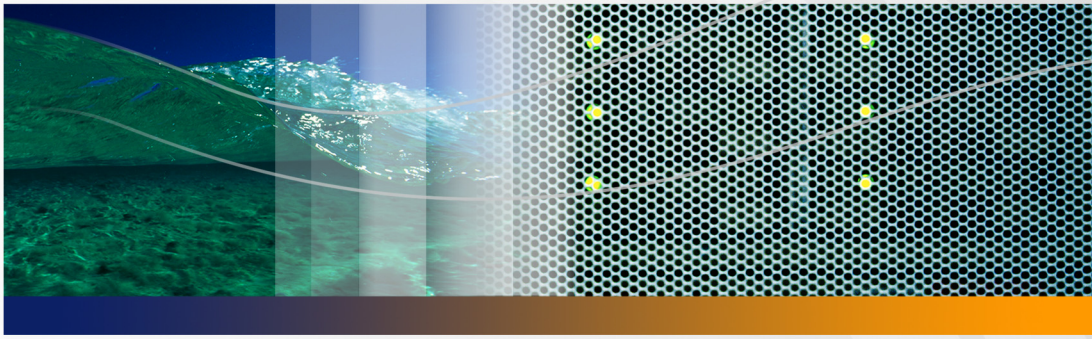
Product Documentation

Peregrine Systems, Inc.

Get-Services™ 4.2



Installation Guide

Part No. DGS-42-EN28

Peregrine SYSTEMS

Peregrine Systems, Inc.
3611 Valley Centre Drive San Diego, CA 92130
858.481.5000
Fax 858.481.1751
www.peregrine.com

PEREGRINE

Contents

PEREGRINE

About this Guide

Get-Services is an application that provides a web-based interface to Peregrine ServiceCenter®. Get-Services enables users to report problems in their work environment by opening incident tickets in the appropriate back-end system.

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

- Install the Peregrine OAA Platform and Get-Services.
- Configure Get-Services for ServiceCenter.

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

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

Book audience

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

- XML and ECMAScript or JScript/JavaScript (if you use the tailoring kit)
- Operating guides, reference manuals, and other documentation for your PC hardware and operating system
- Web server and Application server administration
- ServiceCenter administration and functionality

Related documentation

Refer to the following documentation for additional information.

Document	Description
Get-Services Administration Guide	This describes the Peregrine OAA platform and Get-Services administration.
Get-Services Release Notes	This covers any last-minute documentation or known issues with Get-Services. These documents are constantly updated and posted to the Customer Support web site. See Need further assistance? on page 13 for details on accessing the Customer Support website.

Associated applications

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

Note: ServiceCenter must be installed and configured before you can install and configure Get-Services. Peregrine OAA installs with Get-Services, and only the installation of Peregrine OAA for Get-Services is included in this guide.

Terminology

The terminology used in this guide and in the Get-Services interface is based on ServiceCenter 5.1.x and ServiceCenter 6.0.

Typographical conventions

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

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**.
Italics	Variables and values that you must provide appear in *italics*. New terms also appear in *italics*.

`Monospace`	Code or script examples, output, and system messages appear in a monospace font.
	`var msgTicket = new Message( "Problem" );`
	…
	`msgTicket.set( "_event", "epmc" );`
	
	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.
	Filenames, such as `login.asp`, appear in a monospace 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
Chapter 1, Introduction	Installation requirements and description of different types of installations for Get-Services.
Chapter 2, Installing	Detailed instructions for installing Get-Services in a Windows environment.
Chapter 3, Upgrading Get-Services on Application Servers	Detailed instructions for installing Get-Services in a Unix environment.
Chapter 4, Load Balancing the Servers	Information on load balancing the application server for optimum system memory management.
Chapter 5, Back-end System Configuration	Detailed instructions for configuring ServiceCenter or AssetCenter as a back-end system with Get-Services.
Chapter 6, Get-Services Configuration	Final configuration settings within the Peregrine Portal Administration module to complete the installation of Get-Services.
Chapter 7, Troubleshooting	Troubleshooting installation problems with Apache Web server, Tomcat, OAA, and AssetCenter and ServiceCenter.
Appendix A, Copyright Notices	Additional licensing information.

Need further assistance?

For further information and assistance with this release, you can download documentation or schedule training.

Customer Support

For further information and assistance, contact Peregrine Systems' Customer Support at the Peregrine CenterPoint Web site.

To contact customer support:

1. In a browser, navigate to *http://support.peregrine.com*
2. Log in with your user name and password.
3. Follow the directions on the site to find your answer. The first place to search is the KnowledgeBase, which contains informational articles about all categories of Peregrine products.
4. If the KnowledgeBase does not contain an article that addresses your concerns, you can search for information by product; search discussion forums; and search for product downloads.

Documentation Web site

For a complete listing of current Get-Services documentation, see the Documentation pages on the Peregrine Customer Support Web.

To view the document listing:

1. In a browser, navigate to *http://support.peregrine.com*.
2. Log in with your login user name and password.
3. Click either Documentation or Release Notes at the top of the page.
4. Click the Get-Services link.

5 Click a product version link to display a list of documents that are available for that version of Get-Services.

6 Documents may be available in multiple languages. Click the Download button to download the PDF file in the language you prefer.

You can view PDF files using Acrobat Reader, which is available on the Customer Support Web site and through Adobe at *http://www.adobe.com*.

Important: Release Notes for this product are continually updated after each release of the product. Ensure that you have the most current version of the Release Notes.

Education Services Web site

Peregrine Systems offers classroom training anywhere in the world, as well as "at-your-desk" training using the Internet. For a complete listing of Peregrine's training courses, refer to the following web site:

http://www.peregrine.com/education

You can also call Peregrine Education Services at +1 858.794.5009.

1 Introduction

CHAPTER

This chapter covers the following topics for Get-Services:

Installation requirements

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

Component	for Windows	for Unix
System Processor	Pentium, 400 MHz or faster	Linux: Pentium, 400 MHz or faster AIX: POWER 3, 375 MHz or faster Solaris: Ultra SPARC II, 300 MHz or faster
RAM	512 MB or more	512 MB or more
Hard disk space	100 MB for Get-Services and 200 MB for MSI files	100 MB for Get-Services

For information about software compatibility with Get-Services, see the Peregrine Systems Customer Support site at *http:\\support.peregrine.com*. Click

Documentation > Get-Services > Compatibility Matrices and click the link to your version of Get-Services.

Types of installations

You can optimize the Get-Services custom installation for two types of environments:

- Development environment
- Production environment

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

Tip: Install the same Web server and application server in the development environment that you plan to use in the production environment.

A *production environment* installation is optimized for performance and scalability. Each component, such as an application server and Web server, resides on different servers. In addition, there may be multiple instances of any component. Given the amount of flexibility involved in such an installation, users must manually set up a production environment.

Deploying multiple Peregrine Portal applications

When multiple Peregrine Portal applications, including Get-Services, are deployed, all must be applications that use the same version of OAA. Multiple Peregrine Portal applications that use different versions of OAA are not supported.

Further, when you upgrade from a version prior to 4.1, you first upgrade all Peregrine Portal applications to version 4.1. Then you upgrade all Peregrine Portal applications to version 4.2.

Back-end systems

Important: In order to use Get-Services, you must have a properly configured back-end system.

Get-Services uses ServiceCenter as its back-end database. Get-Services uses the back-end system to:

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

Refer to the compatibility matrix on the Customer Support Web site for a complete list of the versions compatible with Get-Services.

CHAPTER 2 Installing

This chapter explains the installation and configuration of Get-Services on the OAA platform on a Windows or UNIX system using the application servers listed in the Get-Services Compatibility Matrix. These instructions are for a new installation of Get-Services. If you are upgrading Get-Services from a previous version, see Chapter 3.

The installation program no longer deploys files directly into an application server's directory structure. Instead, files are deployed into a *staging area* on the file system. This staging area functions as a common place into which one or more Peregrine Portal applications are deployed. From these deployed files, the installation program builds a WAR file and terminates.

Important: After the installation program executes, you must perform additional steps to complete the installation process.

With the WAR file, you must use the application server's deployment process to install the Peregrine Portal application. Once installed and running in the application server, an administrator can then access the Get-Services Administration page to complete the setup process.

This WAR file distribution complies with the J2EE application deployment process and enables one WAR file, containing the Peregrine Portal application(s), to be conveniently deployed into separate test and production environments.

Before you install the OAA platform on the Web server, see the recommended configurations in the Get-Services Compatibility Matrix, available at the Peregrine Systems Customer Support site.

To view the compatibility matrices:

1 In a browser, navigate to *http://support.peregrine.com*

2 Log in with your user name and password.

3 Click **Documentation** > **Get-Services** > **Compatibility Matrices**.

Warning: Before you begin the installation process, close all anti-virus software programs.

This section describes configurations required for the application servers and Web servers to support Peregrine's Open Application Architecture (OAA) and the Get-It applications.

Application servers	Web servers
Tomcat	Apache
WebSphere Application Server	IBM HTTP Server
WebLogic	IIS
	iPlanet
	SunONE

Note: The following configuration information assumes that the server or servers are installed and the services are enabled and running.

Pre-installation overview

Before installing any of the Get-It 4.2 applications, make note of the following changes.

Note: This section provides an overview only of the configuration changes described in this chapter. Follow the detailed instructions for all of the configurations that Peregrine Systems supports.

- Use JDK 1.4.2 with OAA 4.2 applications when the application is not being deployed to an application server that supplies its own JDK. This simplifies the installation process.
- When using JDK 1.4.2 with any OAA 4.2 application, it is no longer necessary to copy any files to `$JAVA_HOME/jre/lib/ext`.

Note: Placing files in `JRE/lib/ext` with JDK 1.4 is not appropriate when the standard extension to 1.3 has been integrated into 1.4, as is the case of JAAS, JCE, JSSE and JAXP. To override the built-in behavior of the 1.4 VM, you must use the Endorsed Standards Override Mechanism located at *http://java.sun.com/j2se/1.4.2/docs/guide/standards/index.html*. This is required for the Xerces XML parser to be used, but Tomcat already handles this case in its startup script. While JDK 1.4 ships with Xalan, the XML parser included is Crimson, not Xerces.

- When you use a Get-It 4.2 application with JDK 1.4.2 and Tomcat, there is no need to place any files in `JRE/lib/ext` or to directly make use of the Endorsed Standards Override Mechanism. The only files formerly installed in `jre/lib/ext` that are still required are:

  ```
  js.jar
  jai_core.jar
  jai_codec.jar
  mlibwrapper_jai.jar
  ```

  Install these in `WEB-INF/lib`. The Xerces and Xalan supplied with Tomcat are sufficient, as are the versions of `mail.jar` and `activation.jar`. The `oaasecurityproxy.jar` file is no longer needed since the bug that required this work-around was resolved when JAAS was integrated into JDK 1.4. Detailed instructions for doing this are in the sections that follow.

- When using WebSphere with a Get-It 4.2 application, ensure that the Web application server classloading is configured to favor classes under `WEB-INF` over those visible to parent classloaders. This is configured in the WAS administration tool. Use the *parent last* setting. This setting is automatically configured when using WebLogic with the included `weblogic.xml` file.
- When using WebSphere 4 or 5, it is no longer necessary to add the following parameters to the `local.xml` file.

  ```
  <SSLProvider>com.ibm.jsse.JSSEProvider</SSLProvider>
  <HTTPSHandlerPkg>com.ibm.net.ssl.internal.www.protocol</HTTPSHandlerPkg>
  <CryptoProvider>com.ibm.crypto.provider.IBMJCE</CryptoProvider>
  ```

  While these options are still available, you no longer need to manually set them on any of the Peregrine Systems supported platforms for Get-It 4.2 applications.

- When you are using Tomcat with JDK 1.4, do NOT use the LE version of Tomcat. From the Tomcat FAQ at *http://jakarta.apache.org/tomcat/faq/misc.html*

  Should I use the LE version?
  No. It was an experiment, it failed. (YMMV) The original purpose of LE was because jdk1.4 provides a LOT of standard functionality that Tomcat uses that was not in jdk1.3. So to save some space two distributions were made to save some bandwidth. Since then, it has been an exercise in confusion...

  Also, you need to place the `xalan.jar` in the Tomcat `/endorsed` directory.

- When using an application server other than Tomcat with JDK 1.4, you may have to add both Xerces and Xalan using the Endorsed Standards Override Mechanism.
- If you are using WebSphere Application Server (WAS) 5.1, update to WAS 5.1.1 or later.

Configuring the servers to run Get-Services

You must configure your application and Web servers *prior* to running the Get-Services installer. This section describes how to configure some of the common application and Web server combinations with Get-Services.

Tomcat and Apache servers

These instructions describe how to configure Tomcat 4.1.x as an application server and Apache as a Web server, with no other applications running on this configuration. Tomcat 5.0.x uses the same procedures.

To configure Tomcat and Apache:

Step 1 Download and install the Java Software Development Kit and the servers. See Downloading the software on page 24.

Step 2 Download the Web server connectors for Apache. See Downloading the Web server connectors for Apache on page 24.

Step 3 Run the installer. See Running the installer on page 24.

Step 4 Copy the `oaa.war` file. See Copying the oaa.war file on page 25.

Step 5 Start the application server. See Starting the Tomcat application server on page 25.

Step 6 Copy the jar files. See Copying the jar files on page 25.

Step 7 Edit the `httpd.conf`, `mod_jk2.conf` and `workers2.properties` files. See Editing the connector files on page 26.

Step 8 Start the Web server. See Starting the Apache Web server on page 28.

Step 9 Test the configuration. See Testing the configuration on page 29.

Downloading the software

Download and install the Java Software Development Kit, Tomcat application server, and Apache Web server.

Download	Web site
Java J2SE v 1.4.x Software Development Kit (SDK)	*http://java.sun.com*
Tomcat 4.1.x application server or Tomcat 5.0.x	*http://jakarta.apache.org*
Apache Web server	*http://httpd.apache.org*

Downloading the Web server connectors for Apache

To properly configure the servers, you must download the Tomcat Web server connectors (JK2) for Apache and extract the files.

1 Download the following compressed Tomcat Web server connectors (JK2) file for Apache from *http://jakarta.apache.org*:

```
(jakarta-tomcat-connectors-jk2.0.4-[operating system]-apache2.x.x.zip)
```

2 Extract `mod_jk2.so` to your `Apache2\modules` directory.

3 Extract `mod_jk2.conf.sample` to your `Apache2\conf` directory.

   a Copy it to the same directory.

   b Rename it to `mod_jk2.conf`.

4 Extract `workers2.properties.sample` to your `Apache2\conf` directory.

   a Copy it to the same directory.

   b Rename it to `workers2.properties`.

Running the installer

Follow the instructions in Running the installer on page 83 to install Get-Services on a Tomcat application server.

Note: Make sure that you stop Tomcat and Apache before installing Get-Services.

Copying the oaa.war file

The installer creates the oaa.war file that you need to deploy the application.

- Copy the oaa.war file that the installer created from the Portal directory to the <tomcat_home>\webapps directory, where <tomcat_home> is the path to your Tomcat application server.

  The default Tomcat path is: C:\Program Files\Apache Group\Tomcat 4.1.

Starting the Tomcat application server

When you start the Tomcat application server, the application automatically deploys, creating an oaa directory under the webapps directory. The default path is: C:\Program Files\Apache Group\Tomcat 4.1\webapps\oaa.

Once the application finishes deploying, stop the Tomcat application server.

Copying the jar files

The installer generates jar files that you must copy to the application server endorsed directory.

1 Copy the following files from the installation directory C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions

```
xalan.jar
xercesImpl.jar
xml-apis.jar
```

where C:\Program Files\Peregrine\Portal specifies the location of your application installation,

to the <tomcat_home>\common\endorsed directory.

Note: Create the endorsed directory if one does not exist.

2 If necessary, replace any older versions of these files.

3 If `xmlParserAPIs.jar` exists in the `endorsed` directory, delete it.

4 For Tomcat 5.0.x, copy the following files from the installation directory `C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions`

```
activation.jar
mail.jar
pop3.jar
```

to the `<tomcat_home>\webapps\oaa\WEB-INF\lib` directory.

Editing the connector files

You must edit the `httpd.conf`, `mod_jk2.conf`, and `workers2.properties` files to properly configure the servers.

1 From the `Apache\conf` directory, using a text editor, open `httpd.conf`.

a Add the following code to the end of the file:

```
include “C:\Program Files\Apache Group\Apache2\conf\mod_jk2.conf”
```

where `C:\Program Files\Apache Group\Apache2\conf\mod_jk2.conf` specifies the location of the `mod_jk2.conf` file on your Apache Web server.

b Save and close the file.

2 From the `Apache2\conf` directory, using a text editor, open `mod_jk2.conf`.

a Delete any contents that may already exist in this file.

b Add the following code to instruct Apache to load the jk2 module:

```
LoadModule jk2_module modules\mod_jk2.so
```

c Add the following code to instruct Apache to use the `workers2.properties` file for configuration:

```
JkSet config.file "C:\Program Files\Apache Group\Apache2
\conf\workers2.properties"
```

where `C:\Program Files\Apache Group\Apache2\conf\workers2.properties` specifies the location of the `workers2.properties` file on your Web server.

d Add the following code to make Apache aware of the location of the `\oaa` context:

```
Alias /oaa "C:/Program Files/Apache Group/Tomcat 4.1/webapps/oaa"
```

where `C:/Program Files/Apache Group/Tomcat 4.1/webapps/oaa` specifies the location of the `webapps/oaa` directory on your Apache Tomcat application server.

e Add the following entries to set access to the specified content in the `/oaa` directory.

```
<Directory "C:/Program Files/Apache Group/Tomcat 4.1/webapps/oaa">
    AllowOverride None
    Options None
    Order allow,deny
    Allow from all
</Directory>
<Location "/oaa/WEB-INF/">
    AllowOverride None
    deny from all
</Location>

# Use Directory attribute also. Location doesn't work unless case
# matches.
<Directory "C:/Program Files/Apache Group/Tomcat 4.1
  /webapps/oaa/WEB-INF/">
    AllowOverride None
    deny from all
</Directory>

# The following line prohibits users from directly accessing
META-INF
<Location "/oaa/META-INF/">
    AllowOverride None
    deny from all
</Location>

# Use Directory attribute also. Location doesn't work unless case
# matches.
<Directory "C:/Program Files/Apache Group/Tomcat 4.1
  /webapps/oaa/META-INF/">
    AllowOverride None
    deny from all
</Directory>
```

Note: You must update the path specified in the examples with the path that matches your Tomcat installation:
`(C:/Program Files/Apache Group/Tomcat/…).`

f Save and close the file.

3 From the `Apache2\conf` directory, using a text editor, open the `workers2.properties`.

a Add the following lines of code:

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

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

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

[uri:/oaa/answers/attachments/*]
info=Prefix mapping

[uri:/oaa/attachments/*]
info=Attachment mapping
```

b Save and close the file.

Starting the Apache Web server

To activate the new configurations, start the Apache Web server.

Testing the configuration

After the application and Web servers are properly configured, log on to the Get-Services Admin page to configure the settings that are stored in the `local.xml` file.

1 To verify that the Tomcat and Apache servers are properly configured, check the following URLs.

- *http://<server>:80 for the Apache Web server*
- *http://<server>:8080 for the Tomcat application server*

Make sure that the appropriate Tomcat and Apache services or applications are started before proceeding.

2 Follow the steps described in Configuring Get-Services on page 86 to complete the Get-Services configuration.

Tomcat and IIS servers

The following instructions configure Tomcat 4.1.x to connect to an IIS 5.0 Web server and Tomcat 5.0.x to connect to an IIS 5.0 or IIS 6.0 Web server. These instructions set up Tomcat to use a single Java Virtual Machine (JVM). See the Load-Balancing Application Servers chapter in this guide for information about installing multiple JVMs.

To configure Tomcat and IIS:

Step 1 Download and install the Java Software Development Kit and the Tomcat application server. See Downloading the software on page 30.

Step 2 Download the Tomcat-IIS JK2 Web server connector zip file. See Downloading the Web server connector for IIS on page 30.

Step 3 Reboot your system. See Rebooting the system on page 31.

Step 4 Run the installer. See Running the installer on page 31.

Step 5 Copy the `oaa.war` file. See Copying the oaa.war file on page 32.

Step 6 Start the application server. See Starting the Tomcat application server on page 32.

Step 7 Copy the `jar` files. See Copying the jar files on page 32.

Step 8 Copy the `jk2.reg` file. See Copying the jk2.reg file on page 33.

Step 9 Configure the ISAPI Plug-in for IIS. See Configuring the ISAPI Plug-in for IIS on page 33.

Step 10 Configure IIS to use `isapi_redirector2.dll` as an ISAPI Filter. See Configuring the isapi_redirector2.dll as an ISAPI filter on page 35.

Step 11 Create and configure a `jakarta` virtual directory in IIS. See Configuring a jakarta virtual directory in IIS on page 36.

Step 12 Create and configure an `oaa` virtual directory in IIS. See Configuring an oaa virtual directory in IIS on page 36.

Step 13 Edit the `server.xml` file to add performance settings and configure alternate communications ports (Optional). See Editing the server.xml file for IIS on page 37.

Step 14 Install Tomcat as a service using `installservice.bat` (Optional). This file can be found in the `<tomcat_home>\bin` directory. See Installing Tomcat as a service on page 38.

Step 15 Test the configuration. See Testing the configuration on page 40.

Downloading the software

Download and install the Java Software Development Kit and Tomcat application server.

Download	Web site
Java J2SE v 1.4.x Software Development Kit (SDK)	*http://java.sun.com*
Tomcat 4.1.x or 5. 0.x application server	*http://jakarta.apache.org*

Downloading the Web server connector for IIS

To properly configure the servers, you must download the Tomcat Web server connector (JK2) for IIS and extract the files.

1 Download the following compressed Tomcat Web server connector (JK2) file for IIS from *http://jakarta.apache.org*:

`(jakarta-tomcat-connectors-jk2.0.4-[operating system]-IIS.zip)`

2 Extract `workers2.properties.sample` to your `<tomcat_home>\conf` directory.

a Copy it to the same directory.

b Rename it to `workers2.properties`.

c Using a text editor, add the following lines of code to the end of file:

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

[uri:/oaa/*.do]
info=Extension mapping
group=lb

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

[uri:/oaa/answers/attachments/*]
info=Prefix mapping
group=lb

[uri:/oaa/attachments/*]
info=Attachment mapping
group=lb
```

3 Extract `isapi_redirector2.dll` to your `<tomcat_home>\bin` directory.

Rebooting the system

Before continuing with the configuration, reboot your system.

Running the installer

Follow the instructions in Running the installer on page 83 to install Get-Services on a Tomcat application server.

Note: Make sure that you stop Tomcat and IIS before installing Get-Services.

Copying the oaa.war file

The installer creates the oaa.war file that you need to deploy the application.

- Copy the oaa.war file that the installer created from the Portal directory to the <tomcat_home>\webapps directory, where <tomcat_home> is the path to your Tomcat application server.

  The default Tomcat path is: C:\Program Files\Apache Group\Tomcat.

Starting the Tomcat application server

When you start the Tomcat application server, the application automatically deploys, creating an oaa directory under the webapps directory. The default path is: C:\Program Files\Apache Group\Tomcat\webapps\oaa.

Once the application finishes deploying, stop the Tomcat application server.

Copying the jar files

The installer generates jar files that you must copy to the application server endorsed directory.

1 Copy the following files from the installation directory C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions

```
xalan.jar
xercesImpl.jar
xml-apis.jar
```

where C:\Program Files\Peregrine\Portal specifies the location of your application installation,

to the <tomcat_home>\conf directory.

Note: Create the endorsed directory if one does not exist.

2 If necessary, replace any older versions of these files.

3 If `xmlParserAPIs.jar` exists in the `endorsed` directory, delete it.

4 For Tomcat 5.0.x, copy the following files from the installation directory `C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions`

```
activation.jar
mail.jar
pop3.jar
```

to the `<tomcat_home>\webapps\oaa\WEB-INF\lib` directory.

Copying the jk2.reg file

The `jk2.reg` file creates a Windows registry entry.

Copy the `jk2.reg` file from the installation directory `C:\Program Files\Peregrine\Portal\SupportFiles\AppServerFiles` directory to the `<tomcat_home>\common\endorsed` directory,

where `C:\Program Files\Peregrine\Portal` specifies the location of your application installation and where `<tomcat_home>` is the path to your Tomcat application server.

The default Tomcat 4.1 path is: `C:\Program Files\Apache Group\Tomcat 4.1`.

The default Tomcat 5.0.x path is:
`C:\Program Files\Apache Software Foundation\Tomcat 5.0`.

Configuring the ISAPI Plug-in for IIS

The ISAPI plug-in 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.

Use the following procedures to configure the plug-in for your intranet environment.

To configure the ISAPI plug-in for IIS 5.0 or IIS 6.0:

1 Open the file `jk2.reg` in a text editor. The file path for Tomcat 4.1.x is:

```
C:\Program Files\Apache Group\Tomcat 4.1\conf
```

The file path for Tomcat 5.0.x is:

```
C:\Program Files\Apache Software Foundation\Tomcat 5.0\conf
```

2 Verify that the `ServerRoot` and `workersFile` entries list the proper installation path to Tomcat.

- For Tomcat 4.1.x, the default values are:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Group\Jakarta Isapi Redirector\2.0]
“ServerRoot”=”C:\\Program Files\\Apache Group\\Tomcat 4.1”
“workersFile”=”C:\\Program Files\\Apache Group\\Tomcat 4.1\\conf\\
workers2.properties”
```

- For Tomcat 5.0.x, the default values are:

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Group\Jakarta Isapi Redirector\2.0]
“ServerRoot”=”C:\\Program Files\\Apache Software Foundation\\Tomcat 5.0”
“workersFile”=”C:\\Program Files\\Apache Software Foundation
\\Tomcat 5.0\\conf\\
workers2.properties”
```

3 Verify that the HKEY_LOCAL_MACHINE path is correct for your Tomcat version.

- For Tomcat 4.1.x, modify each version of Tomcat as follows:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Group\Jakarta Isapi Redirector\2.0]
"serverRoot"="C:\\Program Files\\Apache Group\\Tomcat 4.1"
"extensionUri"="/jakarta/isapi_redirector2.dll"
"workersFile"="C:\\Program Files\\Apache Group
\\Tomcat 4.1\\conf\\workers2.properties"
```

- For Tomcat 5.0.x, modify each version of Tomcat as follows:

```
Windows Registry Editor Version 5.00

[HKEY_LOCAL_MACHINE\SOFTWARE\Apache Software Foundation\Jakarta Isapi
Redirector\2.0]
"serverRoot"="C:\\Program Files\\Apache Software Foundation\\Tomcat 5.0"
"extensionUri"="/jakarta/isapi_redirector2.dll"
"workersFile"="C:\\Program Files\\Apache Software Foundation
\\Tomcat 5.0\\conf\\workers2.properties"
```

4 Save and close the jk2.reg file.

5 Right-click and select MERGE or double-click 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 must install isapi_redirector2.dll as an ISAPI filter.

To install isapi_redirector2.dll as an ISAPI filter:

1 From Windows Control Panel > Administrative Tools, 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:

Field	Value	Comments
Filter Name	`jakarta`	The filter name must match the name you defined in the `jk2.reg` registry file. By default, the filter name is `jakarta`.

Executable	`isapi_redirector2.dll`	The file path is:
		`C:\<tomcat_home>\bin\`
		`isapi_redirector2.dll`

6 Click **OK**.

Note: Stop and restart the IIS service for changes to take effect. Also, restart your Tomcat service.

7 From the Internet Services management console, right-click the **Default Web Site** node, then select **Properties** > **Isapi Filters** again.

The ISAPI filter in IIS displays a green status arrow to indicate that it is running.

Configuring a jakarta virtual directory in IIS

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

To configure a jakarta virtual directory in IIS:

Use the following guidelines to create the virtual directory on the Default Web Site.

Requirement	Setting
Create virtual directory	`jakarta`
Map to physical path	`<tomcat_home>\bin`
Directory access rights	`Read, Run scripts, Execute`

Configuring an oaa virtual directory in IIS

To run Get-Services from IIS, you need to create a virtual directory that maps to your Tomcat deployment folder. For specific instructions about configuring IIS, refer to your Windows Help.

To configure an oaa virtual directory in IIS:

Use the following guidelines to create the virtual directory.

Requirement	Setting
Create virtual directory	`oaa`
Map to physical path	`<tomcat_home>\webapps\oaa`
Directory access rights	`Read, Run scripts`

Editing the server.xml file for IIS

A default Tomcat installation is sufficient for most Get-Services installations. However, if you are experiencing performance problems or communications port conflicts, you may need to edit the Tomcat `server.xml` file to correct these problems.

Performance settings

The Tomcat `server.xml` file allows you to determine how Tomcat processes Get-Services files. If you are experiencing performance problems, you can change the `<Context>` setting for Get-Services to disable page reloading.

Tip: Make a back up copy of the `server.xml` file before editing.

To edit the server.xml performance settings:

1 Open the file server.xml in any text editor. The default file path is:

```
C:\<tomcat_home>\conf
```

2 Create a `<Context>` element entry from Tomcat to the Get-Services deployment directory to establish a point of reference for docBase.

- For Tomcat 4.1.x, add the entry just above the **examples** Context entry.

Example:

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

- For Tomcat 5.0.x, see the code creating a `<Context>` element on page 15 of Editing the server.xml files.

Setting the `reloadable` attribute to `false` results in faster JSP page processing.

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

Installing Tomcat as a service

After you edit the Tomcat files, you can install Tomcat as Windows services using the `installservice.bat` file. See Installing Tomcat instances as services on page 17 for more information.

This completes the procedure for IIS 5.0.

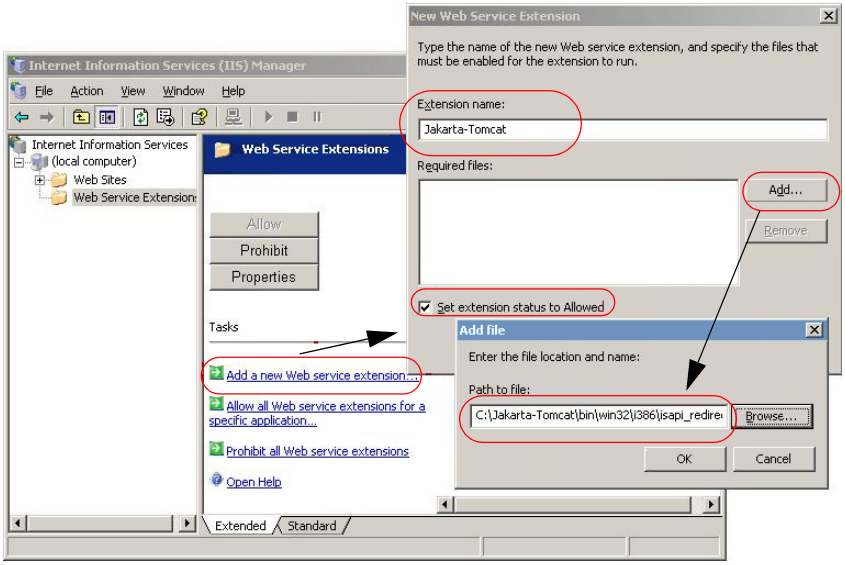
To configure the ISAPI plug-in connection between Tomcat and IIS 6.0, complete the following:

1 Set the ISAPI Redirector Filter DLL in the Web Service Extensions to **Allowed**.

Note: This explicitly allows the redirector DLL to work within IIS. The enhanced security in IIS 6.0 includes **Web Service Extensions**. The redirector DLL, added as an ISAPI Filter, needs to be added to this area of IIS and set as **Allowed**.

2 Go to the management console.

3 Click **Web Services Extensions**.



4 Choose **Add a new Web service extension**.

5 Enter an extension name (for example, Jakarta-Tomcat), and then select the **Set extension status to Allowed** check box.

6 Click **Add**.

7 Enter the path to `isapi_redirector2.dll` and click **OK**.

Testing the configuration

After the application and Web servers are properly configured, log on to the Get-Services Admin page to configure the settings that are stored in the `local.xml` file.

1 To verify that the Tomcat and IIS servers are properly configured, check the following URLs.

- *http://<server>:80 for the IIS Web server*
- *http://<server>:8080 for the Tomcat application server*

Make sure that the appropriate Tomcat and IIS services or applications are started before proceeding.

2 Follow the steps described in Configuring Get-Services on page 86 to complete the Get-Services configuration.

WebLogic 6.1 SP4 and IIS

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

To configure WebLogic 6.1 SP4 with IIS:

Step 1 Stop both WebLogic and your Web server. See Stopping the servers on page 41.

Step 2 Edit the `startWebLogic.cmd` file to set the system password, memory settings, and start mode. See Editing startWebLogic.cmd on page 41.

Step 3 Edit the `Server.Policy` file to set the debug to true. See Editing the Server.Policy file on page 43.

Step 4 Run the Get-Services installer. See Running the installer on page 43.

Step 5 Move `jar` files to the Java development kit `ext` directory. See Moving jar files to the Java Development Kit ext directory on page 43.

Step 6 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 44.

Step 7 Configure IIS to use `iisproxy.dll` as an extension. See Configuring the iisproxy.dll as an extension on page 45.

Step 8 Copy the installed files to your WebLogic directory. See Copying the installed files to your WebLogic directory on page 46.

Step 9 Create a virtual directory for Get-Services in your Web server. See Creating a virtual directory for Get-Services on page 47.

Step 10 Restart WebLogic and your Web server. See Restarting the servers on page 48.

Step 11 Configure Get-Services. See Configuring Get-Services on page 48.

Stopping the servers

Before you begin configuring WebLogic, you must close your WebLogic server and Web server.

To begin configuring WebLogic:

1 Stop the WebLogic application server.

2 Stop the Web server.

Editing startWebLogic.cmd

To edit startWebLogic.cmd:

1 Using a text editor, open the `startWebLogic.cmd` file. The default file path is:

```
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.
set WLS_PW=password
```

3 In the last line, change the word `password` to your WebLogic system password.

4 Search for the `-ms` parameter and set it to `256m` or greater.

5 Search for the `-mx` parameter setting in the file. The recommended setting is `512m`.

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.

6 Set the `STARTMODE` variable to `STARTMODE=false`.

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

7 Add the following phrase to the entry that precedes the `weblogic.Server` entry. Keep the quotation marks.

`"-Djava.security.auth.login.config==<WebLogic>\lib\server.policy"`

Where `<WebLogic>` is the installation path for WebLogic. The default path is: `c:\bea\wlserver6.1`

8 Save the file.

Editing the Server.Policy file

To edit Server.Policy:

1 Using a text editor, open the `Server.Policy` file. The default file path is: `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 and close the file.

Running the installer

Follow the instructions in Running the installer on page 83 to install Get-Services on a WebLogic 6.1 application server.

Moving jar files to the Java Development Kit ext directory

The installer generates `jar` files that you must copy, or FTP, to the Java Development Kit `ext` directory.

To move jar files:

1 Verify that the following directory exists. If it does not, create it: `c:\bea\jdk131\jre\lib\ext`

2 Go to `<install_dir>\Portal\image\WEB-INF\lib` where `<install_dir>` is the location where you installed Get-Services. The default location is `C:\Program Files\Peregrine\Portal\image\WEB-INF\lib`.

   a Move the `log4j-1.2.6.jar` file to `\bea\jdk131\jre\lib\ext`.

   b Copy the following files to `\bea\jdk131\jre\lib\ext`.

```
jai_codec.jar
jai_core.jar
mlibwrapper_jai.jar
```

3 Go to the <install_dir>\Portal\SupportFiles\JavaExtensions directory and copy the following files to the \bea\jdk131\jre\lib\ext directory.

```
jaas.jar
jce1_2_2.jar
jcert.jar
jnet.jar
jsse.jar
ocal_policy.jar
oaasecurityproxy.jar
sunjce_provider.jar
US_export_policy.jar
xalan.jar
xercesImpl.jar
xml-apis.jar
```

4 Copy the following files from the installation directory C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions

```
activation.jar
mail.jar
pop3.jar
```

to the <APP_DEPLOYMENT_DIR>\WEB-INF\lib directory.

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

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

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

1 Open the Internet Services management console.

2 Right-click the <Machine_name> and then click **Properties**.

Note: This is not a Web site; it is the parent node in the tree for the Web sites.

3 Click **Edit** from the Master Properties pane.

4 Click the **ISAPI Filters** tab.

5 Click **Add**.

6 Enter the following information.

Filter Name	`iisforward`

Executable	`iisforward.dll`
	The default file path is:
	`c:\bea\wlserver6.1\bin\iisforward.dll`

7 Click **OK**.

8 Click the **Home Directory** tab.

9 Click **Configuration**.

The Application Configuration page opens on the App Mappings tab.

10 Verify that there is a mapping for the `.wlforward` extension. If it does not exist, click **Add** to map the `.wlforward` extension.

11 Enter the following information.

Executable	`iisforward.dll`
	The default file path is:
	`c:\bea\wlserver6.1\bin\iisforward.dll`

Extension	`.wlforward`

12 Close the Internet Services management console.

Configuring the iisproxy.dll as an extension

To establish a connection between WebLogic and IIS, you install the file `iisproxy.dll` as an extension.

To install iisproxy.dll as an extension:

1 Open Internet Services management console.

2 Right-click the **Default Web Site** node and then click **Properties**.

3 Click the **Home Directory** tab.

   a Select the **Read** check box to enable it.

   b From the **Execute Permissions** drop-down list, select **Scripts and Executables**.

4 Click **Configuration**.

   The Application Configuration page opens on the App Mappings tab.

5 Verify that there is a mapping for the `.jsp` extension and that it is mapped to `c:\bea\wlserver6.1\bin\iisproxy.dll`.

   a If the `.jsp` extension mapping is not there, click **Add**.

   b Enter the following information:

Executable	`iisproxy.dll`
	The default file path is:
	`c:\bea\wlserver6.1\bin\iisproxy.dll`

Extension	`.jsp`

Note: Ensure that the `iisproxy.dll` is located in the same directory as the `iisforward.dll`.

6 Close the Internet Services management console.

Copying the installed files to your WebLogic directory

The installer generates files that you must copy, or FTP, to your WebLogic application directory.

To copy the installed files:

1 Copy the Portal\image directory (the default path is C:\Program Files\Peregrine\Portal\image) to the <WebLogic>\applications directory.

The WebLogic default path is: C:\bea\weblogic6.1\config\<my-domain>

2 Rename <WebLogic>\applications\image directory to <WebLogic>\applications\oaa.

Tip: If you need to FTP files from one server to another to perform the copy, it is more convenient to FTP the oaa.war file in <Portal>\oaa.war to your WebLogic server than to extract the oaa.war files into a <WebLogic>\applications\oaa directory.

Creating a virtual directory for Get-Services

To run Get-Services, you create a virtual directory in your Web server that maps to your WebLogic deployment directory.

To configure a virtual directory:

1 Use the following guidelines to create the virtual directory in WebLogic and map it to your deployment directory with the following settings.

Requirement	Setting
Create virtual directory	<oaa>
Access Permissions	Read, Run scripts
Map to physical path	<WebLogic>\applications\oaa
Set Execute Permissions to	Scripts and Executables

Where <oaa> is the name of the virtual directory you want to use for Get-Services. Peregrine recommends oaa as the virtual directory name. Use this name in your application server configuration.

For <WebLogic>, enter the path to your WebLogic installation. The default file path is:
c:\bea\weblogic6.1\config\<my-domain>\applications\oaa

2 Right-click the newly created virtual directory under Default Web Site.

   a Select **Properties**.

   b Verify that the permission is **Scripts and Executables**.

3 Click **Configuration** on the Home Directory tab.

   The Application Configuration page opens on the App Mappings tab.

4 Verify that there is a mapping for the `.jsp` extension and that it is mapped to `c:\bea\wlserver6.1\bin\iisproxy.dll`.

Restarting the servers

To activate the new WebLogic configurations, restart the servers.

To activate the WebLogic configurations:

1 Restart your Web server.

2 Restart the WebLogic server.

3 Start Get-Services.

Configuring Get-Services

Follow the instructions in Configuring Get-Services on page 86.

WebLogic 8.1

The following sections provide instructions for installing, setting up, and configuring Get-Services on a Windows or Unix platform with WebLogic 8.1 using WebLogic's internal Web server.

Step 1 Install Get-Services, then copy or FTP the `image` directory. See Installing Get-Services on page 49.

Step 2 Copy the `jar` files. See Copying jar files on page 50.

Step 3 Edit the `startmydomain` file. See Editing the startmydomain file on page 51.

Step 4 Edit the `Server.Policy` file. See Editing the Server.Policy file on page 53.

Step 5 Deploy the Web Application Module to the application server. See Deploying to the application server on page 53.

Step 6 Modify the `local.xml` file to configure Get-Services. See Configuring Get-Services using the local.xml file on page 53.

Installing Get-Services

Follow these step-by-step instructions to install Get-Services on a WebLogic 8.1 application server. The installer creates an image directory that you must copy or FTP to a deployment directory on the WebLogic server.

To install Get-Services:

1 Run the installer on the installation CD. (See Running the installer on page 83)

2 Do one of the following:

- On Windows, copy the entire contents of the `image` directory (the default path is `C:\Program Files\Peregrine\Portal\image`) to a deployment directory of your choice on the WebLogic server.

  Example: `C:\Program Files\Peregrine\Portal\applications\oaa`

- On Unix, FTP the entire contents of the image directory (the default path is `C:\Program Files\Peregrine\Portal\image`) to a deployment directory of your choice on the WebLogic server.

  Example: `app\peregrine\applications\oaa`.

Note: These instructions refer to the deployment directory as `<APP_DEPLOYMENT_DIR>`.

Copying jar files

The installer generates `jar` files that you must copy, or FTP, to the application server `endorsed` directory.

- On Windows, copy the following files to the `<jdk_dir>\jre\lib\endorsed\` directory on the WebLogic server, where `<jdk_dir>` represents the path to the JDK home directory.

  ```
  xercesImpl.jar
  xalan.jar
  xml-apis.jar
  ```

  The default path for these files is:

  ```
  C:\Program Files\Peregrine\Portal\JavaExtensions
  ```

  Create the `endorsed` directory if it does not exist in the JDK home directory.

- On Unix, FTP the following files to the `<jdk_dir>\jre\lib\endorsed\` directory on the WebLogic server, where `<jdk_dir>` represents the path to the JDK home directory.

  ```
  xercesImpl.jar
  xalan.jar
  xml-apis.jar
  ```

  The default path for these files is:

  ```
  C:\Program Files\Peregrine\Portal\JavaExtensions
  ```

  Create the `endorsed` directory if it does not exist in the JDK home directory.

- Copy, or FTP, the following files from the installation directory `C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions`

  ```
  activation.jar
  mail.jar
  pop3.jar
  ```

  to the `<APP_DEPLOYMENT_DIR>\WEB-INF\lib` directory.

Editing the startmydomain file

- On Windows, use a text editor to modify the `startmydomain.cmd` file (or whichever file you use to start the WebLogic server).

  Add the following string to the final start up command (led by calling java):

  ```
  -Djava.security.auth.login.config=%WL_HOME%\server\lib\Server.Policy
  ```

  Example:

  ```
  %JAVA_HOME%\bin\java %JAVA_VM% %MEM_ARGS% %JAVA_OPTIONS%
  -Dweblogic.Name=%SERVER_NAME%
  -Dweblogic.ProductionModeEnabled=%PRODUCTION_MODE%
  -Djava.security.policy="%WL_HOME%\server\lib\weblogic.policy"
  -Djava.security.auth.login.config="%WL_HOME%\server\lib\Server.Policy
  " weblogic.Server
  ```

- On Unix, modify the `startmydomain.sh` file according to your operating system.

Operating System	Action

Solaris	Add the following string to the final start up command (led by calling java).
	`-Djava.security.auth.login.config=${WL_HOME}/server/lib/Server.Policy`
	
	Example:
	`java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}`
	`-Dweblogic.Name=${SERVER_NAME}`
	`-Dweblogic.management.username=${WLS_USER}`
	`-Dweblogic.management.password=${WLS_PW}`
	`-Dweblogic.ProductionModeEnabled=${STARTMODE}`
	`-Djava.security.policy="${WL_HOME}/server/lib/weblogic.policy"`
	`-Djava.security.auth.login.config=${WL_HOME}/server/lib/Server.Policy`
	`weblogic.Server`

AIX	At the top of the file, add a single entry for LIBPATH and set it to the path for the appropriate ServiceCenter libraries.
	For example, when running ServiceCenter 5:
	`LIBPATH=<APP_DEPLOYMENT_DIR>/WEB-INF/lib/AIX/ServiceCenter5`
	
	When running ServiceCenter 6:
	`LIBPATH=<APP_DEPLOYMENT_DIR>/WEB-INF/lib/AIX/ServiceCenter6`
	
	Export the completed variable entry using the following command:
	`export LIBPATH`
	
	Add the following string to the final start up command (led by calling java).
	`-Djava.security.auth.login.config=${WL_HOME}/server/lib/Server.Policy`
	
	Example:
	`java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}`
	`-Dweblogic.Name=${SERVER_NAME}`
	`-Dweblogic.management.username=${WLS_USER}`
	`-Dweblogic.management.password=${WLS_PW}`
	`-Dweblogic.ProductionModeEnabled=${STARTMODE}`
	`-Djava.security.policy="${WL_HOME}/server/lib/weblogic.policy"`
	`-Djava.security.auth.login.config=${WL_HOME}/server/lib/Server.Policy`
	`weblogic.Server`

Linux	Add the following string to the final start up command (led by calling Java).
	`-Djava.security.auth.login.config=${WL_HOME}/server/lib/Server.Policy`
	
	Example:
	`java ${JAVA_VM} ${MEM_ARGS} ${JAVA_OPTIONS}`
	`-Dweblogic.Name=${SERVER_NAME}`
	`-Dweblogic.management.username=${WLS_USER}`
	`-Dweblogic.management.password=${WLS_PW}`
	`-Dweblogic.ProductionModeEnabled=${STARTMODE}`
	`-Djava.security.policy="${WL_HOME}/server/lib/weblogic.policy"`
	`-Djava.security.auth.login.config=${WL_HOME}/server/lib/Server.Policy`
	`weblogic.Server`

Editing the Server.Policy file

Edit the `Server.Policy` file to define the back-end system.

- Create or edit the `Server.Policy` file (in the `<WL_HOME>/server/lib` directory for Unix or`<WL_HOME>\server\lib\` for Windows), with the following contents.

```
oaa{
    com.peregrine.oaa.security.OAALoginModule optional  target=sc;
};
getit.admin{com.peregrine.oaa.security.GetitAdminLoginModule required;};
getit.anonymous{com.peregrine.oaa.security.GetitAnonymousLoginModule required;};
sc{com.peregrine.oaa.security.OAALoginModule required target=sc;};
ServerLoginModule{weblogic.security.internal.ServerLoginModule required;};
```

Deploying to the application server

Deploy the Web Application Module from the WebLogic 8.1 Web console using the `<APP_DEPLOYMENT_DIR>` as the target.

Configuring Get-Services using the local.xml file

To configure Get-Services, you need to manually edit the `local.xml` file found in `<APP_DEPLOYMENT_DIR>/WEB-INF/local.xml`. If you do not find the `local.xml` file in `<APP_DEPLOYMENT_DIRECTORY/WEB-INF/`, you need to create it by following the steps in Creating a local.xml file on page 53. If you have a `local.xml` file, go to Editing the local.xml file on page 54.

Creating a local.xml file

1 Start the WebLogic 8.1 server with the `startmydomain.sh` script.

Once the server finishes initializing, open the URL *http://<WebLogic Server>:7001/<Context Root>/admin.jsp* in a browser. This assumes your server port is **7001**. Substitute your port number in the URL if you are not using port **7001**.

2 Log in with username **System**, keeping the password field blank.

a From the Administration menu, click **Settings**.

b Select the **Common** tab.

c Scroll down to the Server URL field and replace the entry with the correct URL.

Example: `http://<WebLogicServer>:7001/<Context Root/login.jsp`

d Click **Save** at the bottom of the tab.

e Log out of the Peregrine Portal.

f Stop the WebLogic server.

This creates the `local.xml` file, which you now need to edit.

Editing the local.xml file

Edit the `local.xml` file found in `<APP_DEPLOYMENT_DIRECTORY/WEB-INF/local.xml`.

To edit the local.xml file:

1 Using a text editor, add the following lines to the `<settings></settings>` section of the `<APP_DEPLOYMENT_DIR>/WEB-INF/local.xml` file.

```
<jaas_config>
        <useStandardJAASConfiguration>true</useStandardJAASConfiguration>
        <jaasConfiguration>oaa</jaasConfiguration>
</jaas_config>
```

2 Start the WebLogic 8.1 server with the `startmydomain.sh` command or the `startmydomain.cmd` command.

3 Once the server finishes initializing, open the URL *http://<WebLogic Server>:7001/<Context Root>/admin.jsp* in a browser.

This assumes your server port is **7001**. Substitute your port number in the URL, if you are not using port **7001**.

4 Log in with username **System**, keeping the password field blank.

5 Follow the steps described in Configuring Get-Services on page 86.

WebLogic 8.1 and IIS 5

The following instructions configure the WebLogic 8.1 application server with the Microsoft Internet Information Server (IIS) Web server.

Step 1 Stop both WebLogic and your IIS Web server. See Stopping the servers on page 55.

Step 2 Run the Get-Services installer. See Running the installer on page 55.

Step 3 Copy the installed files to your WebLogic directory. See Copying the installed files to your WebLogic directory on page 55.

Step 4 Create the `oaa` virtual directory. See Creating a virtual directory for Get-Services on page 56.

Step 5 Copy the `iisproxy.dll` file. See Copying the iisproxy.dll file on page 57.

Step 6 Configure the IIS plug-in. See Configuring the IIS plug-in on page 58.

Step 7 Restart the servers. See Restarting the servers on page 62.

Step 8 Test your configurations. Testing the configuration on page 62.

Stopping the servers

Before you begin configuring WebLogic, you must close your WebLogic server and IIS Web server.

Running the installer

You must do this on a Windows machine. See Running the installer on page 83.

Copying the installed files to your WebLogic directory

The installer generates files that you must copy, or FTP, to your WebLogic application directory.

To copy the installed files:

1 Copy the `Portal\image` directory (the default path is `C:\Program Files\Peregrine\Portal\image`) to the `<WebLogic>\applications` directory.

The WebLogic default path is: `C:\bea\weblogic8.1\config\<my-domain>`

2 Rename `<WebLogic>\applications\image` directory to `<WebLogic>\applications\oaa`.

Tip: If you need to FTP files from one server to another to perform the copy, it is more convenient to FTP the `oaa.war` file in `<Portal>\oaa.war` to your WebLogic server than to extract the `oaa.war` files into a `<WebLogic>\applications\oaa` directory.

Creating a virtual directory for Get-Services

To run Get-Services from IIS, you need to create a virtual directory that maps to your WebLogic deployment directory.

1 Create the virtual directory in WebLogic and map it to your deployment directory with the following settings:

Requirement	Setting
Create virtual directory	`<oaa>`
Map to physical path	`<WebLogic>/applications/oaa`
Directory access rights	`Read, Run scripts, Execute`

Where `<oaa>` is the name of the virtual directory you want to use for Get-Services. Peregrine recommends `oaa` as the virtual directory name. You must replicate the name that you enter as the virtual directory name in your application server configuration.

For `<WebLogic>`, enter the path to your WebLogic installation. The default file path is: `c:/bea/weblogic81`.

2 Copy, or FTP, the following files to the <jdk_dir>/jre/lib/endorsed/ directory on the WebLogic server, where <jdk_dir> represents the path to the JDK home directory.

```
xercesImpl.jar
xalan.jar
xml-apis.jar
```

The default path for these files is:

```
C:/Program Files/Peregrine/Portal/JavaExtensions
```

Create the endorsed directory if it does not exist in the JDK home directory.

3 Copy the following files from the installation directory C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions

```
activation.jar
mail.jar
pop3.jar
```

to the <APP_DEPLOYMENT_DIR>\WEB-INF\lib directory.

Copying the iisproxy.dll file

These instructions use the WL_HOME/server/bin directory as the default location for the DLLs.

- Copy the iisproxy.dll file from the WL_HOME/server/bin directory of your WebLogic Server installation to a convenient directory that is accessible to IIS.

  Where WL_HOME/server/bin is the top-level directory for the WebLogic Platform and Server and contains the WebLogic Server installation files.

  BEA recommends that this directory also contain the iisproxy.ini file that you create in Step 5 on page 61 of Configuring the IIS plug-in.

Configuring the IIS plug-in

WebLogic supports two methods of configuring the IIS plug-in:

- proxying by path
- proxying by MIME type

These instructions focus on the `proxying by path` method. It takes precedence over proxying by MIME type. For more information about these methods for configuring the IIS plug-in, refer to your BEA WebLogic documentation.

1 Start the Internet Information Service Manager by selecting it from the Start menu.

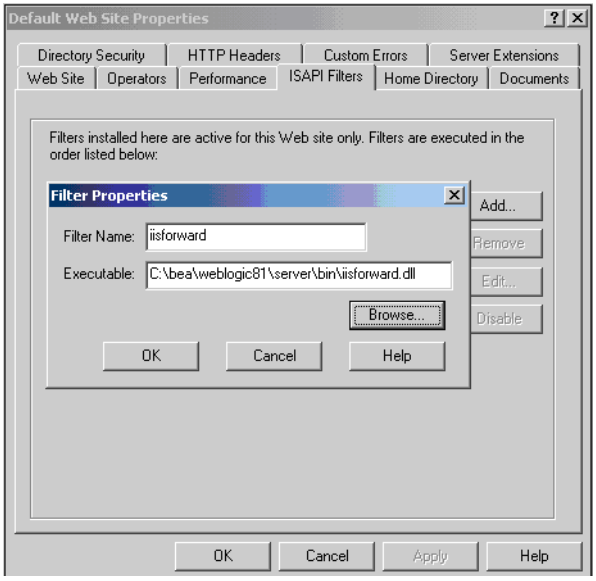
2 Place the `iisforward.dll` file in the same directory as the `iisproxy.dll` file, and add the `iisforward.dll` file as a filter service in IIS.

a Open the properties for the Default Web site by right-clicking the Web site selection in the left panel and selecting Properties.

Note: Substitute the appropriate Website if you do not want to use the Default Website.

b Select the ISAPI Filters tab and click **Add**.

c Define the ISAPI filter as shown in the following example.

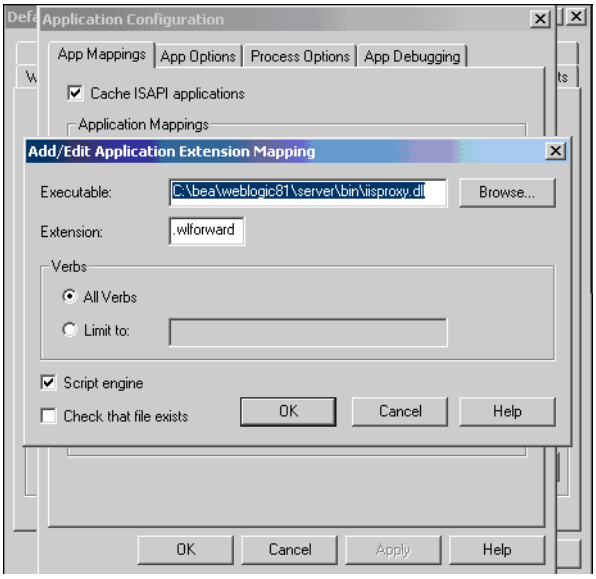


d When you finish, click **OK** to save the configuration.

3 Register `.wlforward` as a special file type that `iisproxy.dll` handles.

a In the Properties panel, select the Home Directory tab, and click **Configuration** in the Applications Settings section.

b Define the Application Mapping as shown in the following example.

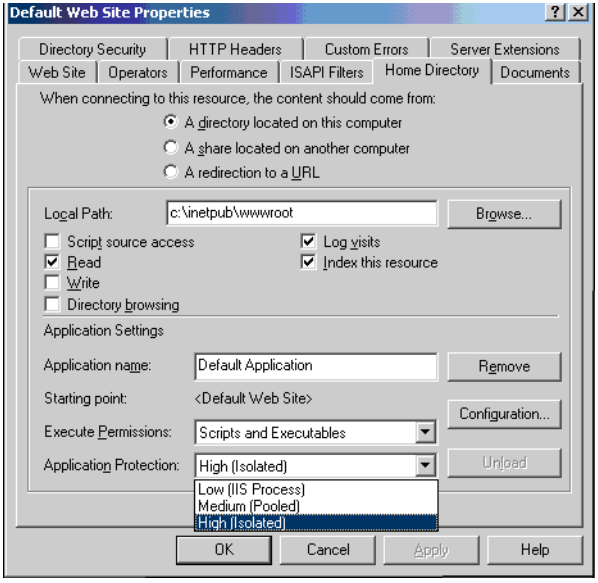


c When you finish, click **OK** to save the configuration.

d In the Application configuration window, select the **Cache ISAPI applications** option.

e Click **OK** to save the configuration.

f At the Website properties window, set the Application Protection to **High (isolated).**



4 Create the `iisproxy.ini` in the same directory where the `iisproxy.dll` is located.

5 Define the following properties in the `iisproxy.ini`.

```
# This file contains the configuration parameters
# for the IIS/WebLogic plug-in.
WebLogicHost=csx602
# substitute your WebLogic hostname above
WebLogicPort=7001
# substitute your WebLogic port number above
ConnectTimeoutSecs=20
ConnectRetrySecs=2
WLIOTimeoutSecs=unlimited
WlForwardPath=/oaa/servlet/archway,/oaa/servlet/oemlicense.OE
MLicenseServlet,/oaa/answers/attachments/*,/oaa/servlet/rpcro
uter,/oaa/servlet/messagerouter,/oaa/servlet/download/*,/oaa/
attachments/*,/oaa/servlet/upload,/oaa/*.do,/oaa/*.jsp,/oaa/*
.jsv,/oaa/*.jsw,/oaa/j_security_check
```

Note: The `WLForwardPath` parameter is all one line, delimited by commas.

If you need to debug your application, set the `Debug=ON` parameter in `iisproxy.ini`. The system generates a `c:\tmp\iisforward.log` file that contains a log of the plug-in's activity that you can use for debugging purposes.

For more information on the available configuration parameters for the `iisproxy.ini` file, refer to your BEA WebLogic documentation.

Restarting the servers

Restart WebLogic and your IIS Web server for the new settings to take effect. The IIS Admin and World Wide Web Publishing services control the IIS plug-in.

1 Start the WebLogic application server.

2 For IIS, from the Start menu, click **Control Panel** > **Administrative Tools** > **Services**, then select IIS Admin and World Wide Web Publishing services.

Testing the configuration

After the application and Web servers are properly configured, log on to the Get-Services Admin page to configure the settings that are stored in the `local.xml` file.

Make sure that the servers and services are started before proceeding.

1 To verify that the WebLogic and IIS servers are properly configured, check the following URLs.

- *http://<server>:7001/oaa/login.jsp* for the WebLogic application server
- *http://<server>/oaa/login.jsp* for the IIS Web server

2 Follow the steps described in Configuring Get-Services on page 86 to complete the Get-Services configuration.

WebLogic 8.1 and Sun ONE

The following instructions configure the WebLogic 8.1 application server with the Sun ONE 6.1 Web server. You must modify two Sun ONE files, `obj.conf` and `magnus.conf`, found in the `<SunONE_Home>/https-<server.domain>/config` directory.

Important: Before you begin using this procedure, ensure that you have completed all of the steps for configuring the WebLogic 8.1 server as described in WebLogic 8.1 on page 48.

To configure WebLogic 8.1 and Sun ONE:

Step 1 Modify the `obj.conf` file. See Modifying the obj.conf file.

Step 2 Modify the `magnus.conf` file. See Modifying the magnus.conf file on page 67.

Modifying the obj.conf file

For the WebLogic and Sun ONE servers to run the Peregrine Portal, you must configure the `obj.conf` file.

To modify the obj.conf file:

Note: These instructions refer to the deployment directory as `<APP_DEPLOYMENT_DIR>`.

1 With a text editor, update the original `obj.conf` file.

Original `obj.conf` file

```
<Object name="default">
AuthTrans fn="match-browser" browser="*MSIE*" ssl-unclean-shutdown="true"
NameTrans fn="ntrans-j2ee" name="j2ee"
NameTrans fn=pfx2dir from=/mc-icons dir="/SunOne61_WS/ns-icons"
name="es-internal"
NameTrans fn=document-root root="$docroot"
PathCheck fn=unix-uri-clean
PathCheck fn="check-acl" acl="default"
PathCheck fn=find-pathinfo
PathCheck fn=find-index index-names="index.html,home.html,index.jsp"
ObjectType fn=type-by-extension
ObjectType fn=force-type type=text/plain
Service method=(GET|HEAD) type=magnus-internal/imagemap fn=imagemap
Service method=(GET|HEAD) type=magnus-internal/directory fn=index-common
Service method=(GET|HEAD|POST) type=*~magnus-internal/* fn=send-file
Service method=TRACE fn=service-trace
Error fn="error-j2ee"
AddLog fn=flex-log name="access"
</Object>

<Object name="j2ee">
ObjectType fn=force-type type=text/html
Service fn="service-j2ee" method="*"
</Object>

<Object name="cgi">
ObjectType fn=force-type type=magnus-internal/cgi
Service fn=send-cgi user="$user" group="$group" chroot="$chroot" dir="$dir"
nice="$nice"
</Object>

<Object name="es-internal">
PathCheck fn="check-acl" acl="es-internal"
</Object>

<Object name="send-compressed">
PathCheck fn="find-compressed"
</Object>

<Object name="compress-on-demand">
Output fn="insert-filter" filter="http-compression"
</Object>
```

a Add the following lines of code to the original `obj.conf` file.

Note: When you add these lines, add them to the file in the order shown in the example `obj.conf` file.

```
NameTrans fn="assign-name" from="/oaa/*.jsp" name="oaaservlet"
NameTrans fn="assign-name" from="/oaa/*.do" name="oaaservlet"
NameTrans fn="assign-name" from="/oaa/servlet/*" name="oaaservlet"
NameTrans fn="pfx2dir" from="/oaa" dir="/<APP_DEPLOYMENT_DIR>/oaa"
<APP_DEPLOYMENT_DIR><APP_DEPLOYMENT_DIR>
PathCheck fn="find-index" index-names="index.html,home.html,index.jsp"
```

Note: The PathCheck line entry instructs the Sun ONE server to reject any request with a URL that contains the path /WEB-INF. Each servlet application (context) has a special WEB-INF directory. This directory contains sensitive configurations data and Java classes and must be hidden from Web users.

b Comment out the following lines of code.

```
#NameTrans fn="ntrans-j2ee" name="j2ee"
#NameTrans fn="pfx2dir" from="/mc-icons" dir="/SunOne61_WS/ns-icons"
name="es-internal"

#Error fn="error-j2ee"
```

c Add the following new configuration object.

```
<Object name="oaaservlet">
ObjectType fn="force-type" type="text/plain"
Service fn=2l_proxy WebLogicHost=<servername> WebLogicPort=7001
</Object>
```

d Verify that the `obj.conf` you modified matches the following sample in both content and the order of the lines of code.

Sample `obj.conf` after you have made the required modifications

```
<Object name="default">
AuthTrans fn="match-browser" browser="*MSIE*" ssl-unclean-shutdown="true"
#NameTrans fn="ntrans-j2ee" name="j2ee"
#NameTrans fn="pfx2dir" from="/mc-icons" dir="/SunOne61_WS/ns-icons"
name="es-internal"
NameTrans fn="assign-name" from="/oaa/*.jsp" name="oaaservlet"
NameTrans fn="assign-name" from="/oaa/*.do" name="oaaservlet"
NameTrans fn="assign-name" from="/oaa/servlet/*" name="oaaservlet"
NameTrans fn="pfx2dir" from="/oaa" dir="/<APP_DEPLOYMENT_DIR>/oaa"
<APP_DEPLOYMENT_DIR><APP_DEPLOYMENT_DIR>NameTrans fn="document-root"
root="$docroot"
PathCheck fn="unix-uri-clean"
PathCheck fn="check-acl" acl="default"
PathCheck fn="deny-existence" path="*/WEB-INF/*"
PathCheck fn="find-pathinfo"
PathCheck fn="find-index" index-names="index.html,home.html,index.jsp"
ObjectType fn="type-by-extension"
ObjectType fn="force-type" type="text/plain"
Service method="(GET|HEAD)" type="magnus-internal/imagemap" fn="imagemap"
Service method="(GET|HEAD)" type="magnus-internal/directory" fn="index-common"
Service method="(GET|HEAD|POST)" type="*~magnus-internal/*" fn="send-file"
Service method="TRACE" fn="service-trace"
#Error fn="error-j2ee"
AddLog fn="flex-log" name="access"
</Object>

<Object name="j2ee">
ObjectType fn="force-type" type="text/html"
Service fn="service-j2ee" method="*"
</Object>

<Object name="cgi">
ObjectType fn="force-type" type="magnus-internal/cgi"
Service fn="send-cgi" user="$user" group="$group" chroot="$chroot" dir="$dir"
nice="$nice"
</Object>

<Object name="es-internal">
PathCheck fn="check-acl" acl="es-internal"
</Object>

<Object name="send-compressed">
PathCheck fn="find-compressed"
</Object>

<Object name="compress-on-demand">
Output fn="insert-filter" filter="http-compression"
</Object>

<Object name="oaaservlet">
ObjectType fn="force-type" type="text/plain"
Service fn=wl_proxy  WebLogicHost=mustang WebLogicPort=7001
</Object>
```

2 Save and close the file.

Modifying the magnus.conf file

For the WebLogic and Sun ONE servers to run the Peregrine Portal, you must configure the `magnus.conf` file.

Note: These instructions refer to the deployment directory as <APP_DEPLOYMENT_DIR>.

To modify the magnus.conf file:

1 Using a text editor, replace the following line of code in `magnus.conf`:

```
Init fn=flex-init access="$accesslog" format.access="%Ses->client.ip% -
%Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

with these two lines.

```
Init fn="load-modules" funcs="wl_proxy,wl_init"
shlib="<AppServerPath>/server/lib/solaris/libproxy.so"

Init fn="wl_init"
```

2 Verify your changes.

Original section of the `magnus.conf` file being modified.

```
Init fn=flex-init access="$accesslog" format.access="%Ses->client.ip% -
%Req->vars.auth-user% [%SYSDATE%] \"%Req->reqpb.clf-request%\"
%Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"

Init fn="load-modules" shlib="/appSunOne61_WS/bin/https/lib/libj2eeplugin.so"
shlib_flags="(global|now)"
```

Updated section of the `magnus.conf` file being modified.

```
Init fn="load-modules" funcs="wl_proxy,wl_init"
shlib="<AppServerPath>/server/lib/solaris/libproxy.so"

Init fn="wl_init"

Init fn="load-modules" shlib="/appSunOne61_WS/bin/https/lib/libj2eeplugin.so"
shlib_flags="(global|now)"
```

3 Save and close the file.

After you have modifed the `obj.conf` and the `magnus.conf` configuration files, you need to update the Sun ONE configuration on the server.

To update the Sun ONE configuration:

1 Use Sun One Web Server Administration Server to update the Sun ONE cofiguration.

2 Restart the Sun One Web server.

3 Restart the WebLogic application server.

4 Verify that the server is running.

http://<servername>/oaa/admin.jsp

5 Log in with username **System**, keeping the password field blank.

6 If necessary, follow the steps described in Configuring Get-Services on page 86.

WebSphere 5.0.2

You must configure your application and Web servers *prior* to running the Get-Services installer.

Assumptions

Before beginning installation, verify that you have the following software.

- WebSphere 5.0.2 using Java version 1.3.1 or later
  - Java(™) 2 Runtime Environment, Standard Edition (build 1.3.1)
  - Classic VM (build 1.3.1, J2RE 1.4.1 IBM Windows 32 build cn1411-20031011 (JIT enabled: jitc))
- IBM HTTP Server 1.3.26 or later

Configuring WebSphere 5.0.2

Use the following procedures to configure WebSphere 5.0.2 to run Get-Services.

Step 1 Run the installer. See Running the installer.

Step 2 Deploy the `oaa.war` file. See Deploying the oaa.war file.

Step 3 Copy the `jar` files. See Copying jar files on page 71.

Step 4 Edit the application settings using the WebSphere Administrative Console. See Editing the application settings on page 72.

Step 5 Edit the `httpd.conf` file to add the plug-in modules. See Editing the httpd.conf file on page 73.

Step 6 For AIX, if you are using ServiceCenter as the back-end, update the path to the ServiceCenter libraries in the `startServer` file. See Editing the startServer file for AIX on page 74.

Step 7 Start the WebSphere Administration server. See Starting the WebSphere Administration server on page 74.

Step 8 Complete the configuration. See Configuring Get-Services on page 75.

Running the installer

You must do this on a Windows machine. See Running the installer on page 83.

Deploying the oaa.war file

The oaa.war file creates the directory structure necessary to deploy Get-Services to your application server.

To deploy the `oaa.war` file using the WebSphere Administrative Console:

1 Start the WebSphere Administration Server.

2 Log on to the WebSphere Administrative Console.

3 Click **Applications** on the Administrative Console menu.

4 Click **Install New Application**.

5 In the Local path field, browse to the `oaa.war` file created at installation.

6 Specify the Context Root, a value such as **oaa** (the virtual directory name).

7 Click **Next**.

This may take several minutes.

8 Accept the default settings and click **Next**.

9 Click **Use Binary Configuration**.

10 In the Application Name field, specify a name, such as **oaa**.

11 Keep all the other settings at their defaults, and click **Next**.

12 Keep the default settings, and click **Next**.

13 Keep the default settings, and click **Next**.

14 Click **Finish**.

15 Click **Save to Master Configuration** to save the server configuration.

16 Click **Save**.

17 Log out of the Administrative Console.

18 Stop the WebSphere Administration Server.

Copying jar files

The installer generates `jar` files that you must copy to the application server `endorsed` directory.

1 Verify that the following files are in the `<APP_DEPLOYMENT_DIR>/WEB-INF/lib` directory, where `<APP_DEPLOYMENT_DIR>` is the path to your Get-Services deployment directory.

```
js.jar
jai_core.jar
jai_codec.jar
mlibwrapper_jai.jar
```

2 Copy, or FTP, the following files from the installation directory
`C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions`

```
activation.jar
mail.jar
pop3.jar
```

to the `<APP_DEPLOYMENT_DIR>\WEB-INF\lib` directory.

3 Copy, or FTP, the following files from the installation directory
`C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions`

```
jaas.jar
xalan.jar
xercesImpl.jar
xml-apis.jar
oaasecurityproxy.jar
jsse.jar
```

where `C:\Program Files\Peregrine\Portal` specifies the location of your application installation,

to the `$JAVA_HOME/jre/lib/ext` directory.

a If necessary, replace any older versions of these files.

b If `xmlParserAPIs.jar` exists in the `ext` directory, delete it.

Editing the application settings

Using the WebSphere Administrative Console, edit the application settings.

To edit the application settings:

1 Start the WebSphere Administration Server.

2 Log on to the WebSphere Administrative Console.

3 Click the Enterprise Application link.

4 Click the Get-Services link from the list for the application that you just installed.

5 Click **Web Modules** under Related Items.

6 Click the **oaa.war** link.

7 Change the **Classloader Mode** to **PARENT_LAST**.

8 Click **OK**, click **Save**, then click **Save** again under **Save to Master Configuration** to save the server configuration.

9 To regenerate the plug-ins, click **Environment** > **Update Web Server Plugin**.

- Click **OK** to update the Web server plugin.
- Wait for confirmation that the plugin is updated.

10 Log out of the Administrative Console.

11 Stop the WebSphere Administration Server.

Editing the httpd.conf file

Modify the httpd.conf file to add the plug-in modules and an alias for the virtual directory.

1 From the IBM HTTP Server configuration `/conf/httpd.conf` file, add the plug-in modules.

On Windows:

```
LoadModule ibm_app_server_http_module
<AppServerPath>\bin\mod_ibm_app_server_http.dll
WebSpherePluginConfig <AppServerPath>\config\cells\plugin-cfg.xml
```

On UNIX:

```
LoadModule ibm_app_server_http_module
<AppServerPath>/bin/mod_ibm_app_server_http.so
WebSpherePluginConfig <AppServerPath>/config/cells/plugin-cfg.xml
```

2 Create an alias in `httpd.conf` for the virtual directory.

```
Alias /oaa <AppServerPath>/AppServer/installedApps/<hostname>
/oaa_war.ear/oaa.war
```

Note: The `/oaa` alias needs to match the Context Root specified in Step 6 on page 70 of Deploying the oaa.war file.

3 Save and close the file.

Restart the IBM HTTP Server.

Editing the startServer file for AIX

When using ServiceCenter as the back-end, the AIX environment requires you to add the path to the ServiceCenter libraries in the `startServer.sh` file.

1 Using a text editor, open the `startServer.sh` file.

2 At the top of the file add a single entry for LIBPATH and set it to the path for the appropriate ServiceCenter libraries.

For example, when running ServiceCenter 5:

```
LIBPATH=/WebSphere/AppServer/installedApps/oaa.ear/oaa.war/
WEB-INF/lib/AIX/ServiceCenter5
```

When running ServiceCenter 6:

```
LIBPATH=/WebSphere/AppServer/installedApps/oaa.ear/oaa.war/
WEB-INF/lib/AIX/ServiceCenter6
```

Export the completed variable entry using the following command:
`export LIBPATH`

3 Save and close the file.

Starting the WebSphere Administration server

Start the WebSphere Administration server for the new settings to take effect.

Configuring Get-Services

To complete the configuration, follow the instructions in Configuring Get-Services on page 86.

WebSphere 5.1 and IBM HTTP server

You must configure your application and Web servers *prior* to running the Get-Services installer.

Assumptions

Before beginning installation, verify that you have the following software.

- WebSphere 5.1 using Java version 1.4.1 or later
  - Java™ 2 Runtime Environment, Standard Edition (build 1.4.1)
  - Classic VM (build 1.4.1, J2RE 1.4.1 IBM Windows 32 build cn1411-20031011 (JIT enabled: jitc))
- IBM HTTP Server 1.3.26 or later

Configuring WebSphere 5.1

Use the following procedures to configure WebSphere 5.1 to run Get-Services.

Step 1 Run the installer. See Running the installer on page 76.

Step 2 Deploy the `oaa.war` file. See Deploying the oaa.war file on page 76.

Step 3 Copy the `jar` files. See Copying jar files on page 77.

Step 4 Edit the application settings using the WebSphere Administrative Console. See Editing the application settings on page 78.

Step 5 Edit the `httpd.conf` file to add the plug-in modules. See Editing the httpd.conf file on page 79.

Step 6 For AIX, if you are using ServiceCenter as the back-end, update the path to the ServiceCenter libraries in the `startServer` file. See Editing the startServer file for AIX on page 80.

Step 7 Start the WebSphere Administration server. See Starting the WebSphere Administration server on page 80.

Step 8 Complete the configuration. See Configuring Get-Services on page 81.

Running the installer

You must do this on a Windows machine. See Running the installer on page 83.

Deploying the oaa.war file

The oaa.war file creates the directory structure necessary to deploy Get-Services to your application server.

To deploy the `oaa.war` file using the WebSphere Administrative Console:

1 Start the WebSphere Administration Server.

2 Log on to the WebSphere Administrative Console.

3 Click **Applications** on the Administrative Console menu.

4 Click **Install New Application**.

5 In the Local path field, browse to the `oaa.war` file created at installation.

6 Specify the Context Root, a value such as **oaa** (the virtual directory name).

7 Click **Next**.

This may take several minutes.

8 Accept the default settings and click **Next**.

9 Click **Continue**.

10 Click **Use Binary Configuration**.

11 In the Application Name field, specify a name, such as **oaa**.

12 Keep all the other settings at their defaults, and click **Next**.

13 Keep the default settings, and click **Next**.

14 Keep the default settings, and click **Next**.

15 Click **Finish**.

16 Click **Save to Master Configuration** to save the server configuration.

17 Click **Save**.

18 Log out of the Administrative Console.

19 Stop the WebSphere Administration Server.

Copying jar files

The installer generates `jar` files that you must copy to the application server `endorsed` directory.

1 Verify that the following files are in the `<APP_DEPLOYMENT_DIR>/WEB-INF/lib` directory, where `<APP_DEPLOYMENT_DIR>` is the path to your Get-Services deployment directory.

```
js.jar
jai_core.jar
jai_codec.jar
mlibwrapper_jai.jar
```

2 Copy, or FTP, the following files from the installation directory
`C:\Program Files\Peregrine\Portal\SupportFiles\JavaExtensions`

```
activation.jar
mail.jar
pop3.jar
```

to the `<APP_DEPLOYMENT_DIR>\WEB-INF\lib` directory.

3 Copy, or FTP, the following files from the installation directory
`C:/Program Files/Peregrine/Portal/SupportFiles/JavaExtensions`

```
xalan.jar
xercesImpl.jar
xml-apis.jar
```

where `C:/Program Files/Peregrine/Portal` specifies the location of your application installation,

to the `$JAVA_HOME/jre/lib/endorsed` directory.

a If necessary, replace any older versions of these files.

b If `xmlParserAPIs.jar` exists in the `endorsed` directory, delete it.

c Create the `endorsed` directory if one does not exist.

Editing the application settings

Using the WebSphere Administrative Console, edit the application settings.

To edit the application settings:

1 Start the WebSphere Administration Server.

2 Log on to the WebSphere Administrative Console.

3 Click the Enterprise Application link.

4 Click the Get-Services link from the list for the application that you just installed.

5 Click **Web Modules** under Related Items.

6 Click the **oaa.war** link.

7 Change the **Classloader Mode** to **PARENT_LAST**.

8 Click **OK**, click **Save**, then click **Save** again under **Save to Master Configuration** to save the server configuration.

9 To regenerate the plug-ins, click **Environment** > **Update Web Server Plugin**.

- Click **OK** to update the Web server plugin.
- Wait for confirmation that the plugin is updated.

10 Log out of the Administrative Console.

11 Stop the WebSphere Administration Server.

Editing the httpd.conf file

Modify the httpd.conf file to add the plug-in modules and an alias for the virtual directory.

1 From the IBM HTTP Server configuration `/conf/httpd.conf` file, add the plug-in modules.

On Windows:

```
LoadModule ibm_app_server_http_module
<AppServerPath>\bin\mod_ibm_app_server_http.dll
WebSpherePluginConfig <AppServerPath>\config\cells\plugin-cfg.xml
```

On UNIX:

```
LoadModule ibm_app_server_http_module
<AppServerPath>/bin/mod_ibm_app_server_http.so
WebSpherePluginConfig <AppServerPath>/config/cells/plugin-cfg.xml
```

2 Create an alias in `httpd.conf` for the virtual directory.

```
Alias /oaa <AppServerPath>/AppServer/installedApps/<hostname>
/oaa_war.ear/oaa.war
```

Note: The `/oaa` alias needs to match the Context Root specified in Step 6 on page 76 in Deploying the oaa.war file.

3 Save and close the file.

4 Restart the IBM HTTP Server.

Editing the startServer file for AIX

When using ServiceCenter as the back-end, the AIX environment requires you to add the path to the ServiceCenter libraries in the `startServer.sh` file.

1 Using a text editor, open the `startServer.sh` file.

2 At the top of the file add a single entry for LIBPATH and set it to the path for the appropriate ServiceCenter libraries.

For example, when running ServiceCenter 5:

```
LIBPATH=/WebSphere/AppServer/installedApps/oaa.ear/oaa.war/
WEB-INF/lib/AIX/ServiceCenter5
```

When running ServiceCenter 6:

```
LIBPATH=/WebSphere/AppServer/installedApps/oaa.ear/oaa.war/
WEB-INF/lib/AIX/ServiceCenter6
```

Export the completed variable entry using the following command:
`export LIBPATH`

3 Save and close the file.

Starting the WebSphere Administration server

Start the WebSphere Administration server for the new settings to take effect.

Configuring Get-Services

To complete the configuration, follow the instructions in Configuring Get-Services on page 86.

Specifying a file path to the local.xml file

The default path to the parent directory of the `local.xml` file is `<APP_DEPLOYMENT_DIR>\WEB-INF\`. You may specify an alternate directory path by adding a JVM runtime property to your Get-Services application through your application server's administrative interface. Consult your application server's documentation on adding a JVM runtime property. The property name is `com.peregrine.sharedSettingsDir`.

For example, if your directory path is `c:\share\oaa\` (or `/usr/share/oaa`), you provide a URI as `file:/c:/share/oaa/` (or `file:/usr/share/oaa`) or a file system path as `c:/share/oaa/` (or `/usr/share/oaa`).

If no administrative interface exists to perform the change, you can use command line switch as part of the JVM runtime property command that executes the Java JVM, as follows:

```
-Dcom.peregrine.sharedSettingsDir=file:/c:/share/oaa/
```

or

```
-Dcom.peregrine.sharedSettingsDir=c:/share/oaa/
```

Alternately, you can set the path as an initialization parameter to the Archway servlet. You can do this through the application server's administrative interface. Consult your application server's documentation on adding a servlet initialization parameter. The parameter name and value are as described above.

If no administrative interface exists to add the initialization parameter, you can edit the `web.xml` file in `<APP_DEPLOYMENT_DIR>/WEB-INF/` manually to contain an init-param for the Archway servlet as follows:

```
<servlet>
    <servlet-name>Archway</servlet-name>
    <display-name>Archway</display-name>
    <description></description>
    <servlet-class>com.peregrine.oaa.archway.ArchwayServlet</servlet-class>
    <load-on-startup>1</load-on-startup>
    <init-param>
        <param-name>com.peregrine.sharedSettingsDir</param-name>
        <param-value>file:/c:/share/oaa/</param-value>
    </init-param>
</servlet>
```

You have to restart the application server for Get-Services application after making any of these.

Running the installer

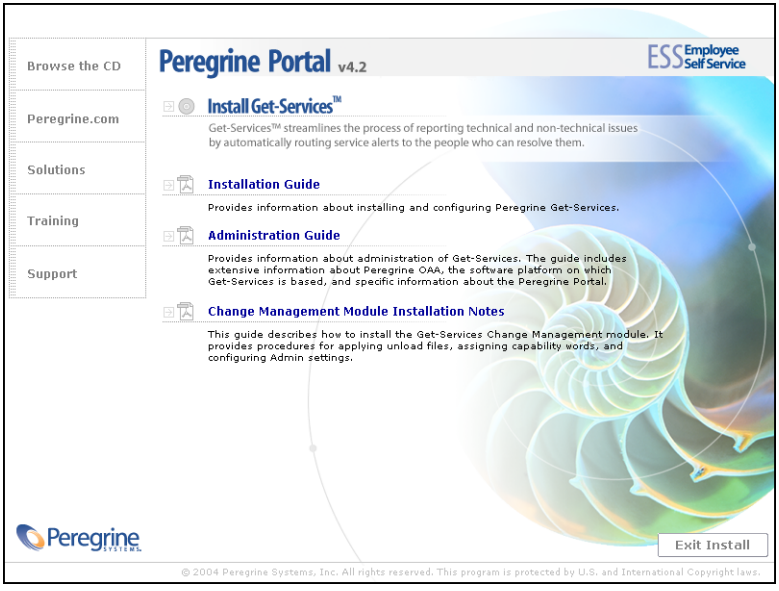
Prior to Get-Services installation, you must install and configure the application and Web servers for your environment.

To install Get-Services:

1 Insert the Get-Services installation CD-ROM into the CD-drive.

If you are installing on a system that has autorun enabled, the CD browser starts automatically. If autorun is disabled, you can manually start the installer from the CD.

- Use Windows Explorer to navigate to the CD-ROM directory. Double-click **Autorun.exe**.
- Start the Get-Services installation from the Windows command prompt. Type **D:\>autorun** where `D` identifies the CD-ROM drive. Substitute your CD-ROM drive identifier.



2 Click **Install Get-Services** to start the setup wizard.

3 At the Welcome screen, click **Next**.

4 Accept the License Agreement and click **Next**.

5 Choose your destination directory and click **Next**.

6 Confirm the setting of your Java installation and click **Next**.

7 Click **Install** to install Peregrine Portal 4.2.

The command prompt window shows what is being deployed and added. This may take several minutes.

8 When the installation is complete, click **Finish**.

9 Click **Exit Install**.

The installer creates the `oaa.war` and `oaadeploy.properties` files in the `Portal` directory. The default location is: `C:\Program Files\Peregrine\Portal`.

File	Purpose
`oaa.war`	Most of the application server configurations require this file to deploy the application.
`oaadeploy.properties`	This file is useful if you run `oaadeploy` to recreate the `image` directory or when you are installing a patch.

Installing multiple Peregrine Portal applications

When multiple Peregrine Portal applications, including Get-Services, are deployed, *all* applications must use the same version of OAA. Multiple Peregrine Portal applications that use different versions of OAA are not supported.

The `oaa.war` file contains the files that you need for each application. When you install additional applications, the installer appends the application files you need to the `oaa.war` file.

Important: Make sure that you use the same installation directory for each Peregrine Portal application.

To install multiple Peregrine Portal applications:

1. Back up your `local.xml` file.
2. Use the application server to stop and uninstall existing Peregrine Portal applications.

   For Tomcat, stop Tomcat, then delete the OAA deployment directory and `oaa.war` file from Tomcat's directory structure.

   For example, if Tomcat 4.1 is your application server, the default path is: `C:/Program Files/Apache Group/Tomcat 4.1/webapps/oaa`
3. Run the installer (see page 83) for the application.
4. Use the application server's deployment mechanism to deploy the WAR file that contains all the applications.

   For Tomcat, merely place the WAR file into the `webapps` directory.
5. Copy the `local.xml` file into the deployment directory's WEB-INF directory.
6. Use the application server to start the Peregrine Portal.

   For Tomcat, start Tomcat.

Configuring Get-Services

This section contains information on how to use the Administration function to configure the adapter connections. The settings are stored in the `local.xml` file.

To configure Get-Services:

1. Log on to the Peregrine Portal Administration login page (`admin.jsp`).
2. From the Administration menu, click **Settings**.
3. Click the **GICommonDB** tab.
   a. Specify the **Alias for** target configuration. For more information, see the Get-Services Configuration chapter in this guide.
   b. Configure the PortalDB and Web Application adapters.
4. Click **Save**.
5. From the Administration menu, click **Control Panel**.
6. Click **Reset Peregrine Portal**.

Uninstalling Get-Services from Windows

Use the Windows **Add or Remove Programs** option from the Control Panel to remove Get-Services.

To uninstall Get-Services:

1 From the Windows Control Panel, select **Add or Remove Programs**.

2 Scroll to **Peregrine Portal 4.2** and click **Remove**.

3 Click **Yes** when asked to remove Peregrine Portal 4.2 from your computer.

4 From Windows Explorer, navigate to the installation files.

The default path is: `C:/Program Files/Peregrine/Portal`.

5 Delete the Portal directory and its contents.

6 From your application server, remove the deployment directory and its contents.

CHAPTER 3 Upgrading Get-Services on Application Servers

The installation program no longer deploys files directly into an application server's directory structure. Instead, files are deployed into a *staging area* on the file system. This staging area functions as a common place into which one or more Peregrine Portal applications are deployed. From these deployed files, the installation program builds a WAR file and terminates.

Important: After the installation program executes, you must perform additional steps to complete the upgrade process.

You must backup any customizations from previous versions and backup certain configuration files, then uninstall the existing Peregrine Portal application from the application server using the application server's uninstall mechanism.

You then merge the backed up customizations and configuration files with the application files in the staging area, and re-create the WAR file. With the WAR file, use the application server's deployment process to install the upgraded Peregrine Portal application. This process of modifying files in the staging area, then generating the WAR file, is a best practice for customizations and future upgrades.

This WAR file distribution complies with the J2EE application deployment process and enables one WAR file, containing the Peregrine Portal application(s), to be conveniently deployed into separate test and production environments.

Once installed and running in the application server, an administrator can then access the Get-Services Administration page to complete the setup.

Upgrading Get-Services on Tomcat 4.1.x and 5.0.x

If you are upgrading from Get-Services 4.1.x to Get-Services 4.2, complete the following steps.

Step 1 Backup the `local.xml` file and any customizations from your current installation of Get-Services. See page 90.

Step 2 Uninstall Get-Services from Tomcat. See page 91.

Step 3 Run the Get-Services installer. See page 91.

Step 4 Apply the backed up `local.xml` file and customizations to the staging area created by the installer. See page 91.

Step 5 Generate the WAR file from the files in the staging area. See page 91.

Step 6 Copy, or FTP, the WAR file to the Tomcat server machine, into the Tomcat's `webapps` directory. See page 92.

Step 7 Verify XML parser files are the same version as those from the installer. See page 92.

Step 8 Start Tomcat and complete the Get-Services upgrade. See page 92.

Backing up the local.xml file and any customizations

Before you make changes to any files, create a backup copy.

1 Backup the `local.xml` file. The default location is:

```
<tomcat_home>webapps\oaa\WEB-INF\local.xml
```

where `<tomcat_home>` is the root directory in which Tomcat is installed.

2 Backup any schema extensions and any customization of themes created for the existing version. First read Completing the Get-Services upgrade on page 101 to familiarize yourself with how certain customizations will be performed later.

Uninstalling Get-Services from Tomcat

Prior to uninstalling Get-Services, make note of the OAA application directory name.

1 Note the name of the OAA application directory installed in `<tomcat_home>\webapps`. The default is **oaa**, such as `<tomcat_home>\webapps\oaa`. You will need to give the WAR file you will later generate the same name.

2 Stop Tomcat.

3 Delete the above application directory.

4 Delete any OAA WAR file existing in `<tomcat_home>\webapps`, like `oaa.war`.

Running the Get-Services Installer

Follow the instructions in Running the installer on page 67.

Applying backed up files to the staging area

The installation program deploys files into a staging area where one or more Peregrine Portal applications are deployed. Back up the following files to this staging area.

1 Copy the backed up `local.xml` file into the `<install_dir>\image\WEB-INF` directory where `<install_dir>` is the directory in which you directed the installer to deploy its files. The default is `C:\Program Files\Peregrine\Portal`.

2 Apply your backed up customizations to the files in the `<install_dir>\image` directory.

Generating the WAR file

The following steps generate a WAR file.

1 From the command prompt, change your current working directory to `c:\Program Files\Peregrine\Portal\image`.

2 From the command line, type:

```
"<Java-path>\bin\jar.exe cvf <directory-where-war-file-will-be>\oaa.war ."
```

where `<Java-Path>` is the path to your Java installation.

Important: The command must contain the period after following `oaa.war`.

3 Press Enter.

Important: Name the WAR file the same name you noted in Step 1 on page 91 of Uninstalling Get-Services from Tomcat. For example, if the directory name was **myOaa**, name the WAR file **myOaa.war**.

Copying the WAR file to the Tomcat server machine

Copy, or FTP, the WAR file you just generated into the `<tomcat_home>\webapps` directory.

Verifying XML parser files

Make sure that you have the correct version of the JAR files.

1 If the Java JDK Tomcat is using version 1.3.X, follow the instructions in Copying the jar files on page 7 to copy the latest version of the XML parser files. However, instead of the destination directory of `<tomcat_home>\common\endorsed`, use `<java_home>\jre\lib\ext`.

where `<java_home>` is the root directory of the Java JDK that Tomcat is using.

2 If the Java JDK Tomcat is using version 1.4.X, follow the instructions in Copying the jar files on page 7 to copy the latest version of the XML parser files.

Starting Tomcat and upgrading Get-Services

You must start the application server to deploy the WAR file.

1 Start Tomcat to have it automatically deploy the upgraded WAR file.

2 Follow the steps in Completing the Get-Services upgrade on page 101.

Upgrading Get-Services on WebLogic 6.1 and 8.1

If you are upgrading from Get-Services 4.1.x to Get-Services 4.2, complete the following steps.

Step 1 Backup the local.xml file and any customizations from your current installation of Get-Services. See page 94.

Step 2 Uninstall Get-Services from WebLogic 8.1. See page 94.

Step 3 Run the Get-Services installer. See page 94.

Step 4 Apply the backed up `local.xml` file and customizations to the staging area created by the installer. See page 95.

Step 5 Generate the WAR file from the files in the staging area. See page 95.

Step 6 Copy, or FTP, the WAR file to the WebLogic server machine, into the WebLogic's `webapps` directory. See page 95.

Step 7 Verify WebLogic has the latest XML parser files. See page 96.

Step 8 Deploy the WAR file to the WebLogic server. See page 96.

Step 9 Complete the Get-Services upgrade. See page 96.

Backing up the local.xml file and any customizations

Before you make changes to any files, create a backup copy.

1 Backup the configuration `local.xml` and `web.xml` files. The default location for the `local.xml` file is

```
<APP_DEPLOYMENT_DIRECTORY>/WEB-INF/local.xml
```

where `<APP_DEPLOYMENT_DIRECTORY>` is the deployment directory of the Peregrine Portal within the WebLogic Server. For example: `app\peregrine\applications\oaa`.

2 Backup any schema extensions and any customization of themes created for the existing version. First read Completing the Get-Services upgrade on page 101 to familiarize yourself with how certain customizations will be performed later.

Uninstalling Get-Services from WebLogic

Prior to uninstalling Get-Services, make note of the Web Module name.

1 Stop WebLogic.

2 Note the name of the Web Module installed. You will need to redeploy Get-Services using the same Web Module later.

3 Delete the Web Application Module using the WebLogic User Interface (UI).

4 Delete OAA files from the file system.

Running the Get-Services Installer

Follow the instructions in Running the installer on page 67.

Applying backed up files to the staging area

The installation program deploys files into a staging area where one or more Peregrine Portal applications are deployed. Back up the following files to this staging area.

1 Copy the backed up `local.xml` file into the `<install_dir>\image\WEB-INF` directory where `<install_dir>` is the directory in which you directed the installer to deploy its files. The default is `C:\Program Files\Peregrine\Portal`.

2 Apply your backed up customizations to the files in the `<install_dir>\image` directory. Follow the appropriate steps provided in the section Migrating Get-Services Customizations in this chapter.

Generating the WAR file

The following steps generate a WAR file.

1 From the command prompt, change your current working directory to `c:\Program Files\Peregrine\Portal\image`.

2 From the command line, type:

```
"<Java-path>\bin\jar.exe cvf <directory-where-war-file-will-be>\oaa.war ."
```

where `<Java-Path>` is the path to your Java installation.

Important: The command must contain the period after following `oaa.war`.

3 Press Enter.

Copying the WAR file to the WebLogic Server machine

Copy, or FTP, the WAR file you just generated into the WebLogic server machine and extract to the `<WebLogic>\applications` directory. See Copying the installed files to your WebLogic directory on page 29 for details.

Verifying WebLogic has the latest XML parser files

Make sure that you have the correct version of the JAR files.

- If you are using WebLogic 6.1, follow the steps in Moving jar files to the Java Development Kit ext directory on page 26.
- If you are using Weblogic 8.1, follow the steps in Copying jar files on page 33.

Deploying the WAR file to the WebLogic server

Use the WebLogic User Interface (UI) to deploy using the same Web Module name noted in Step 2 on page 94 of Uninstalling Get-Services from WebLogic.

Completing the Get-Services upgrade

Follow the steps in Completing the Get-Services upgrade on page 101 to complete the upgrade.

Upgrading Get-Services on WebSphere 5.0.2 and 5.1

If you are upgrading from Get-Services 4.1.x to Get-Services 4.2, complete the following pre-upgrade steps.

To prepare for the WebSphere upgrade:

1 Remove the following lines from the current `local.xml`. file.

```
<SSLProvider>com.ibm.jsse.JSSEProvider</SSLProvider>
<HTTPSHandlerPkg>com.ibm.net.ssl.internal.www.protocol</HTTPSHandlerPkg>
<CryptoProvider>com.ibm.crypto.provider.IBMJCE</CryptoProvider>
```

2 For WebSphere 5.0.2 only, remove the following files from `$JAVA_HOME/jre/lib/ext`.

```
oaasecurityproxy.jar
pop3.jar
js.jar
jai_core.jar
jai_codec.jar
mlibwrapper_jai.jar
xalan.jar
xercesImpl.jar
xml-apis.jar
```

3 Verify that there are no added entries for the following:

- On Windows, remove any path entries to the ServiceCenter libraries placed in the `$PATH` environment variable.
- On Solaris or Linux, verify that `startServer.sh` has no added entries for the `LD_LIBRARY_PATH` variables.
- On AIX, remove any added path entries in `startServer.sh` for the `LIBPATH` variable, except for the path to the ServiceCenter libraries.

  For example, when running ServiceCenter 5, keep this entry:

```
LIBPATH=/WebSphere/AppServer/installedApps/oaa.ear/oaa.war/
WEB-INF/lib/AIX/ServiceCenter5
```

For example, when running ServiceCenter 6, keep this entry:

```
LIBPATH=/WebSphere/AppServer/installedApps/oaa.ear/oaa.war/
WEB-INF/lib/AIX/ServiceCenter6
```

When you complete the pre-upgrade steps for WebSphere, continue with the following instructions.

Step 1 Backup the `local.xml` file and any customizations from your current installation of Get-Services. See page 98.

Step 2 Uninstall Get-Services from WebSphere. See page 99.

Step 3 Run the Get-Services installer. See page 99.

Step 4 Apply the backed up `local.xml` file and customizations to the staging area created by the installer. See page 99.

Step 5 Generate the WAR file from the files in the staging area. See page 99.

Step 6 Copy, or FTP, the WAR file to the WebSphere server machine. See page 100.

Step 7 Copy required runtime libraries. See page 100.

Step 8 Use WebSphere to deploy the WAR file. See page 100.

Backing up the local.xml file and any customizations

Before you make changes to any files, create a backup copy.

1 Backup the `local.xml` file. The default location is

```
<websphere_home>\AppServer\installedApps\<server_name>
\oaa.ear\oaa.war\WEB-INF\local.xml.
```

where `<websphere_home>` is the root directory in which WebSphere is installed and `<server_name>` is the name of your server.

2 Backup any schema extensions and any customization of themes created for the existing version. First read Completing the Get-Services upgrade on

page 101 to familiarize yourself with how certain customizations will be performed later.

Uninstalling Get-Services from WebSphere

Prior to uninstalling Get-Services, make note of the Context Root and Application names.

1 Log on to the WebSphere Administrative Console.

2 Note the name of the Get-Services Context Root installed in WebSphere. The default is **oaa**. Also note the Application Name. You will need to provide these names when you install the upgraded WAR file.

3 Uninstall Get-Services using the WebSphere Administrative Console.

Running the Get-Services Installer

Follow the instructions in Running the installer on page 67.

Applying backed up files to the staging area

The installation program deploys files into a staging area into which one or more Peregrine Portal applications are deployed. Back up the following files to this staging area.

1 Copy the backed up `local.xml` file into the `<install_dir>\image\WEB-INF` directory where `<install_dir>` is the directory in which you directed the installer to deploy its files. The default is `C:\Program Files\Peregrine\Portal`.

2 Apply your backed up customizations to the files in the `<install_dir>\image` directory.

Generating the WAR file

The following steps generate a WAR file.

1 From the command prompt, change your current working directory to `c:\Program Files\Peregrine\Portal\image`.

2 From the command line, type:

```
"<Java-path>\bin\jar.exe cvf <directory-where-war-file-will-be>\oaa.war ."
```

where `<Java-Path>` is the path to your Java installation.

Important: The command must contain the period after following `oaa.war`.

3 Press Enter.

Copying the WAR file to the WebSphere server machine

Copy, or FTP, the WAR file you just generated to the WebSphere server machine, or verify the WAR file is accessible from the WebSphere server machine.

Copying required JVM libraries

The installer generates `jar` files that you must copy to the application server.

1 If your WebSphere version is 5.0.2, follow the instructions in Copying jar files on page 55 to copy required runtime libraries.

2 If your WebSphere version is 5.1, follow the instructions in Copying jar files on page 61 to copy required runtime libraries.

Using WebSphere to deploy the WAR file

The `oaa.war` file creates the directory structure necessary to deploy Get-Services to your application server.

1 Follow the steps in Deploying the oaa.war file on page 54 (for WebSphere 5.0.2) or Deploying the oaa.war file on page 60 (for WebSphere 5.1) to deploy the upgraded WAR file.

Important: Set the Context Root and Application Name to the names you noted in Step 2 on page 99 in Uninstalling Get-Services from WebSphere.

2 Follow the steps Editing the application settings on page 56 (for WebSphere 5.0.2) or Editing the application settings on page 62 (for WebSphere 5.1)to set class loading to `PARENT_LAST` and to regenerate the WebSphere plug-in.

3 Stop and start the Administrative Console.

4 Follow the procedures in Completing the Get-Services upgrade on page 101 to complete the upgrade process.

Completing the Get-Services upgrade

Follow the procedures in this section to complete the upgrade process.

Re-creating customizations in Get-Services 4.2

You cannot directly migrate customizations implemented before version 4.0.1 to Get-Services 4.2. Instead, you must re-create your changes using the new features and methods available in Get-Services 4.2.

The following sections describe how to re-create your customizations from previous versions.

No customizations

If you have made no customizations to Get-Services, you can simply install Get-Services 4.2 on a new system and migrate your data from your existing back-end database.

Note: Get-Services 4.1 requires ServiceCenter 5.1.x or higher.

Customized JSP files

In previous versions, users had to directly modify JSP files in order to add or remove certain functionality. The following table describes how to re-create 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-Services 4.2 offers many more pages that you can personalize directly from the Web interface. If you personalized pages in a previous version, you must re-create your personalized pages in Get-Services 4.2 using DocExplorer. Refer to the Get-Services Administration Guide for information on personalization.

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-Services 4.2 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 can copy over older custom themes to Get-Services 4.2, you may experience rendering errors due to the new images, CSS definitions, frame definitions, and layers. It is recommended that you re-create any custom themes using the Get-Services 4.2 version of the classic theme as your template. Refer to the Get-Services Administration Guide for information on customizing.

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 re-create these customizations using the updated instruction. You can find information about alternate security methods in the Get-Services 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-Services Web interface. The following table describes how to re-create some of the more common tailoring kit changes.

Tailoring kit modification	New method to use
Added or removed fields from a form	Add or remove fields from Personalization
Added a new language or locale to the Get-Services 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 must re-create them in the current version of the Get-Services tailoring kit.

Configuring an existing back-end database for Get-Services 4.2

The following tables lists the options available for data migration.

Get-Services 2.3 to Get-Services 4.2

Back-end version	Migration required
ServiceCenter 5.1.x	Apply Get-Services 4.2 unload files to existing ServiceCenter 5.1.x
ServiceCenter 6.0	Apply Get-Services 4.2 unload files to existing ServiceCenter 6.0

Preserving customized web.xml file settings

You can open the `web.xml.xxx.bak` file, copy its customized configuration settings into the new `web.xml` file, and save the new file to preserve your customizations.

Note: Note that the `<appsrv>\WEB-INF\web.xml` file has been renamed to `<appsrv>\WEB-INF\web.xml.xxx.bak`, where `<appsrv>` is the path to

your application server, and xxx represents a unique sequence of characters. This preserves any customizations that you might have.

After upgrading Get-Services to 4.2, a new file called web.xml.xxx.bak is created in the <appsrv>/WEB-INF directory. You must reconcile the contents of this file with the contents of the new web.xml file. Any portion of the web.xml.xxx.bak file that does not exist in the new web.xml file needs to be added to the web.xml file.

CHAPTER 4 Load Balancing the Servers

This chapter covers the following topics:

Load balancing application servers

A server running a Web application such as Peregrine's Get-Services, Get-Answers, 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 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:

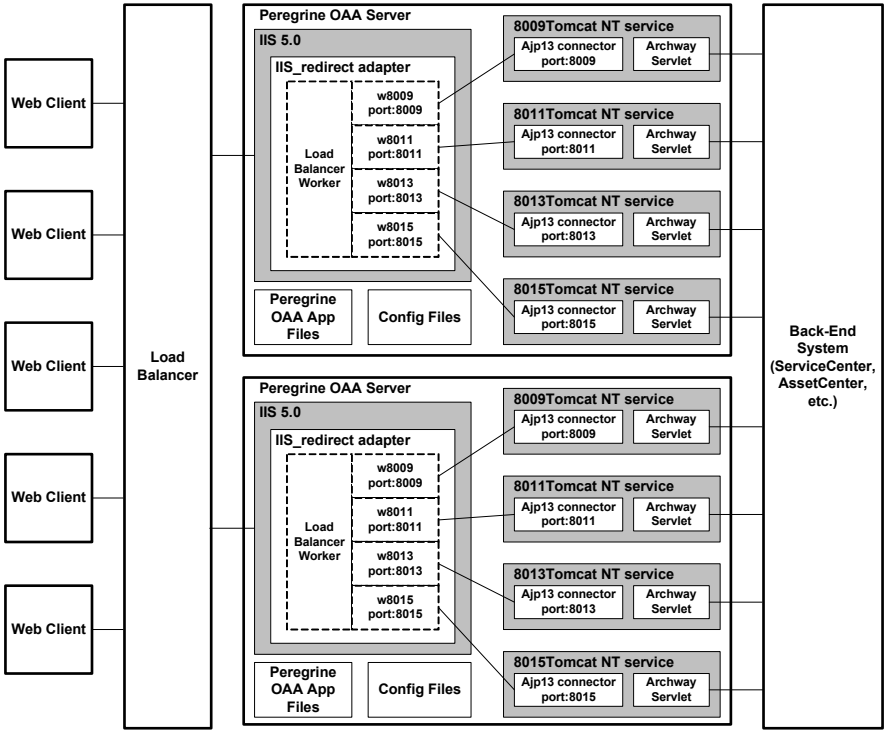
- The number of concurrent users.
- The processing power of the machine hosting the Get-Services Web server.
- The number of processors on the machine.

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 BizDocAdapter, performance beyond the 400-concurrent-user benchmark can be achieved.

The following diagram illustrates the architecture of multiple JVMs.



Note: A white paper on Peregrine OAA architecture and optimization is available on the Customer Support Web site at *http://support.peregrine.com*. In the Search Knowledge Base question field, type optimize and click Go. Click Get-It: Guide for planning and implementing OAA-based applications in the search results list and download the document.

Load balancing with Apache and Tomcat

Note: The following procedures assume that you already installed Get-Services and configured it with Tomcat and Apache. Refer to the installation chapter for more information on installing Get-Services.

Step 1 Create multiple instances of Tomcat. See Creating multiple instances of Tomcat on page 112.

Step 2 Edit Apache's `mod_jk2.conf` file to establish a connection between Tomcat and Apache. See Editing the mod_jk2.conf file on page 108.

Step 3 Edit the `httpd.conf` file to define the Tomcat workers available for Apache. See Editing the httpd.conf file on page 109.

Step 4 Test the configuration. See Testing load balancing on Apache on page 109.

Creating multiple instances of Tomcat

See Creating multiple instances of Tomcat on page 112.

Editing the mod_jk2.conf file

The `mod_jk2.conf` file defines where the `worker` files are available in Apache. Edit `mod_jk2.conf` only after successful deployment of necessary Get-Services files; otherwise the Get-Services mount points, file locations, and directories are not included in the `mod_jk2.conf` file and have to be manually added.

To edit the mod_jk2.conf file:

1 Make a copy of the `mod_jk2.conf` file and rename the copy to `mod_jk2.conf.backup`.

The `mod_jk2.conf` file is located in the `Apache/conf` directory.

2 Open the `mod_jk2.conf` file in any text editor.

3 Verify that JkSet points to the `workers2.properties` file in the `Apache/conf` directory. For example:

```
JkSet config.file "C:\Program Files\Apache Group\Apache2
\conf\workers2.properties"
```

4 Save the file.

Editing the httpd.conf file

The `httpd.conf` file must include `mod_jk2.conf`.

To edit the httpd.conf file:

1 Open the `httpd.conf` file in any text editor.

The default location is: `C:\Program Files\Apache Group\Apache2\conf`.

2 Update the following line to point to Apache's `mod_jk2.conf` file:

```
include "<Apache>/conf/mod_jk2.conf"
```

For `<Apache>` enter the path to your Apache installation. The default file path is: `C:\Program Files\Apache Group\Apache2\Tomcat4.1`

3 Save the file.

Testing load balancing on Apache

After you create additional Tomcat instances, you can test load balancing.

To test load balancing:

1 Start all Tomcat instances.

If you are running on a Windows operating system and you installed Tomcat as a service, open the Windows Control Panel and start each instance from the Services dialog box.

2 Start the Apache web server.

3 Open a browser and log in to Get-Services.

4 Perform an action in Get-Services, such as a search.

5 Leave the browser open.

6 Repeat steps 3 through 5 for the same number of times as the number of Tomcat instances you created.

The load balancing mechanism uses a round-robin algorithm. If load balancing is working correctly, each login should use a different Tomcat instance.

7 Open the `archway.log` file from a Tomcat instance in a text editor. By default, there should be on `archway.log` file in each of the Tomcat instances working directory.

8 Verify that each `archway.log` contains log messages that state the port number on which the Tomcat instance is listening.

For example, if you have set on the Administration page the logging level to DEBUG, there will be log messages such as, "Beginning a new session:<session number>.<hostname>.<port>" where <port> is the port number on which a Tomcat instance is listening.

If a Tomcat instance does not have its own `archway.log` containing text with the rquired port number, then the system is not load balancing and needs troubleshooting.

Load balancing with IIS and Tomcat

You can create multiple instances of Tomcat to load balance request to Get-Services. Although this is not required, it improves performance, makes the instances easier to manage, and provides extra functionality. For example, you can restart the service if it fails or if the machine on which the instances are installed needs to be restarted.

Note: The following procedures assume that you already installed Get-Services and configured it with Tomcat and IIS. Refer to the installation chapter for more information on installing Get-Services.

Step 1 Complete the steps described in Creating multiple instances of Tomcat on page 112.

Step 2 Test the configuration. See Testing load balancing on IIS on page 111.

Creating multiple instances of Tomcat

See Creating multiple instances of Tomcat on page 112.

Testing load balancing on IIS

After you have created additional Tomcat instances, test load balancing.

To test load balancing:

1 Start all Tomcat instances.

   If you are running on a Windows operating system and you installed Tomcat as a service, open the Windows Control Panel and start each instance from the Services dialog box.

2 Start the IIS web server.

3 Open a browser and log in to Get-Services.

4 Perform an action in Get-Services, such as a search.

5 Leave the browser open.

6 Repeat steps 3 through 5 for the same number of times as the number of Tomcat instances you created.

   The load balancing mechanism uses a round-robin algorithm. If load balancing is working correctly, each login should use a different Tomcat instance.

7 Open the `archway.log` file from a Tomcat instance in a text editor. By default, there should be on `archway.log` file in each of the Tomcat instances working directory.

8 Verify that each `archway.log` contains log messages that state the port number on which the Tomcat instance is listening.

For example, if you have set on the Administration page the logging level to DEBUG, there will be log messages such as, "Beginning a new session:<session number>.<hostname>.<port>" where <port> is the port number on which a Tomcat instance is listening.

If a Tomcat instance does not have its own `archway.log` containing text with the rquired port number, then the system is not load balancing and needs troubleshooting.

Creating multiple instances of Tomcat

You can create multiple instances of Tomcat to load balance request to Get-Services. Although this is not required, it improves performance, makes the instances easier to manage, and provides extra functionality. For example, you can restart the service if it fails or if the machine on which the instances are installed needs to be restarted.

To create multiple instances of Tomcat:

Step 1 Create copies of the Tomcat directory, then delete the `\webapps\oaa` directory from the newly copied instances of Tomcat. See Copying the Tomcat directory on page 113.

Step 2 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 on page 114.

Step 3 Edit the `server.xml` files for each Tomcat instance. See Editing the server.xml files on page 115.

Step 4 Edit the `jk2.properties` files for each Tomcat instance. See Editing the jk2.properties files on page 118.

Step 5 For Windows operating systems, install multiple instances of Tomcat as a service using `installservice.bat`. Installing Tomcat instances as services on page 119.

Copying the Tomcat directory

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

Important: Stop Tomcat if it is still running before you begin copying a Tomcat directory.

To copy the Tomcat directory:

1 If you are running on a Windows operating system, copy the

`<Portal_Install_Dir>/SupportFiles/AppServerFiles/`

`installservices.bat` file to the `<Tomcat>/bin` directory,

where `<Portal_Install_Dir>` is the path you directed the installer to deploy its files and `<Tomcat>` is the root of the Tomcat installation.

2 Copy the Tomcat install folder. For example, `C:\Program Files\Apache Group\Tomcat.`

3 Paste the folder into the same root path. For example, `C:\Program Files\Apache Group`

4 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 want four instances of Tomcat listening on ports 8009, 8011, 8013, and 8015, create four copies of the Tomcat folder and name them `\Tomcat_8009`, `\Tomcat_8011`, `\Tomcat_8013`, and `\Tomcat_8015`.

Warning: If you are using more than four Tomcat instances, change the port numbers to avoid conflicts.

5 Delete the \webapps\oaa subdirectory from the newly copied instance of Tomcat. Additional instances will use the same document root as the first or primary Tomcat instance.

6 Repeat steps 1 through 5 for each instance of Tomcat you want to use.

Editing the workers2.properties file

For each Tomcat instance installed, there is only one workers2.properties file needed. This file is shared by all other Tomcat instances on that particular server.

The workers2.properties file specifies the worker threads that the Web server connector creates 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.

To edit the worker2.properties file:

1 If you installed:

- Apache with Tomcat, edit the workers2.properties in the Apache/conf directory.
- IIS with Tomcat, edit the workers2.properties in the Tomcat/conf directory of your primary Tomcat instance.

2 Open the workers2.properties file in any text editor.

3 Create a channel.socket entry for each Tomcat instance.

Example:

```
[channel.socket:<server>:<port>]
lb_factor=1
```

Note: For the existing `channel.socket` entry only, replace `host=localhost port=8009` with the line, `lb_factor=1`.

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

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

The `lb_factor` is a number greater than or equal to 1 used 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 more user sessions to a given worker, then assign it a lower `lb_factor` number relative to the other workers.

4 Create an `AJP13` worker for the `channel.socket` defined in the previous step.

```
[ajp13:<server>:<port>]
channel=channel.socket:<server>:<port>
group=lb
```

Note: For the existing `ajp13` entry, verify `<server>` and `<port>` are correct.

Where `<server>` and `<port>` are the values you have for the `channel.socket` definition.

5 Save the file.

Editing the server.xml files

A separate `server.xml` file is required for each Tomcat instance. The `server.xml` file contains the information that Tomcat needs to connect to the Web server and to find Peregrine OAA Platform Web application files.

Tip: Make a back-up copy of the `server.xml` file before editing it.

To edit the server.xml files:

1 Each Tomcat instance has a `server.xml` file located in the `conf` directory. Open `server.xml` in any text editor.

2 Update the port number attribute of the <Server> element to a unique value that does not conflict with other port numbers used by Tomcat.

Peregrine Systems recommends that you use the port numbers 8005-8008 when configuring four Tomcat instances.

Warning: If you are using more than four Tomcat instances, change the port numbers to avoid conflicts.

For example:

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

3 Update the port number attribute of the Coyote/JK2 AJP 1.3 Connector <Connector> element to a unique value that will not conflict with other port numbers used by Tomcat.

Peregrine Systems recommends that you use port numbers 8009, 8011, 8013, and 8015 when configuring the Coyote Connector. Your primary Tomcat instance may already have this set to 8009, and you will only need to modify the other instances.

Example:

```
<Connector className="org.apache.coyote.tomcat4.1.CoyoteConnector"
   port="8009" minProcessors="5" maxProcessors="75"
   enableLookups="true" redirectPort="8443" acceptCount="10"
   debug="0" connectionTimeout="20000" useURIValidationHack="false"
   protocolHandlerClassName=”org.apache.jk.server.JkCoyoteHandler” />
```

Note: Change port number only. Leave all other values the same.

4 Create a <Context> element entry from the first or primary Tomcat instance and copy it to the other Tomcat instances.

Example:

```
<Server port="...">
   <Service name="...">
      <Engine name="...">
         <Host name="...">
            <Context path="/oaa"
            docBase="<First_Tomcat_install>/webapps/oaa"
            crossContext="false"
            debug="0"
            reloadable="false" >
            </Context>
         </Host>
      </Engine>
   </Service>
</Server>
```

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

5 Update, or add if necessary, the jvmRoute attribute of the <Engine> element with the server name and communications port used by each Tomcat instance using port number 8009, 8011, 8013, or 8015 as appropriate. For example:

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

Important: Verify that you are not modifying an <Engine> element that is surrounded by comment tags (<!-- -->) and thus commented out. Tomcat 5 has one <Engine> element commented out and another that is not.

6 Update the <Host> element with the webapps directory used by the first or primary Tomcat instance. List the server information in the appBase attribute. For 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 primary Tomcat instance.

7 Comment out port 8080 in the non-SSL Coyote HTTP... statement. For example:

```
<!-- Define a non-SSL Coyote HTTP/1.1 Connector on port 8080 -->
<!--
<Connector className="org.apache.coyote.tomcat4.1.CoyoteConnector"
port="8080" minProcessors="5" maxProcessors="75"
acceptCount="10" debug="0" connectionTimeout="20000"
useURIValidationHack="false" />
-->
```

8 Save the server.xml file.

9 Repeat Step 2 through Step 8 for each copy of the server.xml file you created.

Editing the jk2.properties files

You modify the jk2.properties file for each Tomcat instance. The jk2.properties 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 at the bottom of the file for the channelSocket port. The port number must match the port number defined in workers2.properties file for this Tomcat instance. For example:

```
channelSocket.port=8009
```

3 Save the `jk2.properties` file.

4 Repeat Step 1 through Step 3 for each Tomcat instance.

Installing Tomcat instances as services

Important: Perform these steps only if Tomcat 4.1.x is running on a Windows operating system. For Tomcat 5.0.x refer to `Apache/Tomcat5` documentation on Apache's website for instructions on how to install Tomcat as a service.

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

To install Tomcat instances as services:

1 Open a DOS command prompt and change directories to the `bin` directory of your Tomcat instance.

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 assign to 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.

Note: The `<service_name>`, `<tomcat_home>`, and `<jvm_dll_path>` cannot contain spaces.

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

Note: The command to create Tomcat instances cannot accept spaces in the file path. The following example avoids spaces in the file path name by replacing `Program Files` with `Progra~!`.

```
installservice Tomcat8009 C:\Progra~!\Apache~1\Tomcat4_8009
C:\Progra~!\jdk1.3.1_05\jre\bin\server\jvm.dll
```

3 Repeat Step 1 and Step 2 for each Tomcat service you wish to create.

Tip: You can easily remove a service. From the DOS command prompt, change directories to the `bin` directory of your Tomcat instance, then enter the following command: `tomcat -Uninstall <service name>`.

Start each Tomcat instance that you install.

CHAPTER 5 Back-end System Configuration

Get-Services uses ServiceCenter as its back-end database. ServiceCenter authenticates user access rights and stores personalization changes.

- With ServiceCenter as the back-end, Get-Services accesses ServiceCenter Incident Management. Using the Get-Services Admin module, you can add the Service Management component as well.
- Get-Services Change Management Module integrates with ServiceCenter Change Management. It is only available with ServiceCenter 5.x.

Important: To continue the installation for Get-Services, an administrator must perform the procedures in this chapter within the back-end system. For detailed information on how to perform any of the steps listed in the following sections, refer to the ServiceCenter System Administrator's Guide .

Configuring ServiceCenter for Get-Services

Get-Services accesses the ServiceCenter 5.1 and ServiceCenter 6.0 Incident Management and Service Management modules. In addition, Get-Services can access the ServiceCenter 5.1 and ServiceCenter 6.0 Change Management module.

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

Applying the unload files to ServiceCenter

The next step in the Get-Services installation involves installing updated RAD codes, event register records, event maps, and processes that are contained in unload files. The unload files that you apply depend on how you intend to use Get-Services. You apply different unload files to ServiceCenter 5.1 and ServiceCenter 6.0. See the respective tables in the following sections.

The `\oaa` directory referenced in the following tables is a subdirectory of the directory where the Peregrine OAA Server files are installed. For example, if you installed Tomcat 4.1.29 as your Peregrine OAA server, the `\oaa` directory is a subdirectory of the `\Tomcat4\webapps` directory on the machine where you installed Tomcat 4.1.29.

Warning: This procedure overwrites existing ServiceCenter files when you apply the new unload files. If you tailored ServiceCenter, you will lose changes made to the original files.

ServiceCenter 5.1 unload files with Get-Services 4.2

The path for the unload files is in `oaa\WEB-INF\etc\Version51\.`

This file	Contains this information
apm.bg.edit.unl	rad: apm.bg.edit
epmusc5_scr26513_24918.unl	rad: axces.apm apm.bg.edit
gsViewManager.unl	gsView gsCategoryFormat
gsadmincapability.unl	capability
sc51_scr25463.unl	rad:axces.sm
sc51_scr26822.unl	rad: axces.cm3
SCR33126.unl Note: The path to this unload is `oaa\WEB-INF\etc\sc51\.`	capability

ServiceCenter 6.0 unload files with Get-Services 4.2

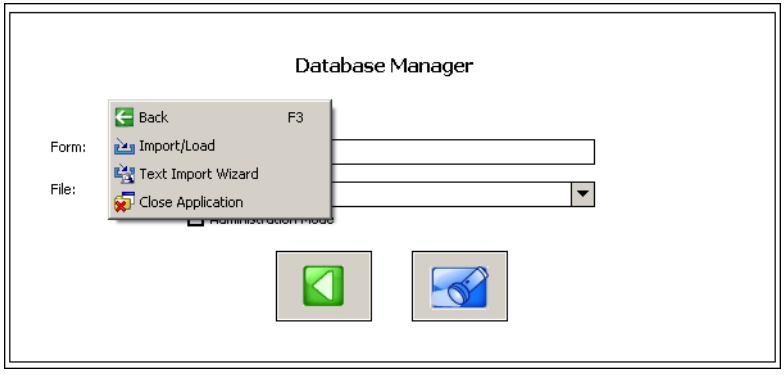
The path for the unload files is in `oaa\WEB-INF\etc\Version6\.`

This file	Contains this information
gsViewManager.unl	gsView gsCategoryFormat
gsadmincapability.unl	capability
sc6_scr30451.unl	eventregister:epmo
sc6_scr30158.unl	eventmap:e problem open e problem update
sc6_scr30981	event:changerequestcancel
SCR33126 Note: The path to this unload is `oaa\WEB-INF\etc\sc6\.`	capability

To apply unload files to ServiceCenter:

1 From the ServiceCenter Navigation, click **Toolkit** > **Database Manager**.

2 Right-click anywhere on the screen to open the pop-up menu.



3 Select **Import/Load**.

4 In the **Filename** text box, browse to, or type, the path for the unload file that pertains to your particular integration. See the previous tables to determine which unload files pertain to your integration.

5 Click **Open**.

6 From the menu, click **Load FG**.

ServiceCenter displays a message indicating that the file is added.

7 Exit from ServiceCenter client.

CHAPTER 6 Get-Services Configuration

Final configuration settings necessary to complete the installation of Get-Services 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-Services.

A more detailed description of additional features necessary for administration and maintenance using the Peregrine Portal Administration utility is available in the Get-Services Administration Guide.

Note: Incident Management is called Problem Management in ServiceCenter versions prior to 4.x. Some parameters in Incident Management use *problem* terminology because they are mapped to *problem* tables in ServiceCenter.

This chapter covers the following topics:

Accessing the Peregrine Portal Admin module

The Peregrine Portal administrator login page accesses the Peregrine Portal Admin module. You use the Admin module to define the settings for your Peregrine system.

A default administrator, System, gives you access to the Admin module without being connected to a back-end system. After you configure your user name on the Common tab, you can also access the Admin module from the Navigation menu.

Important: When you change parameters using the Admin module, a `local.xml` file is created in the `\<appsrv>\WEB-INF` directory to store these parameters. If you reinstall Get-Services, make a copy of this file and store it outside your Get-Services installation. Failure to do this will result in your parameter values being lost during the new installation.

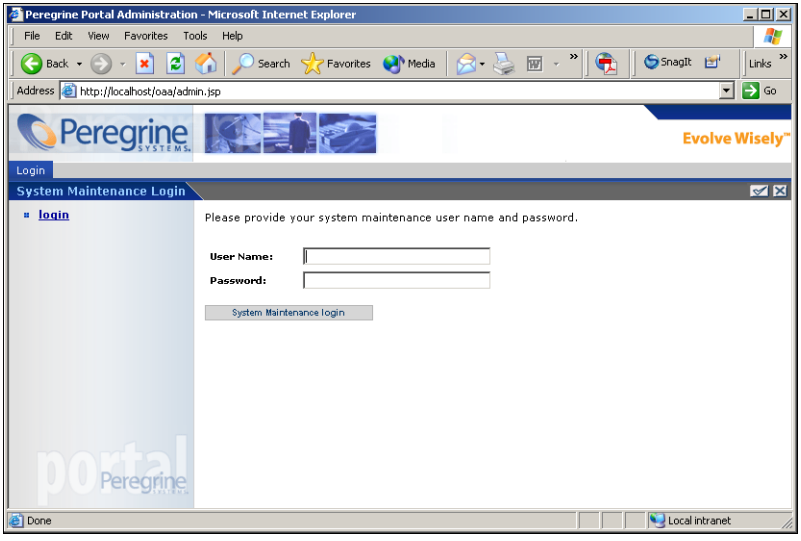
To access the Peregrine Portal administrator login page:

1 Verify that your application server (for example, Tomcat) is running.

2 In your Web browser Address field, type:

<hostname>/oaa/admin.jsp

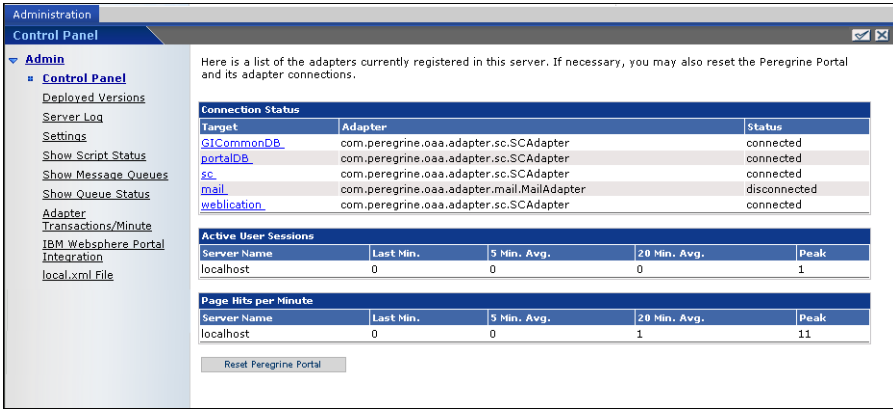
3 Click **Go** to open the Portal administrator login page.



4 In the Name field, type **System**.

No password is required on initial login. Refer to the Get-Services Administration Guide for detailed instructions on how to change the Administrator login name and password.

5 Click **System Maintenance login** to open the Control Panel page.



Activity menu

The left pane Activity menu provides access to a number of different administrative utilities for Get-Services. You need to access some of these during this installation process. The utilities and their functions are described in detail in the Get-Services Administration Guide.

Using the Control Panel

Use the Control Panel page to check the status of the connections to the databases you are accessing with Get-Services and your Web applications. You can also reset the connection between the Archway servlet and the adapters to the back-end systems.

Note: When you first access the Control Panel page, the status for all target adapters is **Disconnected**. The status changes after you define target adapters, as described later in this section.

All changes involving the Control Panel require the following steps.

Step 1 Log in to the Admin page with the username **System**.

Step 2 Go to the Settings page.

Step 3 Make your changes.

Step 4 Reset the portal server.

Using the Settings page

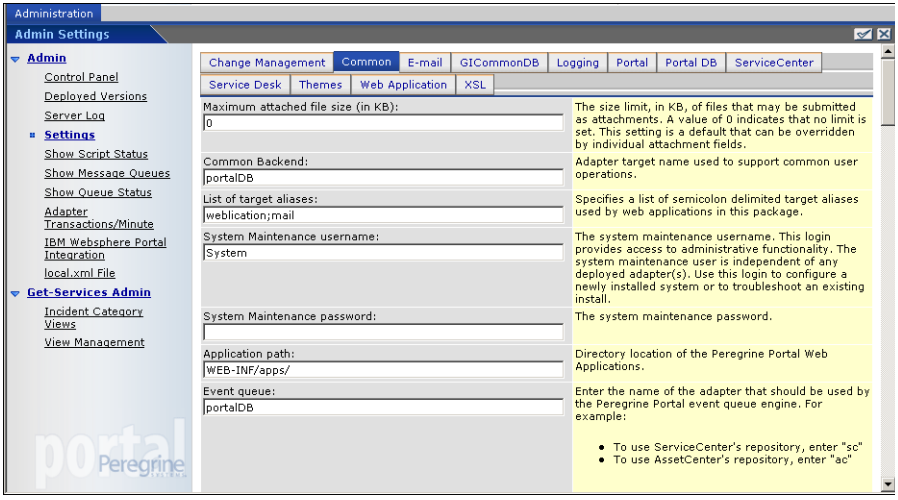
From the Admin module Activity menu, click **Settings** to open the current parameter settings. The Settings page is divided into tabs. The tabs that you see depend on the Web applications that you installed and the adapters that you use. The Common tab is available for all installations.

To open the Settings page:

1 From the Admin module, click **Settings**.

You use the tabs to configure settings for Get-Services.

By default, the Common settings page opens.



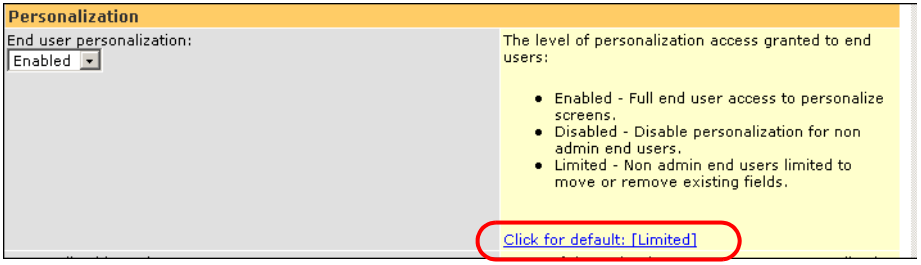
2 Click the appropriate tab to change setting parameters.

When you change the parameter default value for any setting and save it, the corresponding description column adds a link with the default value listed in brackets so that you can return to the default at any time.

To revert to the default setting:

1 Navigate to the appropriate setting parameter.

2 From the description column next to the changed parameter, select the **Click for default** link.



3 To return to the default setting, click **Click for default: [Limited]** and click **Save**.

The parameter returns to the default state.

Resetting the server

After making any configuration changes in the Peregrine Portal Administration module, the following informational message appears at the top of the page.

```
The settings have been saved. To ensure all changes take effect,
you must reset the Peregrine Portal server.
```

Return to the Control Panel to reset the server and apply your configuration changes to Get-Services.

To reset the connection between the Archway servlet and back-end system:

1 From Control Panel, click **Reset Peregrine Portal.**

When the operation is complete, the following message indicates that the connections are reset.

```
The Peregrine Portal and its Adapter connections have been
successfully reset. If you have deployed in a clustered
```

```
environment all web application instances must be reset to
ensure consistent application of the new configuration.
```

2 Verify your changes in the Connection Status table.

Changes include the addition, removal, or connectivity of the target adapters.

Configuring connections to ServiceCenter

Get-Services connects to the ServiceCenter database through a special adapter, the SCAdapter. This section describes how to configure and verify the following settings.

ServiceCenter adapter
Portal DB adapter
Web Application database adapter
Service Desk
Incident Management
Change Management

Connection settings

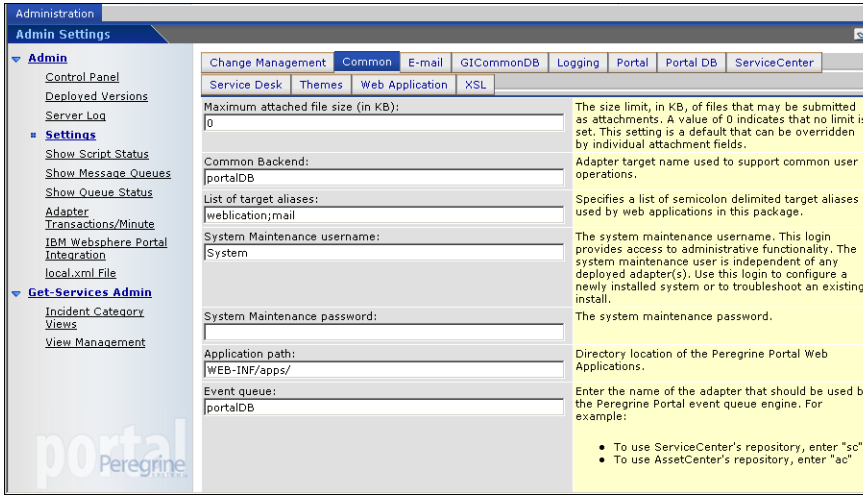
This section describes how to configure and verify back-end system settings.

To define ServiceCenter as the back-end system for Get-Services:

1 From the Peregrine Portal Admin module, click **Settings**.

This displays the Settings page which provides access to the various tabs that configure settings for Get-Services.

The Common settings page opens by default.

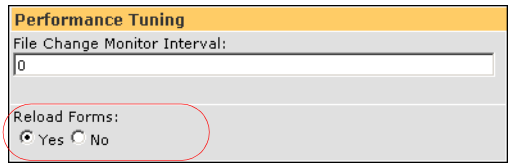


2 Update the fields as required.

The Common tab includes the following settings.

Language Translation
Personalization
Attachments
Server-Side Scripts
Encoding, Locales, and Sessions
Performance Tuning

Tip: You can reload forms without having to reset your Peregrine OAA server. Set the Reload Forms option to Yes under Performance Tuning. This works best in a production environment.



Set the Reload Forms option to **Yes** to automatically reload the forms if their sources are modified after the server started.

3 Click **Save**.

4 Click **Control Panel** > **Reset Peregrine Portal** to apply your changes to the system.

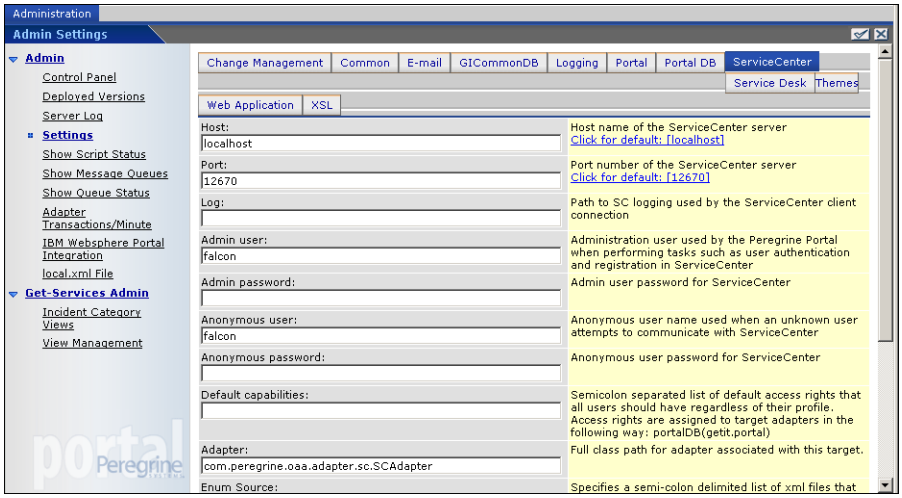
ServiceCenter database settings

Get-Services requires the ServiceCenter adapter to connect to the ServiceCenter database.

To connect to the ServiceCenter database:

1 From the Peregrine Portal Admin module, click **Settings**.

2 At the top of the Settings page, click the **ServiceCenter** tab to open the ServiceCenter settings page.



3 Update the following fields.

Field Name	Value Description	Default Value
Host	Host name of the ServiceCenter server.	localhost
Port	Port number of the ServiceCenter server.	12670
Log	The path to SC logging that the ServiceCenter client connection uses	(none)
Admin user	Administration login name that the Peregrine Portal uses when performing tasks such as user authentication and registration	falcon
Admin password	Administrator password	(none)
Anonymous user	Name to use when unknown user attempts to connect with ServiceCenter	falcon
Anonymous password	Anonymous user password for ServiceCenter	(none)
Default capabilities	Access rights for all users.	(none)

Field Name	Value Description	Default Value
Adapter	Full class path for adapter associated with this target	`com.peregrine.oaa.adapter.sc.SCAdapter`
`WEB-INF/bizdoc/Enum/SysEnums.xml`	The xml file that provides the values for enumeration data types	(none)

4 When finished, click **Save**.

5 Click **Control Panel** > **Reset Peregrine Portal** 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 displays *"Disconnected"* until you connect Get-Services to the ServiceCenter database.

Setting the PortalDB adapter

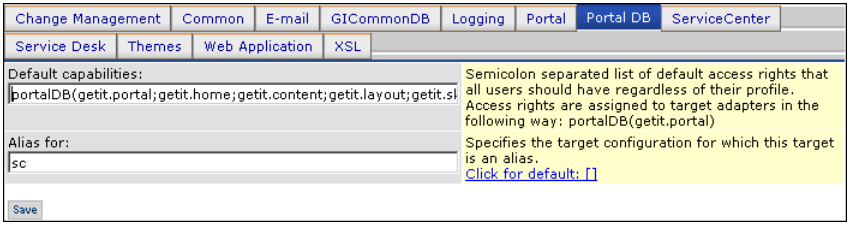
Get-Services lets you personalize portal application screens without manually changing and compiling code. To enable this feature, Get-Services requires a database adapter connection to store portal settings and customizations in the back-end database. Until a database adapter is defined for the portal page, users cannot see or make personalizations to the Peregrine Portal home page.

To configure Get-Services to save personalization settings in the ServiceCenter back-end database:

1 From the Peregrine Portal Admin module, 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 **Save**.

Note: You must manually set this field.

4 Click **Control Panel** > **Reset Peregrine Portal** to apply your changes to the system.

5 When the operation completes, verify that the adapter for the **portalDB** target is `com.peregrine.oaa.adapter.sc.SCAdapter` and displays *Connected* in the Connection Status table.

Important: If you specify one alias and subsequently change that alias, you lose the personalizations of your portal application screens.

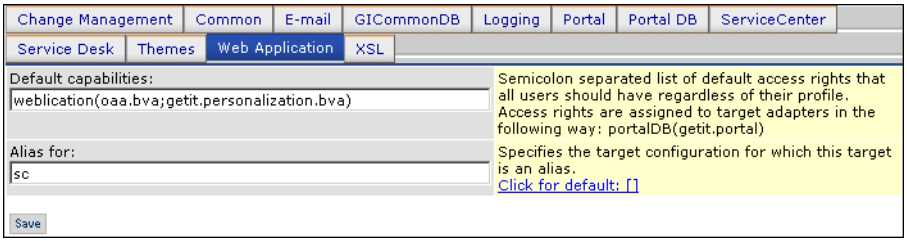
Setting the Web Application database adapter

Get-Services 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.

To configure Get-Services to save personalization to the Web application in the ServiceCenter back-end database:

1 From the Peregrine Portal Admin module, click **Settings**.

2 At the top of the Settings page, click the **Web Application** tab to open the **Web Application** settings page.



3 In the **Alias for** field type **sc**, then click **Save**.

Note: You must manually set this field.

4 Click **Control Panel** > **Reset Peregrine Portal** to apply your changes to the system.

5 When the operation completes, verify that the adapter for the **weblication** target is `com.peregrine.oaa.adapter.sc.SCAdapter` and displays *Connected* in the Connection Status table.

Important: If you specify one alias and subsequently change that alias, you lose personalizations made to forms.

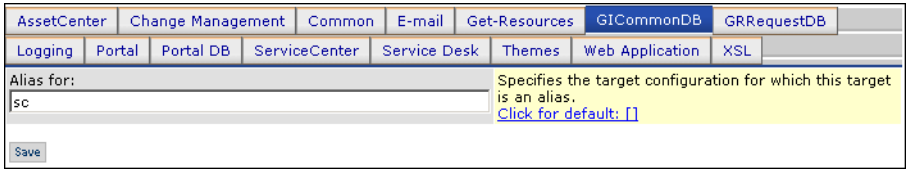
Setting the GICommonDB adapter

Get-Services lets you find information about an employee's assets and reporting structure. To enable this feature, Get-Services requires a database adapter connection to store the information in the back-end database. Until you define a database adapter for the page, users cannot view the personal information on the Peregrine Portal home page.

To configure Get-Services to save personal information in the ServiceCenter back-end database:

1 From the Peregrine Portal Admin module, click **Settings**.

2 At the top of the Settings page, click the **GICommonDB** tab to open the settings page.



3 In the **Alias for** field, type **sc** to specify ServiceCenter as the alias target configuration.

Note: You must manually set this field.

4 Click **Control Panel** > **Reset Peregrine Portal** to apply your changes to the system.

5 When the operation completes, verify that the adapter is `com.peregrine.oaa.adapter.sc.SCAdapter` and displays C*onnected* in the Connection Status table.

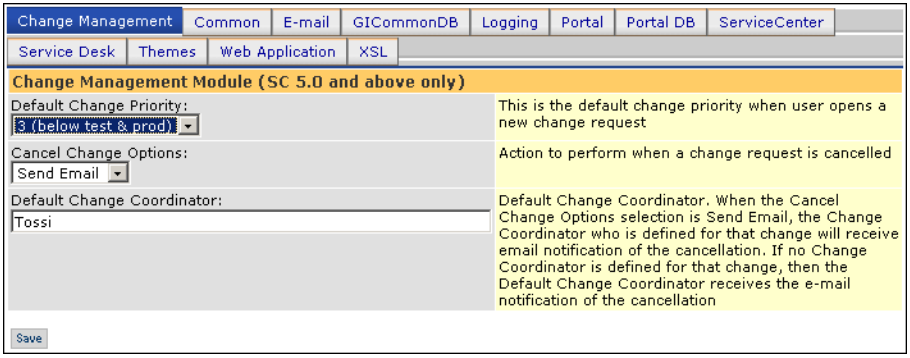
Setting Change Management parameters

The Change Management module shows field technicians their current tasks and a history of all their closed tasks. This section lists Change Management parameters. You configure these parameters with the Change Management tab on the Admin Settings page.

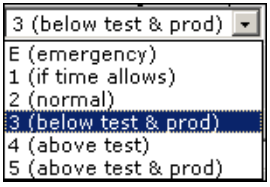
To configure Change Management parameters:

1 From the Peregrine Portal Admin module, click **Settings**.

2 At the top of the Settings page, click the **Change Management** tab to open the **Change Management** settings page.



3 Select the Default Change Priority from the drop-down menu.



This list specifies the default change priority when a user opens a new change request.

4 Select the Cancel Change Options from the drop-down menu.

This list specifies what action to take when a user cancels a change request.

5 Type the name of the change coordinator to notify when a user cancels a change request.

6 Click **Save**.

7 Click **Control Panel** > **Reset Peregrine Portal** to save your changes.

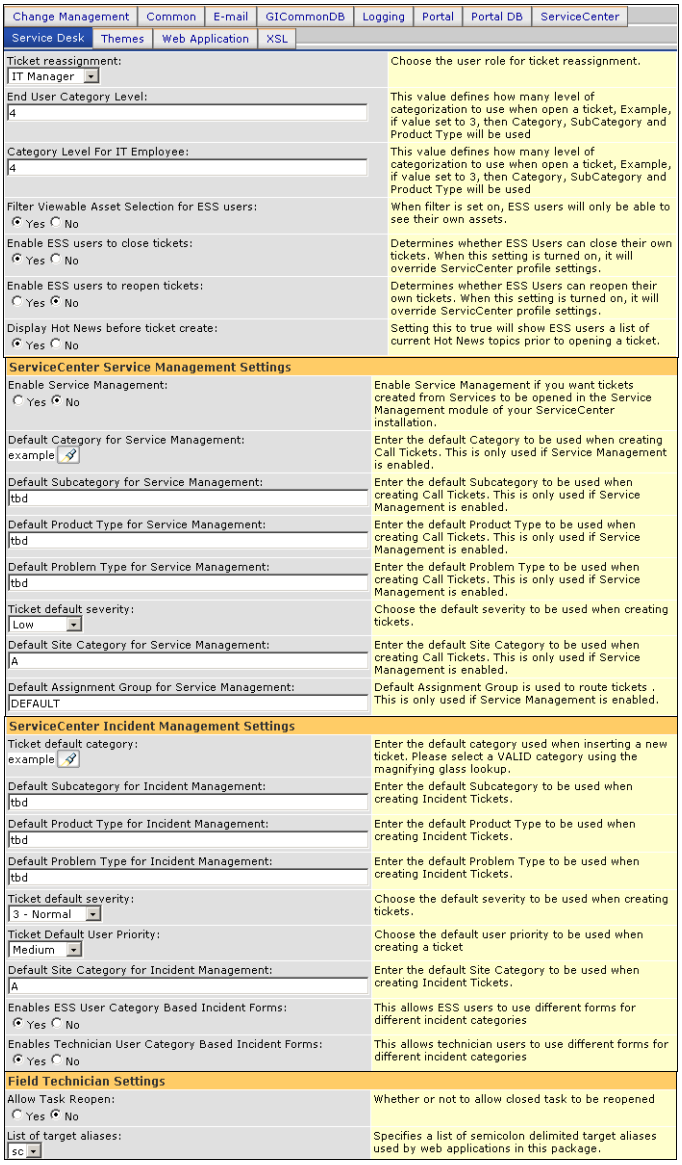
Setting Service Desk parameters

This section lists parameters that are specific to Get-Services. You configure these settings with the **Service Desk** tab on the Admin Settings page.

Incident Management is the default module used for incident (problem) tickets opened in Get-Services with the ServiceCenter adapter. If you want end users to be able to create ServiceCenter call tickets, you enable the Service Management module and configure the appropriate Get-Services settings.

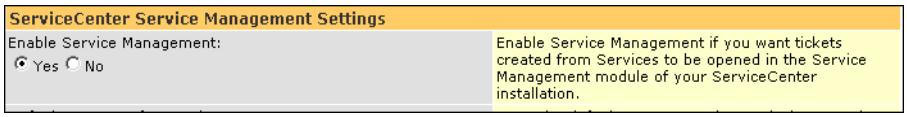
To enable Service Management for Get-Services:

1 From the Peregrine Portal Admin module click Settings, then click the **Service Desk** tab.



2 Select **Yes** in the Enable ESS users to close tickets parameter if you want to let Employee Self Service (ESS) users close their own tickets.

3 Select **Yes** in the Enable Service Management parameter if you want tickets created from Get-Services to be opened in the Service Management module of your ServiceCenter installation.



4 Click **Save**.

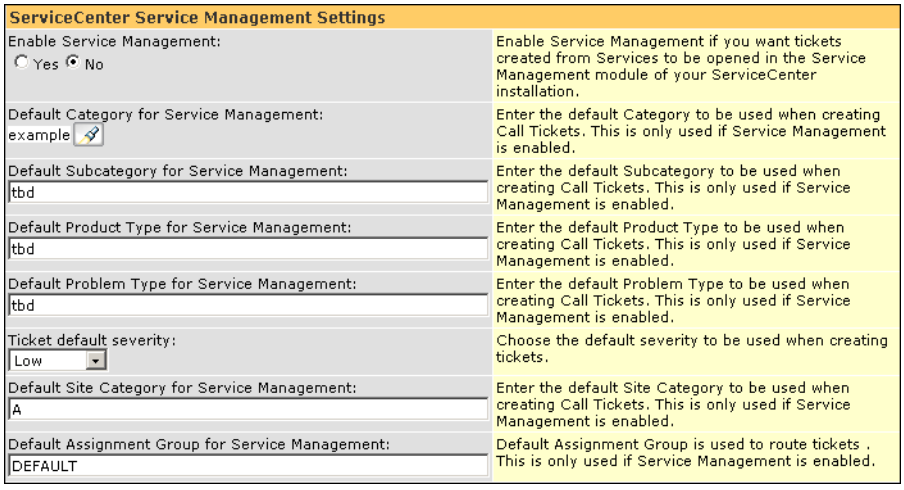
5 Click **Control Panel** > **Reset Peregrine Portal** to save your changes.

Service Management

When you enable the Service Management module, end users can open and view both Incident tickets and Call tickets. You can then change Service Management parameters as needed.

To set Service Management for Get-Services:

1 From the Peregrine Portal Admin module, click Settings. Then click the **Service Desk** tab and scroll to the ServiceCenter Service Management Settings.



2 Update the following fields as needed to define the ServiceCenter Service Management settings for Get-Services.

Field name	Default value	Value description
Enable Service Management	No	When set to Yes, tickets created in Get-Services are opened using the Service Management module of your ServiceCenter installation.
Default Category for Service Management	example	The default ServiceCenter category used for creating Call tickets. This parameter is used only if Service Management is enabled.
Default Subcategory for Service Management	tbd	The default ServiceCenter Subcategory used for creating Call tickets. This parameter is used only if Service Management is enabled.
Default Product Type for Service Management	tbd	The default Product Type used when creating Call tickets. This parameter is used only if Service Management is enabled

Field name	Default value	Value description
Default Problem type for Service Management	tbd	Defines the default ServiceCenter category for Problem tickets.
Ticket default severity	Low	The default severity used when creating tickets from the drop-down list. This parameter is used only if Service Management is enabled.
Default Site Category for Service Management	A	The default Site Category used when creating Call tickets. This parameter is used only if Service Management is enabled.
Default Assignment Group for Service Management	Default	Used to route tickets, this is only used if Service Management is enabled.

3 Click **Save**.

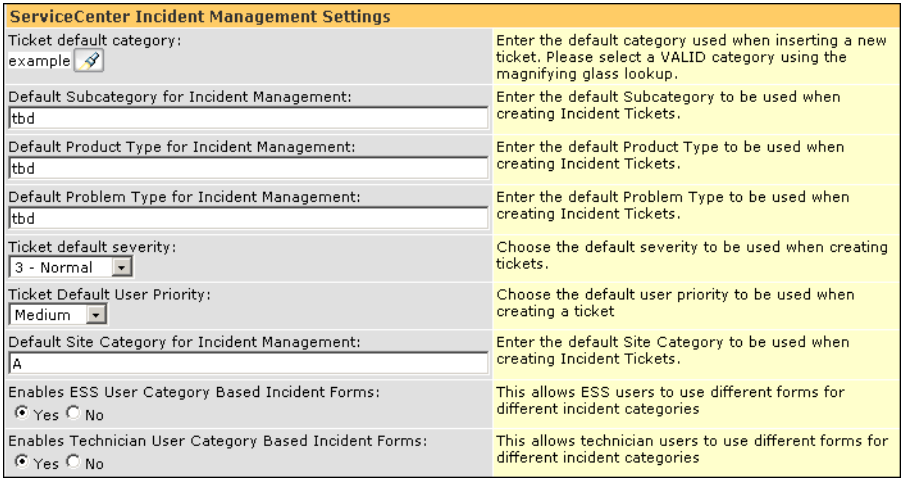
4 Click **Control Panel** > **Reset Peregrine Portal** to save your changes.

Incident Management

The following parameters control the default settings that Incident Management uses when a user opens a Get-Services ticket. These settings are in the Service Desk tab on the Admin Settings page.

To set Incident Management for Get-Services:

1 From the Peregrine Portal Admin module, click Settings. Then click the **Service Desk** tab and scroll to the ServiceCenter Incident Management Settings.



The following table describes the parameters that define the ServiceCenter Incident Management settings for Get-Services.

Note: Only a ServiceCenter user with Admin rights can change the settings.

Field name	Default value	Value description
Ticket default category	example	The default Category used when creating new Incident tickets.
Default Subcategory for Incident Management	tbd	The default Subcategory used when creating Incident tickets.
Default Product Type for Incident Management	tbd	The default Product Type used when creating Incident tickets.
Default Problem Type for Incident Management	tbd	The default Problem Type used when creating Incident tickets.
Ticket default severity	3 - Normal	The default Severity used when creating Incident tickets.

Field name	Default value	Value description
Default Site Category for Incident Management	A	The default Site Category used when creating Incident tickets.
Enables ESS User Category Based Incident Forms	Yes	Allows ESS users to use different forms for different incident categories.
Enables Technician User Category Based Incident Forms	Yes	Allows technicians to use different forms for different incident categories.

2 Change the parameters, as needed.

3 Click **Save**.

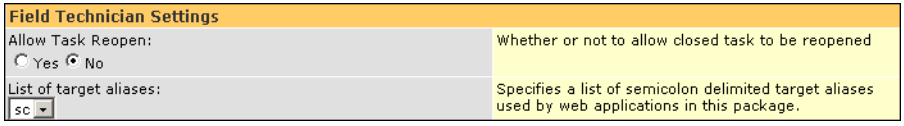
4 Click **Control Panel** > **Reset Peregrine Portal** to save your changes.

Field Technician

The following parameters control the default settings for Field Technicians. These settings are in the Service Desk tab on the Admin Settings page.

To set Field Technicians parameters for Get-Services:

1 From the Peregrine Portal Admin module, click Settings. Then click the **Service Desk** tab and scroll to the Field Technician Settings.



2 Click **Yes** to allow technicians to reopen tasks.

3 Click **Save**.

4 Click **Control Panel** > **Reset Peregrine Portal** to save your changes.

Troubleshooting the ServiceCenter database connection

If the Connection Status table in the Admin Control Panel displays D*isconnected* for any of the target adapters, do the following to troubleshoot possible configuration issues.

To troubleshoot the ServiceCenter database connection:

1 From the Admin module, check the Control Panel page to confirm the database connectivity status.

   If **sc** is disconnected, verify that the ServiceCenter service is running and the ServiceCenter console has been started.

2 From the Settings page ServiceCenter tab:

   a Verify that the parameters for **Host** and **Port** are correct.

   b Verify that the Admin user name and password defined for Get-Services are the same login values used when logging directly into the ServiceCenter back-end system as an Administrator.

3 Verify that you have ServiceCenter full client connectivity by starting a client that points to the port listed on the Settings page in the Admin module.

4 In the ServiceCenter settings page of the Admin module in Get-Services:

   - Verify that the parameters entered for the **Host:** and **Port:** fields are correct.
   - Verify that the Admin user name and password defined for Get-Services are the same login values used when logging directly into the ServiceCenter back-end system as an Administrator.

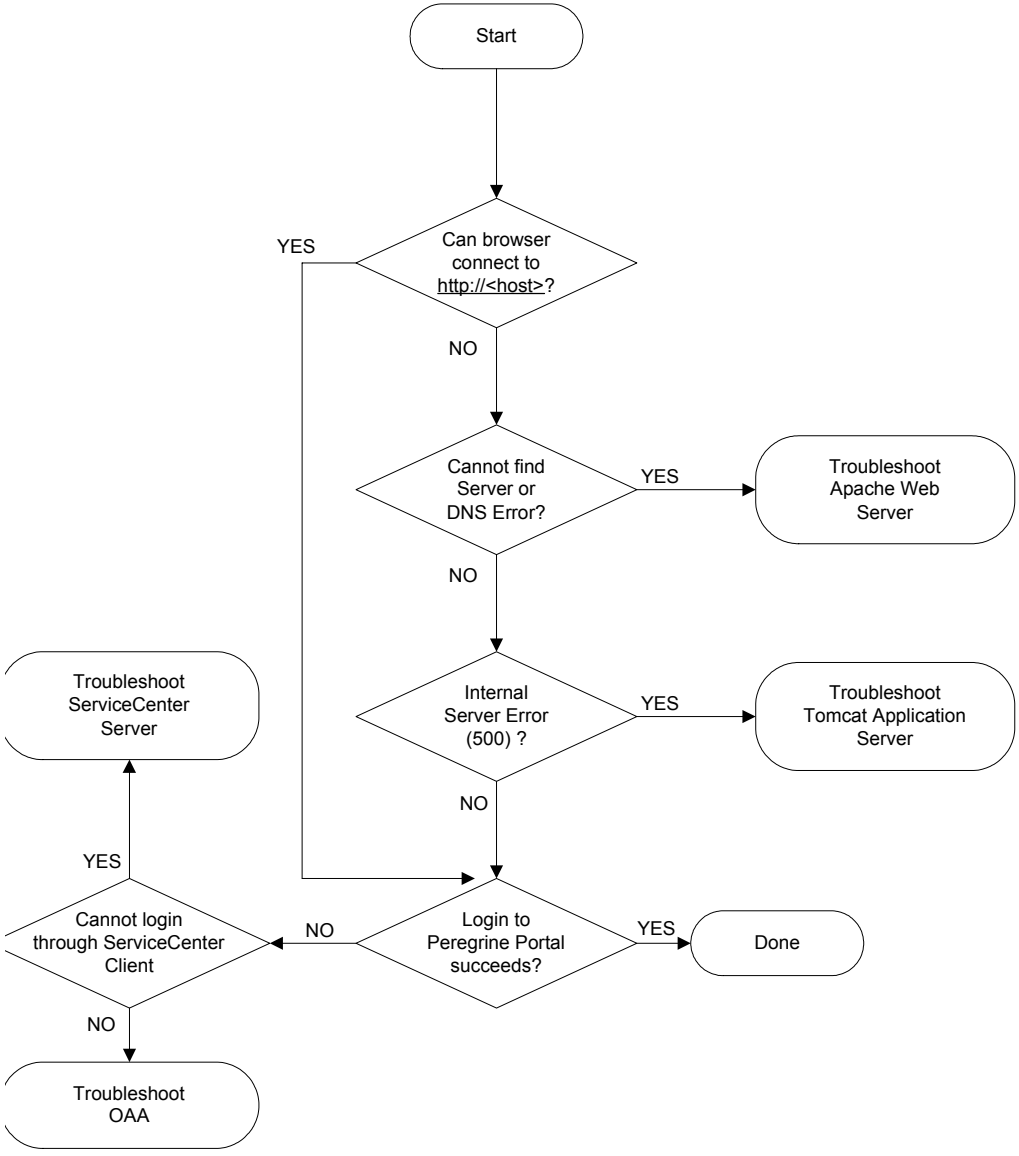
   Refer to ServiceCenter database settings on page 133 for detailed instructions on how to perform these functions.

7 Troubleshooting

CHAPTER

This chapter covers the following topics:

The following troubleshooting diagram shows areas to consider when trying to resolve installation problems.



Troubleshooting Apache Web server for Windows

If you are having trouble with the Apache Web server for Windows, follow these instructions.

The Web server is not responding

If the Web server is not responding:

Step 1 Verify that the network connections are enabled.

Step 2 Verify that the `apache.exe` program is running.

Step 3 Restart Apache service.

Step 4 Make sure the port that Apache uses is not in use by another network service (Apache uses port 80 by default).

To verify that the network connections are enabled:

1 Click **Start**.

2 Point to **Settings**.

3 Click **Network and Dial-up** connection.

4 Click **Local area** connection.

5 In the dialog box, verify that under Connection, Status is listed as Connected.

To verify that the apache.exe program is running:

1 Press Ctrl+Alt+Del.

2 Click **Task Manager**.

3 On the Processes tab, verify that the `Apache.exe` program is listed in the Image Name column.

To restart Apache service:

1 Click **Start**.

2 Click **Programs**.

3 Click **Administrative Tools**.

4 Click **Services**.

5 Locate the **Apache service** in the list and restart it.

To make sure the port that Apache uses is not in use by another network service (Apache uses port 80 by default):

1 Stop Apache.

   a Click **Start**.

   b Click **Programs**.

   c Click **Administrative Tools**.

   d Click **Services**.

   e Locate the Apache service in the list and stop it.

2 Click **Start**.

3 Click **Run**.

4 Enter **cmd** and click **OK**.

5 In the command line window, enter **netstat -a** and press return.

6 Make sure that an entry with `Proto=TCP, Local Address=<host>:http` does not exist.

Note: This ensures that when Apache is not running, no other service is listening on the http port (80).

7 Correct the problem by either changing Apache's default port (refer to customization documentation) or disabling/changing the conflicting service.

Users cannot access the Web server

Users cannot access the Web server even though the server is running, and the network and Internet connections are enabled.

If users cannot access the Web server, follow these instructions.

Step 1 Verify that the WINS server is installed.

Step 2 Verify that the DNS server is installed.

Step 3 Check Apache log files for additional errors.

To verify that the WINS server is installed:

1 Click Start.

2 Point to Settings.

3 Click Control Panel.

4 Click Add/Remove program.

5 Click Add/Remove Windows Components.

6 Click Networking Services.

7 Click Details.

8 Verify that the WINS Server check box is selected and properly configured on the network. Also verify that it is functioning.

To verify that the DNS server is installed:

1 Click Start.

2 Point to Settings.

3 Click Control Panel.

4 Click Add/Remove program.

5 Click Add/Remove Windows Components.

6 Click Networking Services.

7 Click Details.

8 Verify that DNS is installed, and that the DNS servers (or server) are connected and working on the network.

To view Apache log files for additional errors:

- From a text editor, open the Apache log files.

The default files are in `c:\Program Files\Peregrine\Common\Apache2\logs`.

Troubleshooting Apache Web server for UNIX

If you are having trouble with the Apache Web server for UNIX, follow these instructions.

The Web server is not responding

If the Apache Web server is not responding, check the network setup.

To check the network setup:

1 Make sure the port that Apache uses is not in use by another network service.

Note: Apache uses port 80 by default. You can change this by using the **Port** directive in the `httpd.conf` file. Use the `netstat` command to list all ports being listened to after shutting down Apache.

```
$ /etc/init.d/oaactl stop
$ netstat -a | grep 80
```

2 Make sure the IP address and hostname of the server are configured correctly. If so,

- The Ping command successfully gets a response from the server.
- The **nslookup hostname** displays the correct mapping from the hostname to the IP address.
- The **telnet hostname 80** successfully connects to the server.

```
$ /usr/sbin/ping hostname -n 5
$ telnet hostname 80
Trying...
Connected to hostname
Escape character is '^]'.
```

View Apache log files for advanced errors

If you are having trouble with the Apache Web server, view the log files.

To view Apache log files for advanced errors:

- From a text editor, open the Apache log files.

The default Apache log files are in:
`<base install directory>/peregrine/common/apache2/logs.`

Troubleshooting the IBM HTTP Server

The Get-Services installer creates duplicate alias entries in the IBM HTTP Server when you install more than one Peregrine OAA Platform application on WebSphere.

Duplicate entries can also occur if you reinstall Get-Services or install another Peregrine OAA Platform application on a system that formerly had Get-Services installed on it.

Remove any duplicate alias entries from the IBM HTTP Server `httpd.conf` file.

Troubleshooting Tomcat

Before you can troubleshoot problems on Tomcat, you must become familiar with starting and stopping Tomcat on your operating system. You also need know where the Tomcat log files are located.

To start or stop Tomcat on Windows:

1 Click **Start** > **Programs** > **Administrative Tools** > **Services**.

2 Locate the PeregrineTomcat service in the list and start/stop/restart it.

To start or stop Tomcat on UNIX:

- `$ /etc/init.d/oaactl <start/stop/restart>`

The following table contains the default Tomcat log file locations.

Operating system	Default Tomcat log files location
Windows	`C:\Program Files\Peregrine\Common\Tomcat4\logs`
UNIX	`/<installed base directory>/peregrine/common/tomcat4/logs`

Check for Tomcat port conflicts

The following table displays the default Tomcat port usage.

Port number	Tomcat service
8005	Tomcat Administration
8009	Tomcat AJP13 Worker Port

In the Tomcat log file `stderr.log`, the following line indicates the currently succeeded AJP13 port being used:

```
[INFO] ChannelSocket - -JK2: ajp13 listening on tcp port 8009ol
```

To check for Tomcat port conflicts:

1 Stop the Tomcat service.

2 Use **netstat -a** to list ports being listened on. Check for port conflicts.

3 Make necessary modifications to Tomcat port configuration or disable (or modify) the conflicting service. Additional information about Tomcat is available at *http://jakarta.apache.org/tomcat/*.

Checking for Port Conflicts: an example

Check for entries where the Proto value is "TCP" and the State is "Listening." For example, the following output from the netstat -an command shows that ports 80, 8009, 8025, 12670, and 1585 are in use:

```
Active Connections

Proto    Local Address          Foreign Address        State
TCP      0.0.0.0:8              0 0.0.0.0:0            LISTENING
TCP      0.0.0.0:8009           0.0.0.0:0              LISTENING
TCP      0.0.0.0:8025           0.0.0.0:0              LISTENING
TCP      0.0.0.0:12670          0.0.0.0:0              LISTENING
TCP1     0.2.3.154:1032         66.163.173.77:80       ESTABLISHED
TCP1     0.2.3.154:1342         10.2.3.154:12670       ESTABLISHED
TCP1     0.2.3.154:1585         0.0.0.0:0              LISTENING
TCP1     0.2.3.154:1585         10.2.0.112:139         ESTABLISHED
```

Check for Tomcat errors

Make sure that you are working with clean files.

To ensure a clean environment for troubleshooting:

1 Shutdown the Apache and Tomcat services.

2 Remove all log files.

3 Restart the Apache and Tomcat services.

4 Use a browser to connect to the Web server.

File mod_jk.log

This file contains log information regarding the out-of-process TCP connection between the Apache Web server and Tomcat.

This file is empty when there are no errors. It contain hints about connection failures when the AJP13 port is in conflict with another service, or when the Tomcat `mod_jk` connector is configured incorrectly.

File stdout.log

The following is a normal output of this log file.

```
Bootstrap: Create Catalina server
Bootstrap: Starting service
Starting service Tomcat-Standalone
Apache Tomcat/4.1.12
Instantiating Archway Servlet...
2002-12-10 12:22:13,079 INFO [main] - Using application preferences in
/C:/Program Files/Peregrine/Common/Tomcat4/webapps/oaa/WEB-INF/local.xml
2002-12-10 12:22:13,119 INFO [main] - Using default preferences in
/C:/Program
Files/Peregrine/Common/Tomcat4/webapps/oaa/WEB-INF/default/archway.xml
2002-12-10 12:22:13,200 INFO [main] - Using default preferences in
/C:/Program
Files/Peregrine/Common/Tomcat4/webapps/oaa/WEB-INF/default/common.xml
2002-12-10 12:22:13,240 INFO [main] - Using default preferences in
/C:/Program
Files/Peregrine/Common/Tomcat4/webapps/oaa/WEB-INF/default/logging.xml
2002-12-10 12:22:13,270 INFO [main] - Using default preferences in
/C:/Program
Files/Peregrine/Common/Tomcat4/webapps/oaa/WEB-INF/default/themes.xml
2002-12-10 12:22:13,280 INFO [main] - Using default preferences in
/C:/Program
Files/Peregrine/Common/Tomcat4/webapps/oaa/WEB-INF/default/xsl.xml
Bootstrap: Service started
```

Look for the following in this file during an error:

- Archway Servlet is not instantiated.
- The `webapps` location is incorrect.
- Bootstrap service failed to start.

File stderr.log

The following is a normal output of this file.

```
Created catalinaLoader in: C:\Program
Files\Peregrine\Common\Tomcat4\server\lib
[INFO] Registry - -Loading registry information
[INFO] Registry - -Creating new Registry instance
[INFO] Registry - -Creating MBeanServer
 [INFO] ChannelSocket - -JK2: ajp13 listening on tcp port 8009
[INFO] JkMain - -Jk running ID=0 time=0/120  config=C:\Program
Files\Peregrine\Common\Tomcat4\conf\jk2.properties
```

Look for the following problems in this file during an error:

- `catalinaLoader` was not created or is pointing to an incorrect location.
- ChannelSocket – `JK2: ajp13` failed to connect or is connecting on an incorrect port number.
- `JkMain` is not using the right `jk2.properties`.

File localhost_log.<date>.txt

There should not be any Java errors in this log file. This file logs application manager activity in deploying Peregrine OAA Web applications.

Troubleshooting OAA

If you are having trouble with your Peregrine OAA Web application, verify your application's back-end server and view the OAA logs.

OAA back-end configuration

Make sure that the Peregrine OAA application is connecting to the right back-end server and that it is currently functional.

Note: After you have set the Portal DB adapter, you can lose personalizations to your portal application screens if you subsequently change the adapter. After you have set the Web Application database adapter, you can lose personalizations to forms if you subsequently change the adapter.

To check back-end configuration:

1 Browse to *http://hostname/oaa/admin.jsp*.

2 Login as **System** and no password (providing this has not changed after installation).

3 From the Administration module, verify the connection status of the listed adapters.

4 Click on the target for the back-end server, for example, **sc**.

5 Verify that the host and port for the back-end server are correct.

OAA log files

The following table lists the default file locations of the Peregrine OAA log files.

Operating system	Default Peregrine OAA log files location
Windows	`C:\Program Files\Peregrine\Common\Tomcat4\bin\archway.log`
UNIX	`/<installed base directory>/peregrine/common/tomcat4 /archway.log`

Make sure that the log files contain:

- A listing of installed OAA components and their version numbers.
- A correct listing of registered packages.
- An **Archway initialization complete** statement.

If the file contains Java ClassNotFound exceptions, check to see if all the required jar files are found.

Troubleshooting WebSphere

Duplicate alias entries can occur from the IBM HTTP Server `httpd.conf` file during a WebSphere installation. If this happens, the Admin form at *http://hostname/oaa/login.jsp* does not render.

Remove duplicate `Alias /oaa` lines from the `httpd.conf` file under the `conf` directory of the IBM HTTP or Apache web server. You want only one of the following:

```
Alias /oaa "C:\WebSphere\AppServer\installedApps\oaa.ear\portal.war"
Alias /oaa "C:/WebSphere/AppServer/installedApps/oaa.ear/portal.war"
```

Troubleshooting ServiceCenter server

If you are having trouble with the ServiceCenter server, do the following:

Step 1 Check the ServiceCenter Auth code and port setting.

Step 2 Check the ServiceCenter log.

Before you troubleshoot problems, you must become familiar with starting and stopping the ServiceCenter server on your operating system. You also need to know where the ServiceCenter log files are located.

To start or stop ServiceCenter on Windows:

1 Click **Start** > **Programs** > **Administrative Tools** > **Services**.

2 Locate the PeregrineServiceCenter service in the list and start/stop/restart it.

To start or stop ServiceCenter on UNIX:

- `$ /etc/init.d/oaactl <start/stop/restart>`

The following table contains the default ServiceCenter log file locations.

Operating system	Default ServiceCenter log files location
Windows	`C:\Program Files\Peregrine\ServiceCenter\sc.log`
UNIX	`/<installed base directory>/peregrine/servicecenter/sc.log`

Check ServiceCenter Auth code and port setting

The following table contains the ServiceCenter setting file location.

Operating system	ServiceCenter setting file location
Windows	`C:\Program Files\Peregrine\ServiceCenter\RUN\sc.ini`
UNIX	`/<installed base directory>/peregrine/servicecenter /RUN//sc.ini`

To check the ServiceCenter Auth code and port setting:

1 Make sure the auth code set by the **auth:** tag is correct.

2 Make sure the port setting for **system:** matches the setting for the OAA back-end.

View ServiceCenter log

To view the ServiceCenter log:

- View the log file for auth code expiration errors.
- View the log for resource attachment errors.
- Refer to **ServiceCenter Administration Guide** for further troubleshooting if required.

Using ServiceCenter on Oracle

When using ServiceCenter on Oracle as the back-end database, personalization does not display pages correctly. From ServiceCenter, you must map the `giComponentUsers` table to Oracle.

To map the giComponentUsers table to Oracle:

1 Open the ServiceCenter client.

2 Log in as Administrator.

3 Add a `sqlsystemtables` record.

   a Click the Toolkit tab to open the Database Manager dialog box.

b Type **sqlsystemtables** in the File field and click Search.

c Check the **Map as Blob** flag.

4 Go to the `sqlmapping` table and delete all records for `giComponentUsers` table.

5 Map the `giComponentUsers` table to Oracle.

Contact your ServiceCenter Administrator for more information on updating ServiceCenter table definitions.

APPENDIX A Copyright Notices

Peregrine Systems acknowledges the copyrights belonging to the following third parties. (This appendix constitutes a continuation of the copyright page.)

Notices

Jcrypt.java

This product includes software developed by Eric Young (eay@mincom.oz.au).

JAXP

See the W3C license and the Apache version 2 license below.

W3C

W3C IPR SOFTWARE NOTICE

Copyright © 2000 World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved.

The DOM bindings are published under the W3C Software Copyright Notice and License. The software license requires “Notice of any changes or modifications to the W3C files, including the date changes were made.” Consequently, modified versions of the DOM bindings must document that they do not conform to the W3C standard; in the case of the IDL binding, the pragma prefix can no longer be 'w3c.org'; in the case of the Java binding, the package names can no longer be in the 'org.w3c' package.

Note: The original version of the W3C Software Copyright Notice and License could be found at
http://www.w3.org/Consortium/Legal/copyright-software-19980720

This W3C work (including software, documents, or other related items) is being provided by the copyright holders under the following license. By obtaining, using and/or copying this work, you (the licensee) agree that you have read, understood, and will comply with the following terms and conditions:

Permission to use, copy, and modify this software and its documentation, with or without modification, for any purpose and without fee or royalty is hereby granted, provided that you include the following on ALL copies of the software and documentation or portions thereof, including modifications, that you make:

1. The full text of this NOTICE in a location viewable to users of the redistributed or derivative work.

2. Any pre-existing intellectual property disclaimers, notices, or terms and conditions. If none exist, a short notice of the following form (hypertext is preferred, text is permitted) should be used within the body of any redistributed or derivative code: “Copyright © [$date-of-software] World Wide Web Consortium, (Massachusetts Institute of Technology, Institut National de Recherche en Informatique et en Automatique, Keio University). All Rights Reserved. http://www.w3.org/Consortium/Legal/”

3. Notice of any changes or modifications to the W3C files, including the date changes were made. (We recommend you provide URIs to the location from which the code is derived.)

THIS SOFTWARE AND DOCUMENTATION IS PROVIDED “AS IS,” AND COPYRIGHT HOLDERS MAKE NO REPRESENTATIONS OR WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO, WARRANTIES OF MERCHANTABILITY OR FITNESS FOR ANY PARTICULAR PURPOSE OR THAT THE USE OF THE SOFTWARE OR DOCUMENTATION WILL NOT INFRINGE ANY THIRD PARTY PATENTS, COPYRIGHTS, TRADEMARKS OR OTHER RIGHTS.

COPYRIGHT HOLDERS WILL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, SPECIAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF ANY USE OF THE SOFTWARE OR DOCUMENTATION.

The name and trademarks of copyright holders may NOT be used in advertising or publicity pertaining to the software without specific, written prior permission. Title to copyright in this software and any associated documentation will at all times remain with copyright holders.

Apache Version 2

Apache License, Version 2.0, January 2004, http://www.apache.org/licenses/

TERMS AND CONDITIONS FOR USE, REPRODUCTION, AND DISTRIBUTION

1. **Definitions.**

"License" shall mean the terms and conditions for use, reproduction, and distribution as defined by Sections 1 through 9 of this document.

"Licensor" shall mean the copyright owner or entity authorized by the copyright owner that is granting the License.

"Legal Entity" shall mean the union of the acting entity and all other entities that control, are controlled by, or are under common control with that entity. For the purposes of this definition, "control" means (i) the power, direct or indirect, to cause the direction or management of such entity, whether by contract or otherwise, or (ii) ownership of fifty percent (50%) or more of the outstanding shares, or (iii) beneficial ownership of such entity.

"You" (or "Your") shall mean an individual or Legal Entity exercising permissions granted by this License.

"Source" form shall mean the preferred form for making modifications, including but not limited to software source code, documentation source, and configuration files.

"Object" form shall mean any form resulting from mechanical transformation or translation of a Source form, including but not limited to compiled object code, generated documentation, and conversions to other media types.

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. **Grant of Copyright License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. **Grant of Patent License**. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable by

such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. **Redistribution**. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

(a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

(b) You must cause any modified files to carry prominent notices stating that You changed the files; and

(c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

(d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

You may add Your own copyright statement to Your modifications and may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative

Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. **Submission of Contributions**. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. **Trademarks**. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. **Limitation of Liability**. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. A**ccepting Warranty or Additional Liability**. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold

each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.Copyright [yyyy] [name of copyright owner]Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at http://www.apache.org/licenses/LICENSE-2.0 Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.

See the License for the specific language governing permissions and limitations under the License.

PEREGRINE

Index

S

T

U