that manages the full qualification of an off catalog request before it enters the approval cycle</p> <p>REQAPPR_GR; RequestApproval; Sample approval workflow. It is important to note that only the workflow with the REQAPPR_GR reference will be displayed in Get-Resources.</p> <p>REQSTATUS; Request Status; Workflow that, on fulfillment of the request, changes its status to "Received" for better Get-Resources visibility.</p> <p>The REQSTATUS script also imports three employee workgroups. These include:</p> <p>IT: Technical approvers group (any employee in this group can give a technical approval for a request, provided one has the getit.approver user right).</p> <p>Finance: Financial approvers group (any employee in this group can give a financial approval for a request, provided one has the getit.approver user right).</p> <p>Getit: Employees group that will qualify the off catalog request.</p> <p>Note: The off catalog request qualification must be performed in AssetCenter. There is no module for this task in Get-Resources.</p> <p>This script is provided as an example of how workflows can be defined. You can modify the workflows as needed. Refer to page 96 for more information about the AssetCenter workflows provided with Get-Resources.</p>
empldept.scr	<p>This script is optional and is considered demonstration data. It imports one employee per profile, as defined in the following.</p> <p>Valentine: Michael Valentine is a getit.admin.</p> <p>Hartke: Richard Hartke is a getit.full.</p> <p>Tossi: Michaela Tossi is a getit.default.</p>

Refer to *Configuring connections to the back-end system* on page 173 for instructions on how to configure settings within Get-Resources for connectivity to the AssetCenter back-end system.

Configuring ServiceCenter for Get-Resources

Overview of ServiceCenter as a back-end system

Get-Resources accesses the catalog in the Request Management module of ServiceCenter. Requests opened in Get-Resources are stored as quotes in the ServiceCenter database.

Users are authenticated using ServiceCenter Operator records, with special capability words provided for use with Get-Resources. Refer to the Security chapter in the *Get-Resources Administration Guide* for more information about user authentication.

Using ServiceCenter 4.x with Get-Resources

In order to perform as a back-end system for Get-Resources, ServiceCenter 4.x requires a file load (**201patch6.unl**, provided with this installation) that will update tables and security records necessary for the back-end system.

Warning: If you are using a ServiceCenter system which has already been configured for use with a previous version of Get-Resources, follow the instructions below to find out if the **201patch6.unl** file has already been loaded. Reloading this file may cause data and functionality issues with your ServiceCenter system.

To verify whether the **201patch6.unl** file has already been loaded in your ServiceCenter system:

- 1 Login to ServiceCenter with administrative rights.
- 2 Use the command prompt to access the GetResRM input event map and verify if **employee.id** is mapped in position (Pos) 83. Refer to the *ServiceCenter Administration Guide* for detailed instructions on how to perform this function.

If this value exists, you do not need to load the **201patch6.unl** file. Refer to *Configuring connections to the back-end system* on page 173 for instructions on how to configure settings within Get-Resources for connectivity to the ServiceCenter back-end system.

If this value does not exist, follow the instructions below to load the **201patch6.unl** file.

To load the 201patch6.unl file into ServiceCenter:

Step 1 Log into ServiceCenter and load the **201patch6.unl** file located in the following directory:

C:\Program Files\Peregrine\Common\<appsrv>\webapps\oaa\WEB-INF\etc\sc4
where <appsrv> is the name of the application server you are using with Get-Resources (i.e. Tomcat 4.1).

Step 2 Stop and restart the ServiceCenter service.

Step 3 Refer to *Configuring connections to the back-end system* on page 173 for instructions on how to configure settings within Get-Resources for connectivity to the ServiceCenter back-end system.

Using ServiceCenter 5.x with Get-Resources

ServiceCenter 5.x includes all components necessary for use as a back-end system with Get-Resources.

Refer to *Configuring connections to the back-end system* on page 173 for instructions on how to configure settings within Get-Resources for connectivity to the ServiceCenter back-end system.

6 Get-Resources Configuration

CHAPTER

Final configuration settings necessary to complete the installation of Get-Resources are performed in the Peregrine Portal Administration module. This includes parameter configuration and login information for the back-end system, as well as verification of adapter connectivity for Get-Resources.

More information for administration and maintenance of Get-Resources using the Peregrine Portal Administration utility is available in the *Get-Resources Administration Guide*.

This chapter covers the following topics:

- *Logging in to Peregrine Portal Administration* on page 170.
- *Configuring connections to the back-end system* on page 173.
 - *Instructions for AssetCenter* on page 173.
 - *Instructions for ServiceCenter* on page 178.

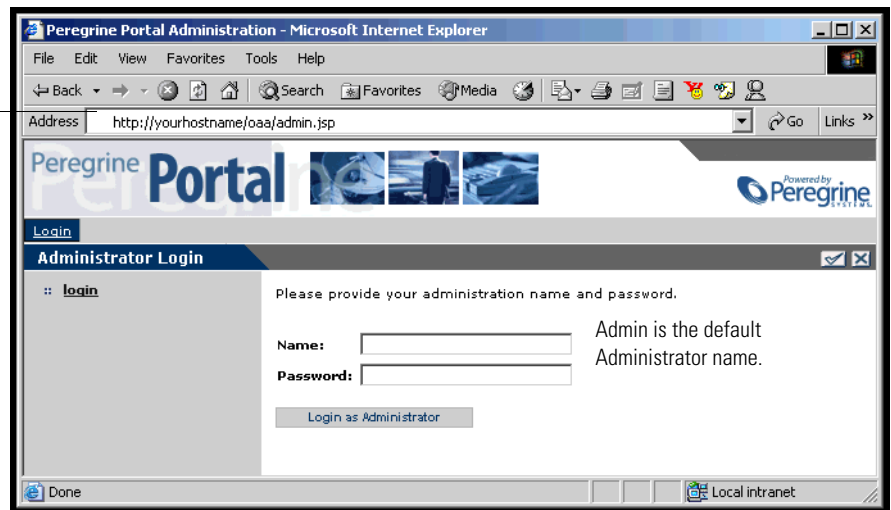
Logging in to Peregrine Portal Administration

The Peregrine Portal Administrator Login page enables access to the Peregrine Portal Administration module.

To access the **Peregrine Portal Administrator Login** page:

- 1 Verify that your application server (for example, Tomcat) is running.
- 2 In your Web browser Address field, type:
`<hostname>/oaa/admin.jsp`
- 3 Press Enter to open the Peregrine Portal Administrator Login page.

Type your
hostname to
connect to your
local server.



- 4 In the Name field, type **Admin**.

No password is required on initial login. Refer to the [Get-Resources Administration Guide](#) for detailed instructions on how to change the Administrator login name and password.

- 5 Click **Login as Administrator** or press Enter to open the **Control Panel** page.

Here is a list of the adapters currently registered in this server. If necessary, you may also reset the server and all its connections.

Connection Status		
Target	Adapter	Status
weblocation	com.peregrine.oaa.adapter.ac.ACAAdapter	connected
mail	com.peregrine.oaa.adapter.mail.MailAdapter	disconnected
portalDB	com.peregrine.oaa.adapter.ac.ACAAdapter	connected
ac	com.peregrine.oaa.adapter.ac.ACAAdapter	connected

Active User Sessions				
Server Name	Last Min.	5 Min. Avg.	20 Min. Avg.	Peak
localhost	0	0	0	1

Page Hits per Minute				
Server Name	Last Min.	5 Min. Avg.	20 Min. Avg.	Peak
localhost	0	0	0	8

Control Panel page

The Control Panel page displays the connection status and additional information for adapters configured for use with Get-Resources. It also provides an option to reset the application server to apply configuration changes or to troubleshoot adapter connectivity.

Note: When you first access the Control Panel page, the status for all targets will be *disconnected*. This will change when you define the targets later on in this section.

Resetting the server

After making any configuration changes in the Peregrine Portal Administration module, the system returns to the Admin Control Panel and displays the following status message at the top of the page:

The settings have been saved. To ensure all changes take effect, you must reset the Peregrine Portal server.

Follow these instructions to reset the server and apply your configuration changes to Get-Resources:

- 1 Click the **Reset Server** button at the bottom of the page.

When the operation is complete, the following status message is displayed at the top of the page:

The Archway servlet and its Adapter connections have been reset successfully.

- 2 Check the Connection Status table to verify the addition, removal or connectivity of the target adapters for which you have made any changes.

Left menu pane

The left menu pane provides access to a number of different administrative utilities for Get-Resources. You will access some of these utilities during this installation process. All of the utilities and their functions are described in detail in the *Get-Resources Administration Guide*.

Configuring connections to the back-end system

Instructions for AssetCenter

By default Get-Resources is installed with all of the connectivity required to interface with AssetCenter through a special adapter, the ACAdapter.

This section describes how to configure and verify back-end system settings and connect to the AssetCenter database.

Connection settings

Follow these instructions to verify settings for AssetCenter as a back-end system:

- 1 In the left menu pane of the Peregrine Portal Administration page, click **Settings**.

This displays the Settings page which provides access to the various tabs used to configure settings for Get-Resources.

The Common settings page is displayed by default.

The screenshot shows the 'Common Backend' configuration page. At the top, there are tabs for 'Portal', 'AssetCenter', 'Common' (selected), 'Portal DB', 'Themes', and 'Web Application'. Below these are sub-tabs for 'Logging', 'Get-Resources' (selected), 'XSL', and 'E-mail'. The main content area has four sections, each with a text input field and a descriptive tooltip:

- Common Backend:** The input field contains 'portalDB'. The tooltip states: 'Adapter target name used to support common user operations.'
- List of target aliases:** The input field contains 'weblication;mail'. The tooltip states: 'Specifies a list of semicolon delimited target aliases used by web applications in this package.'
- Admin name:** The input field contains 'Admin'. The tooltip states: 'The admin login name. This login provides access to administrative functionality without authenticating against backend adapters.'
- Admin password:** The input field is empty. The tooltip states: 'The admin login password.'

- 2 At the top of the page, click the **Get-Resources** tab.

This displays the **Connections Settings** page for Get-Resources:

Get-Resources Connection Settings

Get-Resources Target:

The backend for Get-Resources web applications. This should be either AC if AssetCenter is installed or SC if ServiceCenter is installed.

Request Backend:

Backend used to manage requests. (for example, **sc**, or **ac**.) The default is **ac**.

- 3 Verify that the value is **ac** for the following fields:

Field Name	Default Value	Description
Get-Resources Target:	ac	Back-end for Get-Resources web applications.
Request Backend:	ac	Back-end used to manage requests.

If the values are not set to **ac**, update the fields and click **Save**.

If you made any changes, click the **Save** button to return to the Admin Control Panel page.

AssetCenter database settings

Follow these instructions to connect to the AssetCenter database:

- 1 In the left menu pane of the Peregrine Portal Administration page, click **Settings**, then click the **AssetCenter** tab at the top of the page.

This displays the AssetCenter settings page.

Portal **AssetCenter** Common Portal DB Themes Web Application Logging

Get-Resources XSL E-mail

Date Format: The default date format for data sent to AssetCenter.
 yyyy-MM-dd

Database: Name of the AssetCenter database.
 ACDemo42en
[Click for default: \[ACDemo351ENG\]](#)

Anonymous name: Anonymous user name used when an unknown user attempts to communicate with AssetCenter.
 Admin

Anonymous password: Anonymous user password.

Admin name: Administration user used by the Peregrine Portal when performing tasks such as user authentication and registration.
 Admin

Admin password: Administration password.

2 Update the following fields:

Field Name	Value Description	Default Value
Database:	Name of the AssetCenter database you are using as a back-end for Get-Resources.	ACDemo351ENG
Anonymous name:	Anonymous user name used when an unknown user attempts to communicate with AssetCenter.	Admin
Anonymous password:	Password for the anonymous user name.	(none)
Admin name:	Administration login name used by the Peregrine Portal when performing tasks such as user authentication and registration.	Admin
Admin password:	Administrator password.	(none)

3 When finished updating these fields, click the **Save** button to return to the Admin Control Panel page.

Portal database settings

Get-Resources enables personalization of portal application screens without the need to manually change and compile code. To enable this feature, Get-Resources requires a database adapter connection to store settings and customizations to the portal interface in the back-end database. Until a database adapter is defined for the portal page, users will not be able to see or make personalization changes to the Peregrine portal home page.

Follow these instructions to configure Get-Resources to save personalization settings in the AssetCenter back-end database:

- 1 In the left menu pane of the Peregrine Portal Administration page, click **Settings**.
- 2 At the top of the Settings page, click the **Portal DB** tab.

This displays the Portal Database settings page.

The screenshot shows the 'Portal DB' tab selected in the top navigation bar. Below the navigation bar, there are tabs for 'Logging', 'Get-Resources', 'XSL', and 'E-mail'. The main content area is divided into two sections. The first section, 'Default capabilities:', has a text input field containing 'portalDB(getit.portal)'. To the right of this field is a text box containing the following text: 'Semicolon separated list of default access rights that all users should have regardless of their profile. Access rights are assigned to target adapters in the following way: portalDB(getit.portal)'. The second section, 'Alias for:', has a text input field containing 'ac'. To the right of this field is a text box containing the following text: 'Specifies the target configuration for which this target is an alias. Click for default: []'. At the bottom left of the form is a 'Save' button.

- 3 In the Alias for: field type ac, then click the Save button to return to the Admin Control Panel page.

Web application settings

Get-Resources requires a database adapter connection to store settings and customizations to the web application's interface screens. Until a database adapter is defined for the web application, users cannot make personalization changes to the web application.

Follow these instructions to configure Get-Resources to save personalization to the web application in the AssetCenter back-end database:

- 1 In the left menu pane of the Peregrine Portal Administration page, click **Settings**.
- 2 At the top of the Settings page, click the **Web Application** tab.
This displays the **Web Application** settings page.

- 3 In the **Alias for:** field type `ac`, then click the **Save** button to return to the Admin Control Panel page.

Resetting the server

Follow these instructions to reset the server and apply your configuration changes to Get-Resources:

- 1 In the Admin Control Panel page, click the **Reset Server** button at the bottom of the page.
- 2 When the operation is complete, verify that the **weblication**, **portalDB** and **ac** targets are using the “com.peregrine.oaa.adapter.ac.ACAAdapter” and are “connected” in the Connection Status table.

Troubleshooting the AssetCenter database connection

If you are having trouble making a connection between Get-Resources and the AssetCenter back-end database (i.e. the Connection Status displays “disconnected” for the target adapters in the Admin Control Panel), follow these instruction to verify the AssetCenter Database Settings.

On the AssetCenter settings tab:

- 1 Verify that the parameter defined for the **Database:** field is the same as the database name displayed in the **Connect to database** dialog box when you log into AssetCenter.
- 2 Verify that the correct Admin name and password is defined for Get-Resources to login to AssetCenter.
- 3 Check the ODBC connections to AssetCenter. Depending on the way you run the application server, it looks for either an ODBC User DSN or an ODBC System DSN.
 - If you start your application server as a service (the default method), it references the System DSN for the ODBC connection to the AssetCenter database.
 - If you start your application server as an application , it references the User DSN to determine the ODBC connection.

Refer to [AssetCenter database settings](#) on page 174 for detailed instructions on how to perform these functions.

Instructions for ServiceCenter

Get-Resources connects to the ServiceCenter database through a special adapter, the SCAdapter. This section describes how to configure and verify back-end system settings and connect to the ServiceCenter database.

Connection settings

Follow these instructions to define ServiceCenter as the back-end system for Get-Resources:

- 1 In the left menu pane of the Peregrine Portal Administration page, click **Settings**.

This displays the Settings page which provides access to the various tabs used to configure settings for Get-Resources.

The Common settings page is displayed by default.

Common Backend:
portalDB
Adapter target name used to support common user operations.

List of target aliases:
weblocation;mail
Specifies a list of semicolon delimited target aliases used by web applications in this package.

Admin name:
Admin
The admin login name. This login provides access to administrative functionality without authenticating against backend adapters.

Admin password:

The admin login password.

- 2 At the top of the page, click the **Get-Resources** tab.
This displays the **Connections Settings** page for Get-Resources:

Get-Resources Connection Settings

Get-Resources Target:
sc
The backend for Get-Resources web applications. This should be either AC if AssetCenter is installed or SC if ServiceCenter is installed.

Request Backend:
sc
Backend used to manage requests. (for example, **sc**, or **ac**.) The default is **ac**.

Save

- 3 Update the following fields as follows:

Field Name	Enter Value	Description
Get-Resources Target:	sc	Back-end for Get-Resources web applications.
Request Backend:	sc	Back-end used to manage requests.

- 4 Click the **Save** button to return to the Admin Control Panel page.
- 5 Click the **Reset Server** button at the bottom of the page to apply your changes to the system.
- 6 When the operation is complete, verify that the “sc” target adapter, “com.peregrine.oaa.adapter.sc.SCAdapter”, is displayed in the **Connection Status** table.

Note: Unless the ServiceCenter system you are using as a back-end is configured with default login information, the status for the ServiceCenter target adapter will display “disconnected” until you follow instructions to connect Get-Resources to the ServiceCenter database.

ServiceCenter database settings

Follow these instructions to connect to the ServiceCenter database:

- 1 In the left menu pane of the Admin page, click **Settings**.
- 2 At the top of the Settings page, click the **ServiceCenter** tab.

This displays the ServiceCenter settings page.

Field	Value	Description
Host:	localhost	Host name of the ServiceCenter server
Port:	12670	Port number of the ServiceCenter server
Log:		Path to SC logging used by the ServiceCenter client connection
Admin user:	falcon	Administration user used by the Peregrine Portal when performing tasks such as user authentication and registration in ServiceCenter
Admin password:		Admin user password for ServiceCenter

- 3 Update the following fields as necessary:

Field Name	Value Description	Default Value
Host:	Host name of the ServiceCenter server.	localhost
Port:	Port number of the ServiceCenter server.	12670
Admin user:	Administration user used by the Peregrine Portal when performing tasks such as user authentication and registration in ServiceCenter.	falcon
Admin password:	Admin user password for ServiceCenter	(none)
Anonymous user:	Anonymous user name used when an unknown user attempts to communicate with ServiceCenter	falcon
Anonymous password:	Anonymous user password for ServiceCenter	(none)

- 4 When finished, click the **Save** button at the bottom of the page to return to the Admin Control Panel page.
- 5 Click the **Reset Server** button at the bottom of the page to apply your changes to the system.
- 6 Verify that the Connection Status table displays “connected” for the “sc” target adapter.

Portal database settings

Get-Resources enables personalization of portal application screens without the need to manually change and compile code. To enable this feature, Get-Resources requires a database adapter connection to store settings and customizations to the portal interface in the back-end database. Until a database adapter is defined for the portal page, users will not be able to see or make personalization changes to the Peregrine portal home page.

Follow these instructions to configure Get-Resources to save personalization settings in the ServiceCenter back-end database:

- 1 In the left menu pane of the Peregrine Portal Administration page, click **Settings**.

- 2 At the top of the Settings page, click the **Portal DB** tab.
This displays the Portal Database settings page.

- 3 In the **Alias for:** field type `sc`, then click the **Save** button to return to the Admin Control Panel page.
- 4 Click the **Reset Server** button at the bottom of the page to apply your changes to the system.
- 5 When the operation completes, verify that the adapter used for the **portalDB** target is “com.peregrine.oaa.adapter.sc.SCAdapter” and displays “connected” in the Connection Status table.

Web application settings

Get-Resources requires a database adapter connection to store settings and customizations to the web application’s interface screens. Until a database adapter is defined for the web application, users cannot make personalization changes to the web application.

Follow these instructions to configure Get-Resources to save personalization to the web application in the ServiceCenter back-end database:

- 1 In the left menu pane of the Peregrine Portal Administration page, click **Settings**.
- 2 At the top of the Settings page, click the **Web Application** tab.

This displays the **Web Application** settings page.

Portal AssetCenter Common Portal DB Themes **Web Application** Logging

ServiceCenter Get-Resources XSL E-mail

Default capabilities:

Semicolon separated list of default access rights that all users should have regardless of their profile. Access rights are assigned to target adapters in the following way: portalDB(getit.portal)

Alias for:

Specifies the target configuration for which this target is an alias.
[Click for default: \[\]](#)

Save

- 3 In the **Alias for:** field type `sc`, then click the **Save** button to return to the Admin Control Panel page.
- 4 Click the **Reset Server** button at the bottom of the page to apply your changes to the system.
- 5 When the operation completes, verify that the adapter used for the **weblication** target is “com.peregrine.oaa.adapter.sc.SCAdapter” and displays “connected” in the Connection Status table.

Troubleshooting the ServiceCenter database connection

If the Connection Status table in the Admin Control Panel displays “disconnected” for any of the target adapters, perform the following procedures to troubleshoot possible configuration issues:

In the ServiceCenter settings page of the Admin module in Get-Resources:

- Verify that the parameters entered for the **Host:** and **Port:** fields are correct.
- Verify that the Admin user name and password defined for Get-Resources are the same login values used when logging directly into the ServiceCenter back-end system as an Administrator.

Refer to *ServiceCenter database settings* on page 180 for detailed instructions on how to perform these functions.

In ServiceCenter:

- Verify that the ServiceCenter services are running. If they are not running, access the ServiceCenter Console to start the ServiceCenter services.
- Verify that you have ServiceCenter full client connectivity by starting a client that is pointed to the port listed on the ServiceCenter settings page in the Admin module.

Refer to the *ServiceCenter Administration Guide* for detailed instructions on how to perform these functions.

Index

A

- alternate authentication
 - Unix 85
 - Windows 23
- application servers
 - JRun on Unix 109
 - JRun on Windows 54
 - Tomcat on Unix 87
 - Tomcat on Windows 25
 - WebLogic on Unix 107
 - WebLogic on Windows 50
 - WebSphere on Unix 90
 - WebSphere on Windows 33

B

- book
 - audience 7
 - organization 10

C

- configuring back-end databases
 - migration process, Unix 86
- conventions, typographical 9
- custom installation
 - Unix 127
 - Windows 68
- customer support 10
- customized themes
 - Unix 84
 - Windows 22

D

- Defining 116
- development environment
 - Unix overview steps 80
 - Windows overview steps 18
- documentation, related 8

G

- Get-Resources overview 7

J

- JRun
 - configuring on Unix 109
 - configuring on Windows 54
- JSP file modifications
 - Unix 84
 - Windows 22

M

- migration process
 - alternate authentication, Unix 85
 - alternate authentication, Windows 23
 - configuring back-end databases, Unix 86
 - customized themes, Unix 84
 - customized themes.Windows 22
 - JSP file modifications, Unix 84
 - JSP file modifications, Windows 22
 - personalized pages, Unix 84
 - personalized pages, Windows 22
 - tailoring kit changes, Unix 85
 - tailoring kit changes, Windows 23

- Unix 83
 - Windows 21
- multiple Tomcat instances, installing 31, 148, 157
 - configuring on Unix 90
 - configuring on Windows 33
 - workers.properties file, editing 143

P

- Peregrine Systems customer support 10
- personalized pages
 - Unix 84
 - Windows 22
- Procedure
 - Apply unload files to ServiceCenter 168
- production environment
 - Unix overview steps 82
 - Windows overview steps 20

R

- related documentation 8

S

- server.xml files, editing 29, 146, 155

T

- tailoring kit changes
 - Unix 85
 - Windows 23
- technical support 10
- terminology 8
- Tomcat application server
 - Unix 87
 - Windows 25
- Tomcat instances, installing multiple 31, 148, 157
- typical installation
 - Unix 118
 - Windows 62
- typographical conventions 9

U

- uninstall 77
 - Unix 138

W

- WebLogic
 - configuring on Unix 107
 - configuring on Windows 50
- WebSphere

