Peregrine

Get-Resources 4.0 Administration Guide

For Windows, AIX, Linux, and Solaris

PART NO: GET-RES-4.0-ENG-01026-00247

Peregrine SYSTEMS

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

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

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

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

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

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

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

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

This edition applies to version 4.0 of the licensed program.

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

Contents

About this Guide 7
Book audience 7
Related documentation 8
Associated applications 8
Terminology 8
Typographical conventions 9
Special elements 9
Organization of the guide 10
Contacting Customer Support 10
Peregrine CenterPoint Web site 10
Documentation Web site 11

Chapter 1 **Architecture Overview** **13**
Peregrine OAA Platform architecture 15
Archway internal architecture 18
Archway requests 19
The Document Manager 21

Chapter 2 **Get-Resources Overview** **23**
Get-Resources features 24
Requests 24
Procurement 28
AssetCenter workflows 31
Bundle ordering workflow 31
Routing request workflow 32

Request approval workflow . . . 32
Automatic PO generation workflow . . . 33
Request status workflow . . . 34

Chapter 3 Customizing the Peregrine Portal . . . 35
Deploying the Classic theme variations . . . 36
Changing the header graphic . . . 37
Modifying themes . . . 39
Editing the CSS files . . . 41
Changing the layers . . . 43
Changing the framesets . . . 45
Translating tailored modules . . . 47
Editing existing translation strings files . . . 47
Adding new translation strings files . . . 49

Chapter 4 Using the Peregrine Portal . . . 51
Logging in to the ProductCoreAbbreviated . . . 52
Using the Activity menu . . . 53
Personalizing the ProductCoreAbbreviated . . . 54
Adding components . . . 54
Changing the layout . . . 57
Changing themes . . . 58
Displaying form information . . . 60

Chapter 5 Using the Personalization Interface . . . 61
Supporting Personalization . . . 62
Activating Personalization . . . 62
Using the Personalization form . . . 63
Using the Personalize icons . . . 66
Adding fields to a form . . . 66
Removing fields from the application pages . . . 68
Configuring field attributes . . . 69
Removing fields from a form . . . 70

Chapter 6 Get-Resources Administration . . . 71
Accessing the Peregrine Portal Admin module . . . 72

Using the Control Panel . . . 74
Viewing the Deployed Versions . . . 75
Viewing the Server Log. . . . 76
Using the Settings page. . . . 76
Setting parameters using the Admin module . . . 77
Verifying Script Status . . . 79
Displaying Message Queues. . . . 79
Showing Queue Status . . . 80
Viewing adapter transactions . . . 81
Using the IBM Websphere Portal . . . 81
Displaying form information . . . 82
Displaying form details . . . 83
User self-registration. . . . 84
Changing passwords. . . . 85
Logging and monitoring user sessions . . . 85
Understanding the usage.log file. . . . 86

Chapter 7 Back-end System Administration . . . 89
Configuring the Purchase Order Generation workflow . . . 90
Configuring the Product Catalog . . . 91
Certification . . . 91
Calculated Field: cf_Description. . . . 92
Configuring the Select a type of request form . . . 93
Configuring the Select an item type form . . . 99

Chapter 8 Security . . . 105
Password encoding methods . . . 106
Back-end system security. . . . 107
Authentication with ServiceCenter or AssetCenter . . . 107
ServiceCenter capability words and AssetCenter user right keywords . . . 107
AssetCenter sample security data . . . 109
ServiceCenter password security. . . . 110
Get-Resources global access rights . . . 111
User registration . . . 112
Enabling the E-mail adapter . . . 113

Troubleshooting the MailAdapter connection 114
Authenticating users 114
Default security configuration 115
Custom JAAS configuration 116
JAAS LoginModule control flags 118
JAAS configuration options 120
Example: Defining an LDAP custom configuration 124
Standard Sun Microsystems JAAS configuration 124
Command line options 125
Windows NT Challenge/Response 125
Setting up Windows NT Challenge/Response 126
Testing the settings 131
Creating an alternate login page 131
Creating a login Web page 131
Specifying an alternate authentication method 133

Index 135

About this Guide

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

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

This guide includes the following topics:

- Performing administrative tasks in Get-Resources
- Configuring Get-Resources for ServiceCenter or AssetCenter
- Understanding how users are identified in Get-Resources
- Using the Peregrine Portal
- Personalizing forms

Book audience

This guide is intended for administrators who configure and maintain Get-Resources. To use this guide effectively, you must have a working knowledge of the following:

- XML and ECMAScript (or JScript/JavaScript)

- Operating guides, reference manuals, and other documentation for your PC hardware and operating system
- ServiceCenter or AssetCenter administration and functionality

Related documentation

Refer to the following documentation for additional information:

This manual…	Provides information on…
Get-Resources Installation Guide	installing and configuring the Peregrine OAA platform, Get-Resources, Java SDK, and application and Web servers.
Get-Resources Release Notes	any late- breaking documentation or known issues with Get-Resources. These are constantly updated and posted to the Customer Support web site. See *Contacting Customer Support* on page 10 for details on accessing the Customer Support website.

Associated applications

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

Note: You must install and configure ServiceCenter or AssetCenter before you can install and configure Get-Resources. Refer to the *Get-Resources Installation Guide* for instructions.

Terminology

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

Typographical conventions

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

Convention	Meaning
Bold	Information that you must type exactly as shown appears in **bold**. The names of buttons, menus, and menu options also appear in **bold**.
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.

Sans Serif	Filenames, such as login.asp, appear in a sans serif font.

Special elements

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

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

Organization of the guide

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

To find this …	Look here…
Peregrine OAA architecture overview	*Chapter 1, Architecture Overview*
Get-Resources features and architecture overview	*Chapter 2, Get-Resources Overview*
Customizing the Get-Resources interface	*Chapter 3, Customizing the Peregrine Portal*
Configuring and using the Peregrine Portal	*Chapter 4, Using the Peregrine Portal*
Activating and using end user interface personalization	*Chapter 5, Using the Personalization Interface*
Administering Get-Resources using the Admin module	*Chapter 6, Get-Resources Administration*
Configuring Get-Resources to work with ServiceCenter	*Chapter 7, Back-end System Administration*
Security features	*Chapter 8, Security*

Contacting Customer Support

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

Peregrine CenterPoint Web site

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

http://support.peregrine.com

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

To contact Peregrine customer support

1 Log in to the Web site with your login user name and password.

2 Click **Go** beside the CenterPoint support area.

3 From **Contents** on the left, select **Whom Do I Call?** to display the **Peregrine Worldwide Contact Information**.

Documentation Web site

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

http://support.peregrine.com

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

CHAPTER 1 Architecture Overview

Peregrine® Open Application Architecture (OAA) platform is a software platform that enables the hosting of a variety of Web applications over a corporate intranet. The platform is Java based, encompassing the latest in Java technology including Java servlets, JAAS login authentication, and JSP pages that enable Web pages to display data dynamically.

Peregrine OAA Platform is the underlying architecture for many Peregrine products, including the Get-It™ suite of Employee Self-Service products which offers:

Get-It Product	Description
Get-Answers™	Web-based knowledge management solution for IT that delivers a single, integrated application designed for your service desk technicians, as well as the employees you serve.
Get-Resources™	Web-based solution that integrates with AssetCenter Procurement or ServiceCenter Request Management to enable employees to create requests for resources and to streamline the approval workflow of those requests throughout the organization.
Get-Services™	Web-based extension of ServiceCenter which allows employees to address their IT service requests on-line. Provides a simple interface on your corporate intranet that employees can use to report a problem, check status, request help and reset a password.

Peregrine OAA Platform provides a Web portal, Peregrine Portal, from which users can access their Web applications. The Peregrine Portal also provides access to the Admin module, from which all aspects of the Peregrine OAA Platform are monitored and maintained.

The base of Peregrine OAA Platform includes:

- Archway—a Java servlet that processes HTTP requests from a browser, sends the requests through an adapter to a back-end system, and returns XML data to be displayed in the browser.
- Core files—the Peregrine OAA Platform contains jsp and XML. The core consist mainly of low level Java utility classes used by the Portal Web applications built on the base OAA framework.
- Peregrine Portal—includes a login page and provides access to your Peregrine Web applications and to the Admin module for configuration of your application.
- Skins and style sheets—provide a choice for the appearance of the Web pages.

The Peregrine OAA Platform includes a number of optional components that are configured for use with Web applications as they are needed. These include:

- Adapters—enables connection to the back-end system database. The adapter required by your Web application is deployed during the installation.
- OAA Persistence—provides a general purpose database that is used by certain Peregrine Web applications. OAA Persistence provides data persistence to a database.
- OAA Workflow—enables workflow capabilities used by some Peregrine OAA Platform Web applications.
- Notification Services—a centralized service for sending and receiving notifications through multiple communication devices and for tracking the status of these notifications.

Separate documentation for Notification Services is provided with the Web applications that use this feature.

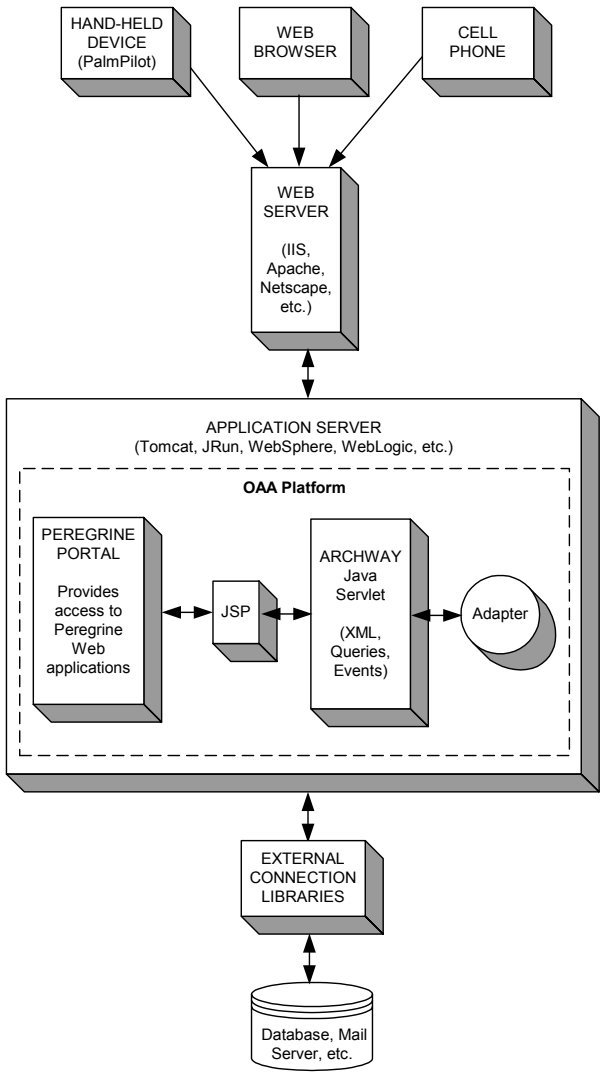
Peregrine OAA Platform architecture

Peregrine OAA Platform applications and interfaces use Web-based building blocks that include:

HTTP	A simple and widely supported protocol for sending client requests to a server. Variations such as HTTPS provide security as well.
XML	Extensible Markup Language. A documentation meta-language that allows you to format data, which can then be displayed through a Web browser. Unlike HTML, you create your own XML tags and define them any way you want.
Commercial web servers	The services provided by the Archway architecture can be served from any commercial Web server, including IIS, Apache, Netscape Enterprise Server, or the Java Web Server.
Application servers	Peregrine OAA Platform supplies Apache Tomcat for an application server with the installation. JRun, WebSphere, and WebLogic are also supported.
Common clients	Applications can be deployed via Web browsers (IE, Netscape), handheld devices (Palm Pilot), or mobile phones (through HDML).

The application server processes data (JSP pages, XML, and so forth) that it receives from the database or client that is specifically related to the Peregrine Systems Web applications. The Web server converts the data into a form (HTML) that can be displayed in a Web browser.

The following diagram illustrates the architecture:



The Archway component listens to HTTP requests from clients, routes the requests to an appropriate server, and returns data or documents. The requests supported by Archway can vary, but they fundamentally consist of queries, data updates, or system events.

For example, a client can contact Archway and ask to query a database for a list of problem tickets. Another client could contact Archway and supply it with a new purchase request to be entered into the database.

All requests and responses are formatted using XML. For example, a problem ticket expressed in XML could appear as follows:

```
<problem>
  <number> PM5670 </number>
  <contact> Joe Smith </contact>
  <description> My printer is out of paper </description>
</problem>
```

Clients that interact with Archway can do anything they need with the XML that is returned as a response. Very frequently, the client initiating the request is a user interface such as a Web browser. Such a client could easily display the XML documents returned by Archway. However, to be of better use, the XML documents are often displayed within a formatted HTML page. This is accomplished by using Java Server Pages (JSP).

JSP provides a syntax for creating HTML pages that is pre-processed by the Web server before being sent to the browser. During this processing, XML data obtained from Archway is merged into the HTML page.

Archway's architecture includes special support for automatically generating the HTML and JSP pages that make up a Web application.

Archway internal architecture

Archway is implemented as a Java servlet. The Java servlet is an application executed by a Web server that processes HTTP requests from client Web browsers and sends the request, by way of an adapter, to a database. It then retrieves the requested information from the database and returns it to the client. Archway requires both a Java environment and a Web server.

Each request is interpreted to determine its destination. Archway is able to communicate with a variety of back-end systems, including the AssetCenter or ServiceCenter products from Peregrine.

Requests can be handled in one of three ways:

- A request can be sent directly to an adapter that talks to a back-end server. For instance, a query request for opened tickets could be forwarded to an adapter capable of communicating with ServiceCenter.
- A request can be sent to a script interpreter hosted by Archway. This enables you to define your own application-specific services. Within a script, calls can be made back to Archway to access the back-end system with database operations and events.
- Finally, a request can be sent to a component known as a Document Manager. This component provides automated services for combining logical documents.

Archway communicates with back-end systems with the help of specialized adapters that support a predefined set of interfaces for performing connections, database operations, events, and authentication. All adapters use DLLs to communicate with each application.

Messages can be routed to a script interpreter hosted by Archway. The interpreter supports ECMAScript, a European standard based on the Core JavaScript language used by Netscape (JavaScript) and Microsoft Internet Explorer (JScript).

Messages can be routed to the Document Manager component. This component reads special schema definitions that describe application documents for logical entities such as a purchase request, problem ticket, or product catalog. The script interpreter uses these schemas to automatically generate database operations that query, insert, or update such documents.

Each form displayed by a Web application using Peregrine OAA Platform has a related JSP. A virtual directory tells the URL the location of the JSP pages the Web browser will use to display the Web application forms.

Archway requests

Archway supports a variety of requests, all of which are based on two basic technologies: HTTP and XML. The HTTP protocol defines a simple way for clients to request data from a server. The requests are stateless and a client/server connection is maintained only during the duration of the request. All this brings several advantages to Archway, including the ability to support a large number of requests with the help of any of today's commercial Web servers.

Another important advantage is that any system capable of making HTTP requests can contact Archway. This includes Web browsers, of course. But in addition, all modern programming environments support HTTP. This makes it very simple to write new adapters that communicate with Peregrine servers without the need of specialized APIs.

An HTTP connection consists of:

- A client request
- A server response

The messages exchanged normally have a number of header lines and some content lines. For example, consider the following two principal parts of a request:

Query String	The parameters sent with the URL for the HTTP connection. For example: http://prgn/servlet/archway?hello&world This URL is made up of a server locator (http://prgn/servlet/archway) and a query string (hello&world).
Content	The data appended to the request. This data can be in any format, but for Archway, the data is always formatted as XML.

Archway uses the query string of a request to determine what it has been asked to do. The following query string syntax is expected:

archway?target.command&param=value&param=value&…

Consider each part of the request:

Target	The name of the target object that should handle the request. Archway forwards requests to a system and returns the response. Thus, the target could be ServiceCenter, AssetCenter, or another database. The target may also be the name of a Script Object that contains customizable logic for handling the request.
Command	The action that the target object should take. By default, five basic actions are supported: query, update, insert, delete, and event. However, when the target is a Script Object, the action can be any function defined by the script.
Param=Value	Parameter values included in the request. An arbitrary number of parameters can be passed along with the request. The encoding of these parameters is the same as that used by CGI (Common Gateway Interface). As with CGI, data sent by a browser is provided by fields embedded in an HTML form. This data is automatically formatted as a CGI request in a way that Archway understands.

The following are sample URLs that query Archway with HTTP requests. These queries return data in XML documents.

- host name/servlet/archway?sc.query&_table=probsummary&priority.code=1

  This sends a query request to ServiceCenter for all records in the probsummary table with a priority code of 1.

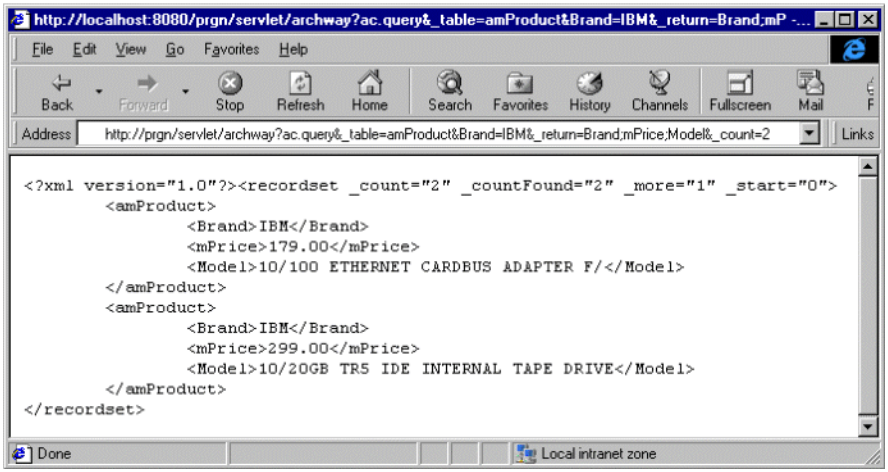
- host name/servlet/archway?ac.query&_table=amAsset&_return=Brand;mPrice;Model&_count=2

  This sends a query request to AssetCenter for the first two records in the amProduct table. Only the Brand, mPrice, and Model fields are returned for each record.

- host name/servlet/archway?test.helloWorld&greeting=Hello

  This sends a *helloWorld* request to a script object named *test*.

The screen below shows the XML results of a query for products from AssetCenter.



The Document Manager

Archway uses XML to exchange data and documents between clients and the supported back-end systems. Fundamentally, the XML data returned by Archway is obtained by executing queries against one or more systems. The queries can be executed by a direct URL request or indirectly within an ECMAScript.

Simple queries are only capable of returning record sets of data. However, clients are more often interested in exchanging documents. A Document is a logical entity built up of several pieces of data that can come from various physical database sources.

The Document Manager uses schemas to determine which XML elements to use and what data should be contained in the elements. The data used by the Document Manager depends on the back-end system being used.

CHAPTER 2 Get-Resources Overview

Get-Resources™, part of the Get-It™ suite of Employee Self-Service (ESS) products from Peregrine Systems®, enables employees to create requests for resources and services and streamlines the approval workflow of those requests throughout the organization.

Get-Resources integrates with AssetCenter or ServiceCenter to:

- Provide a web-based interface for employees to create and monitor the status of requests for resources and services.
- Streamline the request process by presenting employees with corporate approved assets and services.
- Manage the approval workflow of requests throughout the organization.

Additional functionality is available with AssetCenter to:

- Fulfill requests based upon available stock prior to initiating a purchase.
- Create purchase orders for requested items that are not available in existing stock.
- Update and track the procurement process from the time that an order is placed to the time it is received.
- Enable employee acceptance or rejection of a request once it has been fulfilled.

Get-Resources features

This section describes basic features that are available in Get-Resources.

Requests

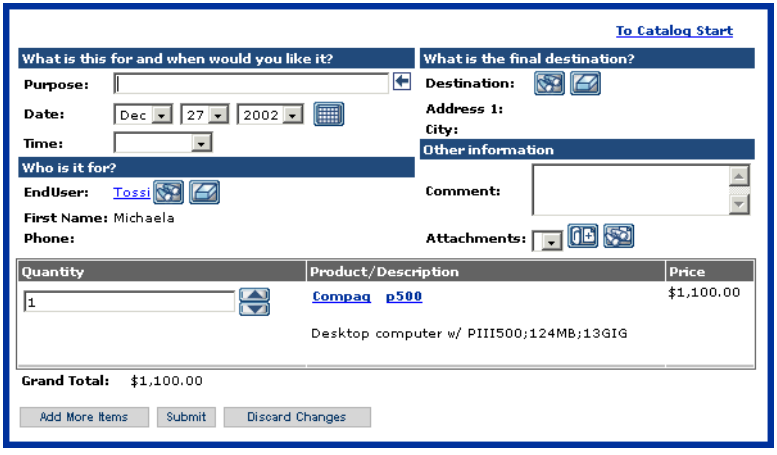
The Request module in Get-Resources provides a web-based interface that enables users to create and view the status of requests for resources and services. Users with appropriate access rights can use this module to approve submitted requests.

When using AssetCenter as a back-end system, the Request module displays items available from AssetCenter's product catalog. Users have the ability to create requests by selecting from pre-configured bundles or individual items. Additional links provide access to saved requests in progress as well as a history of all requests previously made by the user.

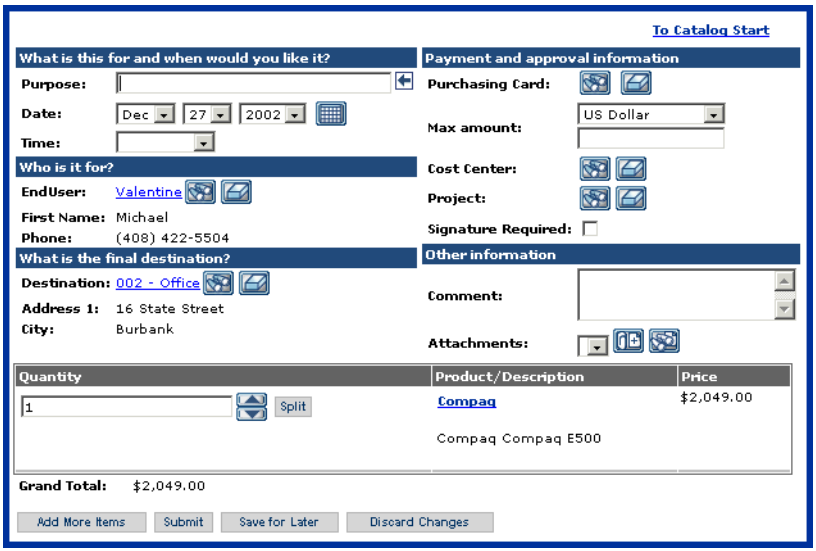
When Get-Resources is configured to work with ServiceCenter as a back-end system, the Request module displays items available from ServiceCenter's Request Management catalog. Users begin creating requests by selecting a request type, a feature which determines the options and screens that will be displayed for the rest of the creation process.

Creating a new request

When a user selects an item from the catalog, the request detail screen is displayed for users to enter additional information for the request. Users can update values for the quanitity or date the item is needed, as well as add financial information such as a cost center or maximum amount to be paid for the item before submitting the request.



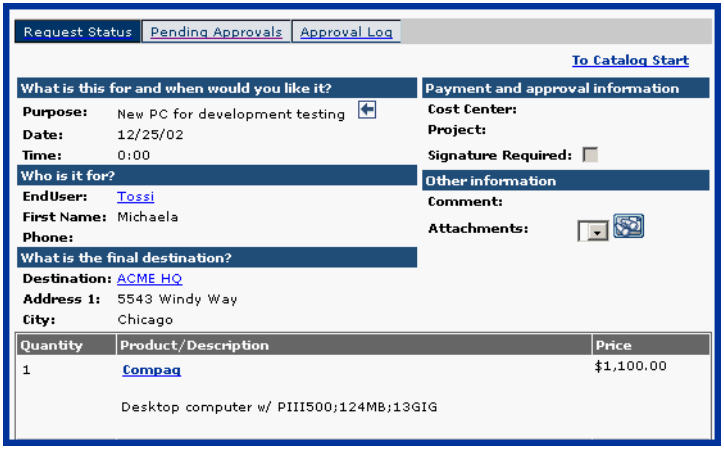
When using AssetCenter as a back-end system, Get-Resources provides additional functionality for adding purchasing card (PCard) information as well as the ability to save the request for completion at a later date.



My submitted requests

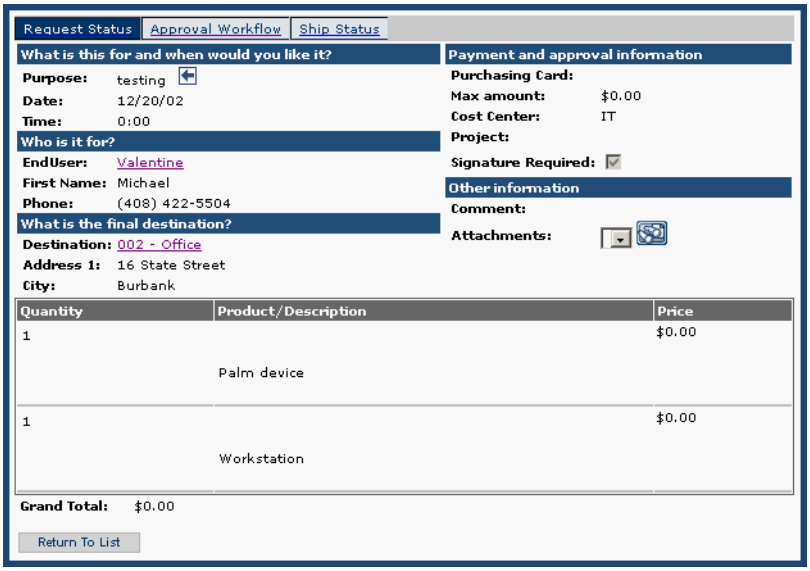
Once a request has been submitted, the user can view the status of that request by accessing My submitted requests from the left menu pane.

When using ServiceCenter as a back-end system, users can view pending approvals and access the approval log from the request status screen.



When using AssetCenter as a back-end system, users can access the approval workflow and ship status from the request status screen.

Refer to *AssetCenter workflows* on page 31 for more information describing the graphical workflows available with AssetCenter.



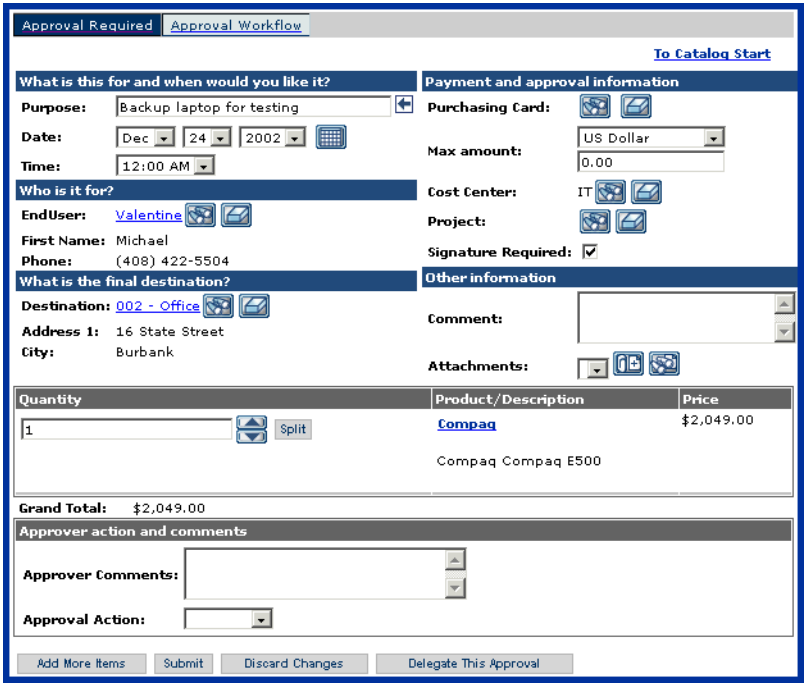
Get-Resources also provides the ability to add the Status Review utility, a list of the user's active requests, to the Peregrine Portal.

Refer to *Personalizing the ProductCoreAbbreviated* on page 54 for detailed instructions on this function.

Approve Requests

Users with appropriate access can access a detailed view of requests submitted for approval from the Request module.

When using AssetCenter as a back-end system, users have the ability to update any details, perform a stock check to reserve an item from stock for the request before approval, or delegate approval for the request to another user.



Users can also delegate approval for all requests from the main menu of the Request module.

Get-Resources provides the ability to add a list of requests waiting for approval the user's active requests, to the Peregrine Portal. Refer to *Personalizing the ProductCoreAbbreviated* on page 54 for detailed instructions on this function.

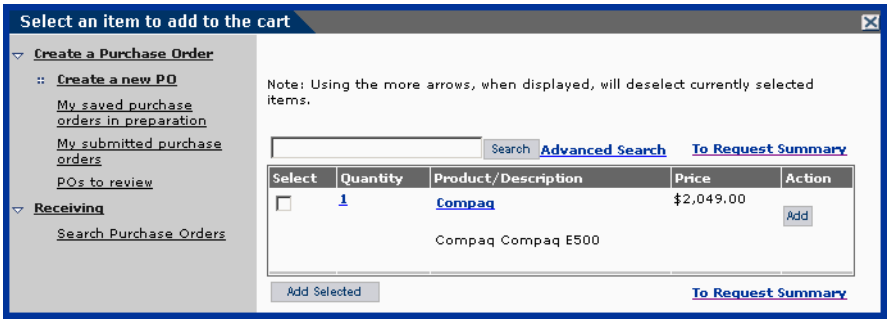
Procurement

For users with appropriate access, the Procurement module provides the ability to create purchase orders for requested items and track the status of those purchase orders to the end of the receiving process.

This feature is only available when using AssetCenter as a back-end system for Get-Resources.

Create a Purchase Order

When a request is approved, if there is no reservation for the requested item(s) from existing stock, Get-Resources displays the item(s) in a list when accessing the Procurement module. Users can select the item(s) from this list to create a purchase order.



The purchase order detail screen includes all of the information originally entered when the request was created. Users have the ability to update the purchase order with changes, save it for completion at a later time or submit it for the next procurement phase, the receiving process.

Users can access purchase orders that have been saved, submitted or automatically approved by a workflow within AssetCenter from the left menu pane of the Procurement module.

Receiving

Information for requested items that have been received are entered in the Receiving module of Get-Resources. Users can enter details such as quanity received, asset tags and serial numbers for items before submitting to the final phase of employee acceptance.

End-user acceptance

As items are received an end-user can access **My submitted requests** to review the item details for accuracy and accept or reject the item as necessary.

AssetCenter workflows

AssetCenter provides workflows to help you automate and formalize your business procedures. The following AssetCenter workflows are available for use with Get-Resources and are used by the system in this order:

- Bundle Ordering (AssetCenter 3.x only)
- Routing Request
- Request Approval
- Automatic PO Generation
- Request Status

Each of these workflows follows a default process established in AssetCenter for Get-Resources as shipped. You can modify the workflows to suit your business needs.

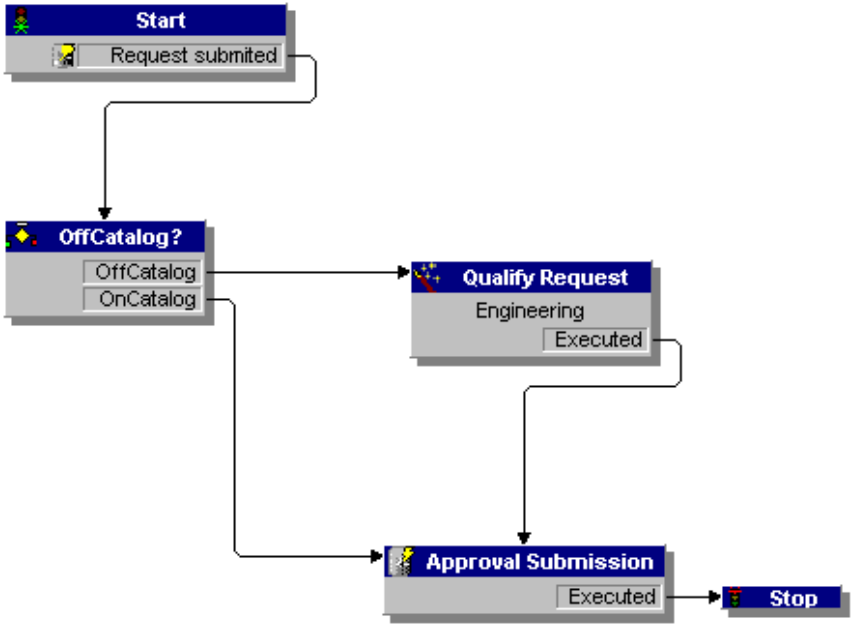
Note: It is important that you leave the first and last boxes in a workflow unchanged, since these boxes are linked to the workflows that precede and follow each workflow. Changing these boxes will break the links between the workflows and render them unworkable. Refer to your AssetCenter documentation for information on creating and modifying workflows.

Bundle ordering workflow

When a request is submitted, it is checked by the Bundle Ordering workflow to see if a bundle request has been submitted. If true, the workflow then removes the bundle from the purchase order part of the request process. This is done so that the individual items that make up the bundle will be processed through to the purchase order, rather than the bundle itself.

Routing request workflow

The Routing Request workflow is activated when the status of a request is set to *submit*. By default, all requests are processed as OnCatalog request, so the Qualify Request box is not used. The last box in this workflow starts the approval cycle.

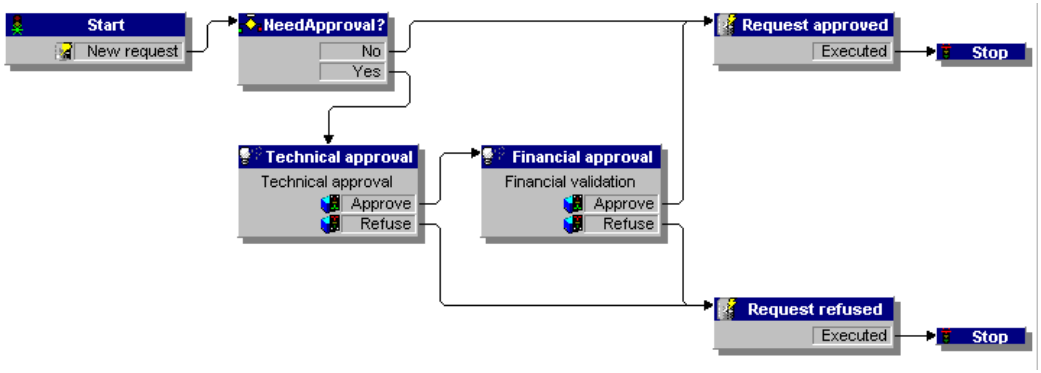


Request approval workflow

The Request Approval workflow shows the approval steps for a request that has been submitted. After the request has gone through the approval process, there are two possible results at the end of this workflow.

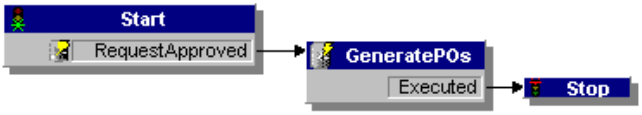
- If the request was approved, the approval status and the request status are set to *approved*. The request status determines what is displayed on the screen when the status of a request is viewed.
- If the request was not approved, the approval status and request status are set to *refused*.

If you change this workflow in AssetCenter, make sure you retain the first part of the reference field designation: REQAPPR_GR. Failure to do this will make it impossible for the system to process the workflow.



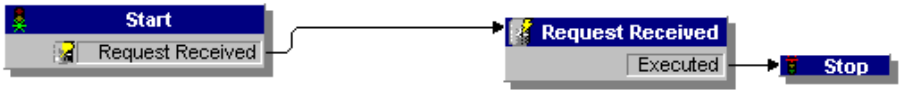
Automatic PO generation workflow

When the approval status of a request is changed to *approved*, the Automatic PO Generation workflow is activated. One purchase order per request is automatically created. You can change the process to create multiple POs by vendor or to have the system bundle several requests into one PO. If you do not want purchase orders to be generated automatically, refer to *Configuring the Purchase Order Generation workflow* on page 90 for instructions for disabling this workflow.



Request status workflow

After ordered items are received, the Request Status workflow changes the status of the request to *received*. When the status of a request is viewed, the status is now shown as *received*.



CHAPTER 3 Customizing the Peregrine Portal

Peregrine OAA provides a number of ways to customize the interface of an application built on the platform. You can make a quick change, such as replacing the logo with your company logo, or a more complex change such as rewriting the code that defines layer placement or frameset size.

This chapter includes advanced procedures for changing the ProductCoreAbbreviated interface. To use this information effectively, you should have knowledge of XML and the CSS2 specifications established by the W3C as outlined at www.w3.org.

Topics in this chapter include:

- *Deploying the Classic theme variations* on page 36
- *Changing the header graphic* on page 37
- *Modifying themes* on page 39
- *Editing the CSS files* on page 41
- *Changing the layers* on page 43
- *Changing the framesets* on page 45
- *Translating tailored modules* on page 47

Deploying the Classic theme variations

The Classic theme is the default theme used by applications built on Peregrine OAA. This is the theme you will use to create a customized theme for your enterprise.

There are five variations of the Classic theme:

- *Classic*, the neutral gray design, which is the version shown in this document.
- *Accessible*, which makes the screen available to users who need high contrast colors or better accessibility support.
- *Baja*, which adds southwestern green and beige hues to the Classic design.
- *Quicksilver*, which adds silver and blue hues to the Classic design.
- *Sierra*, which adds teal hues to the Classic design.

These themes, as well as a number of other optional themes, are deployed with the application installation. However, once you have created your customized theme, it is recommended that you delete all other themes to prevent users from selecting one of them and overriding your custom theme. If you decide later that you want to manually deploy a theme that has been deleted, or if you did not deploy all themes during the installation, use the following procedure to deploy the theme(s) from the oaa\packages directory.

To deploy an alternate Classic design:

1. In a command prompt window, change directories to C:\oaa\packages.
2. Type:

   java -jar OAADeploy.jar <name of the theme>

   Note: If you want to deploy both of the variations on the classic theme, type both of the theme names, separated by a space.

3. Press ENTER.
4. Open your Web browser and log in to the Admin module (localhost/oaa/admin.jsp).
5. Click **Settings**. On the Themes tab, change the following parameters:
   a. In the **Default skin/Theme** field, change the parameter to the name of the theme you want to use (for example, *Baja*).

b In the **Default stylesheet** field, change the parameter to the appropriate name for the CSS file (e.g., baja.css).

6 Scroll to the bottom of the page, and then click **Save**.

7 When the Control Panel is displayed, click **Reset Server**.

8 Refresh your browser to see the new theme.

Changing the header graphic

One of the easiest changes you can make is to add your corporate logo to the ProductCoreAbbreviated.

To change the header graphic:

1 Create the header graphic as needed to customize it for your enterprise.

Note: Your customized header log must be 514 pixels wide and 59 pixels high.

2 Save your changes to the following file:

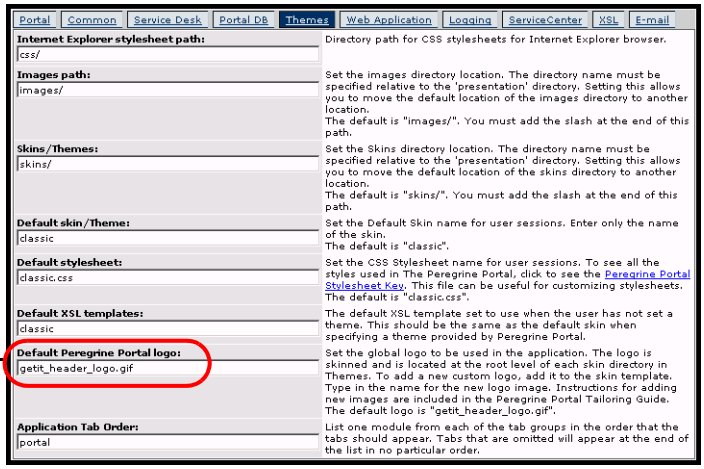
C:\Program Files\Peregrine\Common\Tomcat4\\webapps\oaa\images\skins\classic

Note: The Classic theme is the default theme.

3 Log in as an administrator (admin.jsp login page).

4 Click **Settings**.

5 On the **Themes** tab, type the new image name in the **Default Peregrine Portal logo** field.



Type your new image name.

6 Scroll to the bottom of the page, and then click **Save**.

7 When the Control Panel is displayed, click **Reset Server**.

8 Refresh the browser to view your changes.

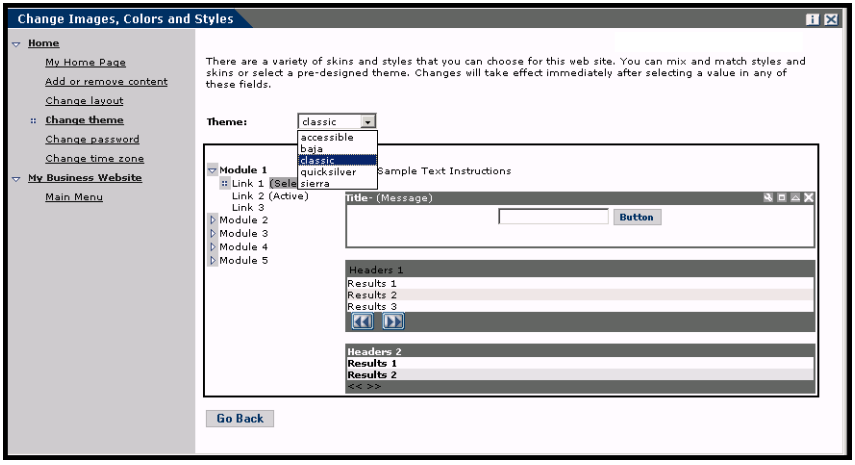
Modifying themes

For each theme folder name, there is a corresponding **css** file of the same name. For example, the classic theme has the corresponding **classic.css** file e for Internet Explorer 5.5 SP2 and 6.0, Netscape 6.x and 7.0, and Mozilla. The **css** folder is at: C:\Program Files\Peregrine\Common\Tomcat4\webapps\oaa\css.

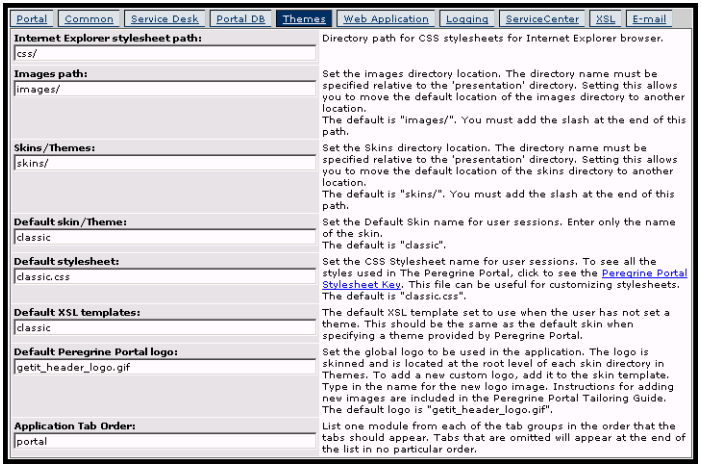
To create your own theme:

1. Copy the skins\classic folder
   C:\Program Files\Peregrine\Common\Tomcat4\webapps\oaa\css.
2. Rename the copy of Classic to *<your corporate theme>*.
3. Edit the skin images.
   - File names must remain the same.
   - Image sizes must remain the same.

You can view the changes you made on the *Change theme* page, available from the Peregrine Portal Home page.



You can also refer to the more detailed style sheet key in the themes Administration section of the Portal. To access the style sheet key, locate the Default stylesheet field on the Themes tab of the Admin Settings page and click the **Peregrine Portal Stylesheet Key** link.



Editing the CSS files

The following shows a portion of the classic.css file.

```
A,BODY,FONT,DIV,SPAN,TD,INPUT,TEXTAREA,SELECT {FONT-FAMILY: Verdana,
Tahoma, Arial, Helvetica, sans-serif;  FONT-SIZE: 10px;}
A:HOVER {COLOR; #000099;}
.ActionBar { TEXT-ALIGN: Center;}
.ActionButton {BORDER: #B5B5B5 Solid 1px; BACKGROUND-COLOR: #D6D6D6;
COLOR: #003366; CURSOR: hand; FONT-SIZE: 9px; FONT-WEIGHT: bold; }
INPUT.ActionButton:HOVER {BORDER: #B5B5B5 Solid 1px; CURSOR: hand;
BACKGROUND-COLOR: #85A5AE; FONT-SIZE: 9px;}
BUTTON.ActionButton:HOVER {BORDER: #B5B5B5 Solid 1px; CURSOR: hand;
BACKGROUND-COLOR: #85A5AE; FONT-SIZE: 9px;}
.ActionSeparator {COLOR: #1C326B;}
.ActiveHeaderMenu { BORDER-BOTTOM: #330099 Solid 1px; BORDER-RIGHT:
#330099 Solid 1px; BORDER-TOP: #330099 Solid 1px; BORDER-LEFT: #330099 Solid
1px; BACKGROUND-COLOR: #F8F8F8; COLOR: #ffffff;  }
.ActiveHeaderLink { COLOR: #ffffff;   TEXT-TRANSFORM: uppercase;}
.ActiveTabMenu {line-height: 18px; color: #FFFFFF; background-color: #666666;
padding-top: 2px; padding-right: 6px; padding-bottom: 2px; padding-left: 6px;
border-color: #CCCCCC #999999 #000000 #666666; cursor: hand; border-style:
groove; border-top-width: 1px; border-right-width: 1px; border-bottom-width: 1px;
border-left-width: 1px}
.ActiveMenuLink {COLOR: #0F2091;CURSOR: hand;  FONT-WEIGHT: Bold; }
.ActiveTableRow {COLOR: #0F2091; }
.ActiveTableNavigationRow { BACKGROUND-COLOR: #666666; COLOR: #B7C2E8;
FONT-WEIGHT: Bold; TEXT-ALIGN: Center; }
.BackgroundImage {BACKGROUND-IMAGE:
url(../images/skins/classic/backgrounds/bkg_sideframe_tile.gif);}
.Body,.BodyAlt,.BodyHead { BACKGROUND-COLOR: #ffffff; }
.BodyHeadAlt {BACKGROUND-COLOR: #D0D0D0; BACKGROUND-IMAGE:
url(../images/skins/classic/backgrounds/bkg_sideframe_tile.gif);}
.CurrentModuleMenu  { COLOR: #1A1A1A;  FONT-WEIGHT: Bold; }
A.CurrentModuleMenu:HOVER { COLOR: #000099; }
.DocTableRow {BACKGROUND-COLOR: #ffffff; COLOR: #000000;  FONT-SIZE: 10px;
BORDER-BOTTOM: #C0C0C0 Solid 2px;}
.EntryTableHeading { BORDER-BOTTOM: #e0e0e0 Solid 1px; BORDER-RIGHT:
#e0e0e0 Solid 1px; BORDER-TOP: #e0e0e0 Solid 1px; BORDER-LEFT: #e0e0e0 Solid
1px; BACKGROUND-COLOR: #09355A; COLOR: #ffffff;  FONT-WEIGHT: Bold;
TEXT-ALIGN: Left; }
```

The font family, size, weight and color, as well as other parameters such as the background color for the style are defined in this file. Some of the more commonly-modified styles are in the following table.

Style Name	Style Description
.ActionButton	The style used on buttons throughout the Portal.
.ActiveMenuLink	Used when the mouse hovers over a menu link.
.ActiveModuleMenu	Designates the currently-selected page within the navigational subset.
.CurrentModuleMenu	Designates the currently-selected navigational subset.
.FormTitle	Used for the title of forms. Normally used to title DocExplorer window content.
.ListboxEvenRow	A bolded version of TableEvenRow.
.ListboxHeading	A bolded version of Table Heading.
.ListboxOddRow	A bolded version of TableOddRow.
.MenuLink	Used within all module menus.
.ModuleMenu	Used for the left-hand navigational menu.
.ModuleMenuTitle	Designates the navigational subsets title.
.PageTitle	Used on the page title located directly below the logo and tabs.
.TableEvenRow	Used within the table heading with alternating background colors for ease of reading. Has a background color of white.
.TableHeading	Used for application headings for both search and results functions.
.TableOddRow	Used within the table heading with alternating background colors for ease of reading. Has a background color of light gray.
a.ListBoxEvenRow	Designates the style with a link attribute.
a.ListBoxOddRow	Designates the style with a link attribute.
a.TableEvenRow	Designates the style with a link attribute.
a.TableOddRow	Designates the style with a link attribute.

Important: Make a copy of **classic.css** before you modify it. You can then create a new **css** file.

Modify the style sheets after you complete the overall design using the image editor's color picker to ensure that the hexadecimal choices are as true to the logo and design as possible.

Changing the layers

Layers are implemented using the *div* tag. Each div tag has an id attribute describing its function in the header. Decide which functionality you wish to retain or omit in your new header, and then modify the appropriate tags.

The following div tags are required:

- logo div
- time div
- toolbar div
- user div
- tabs div
- form titles div

Save your modifications to your deployment directory and then refresh your browser to view the changes.

The following code samples show how the div tags work. These samples are from the **layers_ie.jsp** file located at C:\Program Files\Peregrine\Common\Tomcat4\webapps\oaa\images\skins\classic and are for Internet Explorer. Corresponding files for Netscape have a **_gecko** or **_ns** designation. Make sure that you modify all files if your application is used in both Internet Explorer and Netscape browsers, and that you make the modifications in the corresponding JSP file in your corporate theme folder.

Positions the logo layer in the header:

```
<div id="logo" style="position:absolute; left: 0px; top: 0px; width: 100%; height: 40px;
z-index: 3;">
<img name="logo" border="0" src="<%= logo %>" alt="logo">
</div>
```

Positions the time and alert line layer used for debugging and alerts:

```
<div id="time" style="position:absolute; right: 4px; top: 84px; width: 100%; z-index:
13;" onmouseover="_pauseAlert()" onmouseout="_startAlert()"
class="userBarText">
</div>
```

Positions the toolbar buttons layer located currently in the upper right section of the header:

```
<div id="toolbar" style="position:absolute; width: 50px; top: 59px; right: 0px; z-index:
12;"></div>
```

Positions the name of the currently logged-in user in the header:

```
<div id="user" style="position:absolute; top: -4px; right: 0px; z-index: 14;">
<table width="100%" border="0" cellpadding="0" cellspacing="0" align="right">
<tr>
<td width="50%"> </td>
<td nowrap width="3" align="right" valign="top">
<img src="<%= Archway.getSkinImagePath("backgrounds/rt_l.gif", user ) %>">
</td>
<td nowrap align="right" valign="top" width="100%" background="<%=
Archway.getSkinImagePath("backgrounds/rt_tile.gif", user ) %>">
<img src="<%= Archway.getSkinImagePath("backgrounds/rt_tile.gif", user ) %>">
</td>
<td nowrap><font class="userBarText" size="1" face="Arial, Helvetica,
sans-serif"><%=userTitle%></font>  </td>
</tr>
</table>
</div>
```

Positions the navigational tabs in the header:

```
<div id="tabs" style="position:absolute; left: 0px; top: 60px; width: 100%; z-index: 11;"
>
</div>
```

Positions the page title layer in the header:

```
<div id="formTitles" style="position:absolute; left: 10px; top: 81px; width: 200px;
z-index: 16;"> 
</div>
```

Changing the framesets

Important: You must have advanced knowledge of HTML, JSP, and framesets to modify these files. Keep all of the frames and do not change the names of any of the frames. To do so will result in JavaScript errors in your application.

There are two framesets to be modified for each browser. These files are in C:\Program Files\Peregrine\Common\Tomcat4\webapps\oaa\images\skins\<your corporate theme>.

The frames_xx.jsp files are for the pages that you access when logging in as an end-user (login.jsp). The admin_frames_xx.jsp files contain the configuration for the Admin module (accessed when you log in using admin.jsp).

To change framesets:

1. Stop your application server.
2. Open the browser-specific frameset file frames_<xx>.jsp in a text editor (where <xx> is ie for Internet Explorer and gecko for Netscape).
3. Modify the frameset properties.
4. Save the file.
5. Restart your application server.

   You can now test your changes in your Web browser.

The following sections show the complete -ie.jsp files as examples of the frameset files.

frames_ie.jsp

```
<%@ include file="../../../jspheader_2.jsp" %>
<%@ include file="../../../message_special.jsp" %>

    <frameset onload="setTopFrames()" onunload="closeChildWindows()"
border="0" framespacing="0" frameborder="NO" cols="*" rows="102,*">
     <frame scrolling="NO" marginwidth="0" marginheight="0" src="oaa_header.jsp"
name="getit_main_head">
     <frameset cols="185,10,*" rows="*" frameborder="no" border="0"
framespacing="0">
```

```
        <frame scrolling="AUTO" marginwidth="0" marginheight="0"
src="apphead.jsp" name="getit_header">
        <frame name="framesep" scrolling="no" marginheight="0" marginwidth="0"
src="framesep.jsp">
       <frameset rows="*,0">
        <frame scrolling="AUTO" marginwidth="6" marginheight="6"
src="e_login_main_start.jsp?<%= user.getADW(msg,"Params" ) %>"
name="getit_main">
        <frame noresize scrolling="NO" marginwidth="0" marginheight="0"
src="backchannel.htm" name="backchannel">
       </frameset>
      </frameset>
    </frameset>
```

admin_frames_ie.jsp

```
<%@ include file="../../../jspheader_2.jsp" %>
<%@ include file="../../../message_special.jsp" %>


    <frameset onload="setTopFrames()" onunload="closeChildWindows()"
border="0" framespacing="0" frameborder="NO" cols="*" rows="102,*">
     <frame scrolling="NO" marginwidth="0" marginheight="0" src="oaa_header.jsp"
name="getit_main_head">
      <frameset cols="185,10,*" rows="*" frameborder="no" border="0"
framespacing="0">
        <frame scrolling="AUTO" marginwidth="0" marginheight="0"
src="apphead.jsp" name="getit_header">
        <frame name="framesep" scrolling="no" marginheight="0" marginwidth="0"
src="framesep.jsp">
       <frameset rows="*,0">
        <frame scrolling="AUTO" marginwidth="6" marginheight="6"
src="e_adminlogin_login_start.jsp?<%= user.getADW(msg, "Params") %>"
name="getit_main">
        <frame noresize scrolling="NO" marginwidth="0" marginheight="0"
src="backchannel.htm" name="backchannel">
       </frameset>
      </frameset>
    </frameset>
```

Translating tailored modules

Out of the box, all Peregrine OAA web applications are provided in English. You can order translated versions of the core Peregrine OAA web applications by purchasing a language pack. Peregrine OAA 4.0 language packs are available in the following languages:

- French
- Italian
- German

Note: Not all Peregrine OAA web applications will offer language packs. Refer to the Peregrine support web site to determine the availability of language packs for your Peregrine OAA web applications.

If you tailor your translated web applications, however, then you will need to translate any strings that you added. The following sections describe how you can translate your tailored modules.

If you have a language pack version of a Peregrine OAA web application, you will need to edit the existing string files for these applications and add any new strings that resulted from your tailoring efforts. For more information on the process, refer to *Editing existing translation strings files* on page 47.

If you do not have a language pack version of your Peregrine OAA web applications and you want to create a new translation, refer to the instructions in *Adding new translation strings files* on page 49.

To configure the Peregrine OAA Platform to use your new translation, see *To configure the Peregrine OAA Platform to use new string files:* on page 49.

Editing existing translation strings files

You can make edits, additions, and deletions to string files outside of Peregrine Studio using any text editor or standard translation software.

To edit an existing translation string file:

1 Open the English string file for your Studio prject in a text editor or translation program.

You can find all the translation string files in your application server's installation directory:

<application server install>\webapps\oaa\WEB-INF\strings

Note: The English string file will have the ISO two letter abbreviation EN in the file name.

2 Search for any new text that you added to your tailored Get-It web application.

The string file uses the format illustrated below:

String_label, "translated string"

Where *String_label* is the Peregrine Studio name given to the string, and

Where *translated string* is the actual value of the string to be translated.

For example if you added a new button, you might look for:

EMPLOOKUP_EMPLOYEELOOKUP_SEARCH_LABEL, "Search"

3 Copy the entire line containing the English string.

4 Open the string file for the target language in which you want to add a translation.

Note: The string file will use the ISO-639 two letter abbreviation for the language in the file name.

5 Paste the copied English string into the target string file. You can paste the string at the end of the string file.

6 Change the "*translated string*" portion of the new string to the target language of your translation. For example, to change the string listed above to French, you might enter the following:

EMPLOOKUP_EMPLOYEELOOKUP_SEARCH_LABEL, "Recherche"

7 Save the new string file.

The new translation strings will be available as soon as you stop and restart the application server.

Adding new translation strings files

You can add new string files to the Peregrine OAA Platform to provide additional language support to your Get-It web applications. The translsation process can be accomplished using any text editor or standard translation software.

Important: Peregrine does not support Get-It web applications translated into any languages other than those listed in *Translating tailored modules* on page 47.

To add an existing translation string file:

1 Open the English string file for your Studio prject in a text editor or translation program.

You can find all the translation string files in your application server's installation directory:

<application server install>\webapps\oaa\WEB-INF\strings

Note: The English string file will have the ISO-639 two letter abbreviation EN in the file name.

2 Copy the entire the English string file.

3 Create a new string file for the target language in which you want to add a translation.

Note: The string file must use the ISO two letter abbreviation for the language in the file name.

4 Paste the copied English string file into the new file.

5 Change the "*translated string*" portion of each string to the target language of your translation.

6 Save the new string file.

The new translation strings will be available as soon as you stop and restart the application server.

To configure the Peregrine OAA Platform to use new string files:

1 Log in as an administrator (the administrator login page is located at admin.jsp).

2 Click **Settings**.

3 Click the **Common** tab.

4 Enter the two letter ISO-639 language code for the languages you want to support in the **Locales** field. The first code entered will be the default language used. The other languages you define will be available in a drop-down list.

5 In the **Content type encoding** field, enter the character encoding to be used for the display language. The following table lists some of the common character encoding formats.

Character Encoding	Character Set
ISO-8859-1	U.S. and Western European character sets. This is the default character set used by Studio.
Shift_JIS	Japanese character set
ISO-8859-2	Polish and Czech character set

6 Click **Save** at the bottom of the Settings form to save your changes.

7 On the Console form, click **Reset Server** to implement your changes.

Users will now be able to select the display language for their session used when they login to the Peregrine OAA Platform.

CHAPTER 4

Using the Peregrine Portal

The ProductCoreAbbreviated includes a Navigation menu, an Activity menu, and buttons that enable you to customize your Portal and to end your session.

Your installed Web applications determine the contents of the Navigation menu. However, if you log in as an administrator, all Navigation menus include an Administration tab that provides access to the Admin module.

The graphics in this chapter use the Classic stylesheet and are examples of a generic interface. Also, the Admin module displays only those features that Get-Resources uses. For more advanced changes to the portal, see the chapter on *Customizing the Peregrine Systems Portal*.

Topics in this chapter include:

- *Logging in to the ProductCoreAbbreviated* on page 52
- *Using the Activity menu* on page 53
- *Personalizing the ProductCoreAbbreviated* on page 54

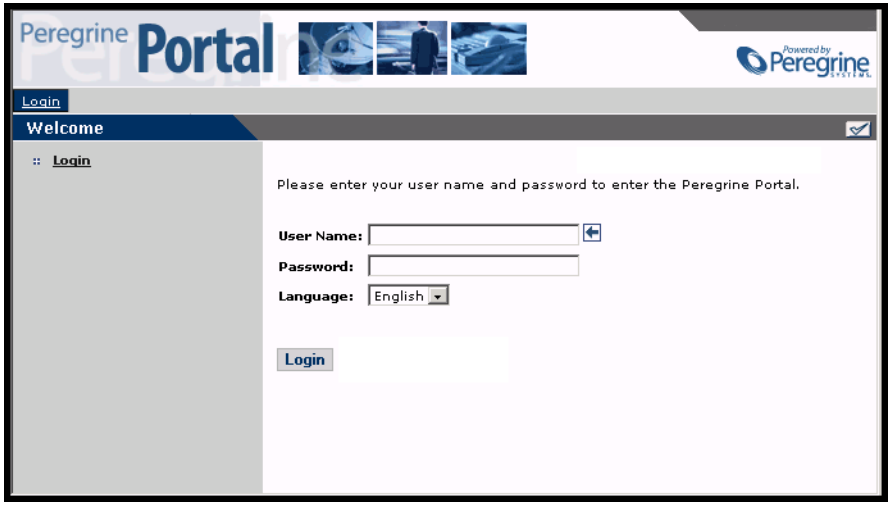
Logging in to the ProductCoreAbbreviated

There are two login screens that provide access to the Peregrine Portal:

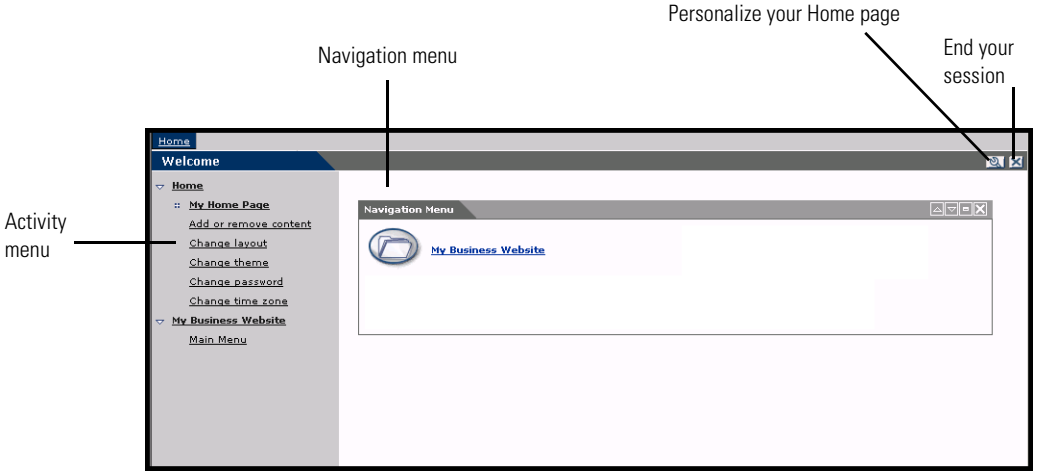
- A user login screen—http://<server>/oaa/login.jsp
- An administrator login screen—http://<server>/oaa/admin.jsp

This chapter discusses the features available with a user login. For more information about the administrator login, see the chapter on *Get-Resources Administration* in this guide.

The following is an example of the user login interface.



The following graphic shows a Portal without any applications installed. The Navigation menu includes modules for your particular application. All applications have the Admin module.



Using the Activity menu

The Activity menu provides access to a number of tasks as you navigate through your Web application. The menu remains visible as you change screens.

The default Activity menu includes the following choices:

Use this option	When you want to
My Home Page	Return to the Peregrine Portal Home page.
Add or remove content	Access the same page as the **Personalization** button, allowing you to customize your Home page.
Change layout	Change the location of a component or remove it from the Peregrine Portal.
Change theme	Select from several options. Changes take effect immediately after selecting a value in any of these fields. **Note:** Select the **accessible** theme to access the alternate text-based interface.
Change time zone	Select the time zone.

Personalizing the ProductCoreAbbreviated

By default, the Navigation menu is displayed on the Peregrine Portal. You can personalize the Peregrine Portal to add Get-Resources utilities as well as personal tools such as a calendar, calculator, or the date and time. You can also change the layout of these components or minimize a component to hide the component details.

See the chapter on *Using the Personalization Interface* in this guide for more information on personalization.

Adding components

The following components are available:

Get-Resources Utilities

This component	Provides
Requests to Approve	List of requests that require your approval.
Status Review	Displays the list of your active requests so you may review their status.

Personal Utilities

This component	Provides
Calculator	A tool using standard arithmetic functions.
Calendar	A monthly calendar.
Theme Selector	A drop-down list to change themes.
Date and Time	A date and time display for the local time zone.

Peregrine Portal Web application components

This component	Provides
Navigation Menu	Quick links to the various modules that make up this application.

This component	Provides
Document List	A display of a document search, list, or detail screen. Configure the component by choosing the document type you want to expose and the type of screen desired.
My Menu	A menu of links that can be configured dynamically. Links can point to arbitrary web sites, other menus, or document explorer screens.

Note: The Calendar and Calculator require Microsoft Internet Explorer 5.0+ or Netscape 6.1+.

Administration components

Only users with Admin capability have access to the Admin components.

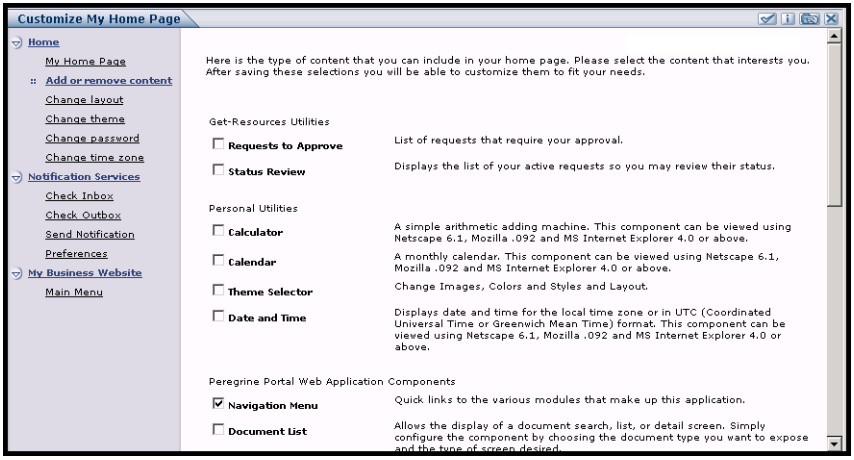
This component	Provides
Connection Status	A list of the adapters currently registered in this server and their connection status.
Control Panel	A button to reset the server and all its connections.
Page Hits / Minute	A list of the total number of pages accessed per minute.
Adapter Transactions / Minute	A list of the number of transactions performed against adapters.
Active User Sessions	A list containing the number of active user sessions.

To add Peregrine Portal components:

1 Click the **Personalize** (wrench) icon.

Note: You can also select the **Add or remove content** link from the Activity menu.

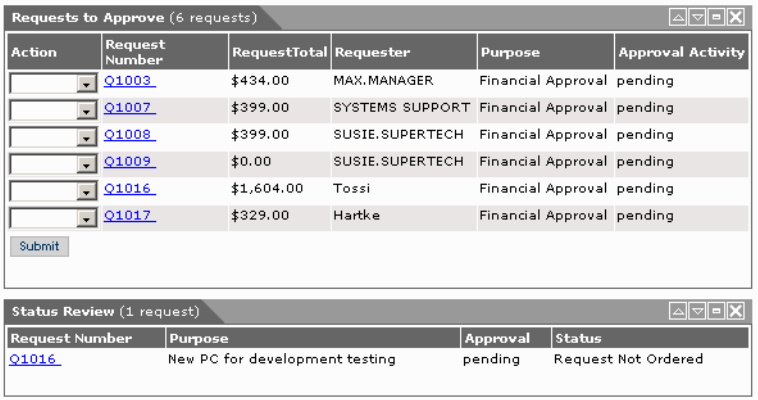
The **Customize My Home Page** opens containing a list of the available components.



2 Select the components you want to add to your Peregrine Portal.

3 When you complete your selections, scroll to the bottom of the page, and then click **Save**. To return to the Peregrine Portal without making any changes, click **Go Back**.

When you return to the Peregrine Portal, the new components are displayed. The following example shows the **Requests to Approve** and **Status Review** utilities for Get-Resources.

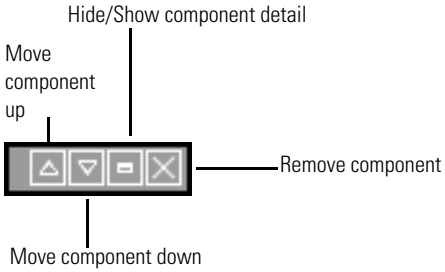


Changing the layout

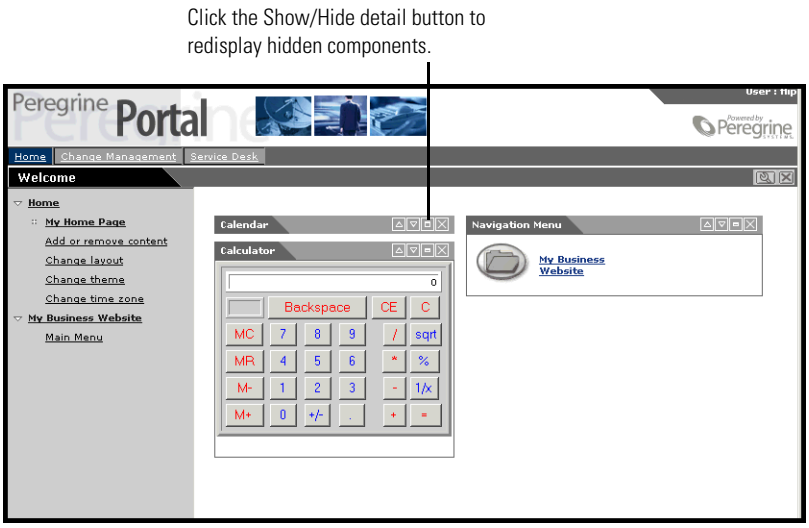
The following sections contain procedures for changing the location of the components or removing them from the Peregrine Portal. The procedure you use is determined by the Web browser you are using.

Microsoft Internet Explorer

If you are using Microsoft Internet Explorer as your Web browser, use the buttons in the upper right corner of each component to move the component up or down, remove the component, or hide/show the component detail.



In the following screen, the Calendar is minimized.

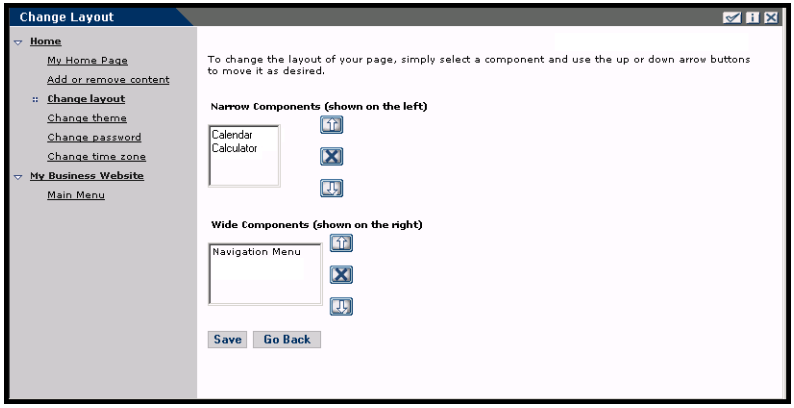


Netscape Navigator

If you are using Netscape Navigator as your Web browser, use the following procedure to change the status of the components on the Peregrine Portal. You can move a component up or down, or remove the component.

1 From the Activity menu, select **Change layout**.

A **Change Layout** page opens where you select the components you want to change.



Components can be Narrow (for example, Calendar or Calculator) and are on the left side of the Peregrine Portal. Other components (for example, Navigation Menu) are Wide and are on the right side of the Peregrine Portal.

2 Select the component you want to modify, and then click the appropriate button to activate the change.

- Up arrow moves the component up.
- Down arrow moves the component down.
- X removes the component from the Peregrine Portal.

3 Click **Save**.

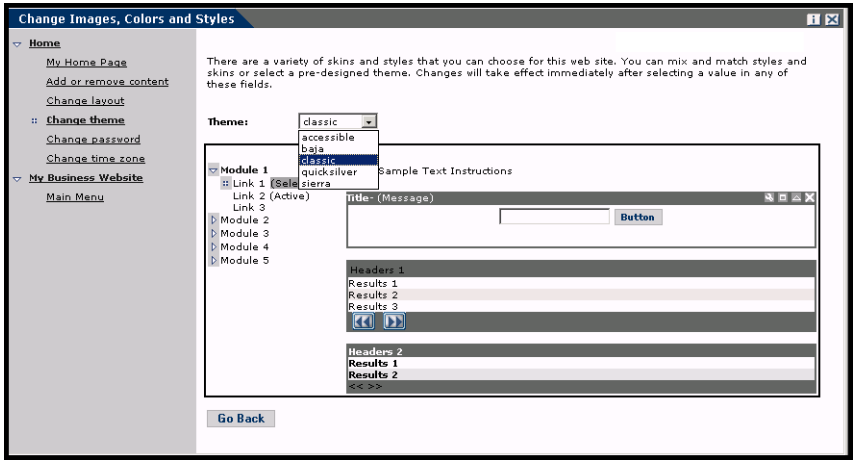
Changing themes

You can choose from a number of themes to change the look of your Web pages.

To change the theme:

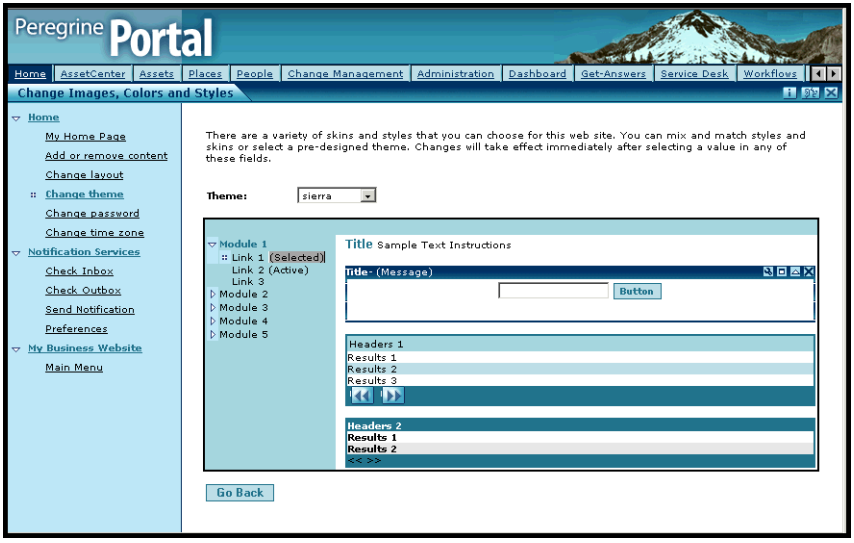
1 From the Activity menu on the Portal Home page, select **Change theme**.

The following page opens.



2 Choose from the drop-down list.

As soon as you make your selection, the page updates to reflect your selection. The following example shows the Sierra theme.



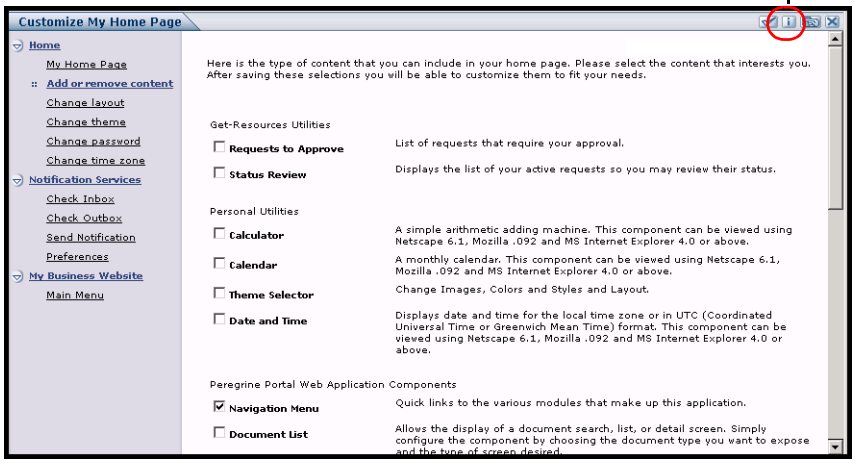
This new configuration remains through subsequent work sessions until changed.

Displaying form information

You can view information about the form you are using. Set this parameter from the Logging tab on the Settings page of the Admin module. See the Get-Resources Administration chapter in this guide for more information.

When the **Show form info** parameter is set to Yes, a **Display Form Info** button appears on the upper right corner of forms.

The Display Form Info button shows information about the form you are using.



CHAPTER 5

Using the Personalization Interface

Personalization is available to both administrators and end users in Peregrine Systems Web applications built using Document Explorers (Doc Explorers). Authorized users can change the appearance and functionality of certain Web applications directly from the application interface.

With a Personalization interface, users can add or remove fields, rearrange how fields are displayed, or add, change, or delete records from a back-end database.

This section includes:

- *Supporting Personalization* on page 62
- *Using the Personalization form* on page 63

Supporting Personalization

To support Personalization, you must have these components.

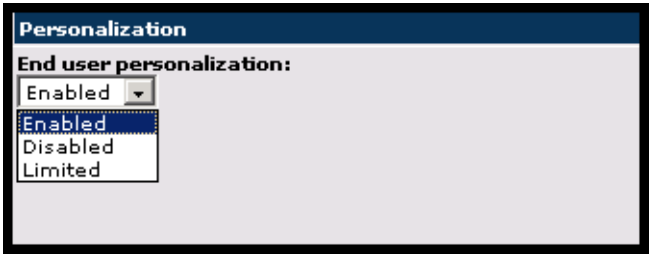
- An AssetCenter or ServiceCenter back-end database. Personalization requires you to store users' login rights and Personalization changes in one of these two databases.
- A user account with Personalization rights enabled. A user's login profile determines the level of Personalization rights Peregrine OAA grants a user. Users' Personalization rights determine not only what personalized components they can see and change, but also determine whether other users will see their Personalization changes. Peregrine OAA can display Personalization changes for an entire organization or specific user groups, but it can also limit the display of changes to a specific user.

Activating Personalization

By default, Peregrine OAA defines all users as personalization end users. Peregrine OAA grants access to personalization features using the **getit.personalization** access right/capability word stored in the Web application database. You can define the scope of end users' personalization rights by selecting one of the options from the End user personalization options on the Settings page Common tab of the Admin module.

To activate End user personalization options:

1. From the Common tab of the Admin module Settings page, scroll to the **End user personalization** parameter.
2. Select the level of personalization from the drop-down list.



You can also grant users administrator rights by adding the **getit.personalization.admin** access right/capability word to the user profile stored in the Web application database. Peregrine OAA administrator users have the following additional rights.

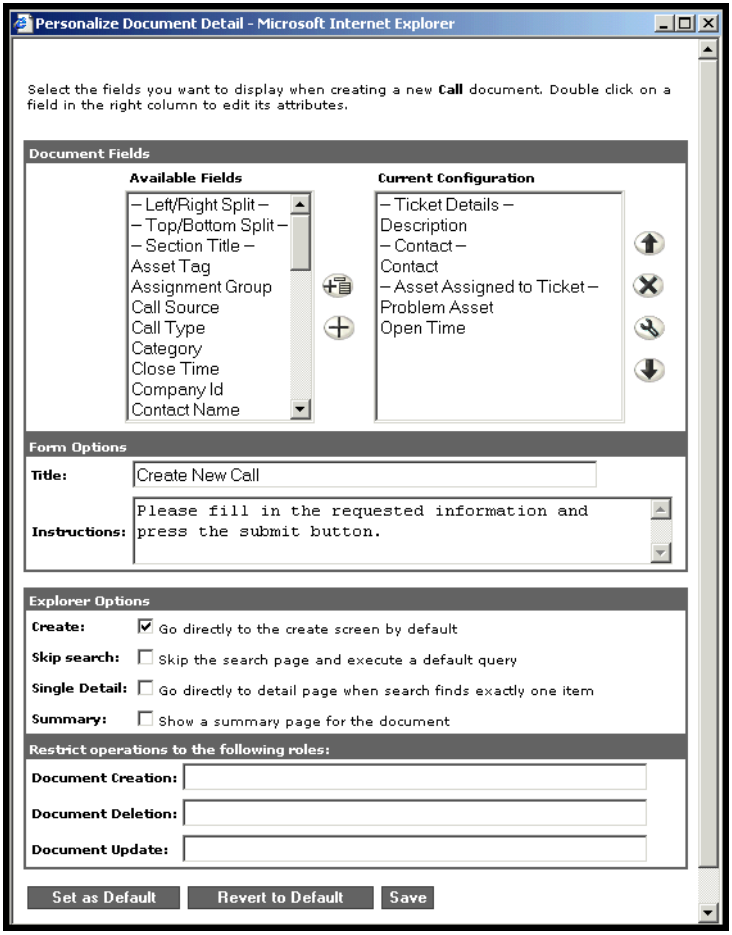
Admin capability	Description
Create	A Create button displays in search forms so that users can create new entries in the back-end database.
Update	An Update button displays in detail forms so that users can change the information that is listed for entries. Peregrine OAA also submits changes to the back-end database.
Delete	A Delete button displays in detail forms so that users can delete records from the list of entries. Peregrine OAA also deletes the record from the back-end database.
Settings are inherited by other users	This setting gives Personalization end users the proper access rights when an administrator makes Personalization changes.

Using the Personalization form

Administrators determine what available data fields display on each form. You can personalize any Web application interface that has a wrench icon in the top right of the Peregrine OAA frame. The wrench icon only appears in activities where a Personalization form has been defined. The Personalization form determines what options display in the Personalization pop-up window described below.

When you click on the Personalization icon, Peregrine OAA displays a Personalization pop-up window for the current form you are viewing.

The left column contains all the available fields of a form. The right column contains the fields that actually are on the form.



The Personalization pop-up window contains these fields:

Field	Description
Available Fields	Shows all the document fields and subdocument collections that can be added to the current form. Peregrine OAA generates the list of available fields by dynamically reading the schema that the form uses. Any items listed between dashes are form components you can use to organize and arrange how document fields are displayed in the form.
Current Configuration	Shows all the document fields, subdocument collections, and displays components that are on the current form. The first time a form is personalized, this column is empty.
Form Options Title Instructions	Defines the form name and specific instructions to follow when using the form.
Explorer Options Skip search Create Single Detail Summary	Defines how Peregrine OAA displays results. Users only see the Options section if they have sufficient Personalization rights.
Restrict operations to the following roles Document Creation Document Deletion Document Update	Determines whether users can update, create, or delete records from the back-end database system.
Set as Default	Sets the revised configuration as the default.
Revert to Default	Removes all Personalization entered by the end user and returns the form to the default state. A default form may still display fields if the Peregrine OAA administrator or the form's schema has defined any default fields to be displayed.
Save	Saves and applies your Personalization changes to the current form.

Using the Personalize icons

You personalize the Get-Resources pages by adding, moving, and removing fields. Select the page you want to personalize, then select the fields you want to display on the screen.

Icon	Description
	The Personalize (wrench) icon allows you to edit the field label. Not all components contain the edit option. Press Save at the bottom of the page to save your changes.
	The Move (vertical arrows) icons move the component either up or down on the page. Moving components is always performed with these arrows.
	The Plus (+) icon adds a component to your current configuration.
	The Insert icon adds a component in the specified place on the screen.
	The Remove (X) icon removes the component from the page. Removing a component does not delete the component; it only does not display it.

Adding fields to a form

You can add fields to a form, and then change the layout.

To add fields to a form:

1. Do one of the following:
   - From the upper right corner of the active form, click the **Personalize** icon.
   - From the lookup page, click **Personalize this page**.
2. Select a field from the **Available Fields** list.
3. Click the **Plus** (+) icon.

   The field appears in the Current Configuration list.
4. Optionally, click the Insert icon to insert a component.
5. Click **Save**.

To arrange the field order:

1. Select a field from the Current Configuration list.
2. Click the up arrow or down arrow to change the field's position in the Current Configuration list.
3. Click **Save**.

To change the field layout:

1. From the **Available Fields** list, select **Left/Right Split**.
2. Click the **Plus** (+) icon.

To add a new section:

1. From the **Available Fields** list, select **Section Title**.
2. Click the **Plus** (+) icon.

   Note: See *To rename labels* for information on editing the **Section Title** field.

3. From the **Current Configuration** column, arrange the order of the section with the **Up**, **Down**, and **Remove** icons.

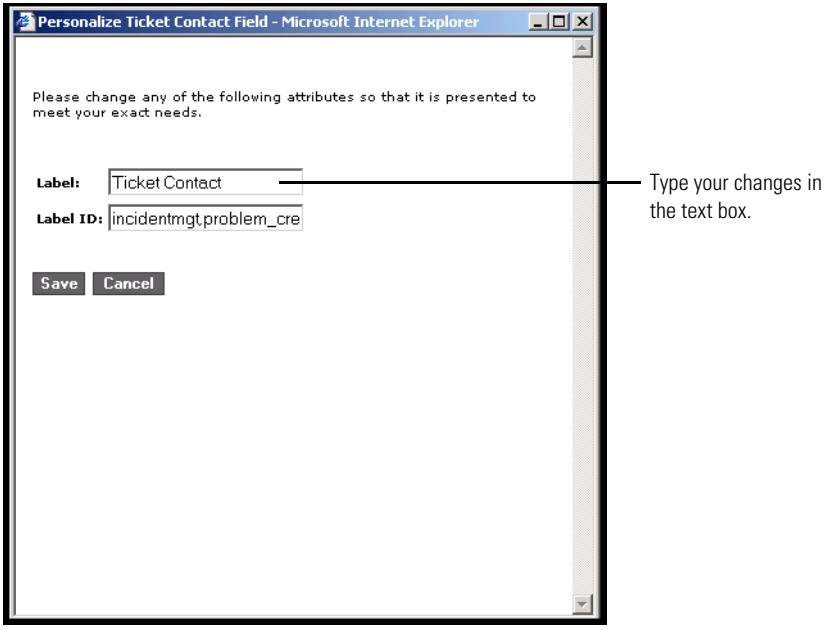
   Note: These icons either move or delete a field. Deleting a field removes the item from the form.

4. Click **Save** to keep your changes and return to the form.

To rename labels

1. From the Current Configuration column, select the label you want to change.
2. Click the **Personalize** (wrench) icon.

The Personalization window opens.



3 Type the information in the **Label** text box, then click **Save** to save your changes and return to the previous pages.

Cancel returns you to the previous page without saving your changes.

Removing fields from the application pages

You can remove a field so that it does not display on the application pages.

To remove a field

1 From the **Current Configuration** list select a field.

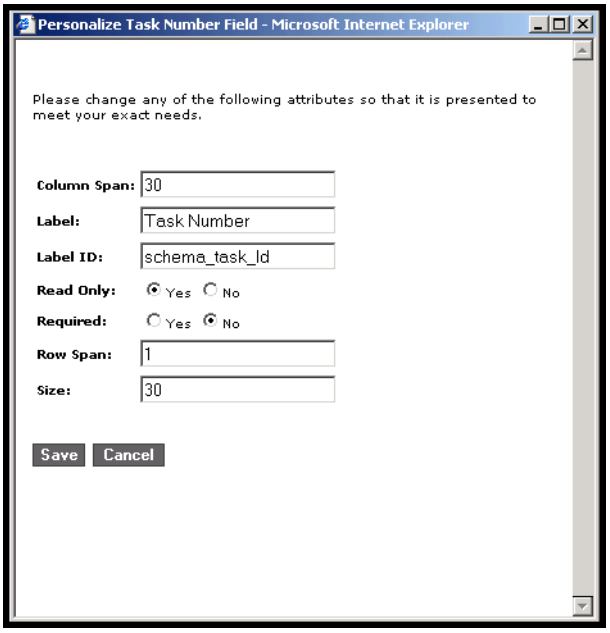
2 Click the **X** button to remove the field.

3 Click **Save**.

Configuring field attributes

To configure field attributes:

1 Double click a field from the Current Configuration list to open an edit window.



2 Enter the new field attributes:

Field	Description
Column Span	The number of data cells in a column.
Label	The name to be used as the field label. This name appears next to the field in the Peregrine OAA interface.
Label ID	The schema name.
Read Only	**Yes** prevents users from updating information displayed in the field.
Required	**Yes** requires the field to have a value before the form can be submitted.
Row Span	The number of data cells in a row.
Size	The number measurement of a component in a cell.

3 Click **Save** to save your changes and return to the previous pages.

Cancel returns you to the previous page without saving your changes.

Removing fields from a form

To remove fields from a form:

1 Select a field from the Current Configuration list.

2 Click the X button to remove the field.

3 Click **Save**.

CHAPTER 6 Get-Resources Administration

This chapter includes instructions for administering your Get-Resources system.

Topics in this chapter include:

- *Accessing the Peregrine Portal Admin module* on page 72
- *Using the Control Panel* on page 74
- *Viewing the Deployed Versions* on page 75
- *Viewing the Server Log* on page 76
- *Using the Settings page* on page 76
- *Verifying Script Status* on page 79
- *Displaying Message Queues* on page 79
- *Showing Queue Status* on page 80
- *Viewing adapter transactions* on page 81
- *Using the IBM Websphere Portal* on page 81
- *Displaying form information* on page 82
- *User self-registration* on page 84
- *Changing passwords* on page 85
- *Logging and monitoring user sessions* on page 85

Accessing the Peregrine Portal Admin module

The Peregrine Portal administrator login page enables access to the Peregrine Portal Admin module. You use the Admin module to define the settings for your Peregrine system.

Note: After installing and building Get-Resources, you must log in as a ServiceCenter or AssetCenter user with **getit.admin** rights to access the Admin module and administer the Get-Resources integration with ServiceCenter or AssetCenter. For a list of access capability words and Adapter configuration instructions, see the section on Get-Resources security in this guide.

A default administrator, Admin, 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 \<*application server*>\webapps\oaa\WEB-INF directory to store these parameters. If you reinstall Get-Resources, make a copy of this file and store it outside your Get-Resources installation. Failure to do this will result in your parameter values being lost during the new installation.

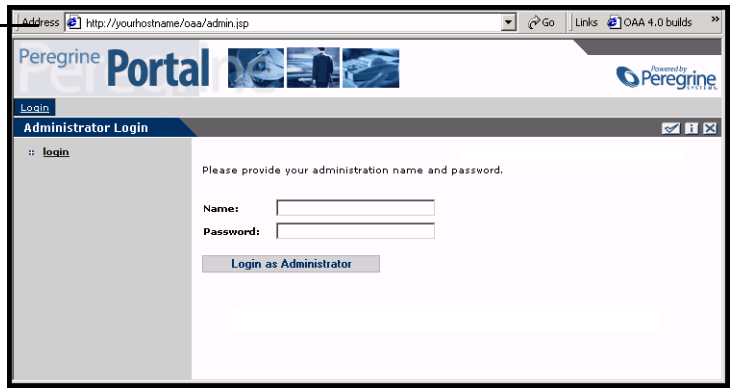
To access the Peregrine Portal administrator login page:

1. Verify that your application server (for example, Tomcat) is running.
2. In your Web browser Address field, type:

   <*hostname*>**/oaa/admin.jsp**

3 Press Enter to open the Portal administrator login page.

Type your hostname to connect to your local server.



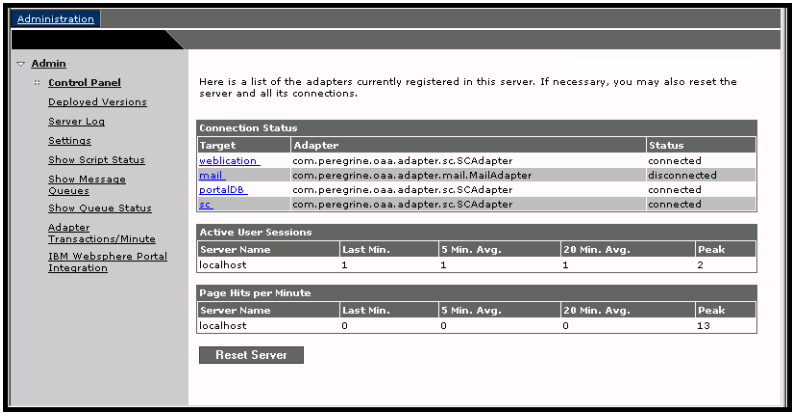
Admin is the default administrator name.

4 In the Name field, type **Admin**.

No password is required on initial login.

5 Click **Login as Administrator** to open the **Control Panel** page.

Administrators use the Admin module to define settings to the system.



The activities available in the Admin module include:

Select this option	To view
Control Panel	the status of connections to the back-end systems.
Deployed Versions	the list of deployed applications with version numbers on this server.
Server Log	activity on the Get-Resources server.
Settings	and change settings for the Peregrine Portal.
Show Script Status	and verify which scripts are running. You can also start and stop scripts from this window.
Show Message Queues	a list of all message queues.
Show Queue Status	the current status of the queues: operational and unlocked, or suspended.
Adapter Transactions/Minute	the transactions per minute for the SC adapter.
IBM Websphere Portal Integration	the installed OAA portal components in the IBM WPS environment

Using the Control Panel

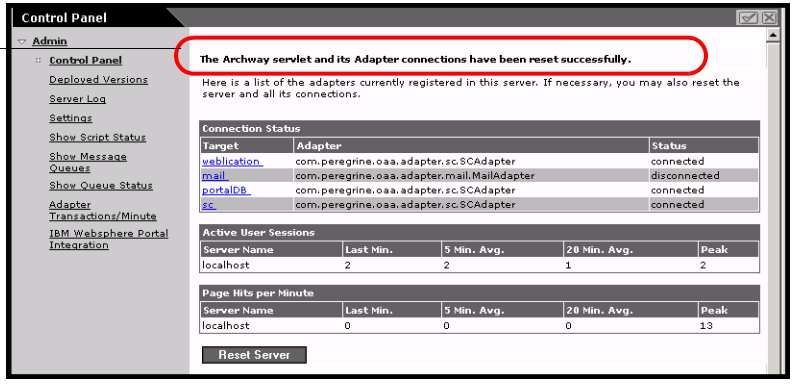
Use the Control Panel page to check the status of the connections to the databases you are accessing with Get-Resources and your Web applications. You can also reset the connection between the Archway servlet and the adapters to the back-end systems.

To reset the connection between the Archway servlet and back-end system:

- Click **Reset Server.**

A message at the top of the page indicates that the connections are reset.

Informational, warning, and error messages appear at the top of the page.



Viewing the Deployed Versions

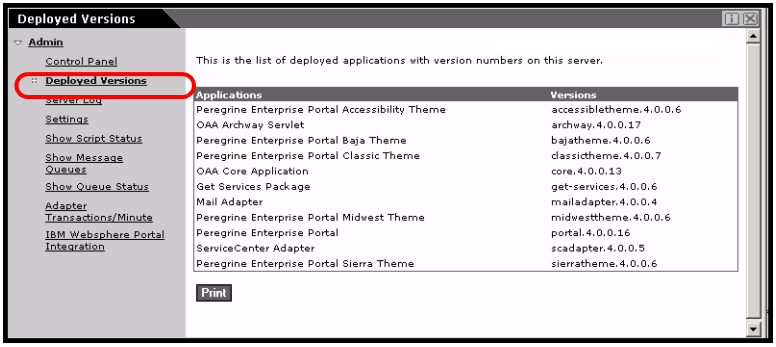
The Deployed Versions screen lists all of the packages that deploy during the installation, including the version number of each package.

To view the Deployed Versions list:

1 From the Activity menu, select **Deployed Versions**.

A list of the installed packages opens.

Current applications and their versions are available for viewing with the Deployed Versions option.



2 Click **Print** for a printout of this list.

Viewing the Server Log

The Server Log provides a history of server events. The default file name is archway.log.

To view the Server Log:

1 From the Activity menu, select **Server Log**.

A form opens with a drop-down list for you to select the log you want to view.

You can view the log file from your Web browser or download it to your preferred location.

2 Click the drop-down and select the log file you want to view.

3 Set the number of lines to view.

4 Do one of the following:

- Click **View** to see the log file from your Web browser.
- Click **Download** to initiate the File Download wizard that downloads the archway.log file to a location of your choice.

Using the Settings page

On the 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.

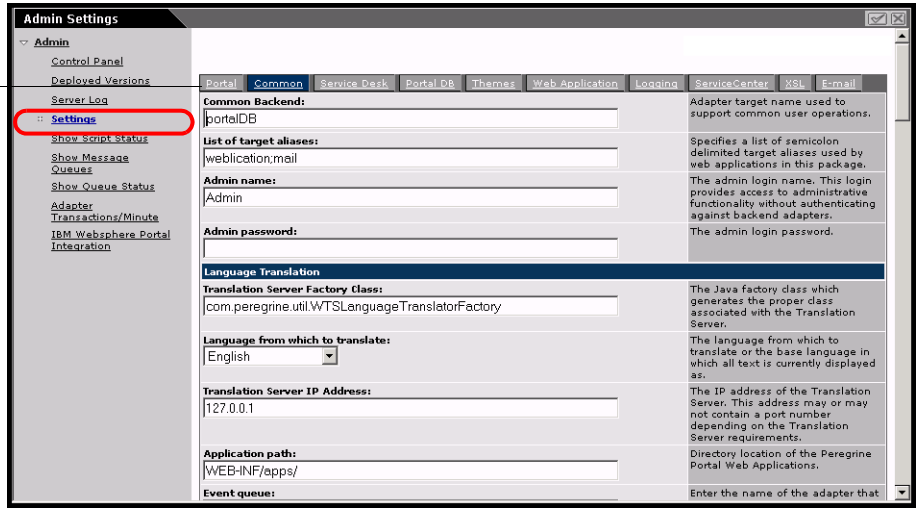
Settings for the Portal, PortalDB, Web Application, AssetCenter (ACadapter), and Service Center (SCadapter) tabs are set during the installation (refer to the *Get-Resources Installation Guide*). You can access the Settings page at any time to change the installation settings. Set the E-mail tab only when users have access to self-registration (see *User self-registration* on page 84).

To view Settings:

- From the Activity menu, click **Settings**.

  Each parameter on the tab has a description that guides you through the settings.

The tabs you see on the Settings page depend on the Web applications you installed.



Setting parameters using the Admin module

When you make changes using the Admin Settings page, a local.xml file is created in the C:\<*application server*>\webapps\oaa\WEB-INF directory. All changes to property settings are stored in this file. Restart Tomcat after making changes that are stored in local.xml.

Important: If you change parameters on the Admin Settings page and then need to reinstall Get-Resources, it is important that you copy the local.xml file to a location other than your Get-Resources installation, or all of your settings will be lost when you redeploy Get-Resources. After the installation, move the copy back to the WEB-INF directory.

To define a parameter:

1 Locate the setting you want to change and type the new parameter.

Note: If you have previously changed a setting and want to return to the default setting, click the **Click for default** link displayed in the description area for the parameter you want to revert. This link appears only when a setting is different from the default.

2 Scroll to the bottom of the page, and then click **Save**.

The Control Panel opens.

3 Click **Reset Server**.

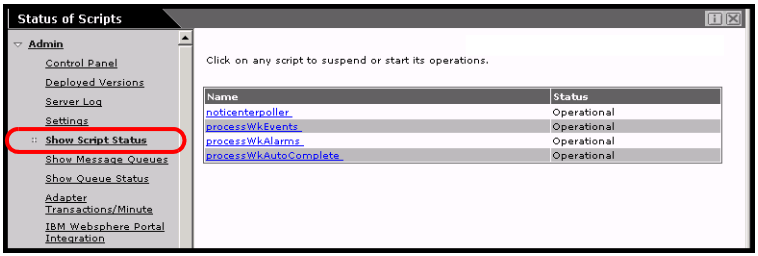
An information message at the top of the Control Panel indicates that the server has been reset.

Verifying Script Status

The Script Status page lists the name and status of any script that is currently running.

To verify the script status:

1 From the Administration Activity menu, click Show Script Status to display the **Status of Scripts** page that shows the name of each script.



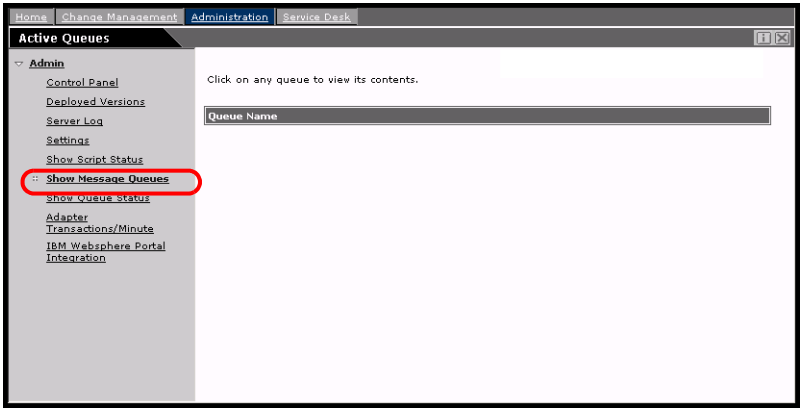
2 Click on the script to suspend it.

Displaying Message Queues

The Message Queues are displayed whenever a queue has data waiting to be transferred.

To display message queues:

1 From the Administration Activity menu, click Show Message Queues to display the Active Queues page.



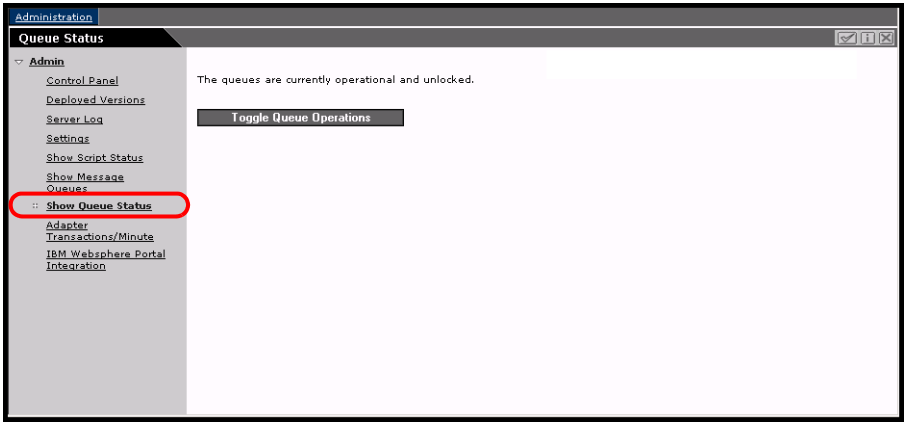
2 Click the queue name in the list to view the contents of a queue.

Showing Queue Status

Use the Show Queue Status option to verify or change the status of the message queues.

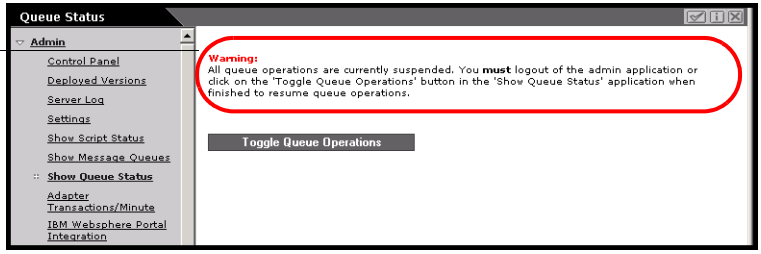
To show queue status:

1 From the Activity menu, click **Show Queue Status** to open the Queue Status page.



2 Click **Toggle Queue Operations** to change the status to **suspended**.

A warning message indicates that the Queue Status is suspended.



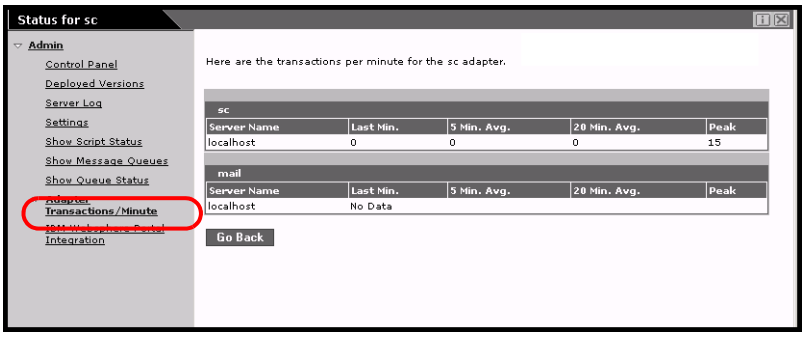
3 Click **Toggle Queue Operations** to return to the operational status.

Viewing adapter transactions

You can track your adapter transactions by viewing the adapter Status page.

To view adapter transactions per minute:

- From the Activity menu, click **Adapter Transactions/Minute** to open the adapter **Status** page.

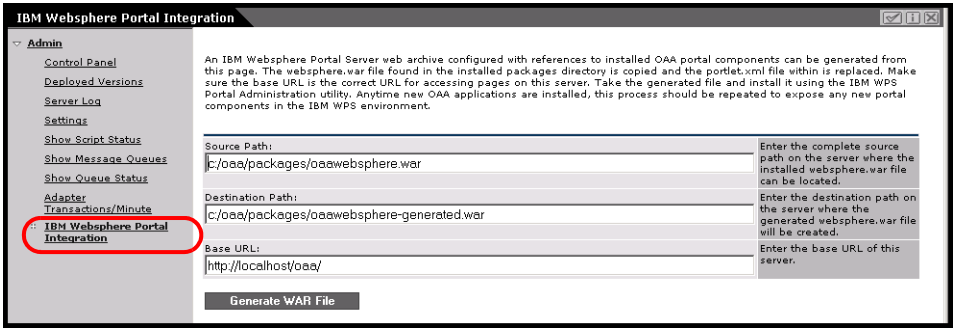


Using the IBM Websphere Portal

You can generate an IBM Websphere Portal Server web archive (war) file configured with references to installed OAA portal components.

To generate a war file:

1. From the Activity menu, click **IBM Websphere Portal Integration** to open the **Portal Integration** page.



2. Enter the following information:

- source path
- destination path
- base URL

3 Click **Generate WAR File**.

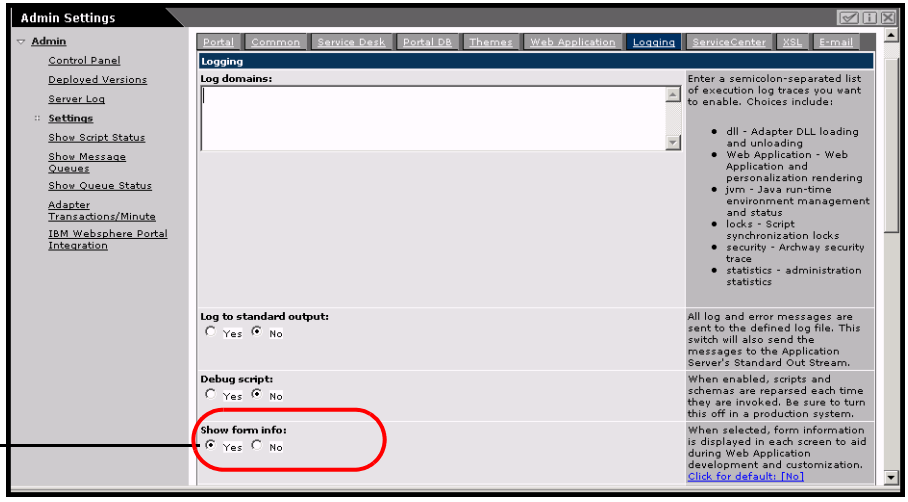
Displaying form information

You can use the Admin module to configure Web application forms to display the location and file name of the current form.

To display form information:

1 From the Admin module, click **Settings**, then **Logging**.

2 Scroll to the **Show form info** field, and click **Yes** if necessary.



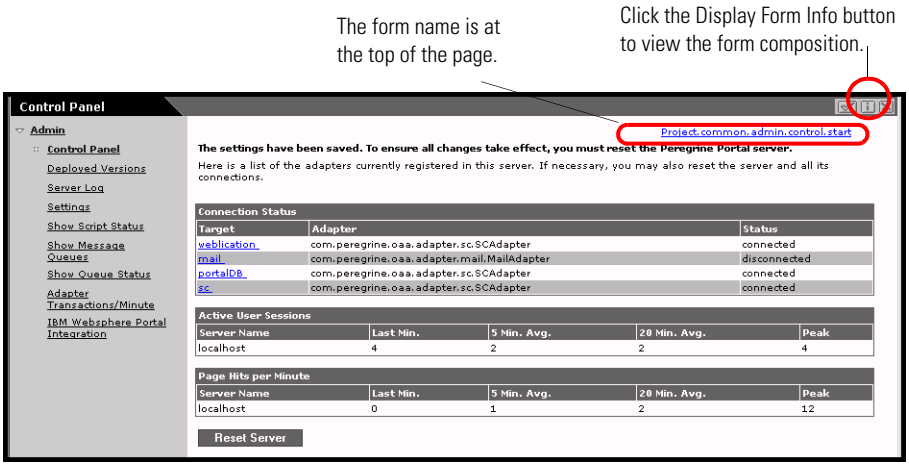
Set Show Form Info to Yes.

3 Click **Save**.

The Control Panel opens.

4 Click **Reset Server**.

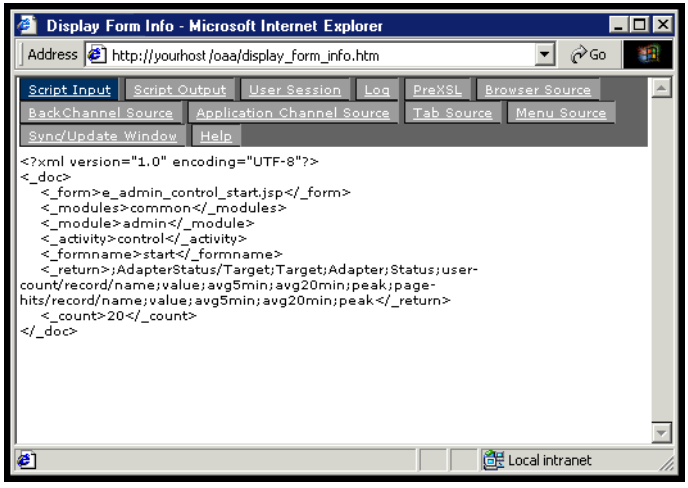
The name of the form is at the top of each form.



Displaying form details

You can also display detailed information about the current form. Click the **Display Form Info** button at the top right of the form. A separate window opens.

View the contents in each tab for more information about the form.



The form has the following tabs:

This tab	Contains
Script Input	the script that sends a request to the back-end system.
Script Output	the information returned by the script request to the back-end system.
User Session	details about the current user session, including browser type, back-end system version, and the access rights established for this user.
Log	a list of actions taken by the script to execute the form.
PreXSL	output from XSL before it gets rendered to the browser.
Browser Source	HTML source code for the current page.
BackChannel Source	HTML source code for frames where the data is stored.
Application Channel Source	HTML source code for the shared applications.
Tab Source	HTML source code for tabs.
Menu Source	HTML source code for menus.
Sync/Update Window	HTML source code to synchronize with the page and reload.
Help	Help for debugging the window.

User self-registration

With the Admin module, administrators can choose to have end users self-register for new accounts from the login screen if the user is not already in the ServiceCenter database. When the user registers, ServiceCenter creates an Operator record and a Contact record for the new user with basic user login rights. See the chapter on *Security* in this guide for more information on the registration process.

To enable users to self-register from the Login screen:

1 From the Admin module Settings page, click **Common**.

2 Scroll to **Enable User Registration.**

Click Yes to give users the ability to self-register for new accounts.

3 Click **Yes.**

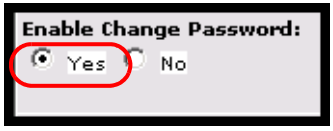
Changing passwords

Using the Admin module, administrators can choose to have end users change their own passwords from the Home page.

To enable users to change passwords:

1 From the Admin module Settings page, click **Common.**

2 Scroll to **Enable Change Password.**



Click Yes to give users the ability to change their own passwords.

3 Click **Yes.**

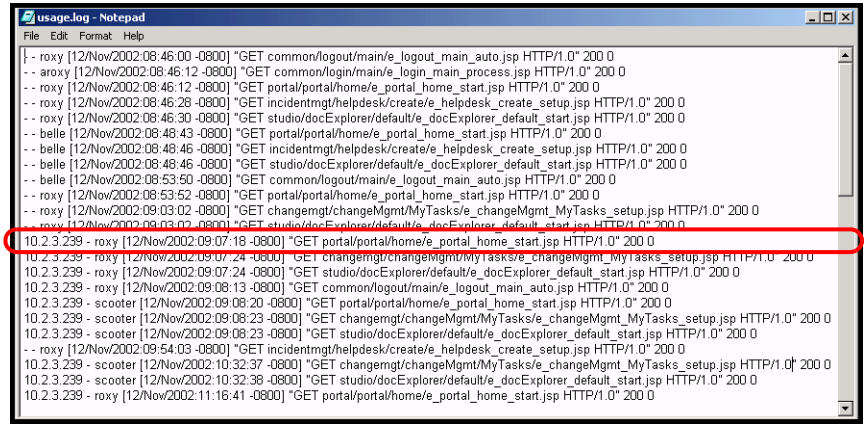
Logging and monitoring user sessions

The usage.log file has a record of user logins that is in the bin directory of your application server installation. With this file, you can determine which application is in use and how many users access an application during a day.

Understanding the usage.log file

The following line shows an excerpt from a usage.log file:

10.2.3.239 - roxy [12/Nov/2002:09:07:18 -0800] "GET portal/portal/home/e_portal_home_start.jsp HTTP/1.0" 200 0



Each login is on a line. Within one user session, each module logs only one line.

The following table shows the meaning of each element in the log entry:

Remote Host	Rfc931	User Login	Date	Request	Status	Bytes
10.2.3.239	-	roxy	[12/Nov/2002:09:07:18 -0800]	"GET portal/portal/home/e_portal_home_start.jsp HTTP/1.0"	200	0

This element	Contains
Remote Host	the remote host name or IP address if the DNS host name is not available or was not provided.
Rfc931	the remote login name of the user. This is always a dash because this information is not needed.
User Login	the user name authenticated to log in to the Peregrine Portal.
Date	the date and time of the request.
Request	the module accessed by the user. The name of the module is the first part of the GET parameter. In the previous above, the module accessed is *notificationservices*, the location of the login script.
Status	the HTTP response code returned to the client. This value is always 200 to specify that it was a valid request.
Bytes	the number of bytes transferred. The number is always entered as 0, because this information is not needed.

CHAPTER 7 Back-end System Administration

The following sections describe administrative functions that must be performed outside of Get-Resources to support certain features available with ServiceCenter or AssetCenter:

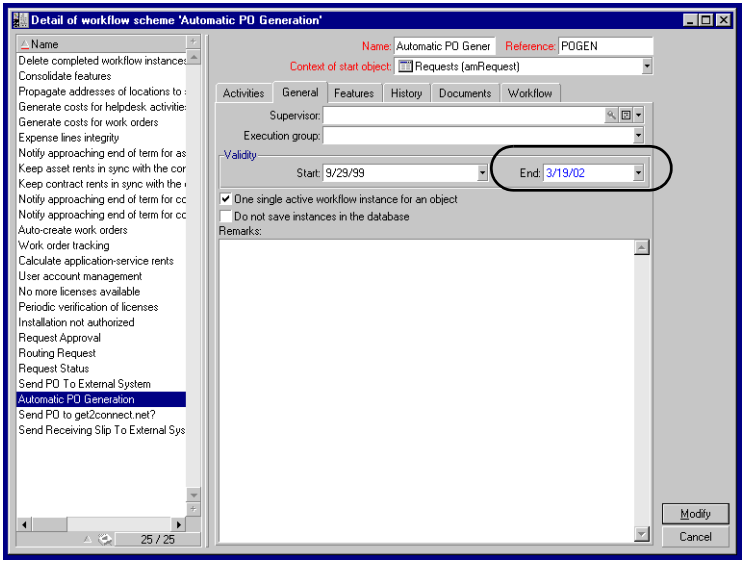
- *Configuring the Purchase Order Generation workflow* on page 90
- *Configuring the Product Catalog* on page 91
- *Configuring the Select a type of request form* on page 93
- *Configuring the Select an item type form* on page 99

Configuring the Purchase Order Generation workflow

The Automatic PO Generation workflow in AssetCenter causes a purchase order to be created automatically each time a request is submitted in Get-Resources. You may want to disable this workflow in AssetCenter.

To disable the Automatic PO Generation workflow:

1. In AssetCenter, go to Tools>Workflow>Workflow Schemes.
2. In the list of Workflows, select Automatic PO Generation.
3. Select the General tab.
4. In the Validity section, End field, set the date to a time in the past.



5. Click Modify.
6. Restart your application server.

Configuring the Product Catalog

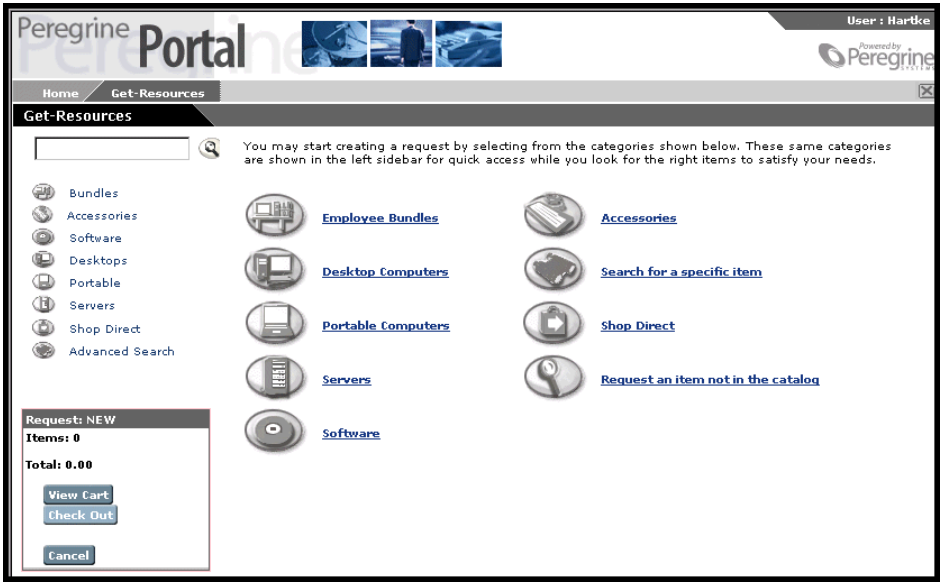
This information applies only to AssetCenter 3.6.

Get-Resources uses the AssetCenter product catalog contained within the amProduct table. There are two areas in the catalog which require special configuration:

- Certification field
- Calculated field

Certification

Get-Resources uses the Certification field to determine the availability of items in the AssetCenter catalog.



These buttons from the Get.It! menu each drive a database call against the amProduct table. The queries executed are similar to the following for Desktop Computers:

SELECT lProdId,Brand,Model,mPrice FROM amProduct WHERE (Certification LIKE 'Desktop%')

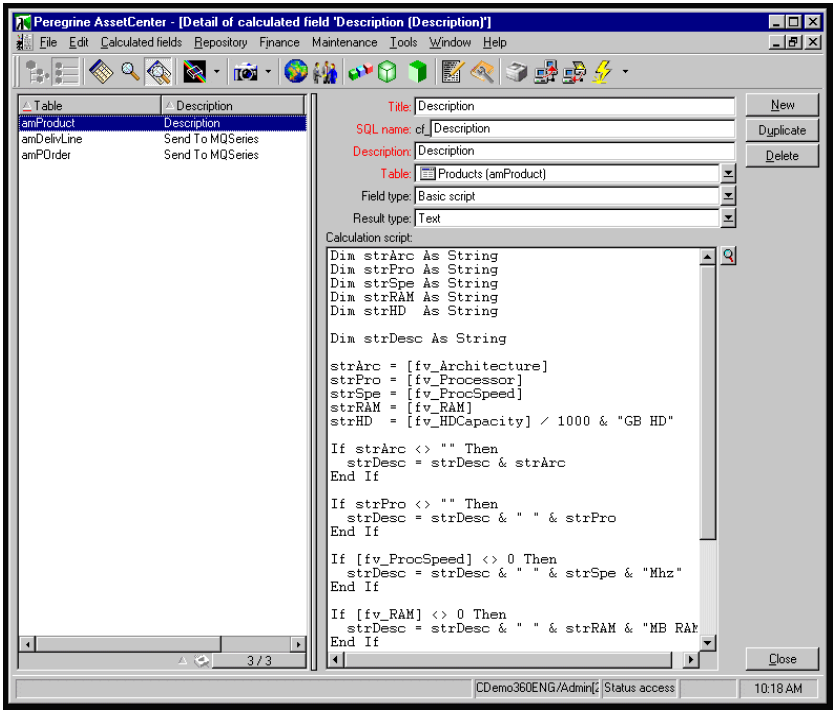
With the exception of the Bundle certification, all may be easily changed to meet your business requirements.

The Bundle certification is special within the weblication. Bundles are groups of items tied together for a specific purpose. For example, a Sales Laptop Bundle may consist of a laptop, PCMCIA NIC, Operating System software, and some applications. This relationship is built within the amProdCompo table, tying together several records from the amProduct table.

Calculated Field: cf_Description

A calculated field is used as a descriptive name for records within the catalog. As identified in the Product schema, the field, Description, maps to the field cf_Description, a calculated field.

The following screen shows a sample of how AssetCenter's calculated fields may be used within Get-Resources to ease data presentation. Refer to your AssetCenter documentation for information about calculated fields.



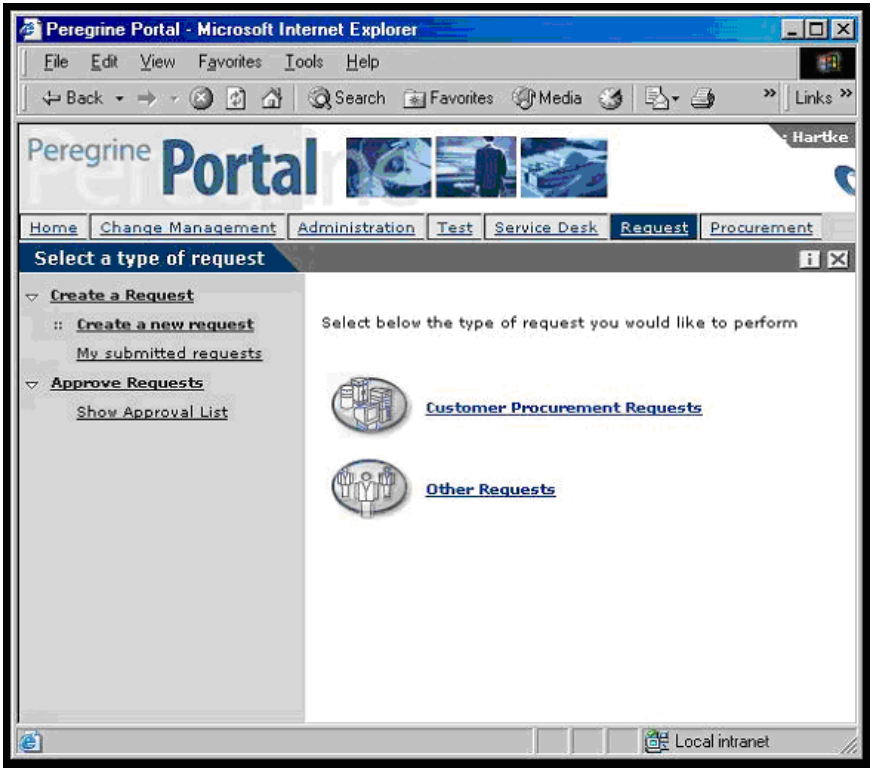
Configuring the Select a type of request form

This section describes:

- The two forms that can be customized to help the user choose what type of request they are about to make
- The XML file used to build the forms
- Steps for modifying the **Select a type of request** forms
- The file format of the XML file

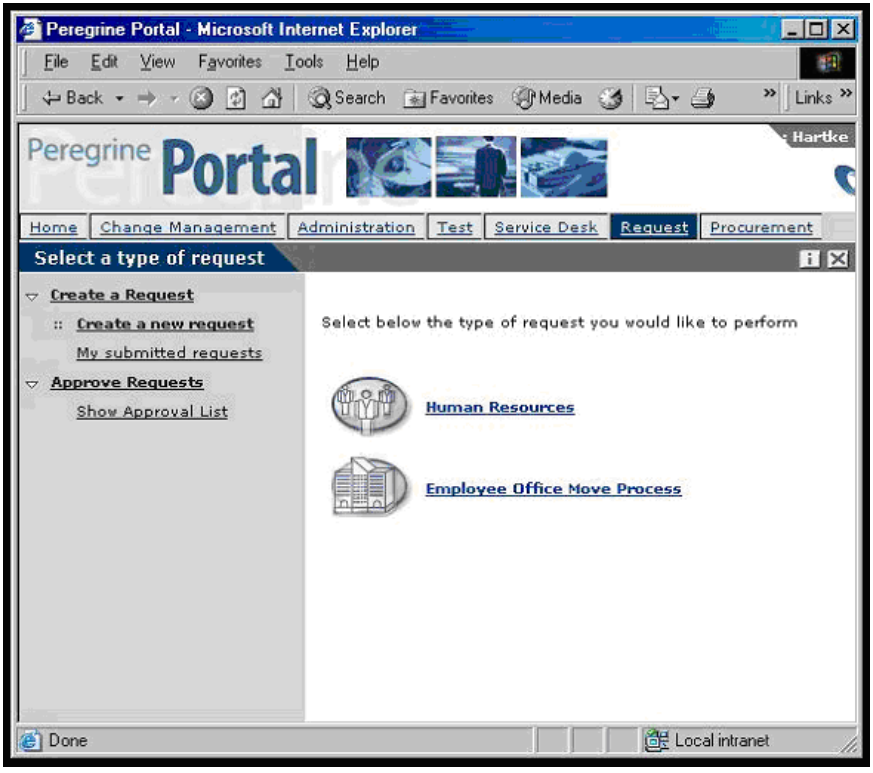
Select a Type of Request forms

When Get-Resources is integrated with ServiceCenter, the **Select a type of Request** form is the first form a user sees when they create a new request. The graphic below presents a possible way for the form to be presented.



The **Select a type of request** form presents the quote categories in ServiceCenter. This selection controls which item categories the user can add to the current request. It also controls the way the request summary screen appears.

Furthermore, another form can be presented to users who click the **Other Requests.** An example of a possible **Other Requests** form is presented in the following graphic:



How the forms are built

The **Select a type of request** forms are not built dynamically from the data in ServiceCenter. The contents are derived from the following file:

../WEB-INF/etc/grtrees/screquestcategory.xml.

The two graphics presented above were created from the following screquestcategory.xml file:

```
<?xml version="1.0" encoding="UTF-8"?>
<WizardMenu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../menu.xsd">
 <Answers>
  <WizardGRTree>
   <Id>customer</Id>
   <!-- <Title>Customer Procurement Requests</Title> -->
   <Title_ids>resources,menuTitleRequestProcurement</Title_ids>
   <Image>icons/oaa_assets.gif</Image>
  </WizardGRTree>
  <WizardMenu>
   <Id>other</Id>
   <Title>Other Requests</Title>
   <Image>icons/oaa_org.gif</Image>
   <Answers>
    <WizardGRTree>
     <Id>hr</Id>
     <!-- <Title>Human Resources</Title> -->
     <Title_ids>resources,menuTitleRequestHR</Title_ids>
     <Image>icons/oaa_org.gif</Image>
<!-- The following element allows to personalize the request summary screen
differently than the standard screen -->
     <SubType>hr</SubType>
    </WizardGRTree>
    <WizardGRTree>
     <Id>office move</Id>
     <!-- <Title>Employee Office Move Process</Title> -->
     <Title_ids>resources,menuTitleRequestMove</Title_ids>
     <Image>icons/oaa_places.gif</Image>
     <!-- The following element allows to personalize the request summary screen
differently than the standard screen -->
     <SubType>move</SubType>
    </WizardGRTree>
   </Answers>
  </WizardMenu>
 </Answers>
</WizardMenu>
```

Modifying the Select a type of request form

1. Copy the WEB-INF/etc/grtrees/screquestcategory.xml file.
2. Paste the file into the WEB-INF/etc/grtrees/user/ directory. Make sure the name of the file is screquestcategory.xml

   Note: When a file is present in the user direstory, Get-Resources uses it instead of the one delivered with Get-Resources.

3. Find menu.xsd and replace it with ../menu.xsd
4. Determine how many of the quote categories in ServiceCenter should appear on the Get-Resources **Select a type of request** form. Write them in the space below:

5. For each quote category, make sure there is an appropriate *.gif graphic in the icons directory. For example, if you are using the baja skin, make sure that there is a graphic in the oaa\images\skins\baja\icons directory.
6. For each of the quote categories, you will need a section of code that resembles the following:

```
<WizardGRTree>
  <Id>customer</Id>
  <!-- <Title>Customer Procurement Requests</Title> -->
  <Title_ids>resources,menuTitleRequestProcurement</Title_ids>
  <Image>icons/oaa_assets.gif</Image>
</WizardGRTree>
```

   Note: <Id>customer</Id> must be the exact quote category **name** in ServiceCenter. <Title>Customer Procurement Requests</Title> is simply text that appears on the form. <Title_Ids> becomes relevant if you are localizing.

7. Make changes as necessary. Refer to the next section for a detailed explanation of the file format.
8. Restart your the server in Peregrine Portal Administration.
9. Log in as a user with getit.requester access rights.
10. Navigate to the **Create a new request** activity to validate your changes.

File format of screquestcategory.xml

The root XML element is WizardMenu. It refers to the XML schema that is used to validate the file.

```
<?xml version="1.0" encoding="UTF-8"?>
<WizardMenu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../menu.xsd">
…
</WizardMenu>
```

The **WizardMenu** element contains one **Answers** element.

The **Answers** element contains a list of elements that represent the icons that you can see on the **Select a type of request** form. This sub-element can be configured two ways:

- **WizardGRTree**. This element will be represented on the screen as a clickable icon that will lead to the item category selection.
- **WizardMenu**. This element will be represented on the screen as a clickable icon that will lead to a sub-menu, in case the administrator would like to organize the request type selection in a hierarchical manner.

The **WizardGRTree** element is described by the following elements:

- **Id**: Required tag; must be the ServiceCenter quote category name.
- **Title**: Optional; contains the text displayed in Get-Resources.
- **Title_ids**: Optional; contains the string id representing the text displayed for the menu entry. This value is expressed as *module,stringname* where *module* corresponds to the file name containing the string, and *stringname* is the id of the string in this file.

Note: Title or Title_ids must be specified. If Title is specified, it has precedence over Title_ids, and all users, regardless of the locale they choose when they log in, will view the same exact text.

- **Image**: This required element is the path to the image that is displayed on the screen in front of the text for this menu entry. The value is a path to the icon, relative to the skin directory (e.g. icons/oaa_assets.gif).
- **SubType**: This optional element contains an alphanumeric string. If set, the personalization performed on the request summary screen for this request type will be saved separately. Any two request types that have the same SubType will share the same personalization of the request summary screen.

The **WizardMenu** sub-element is described by the following elements:

- **Id**: Required element which must be unique among the siblings of the WizardMenu element.
- **Title**: Optional element which contains the text displayed for the menu entry.
- **Title_ids**: Optional element which contains the string id representing the text displayed for the menu entry. This value is expressed as *module,stringname* where *module* corresponds to the file name containing the string, and *stringname* is the id of the string in this file. Note that either Title or Title_ids must be specified. If Title is specified, it has precedence over Title_ids, and all users, regardless of the locale they choose when they log in, will view the same exact text.
- **Image**: This required element is the path to the image that is displayed on the screen in front of the text for this menu entry. The value is a path to the icon, relative to the skin directory (e.g. icons/oaa_assets.gif).
- **Answers**: This element describes the sub-menu entries and its structure is similar to the Answers element described above.

Configuring the Select an item type form

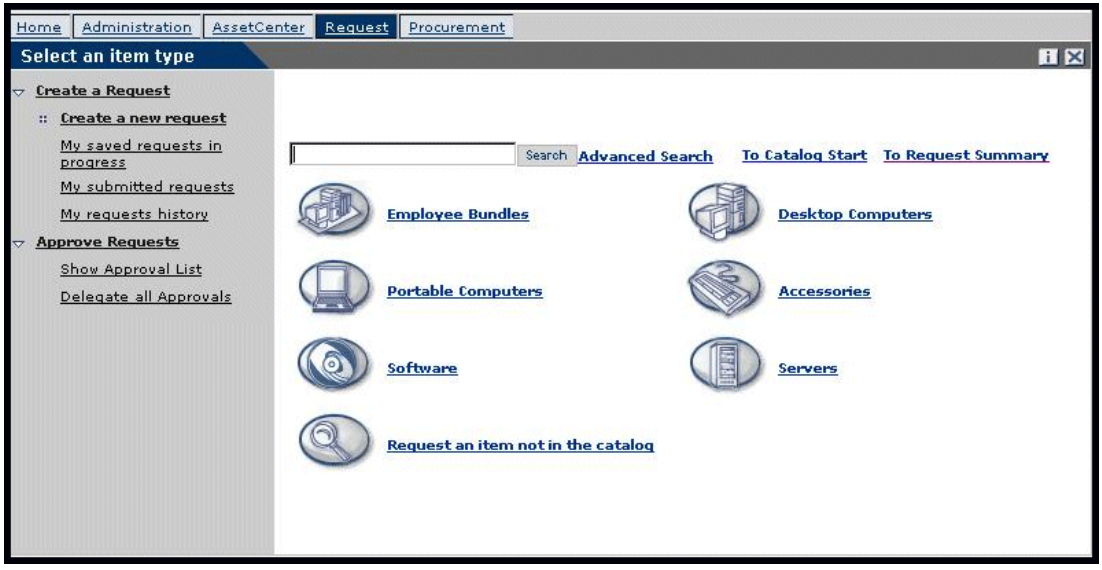
This section describes:

- The a form that allows a user to select an item type.
- The XML file used to build the forms
- Steps for modifying the **Select a an item type** form
- The file format of the XML file

Select an item type form

When Get-Resources is integrated with AssetCenter, the contents of the **Select an item type form** is defined in a configuration file. The configuration file describes the type of action to perform when you click the icon. Furthermore, an option to show a list of catalog items is defined by a file which describes what table it should pull the items from and what subset of the record should be shown.

Note that these configuration files replace the settings performed in the AssetCenter 4 Administration application . Following is a sample **Select an item type** form:



How the forms are built

The **Select an item type** form is not built dynamically from the data in AssetCenter. Its contents are derived from the file:

../WEB-INF/etc/grtrees/ac#reqlineitemcategory.xml

where # is 3 or 4 depending upon the AssetCenter version you are using as a back-end system.

The graphic presented above presents the ac3reqlineitemcategory.xml file. This file contains the following text:

```
<?xml version="1.0" encoding="UTF-8"?>
<WizardMenu extends="ac3lineitemcategory.xml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="menu.xsd">
 <Answers>
  <WizardGRTree>
   <Id>offcatalog</Id>
   <!-- <Title>Request an item not in the catalog</Title> -->
   <Title_ids>resources,menuTitleOffCatalog</Title_ids>
   <Image>icons/catoffcat.gif</Image>
   <CatalogId>offcatalog</CatalogId>
  </WizardGRTree>
 </Answers>
</WizardMenu>
```

Notice that the second line of the file indicates this is an extension of another file: ac3lineitemcategory.xml. The contents of this file are as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<WizardMenu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="menu.xsd">
 <Answers>
  <WizardGRTree>
   <Id>bundle</Id>
   <!-- <Title>Employee Bundles</Title> -->
   <Title_ids>resources,menuTitleCatBundle</Title_ids>
   <Image>icons/catbundle.gif</Image>
   <QueryParam>
```

```
  <Certification operator="=">Bundle</Certification>
 </QueryParam>
</WizardGRTree>
<WizardGRTree>
 <Id>desktop</Id>
 <!-- <Title>Desktop Computers</Title> -->
 <Title_ids>resources,menuTitleCatDesktop</Title_ids>
 <Image>icons/catdesktop.gif</Image>
 <QueryParam>
  <Certification operator="=">Desktop</Certification>
 </QueryParam>
</WizardGRTree>
<WizardGRTree>
 <Id>laptop</Id>
 <!-- <Title>Portable Computers</Title> -->
 <Title_ids>resources,menuTitleCatPortable</Title_ids>
 <Image>icons/catportable.gif</Image>
 <QueryParam>
  <Certification operator="=">Laptop</Certification>
 </QueryParam>
</WizardGRTree>
<WizardGRTree>
 <Id>accessories</Id>
 <!-- <Title>Accessories</Title> -->
 <Title_ids>resources,menuTitleCatAccessories</Title_ids>
 <Image>icons/cataccessories.gif</Image>
 <QueryParam>
  <Certification operator="=">Accessories</Certification>
 </QueryParam>
</WizardGRTree>
<WizardGRTree>
 <Id>software</Id>
 <!-- <Title>Software</Title> -->
 <Title_ids>resources,menuTitleCatSoft</Title_ids>
 <Image>icons/catsoftware.gif</Image>
 <QueryParam>
  <Certification operator="=">Software</Certification>
 </QueryParam>
</WizardGRTree>
<WizardGRTree>
```

```
    <Id>servers</Id>
    <!-- <Title>Servers</Title> -->
    <Title_ids>resources,menuTitleCatServers</Title_ids>
    <Image>icons/catserver.gif</Image>
    <QueryParam>
     <Certification operator="=">Server</Certification>
    </QueryParam>
   </WizardGRTree>
  </Answers>
</WizardMenu>
```

Modifying the Select an item type form

1. Copy the file approprate for your version of AssetCenter:
   - version 3.6 = WEB-INF/etc/grtrees/ac3reqlineitemcategory.xml.
   - version 4.x = WEB-INF/etc/grtrees/ac4reqlineitemcategory.xml.
2. Paste the file into the WEB-INF/etc/grtrees/user/ directory. Make sure the file name did not change from step #1.

   Note: When a file is present in the user direstory, Get-Resources uses it instead of the one delivered with Get-Resources.

3. Find menu.xsd and replace it with ../menu.xsd
4. Make changes as necessary. Refer to the next section for an in depth description of the file format.
5. Restart your the server in Peregrine Portal Administration.
6. Log in as a user with getit.requester access rights.
7. Navigate to the **Create a new request** option to validate your changes.

File format of ac#reqlineitemcategory.xml

The root XML element is WizardMenu. It refers to the XML schema that is used to validate the file.

```
<?xml version="1.0" encoding="UTF-8"?>
<WizardMenu xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="../menu.xsd">
…
</WizardMenu>
```

The **WizardMenu** element contains one **Answers** element.

The **Answers** element contains a list of elements that represent the icons that you can see on the **Select an item type** screen. This sub-element can be configured two ways:

- **WizardGRTree**. Depending on how it is configured, this element will be represented on the screen as a clickable icon that can display one of the following:
  - a list of catalog items to select
  - a search screen
  - a free form item creation screen
- **WizardMenu**. This element will be represented on the screen as a clickable icon that will lead to a sub-menu, in case the administrator would like to organize the item type selection in a hierarchical manner.

The **WizardGRTree** element is described by the following elements:

- **Id**: Required element; must be unique amongst the siblings of the WizardGRTree element.
- **Title**: Optional; contains the text displayed in Get-Resources.
- **Title_ids**: Optional; contains the string id representing the text displayed for the menu entry. This value is expressed as *module,stringname* where *module* corresponds to the file name containing the string, and *stringname* is the id of the string in this file.

Note: Title or Title_ids must be specified. If Title is specified, it has precedence over Title_ids, and all users, regardless of the locale they choose when they log in, will view the same exact text.

- **Image**: This required element is the path to the image that is displayed on the screen in front of the text for this menu entry. The value is a path to the icon, relative to the skin directory (e.g. icons/catbundle.gif).
- **CatalogId**: This optional value is the name of the catalog script used to retrieve the catalog data from the database. There are three catalogs:
  - **catalogbase**: The default value (used when CatalogId is not specified). WEB-INF/apps/resources/schema/Product.xml is the database schema used for this catalog. For AssetCenter 3.6, amProduct table is mapped. For AssetCenter 4.x, amCatRef table is mapped.

  - **ac4bundlecatalog**: Only available for use with AssetCenter 4.x. WEB-INF/apps/resources/schema/ac4bundle.xml is the database schema used to retrieve the values and it is mapped to the amRequest table.
  - **offcatalog**: Clicking on an icon with this CatalogId will open a window from which the user can request an item in free text format.

- **DestType**: This optional element can be configured with two values:
  - **list**: Clicking this menu entry presents a list of catalog items available to add to your request. The database schema, and therefore the table used to retrieve the data, depends on the CatalogId.
  - **search**: Clicking this menu entry presents a search page.
- **QueryParam**: This optional element represents the search parameters that will be used to filter the list of catalog items. The actual search parameters that can be used depends on the database schema that the CatalogId uses. This element contains one or more sub-elements. The name of these sub elements are the attribute name (which can be found in the schema), and the value set is used in the query that retrieves the catalog items.

The **WizardMenu** sub-element is described by the following elements:

- **Id**: Required element which must be unique among the siblings of the WizardMenu element.
- **Title**: Optional element which contains the text displayed for the menu entry.
- **Title_ids**: Optional element which contains the string id representing the text displayed for the menu entry. This value is expressed as *module,stringname* where *module* corresponds to the file name containing the string, and *stringname* is the id of the string in this file. Note that either Title or Title_ids must be specified. If Title is specified, it has precedence over Title_ids, and all users, regardless of the locale they choose when they log in, will view the same exact text.
- **Image**: Required element which defines the path to the image that is displayed on the screen in front of the text for this menu entry. The value is relative to the skin directory (e.g. icons/oaa_assets.gif).
- **Answers**: This element describes the sub-menu entries and its structure is similar to the Answers element described above.

8 Security

CHAPTER

This chapter describes the different security configuration options available in Get-Resources. Topics in this chapter include:

- *Password encoding methods* on page 106
- *Back-end system security* on page 107
- *Get-Resources global access rights* on page 111
- *User registration* on page 112
- *Authenticating users* on page 114
- *Default security configuration* on page 115
- *Custom JAAS configuration* on page 116
- *Standard Sun Microsystems JAAS configuration* on page 124
- *Windows NT Challenge/Response* on page 125
- *Creating an alternate login page* on page 131

Password encoding methods

By default, Get-Resources does not encode passwords sent over the network. Get-Resources sends plain text passwords to the authenticating back-end databases and stores plain text passwords in a browser cookie if the user selects to **enable automatic login**. If you want to secure your Get-Resources passwords, you have three options:

- Enable Secure Sockets Layer (SSL) on your Web server
- Configure Get-Resources to use a directory service such as LDAP
- Enable your Web server to use Windows NT Challenge/Response

In order to use SSL, you need to acquire a digital certificate. If your Web server has a certificate, then your Get-Resources login URL must include the https protocol indicator. After the user browser has made a secure connection to the Web server, all data transferred is encrypted. Refer to your Web server documentation for information on configuring SSL.

Get-Resources also supports authentication via a directory service such as LDAP. When you authenticate to a directory service, Get-Resources passes SHA hash encoding passwords to the service. For instructions configuring a directory service see *Custom JAAS configuration* on page 116.

Get-Resources also supports Windows NT Challenge/Response authentication. When this form of authentication is used, passwords are not actually exchanged between the browser and Web server, and the authentication process is kept secure. However, Windows NT Challenge/Response is only supported by Internet Explorer browsers on Windows systems. For instructions configuring Windows NT Challenge/Response see *Windows NT Challenge/Response* on page 125.

Back-end system security

This section includes information about how Get-Resources authenticates users and stores personalization changes in the ServiceCenter or AssetCenter back-end system.

Authentication with ServiceCenter or AssetCenter

When a user logs in to Get-Resources, the user name and password are validated against a corresponding **operator record** in ServiceCenter or the Employee table in AssetCenter. Each operator record or employee profile must contain appropriate **capability words** or **user right keywords** in order to gain access to and use different features within Get-Resources.

Refer to the *ServiceCenter Administration Guide* for more information on operator records, or the *AssetCenter Administration Guide* for more information on profiles.

ServiceCenter capability words and AssetCenter user right keywords

Following is a list of available capability words and user rights keywords for Get-Resources functionality that can be assigned to an operator record in ServiceCenter or a profile in AssetCenter:

ServiceCenter capability word or AssetCenter user right keyword	Description
getit.admin	Provides access to the OAA Admin module.
getit.advancedrequester	Enables access to advanced request form features in Get-Resources, including the ability to split request lines and assign request line items to different end users. This is useful for requesters who typically request items for a group of people.
getit.approver	Enables access to approve requests.
getit.buyer	Enables user to create and change the status of purchase orders. Only available with AssetCenter.

ServiceCenter capability word or AssetCenter user right keyword	Description
getit.pcardmanager	Enables users to create new Pcards and manage the rights for the Pcards they created. Only available with AssetCenter.
getit.pcarduser	Gives the user access to Pcards on the request and purchase order screens. These fields are not displayed if this access is not granted. Only available with AssetCenter.
getit.personalization.admin	Can personalize the OAA interface with admin privileges.
getit.personalization.default	Can personalize the OAA interface.
getit.personalization.limited	Can do limited personalization.
getit.portal	Can view the OAA home page and portal components.
getit.receiver	Enables access to receive items associated with a request. Only available with AssetCenter.
getit.requester	Enables access to create requests in Get-Resources.
getit.reserve	Enables approver to reserve resources from existing stock. Must have getit.approver access to function.
oaa.forbidden	Reserved capability word to prevent access to all OAA users.
oaa.root	Grants access to all capability words except those listed with oaa.forbidden.

Refer to the *ServiceCenter Administration Guide* for detailed instructions on assigning capability words to operator records, or the *AssetCenter Administration Guide* for detailed instruction on how to add user rights to profiles.

AssetCenter sample security data

Sample profiles

Get-Resources provides several pre-set sample profiles in AssetCenter which combine user rights to give access to various modules. You can create profiles in AssetCenter with any combination of rights for each user.

Get-Resources includes the following sample profiles:

Profile	Module you can Access
getit.admin	Administration, Resources, Status, Approval, Receiving, Shop Direct, Pcard
getit.default	Resources, Status
getit.full	Resources, Status, Approval, Receiving, ShopDirect, Pcard
getit.buyer	Purchasing, Pcard, Resources
getit.requester	Resources, Status, Pcard. This profile provides the minimum rights needed to complete a request.

When users self-registers, they are initially assigned *getit.default* authority. You can update the AssetCenter employee records of those users who will need full or administration access.

Sample users

There are three sample user profiles provided within the AssetCenter demo database to illustrate various access rights for Get-Resources:

Full Name	User Name	Priveleges
Michaela Tossi	Tossi	■ Requesting
Richard Hartke	Hartke	■ Requesting ■ Approving ■ Receiving
Michael Valentine	Valentine	■ Requesting ■ Approving ■ Receiving ■ Purchasing ■ Administration

From each user's profile tab in AssetCenter, you can also view the accessible tables for that user. Refer to the *AssetCenter Administration Guide* for detailed instructions on viewing user rights.

ServiceCenter password security

You can set the ServiceCenter parameter **securepassword** in the ServiceCenter sc.ini file to prevent advanced users from submitting a Get-Resources query that returns a list of user passwords.

To set the password security parameter in ServiceCenter:

1 Open the sc.ini file in a text editor.

2 Add the parameter securepassword to the file, and save the file.

A request for a list of passwords in Get-Resources returns a list with the passwords masked.

Get-Resources global access rights

Although initial login access for Get-Resources is validated against the user's corresponding operator record in ServiceCenter or profile in AssetCenter, global access rights can be granted for all users regardless of how their individual security is defined. For example, defining **getit.requester** as a global access right gives all users the ability to create requests in Get-Resources even if it were not initially assigned to each user's operator record in ServiceCenter or profile in AssetCenter.

Global access rights are defined on the ServiceCenter or AssetCenter settings page of the Peregrine Portal Administration module.

To define global access rights within Get-Resources:

1. Open the Peregrine Portal Administration module in Get-Resources.
2. In the left menu pane, click **Settings**.
3. On the **Settings** page:
   - Click the **ServiceCenter** tab if ServiceCenter is your back-end system.
   - Click the **AssetCenter** tab if AssetCenter is your back-end system.
4. In the ServiceCenter or AssetCenter settings page, update the appropriate field with the global access right(s) that you want to grant to all users in the following format:

   <backend>(capability word)

   where <backend> represents **ac** for AssetCenter or **sc** for ServiceCenter as a back-end database.

   Multiple default access rights may be granted by separating the capability parameter values with a semicolon (;). For example:

   ```
   sc(getit.requester);sc(getit.approver)
   ```

   Following is an example of how you would update the appropriate settings page field for ServiceCenter or AssetCenter to grant all users the default right to create requests in Get-Resources:

Settings page	Field name	Sample field value
AssetCenter	Default Capability Words:	ac(getit.requester)
ServiceCenter	Default capabilities:	sc(getit.requester)

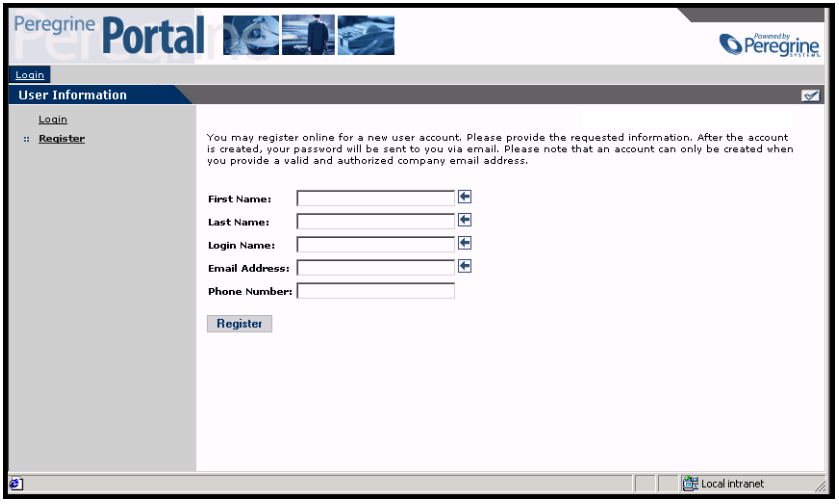
5 Scroll to the bottom of the form and click **Save**.

6 When the Control Panel page is displayed, click **Reset Server** to apply your configuration changes.

User registration

All Get-Resources users need a login account in the back-end database providing authentication. For example, if you are using ServiceCenter as your back-end database, then the appropriate capability words must be defined in the user's Operator record. In AssetCenter, the appropriate user rights are defined in the user's Profile. Similar access rights can be defined in any back-end system that you are using. The user login is automatically authenticated in the back-end system.

However, if a user is attempting to log in for the first time without back-end authentication, the user is prompted for certain default information as shown in the following screen. Note that the first four fields are required, as indicated by the arrows to the right of each field.



When the user clicks **Register**, the information is stored in the appropriate database. In AssetCenter, Get-Resources transforms this data into a Profile record that then passes to your AssetCenter system. An amEmplDept record is created with the user-supplied data and the default Profile **getit.default** is assigned. In ServiceCenter, Get-Resources creates an operator and contact record for the new user.

Note: The appropriate back-end system adapter must be defined before the capability words are recognized. For example, if no adapter is defined for ServiceCenter, the ServiceCenter capability words are not used.

Basic registration information and login scripts are stored in the .../oaa/apps/common/jscript/ directory. Login scripts are in the login.js file. If you want to make changes to the registration process, such as changing the way a user's password is defined, you can change the scripts in this directory.

Enabling the E-mail adapter

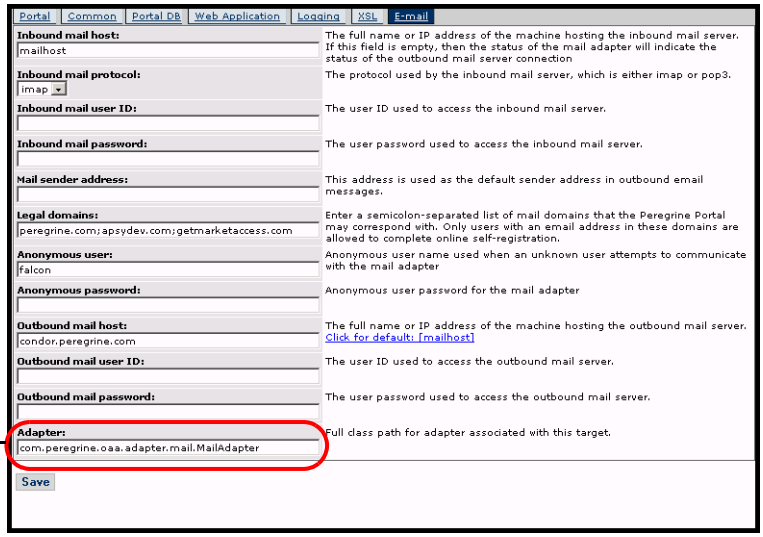
If users have the ability to self-register, you must make sure that the E-mail tab from the Get-Resources Admin module Settings page contains the MailAdapter name.

The MailAdapter is an implementation of JavaMail API 1.2 and supports the following mail protocols:

- POP3 for inbound mail
- IMAP for inbound mail
- SMTP for outbound mail

The MailAdapter also supports MIME type attachments in outbound e-mail.

Set the following parameters, as needed, on the E-mail tab of the Admin module Settings page.



Type the name of your MailAdapter in the Adapter field.

Troubleshooting the MailAdapter connection

You can check the status of the MailAdapter connection on the Control Panel. If the adapter shows as *disconnected*, check that the settings on the E-mail tab of the Settings page are correct. If you are still unable to connect, contact your system administrator for verification of the parameter values.

Authenticating users

You can configure the Peregrine OAA Platform to use one of five security authentication options:

- Use the default configuration to authenticate users against Peregrine adapters. See *Default security configuration* on page 115.
- Use a custom configuration to authenticate users against user-defined adapters such as LDAP or JDBC compliant databases. See *Custom JAAS configuration* on page 116.
- Use a standard JAAS configuration to authenticate users against the Sun Microsystem's standard Java Authentication and Authorization Service (JAAS). See *Standard Sun Microsystems JAAS configuration* on page 124.

- Use Windows NT Challenge and Response to authenticate users and pass the information to the Web application. See *Windows NT Challenge/Response* on page 125.
- Use an alternate login page and authenticate users against any of the other login options. See *Creating an alternate login page* on page 131.

Once a user is authenticated, the modules to which the user has access are defined by the back-end system. For example, if you are using AssetCenter and a user does not have access rights to a particular table in AssetCenter, the user cannot access the corresponding module in the Web application. If you are using ServiceCenter for the back-end system, the user must have the appropriate capability words set in the Operator record in ServiceCenter in order to see the corresponding module in the web application.

Default security configuration

The default configuration authenticates users against a set of pre-configured JAAS login modules. By default, one JAAS login module is configured for each registered Peregrine adapter. For example, if you are using both AssetCenter and ServiceCenter, then Get-Resources creates login modules for *both* the ACAdapter and the SCAdapter.

These login modules are *only* used to authenticate users. User access rights are derived from user profile records in the back-end systems (for example, ServiceCenter or AssetCenter). User access rights determine which modules the user can access and what tasks they can perform within those modules. For example, one user can open tickets only, while another has rights to approve tickets as well.

You do not have to do any additional configuration to use the default security configuration. Get-Resources automatically generates login modules for each Peregrine adapter installed on the system.

The default login module settings are:

- loginModule=com.peregrine.OAA.security.OAALoginModule
- control flag=OPTIONAL
- options=<none>

Custom JAAS configuration

A custom JAAS configuration authenticates users against a set of JAAS LoginModules you define in a **local.xml** file. This file describes the settings to use for each JAAS LoginModule. A **local.xml** file has the following format.

```
<jaas_config>

  <jaasConfiguration>CustomConfig</jaasConfiguration>
  <CustomConfig>adapter1;adapter2</CustomConfig>

  <adapter1>
    <loginModule>Java class of login module</loginModule>
    <controlFlag>authentication behavior</controlFlag>
    <options>semicolon separated list of options</options>
  </adapter1>

  <adapter2>
    <loginModule>Java class of login module</loginModule>
    <controlFlag>authentication behavior</controlFlag>
    <options>semicolon separated list of options</options>
  </adapter2>

</jaas_config>
```

The following table describes how to use the XML tags and assign appropriate values.

Use these XML tags	To do this
<jaas_config> </jaas_config>	Define a custom JAAS configuration. All JAAS configuration settings must be between these two tags.
<jaasConfiguration> </jaasConfiguration>	Define the name of your custom JAAS LoginModule. The value of this tag determines the tag name to use for the next tag. For example, if you create a custom configuration with the value **CustomConfig**, then you must use the tags **<CustomConfig>** and **</CustomConfig>** to define the list of adapters used.

Use these XML tags	To do this
<CustomConfig> </CustomConfig> *This is a user definable tag*	Define the list of *all* adapters that you want to use for authentication. Use semicolons between entries to specify multiple adapters. If the adapter name you list does not match a registered AdapterPool, then Get-Resources assumes the name is a logical name to be defined in a separate adapter tag. Get-Resources attempts to authenticate users against each adapter you list. The values listed in this tag determine the tags names to use for each adapter. For example, if you create two adapters adapter1 and adapter2, then you must use the tags **<Adapter1>**, **</Adapter1>**, **<Adapter2>**, and **</Adapter2>** to define your adapters.
<adapter1> </adapter1> <adapter2> </adapter2> *These are user definable tags*	Define the JAAS LoginModule settings for each adapter. Each adapter *must* have both **<loginModule>** and **<controlFlag>** tags defined for it.
<loginModule> </loginModule>	Define the fully qualified class name of the JAAS LoginModule. This is a *required* setting for each JAAS LoginModule you use.
<controlFlag> </controlFlag>	Define what information is necessary to authenticate a user. This is a *required* setting for each JAAS LoginModule you use. See *JAAS LoginModule control flags* on page 118 for a description of available options.
<options> </options>	Define the list of authentication options. Use semicolons between entries to specify multiple options. This is an *optional* setting for each JAAS LoginModule you use. See *JAAS configuration options* on page 120 for a description of available options.

JAAS LoginModule control flags

The following table lists the possible settings for the <controlFlag> tag. A JAAS LoginModule can have one of four behaviors:

Control flag	Authentication behavior
REQUIRED	If the user cannot be authenticated against the adapter, the login fails. Whether it succeeds or fails, authentication continues to the next LoginModule in the list.
REQUISITE	If the user cannot be authenticated against the adapter, the login fails. If it succeeds, authentication continues to the next LoginModule in the list.
SUFFICIENT	Authentication can proceed even if this LoginModule fails. It it succeeds, authentication does not continue to the next LoginModule in the list. If it fails, authentication continues to the next LoginModule in the list.
OPTIONAL	Authentication can proceed even if this LoginModule fails. Whether it succeeds or fails, authentication continues to the next LoginModule in the list.

Note: ControlFlag settings are case insensitive.

The overall authentication succeeds only if all Required and Requisite LoginModules succeed. If a Sufficient LoginModule is configured and succeeds, then only the Required and Requisite LoginModules prior to that Sufficient LoginModule need to have succeeded for the overall authentication to succeed. If no Required or Requisite LoginModules are configured for an application, then at least one Sufficient or Optional LoginModule must succeed.

By default, the controlFlag setting of all Get-Resources Web applications LoginModules is Optional. For most enterprises, this is the desired configuration.

The following table shows some sample scenarios and how the login process works.

Module Name	Status	Scenario 1	Scenario 2	Scenario 3
LoginModule1	required	pass	pass	fail
LoginModule2	sufficient	fail	fail	fail

Module Name	Status	Scenario 1	Scenario 2	Scenario 3
LoginModule3	requisite	pass	pass	pass
LoginModule4	optional	pass	fail	fail
Final Authentication		pass	pass	fail

In Scenario 1, authentication succeeds even though LoginModule2 fails. This is because the Required loginModule takes precedence over the sufficient loginModule.

In Scenario 2, authentication succeeds because the loginModules that failed are only Sufficient and Optional.

Scenario 3 authentication fails because a loginModule with a status of Required failed.

JAAS configuration options

The following tables list the possible settings for the <options> tag.

Standard JAAS Options

The following table lists the standard JAAS options available for all adapters.

Option	Use	Description
debug=true	optional	Instructs a LoginModule to output debugging information. The OAALoginModule logs debugging information to stdout and not to archway.log.
tryFirstPass=true	optional	The first LoginModule in the list saves the password entered and this password is used by subsequent LoginModules. If authentication fails, the LoginModules prompt for a new password and repeats the authentication process.
useFirstPass=true	optional	The first LoginModule in the list saves the password entered and this password is used by subsequent LoginModules. If authentication fails, LoginModules do not prompt for a new password.
storePass=true	optional	Stores the password for the user being authenticated.
clearPass=true	optional	Clears the password for the user being authenticated.

Peregrine JndiLoginModule options

The following table lists the options available to custom JAAS LoginModules using the Peregrine JndiLoginModule.

Option	Use	Description
user.provider.url	required	Use this option to provide the URL to the starting point in your directory service where you want to search for users. For example, ldap://server/dc=peregrine,dc=com **Note:** This option corresponds to the Java constant Context.PROVIDER_URL.
security.principal	optional	Use this option to specify which directory service user you want to use to authenticate non-anonymous queries of your directory service. Use the DN of the directory service user. For example, uid=user,dc=peregrine,dc=com **Tip:** To prevent user passwords from being visible to users, you should only set this option if you are using a directory server such as IPlanet where user passwords are SHA hashed by default. **Note:** This option corresponds to the Java constant Context.SECURITY_PRINCIPAL.
security.credentials	optional	Use this option to define the password for the security.principal user. This option should only be used in conjunction with the security.principal option. **Important:** If you are using a simple security authentication protocol, then this password may be passed as plain text. **Tip:** To safeguard this password, either enable SSL (set the security.protocol=ssl option) or use an security.authentication that protects passwords. **Note:** This option corresponds to the Java constant Context.SECURITY_CREDENTIALS.

Option	Use	Description
security.protocol	optional	Use this option to enable or disable an SSL connection between the JndiLoginModule and your directory server. This option has two possible values: ■ simple (Default setting) ■ ssl **Note:** This option corresponds to the Java constant Context.SECURITY_PROTOCOL
security. authentication	optional	Use this option to enable or disable anonymous binding to your directory service. Typically, this option has one of two values: ■ none (Default setting) ■ simple **Note:** If you do not specify a value for security.principal then security.authentication defaults to a value of none. Likewise, if you set security.authentication to simple but security.credentials is omitted or has zero length, then security.authentication resets to none. **Note:** This option corresponds to the Java constant Context.SECURITY_AUTHENTICATION.
user.search.scope	optional	Use this option to specify the number of levels to descend when searching for the user being authenticated by user.provider.url. This value must be an integer. The default value is 1. **Note:** This option corresponds to the Java constant SearchControls.ONELEVEL_SCOPE.
group.provider.url	optional	Use this option to provide the URL to the starting point in your directory service where you want to search for groups. For example, ldap://server/dc=peregrine,dc=com **Note:** This option corresponds to the Java constant Context.PROVIDER_URL.

Option	Use	Description
group.search.scope	optional	Use this option to specify the number of levels to descend when searching for a group. This option should only be used with group.provider.url. This value must be an integer. The default value is 1. **Note:** This option corresponds to the Java constant SearchControls.ONELEVEL_SCOPE.
group.search.objectClass	optional	Use this option to specify the name of the LDAP group objectClass. Valid values are: ■ groupOfNames (Default value) ■ groupOfUniqueNames. ■ groupOfUrls **Note:** Either groupOfNames or groupOfUniqueNames can be used to define static groups in LDAP, but they may not be used together. If you choose the groupOfUrls option, then you are configuring dynamic groups. No additional configuration settings are required to recognize dynamic groups.
storeIdentity=true	optional	Use this option to store a reference to the User being authenticated.
clearIdentity=true	optional	Use this option to clear a reference to the User being authenticated.

Example: Defining an LDAP custom configuration

The following XML code illustrates you could define a loginModule to authenticate users against an LDAP directory service.

```
<jaas_config>

  <jaasConfiguration>myconfig</jaasConfiguration>
    <myConfig>ldap</myConfig>

  <ldap>
    <loginModule>
    com.sun.security.auth.module.JndiLoginModule
    </loginModule>
    <controlFlag>REQUIRED</controlFlag>
    <options>
    user.provider.url="ldap://server/dc=peregrine,dc=com";
    group.provider.url="ldap://server/dc=peregrine,dc=com"
    </options>
  </ldap>

</jaas_config>
```

Standard Sun Microsystems JAAS configuration

The standard JAAS configuration authenticates users against the Sun Microsytems JAAS configuration loginModule. To enable the standard JAAS configuration, you must edit the local.xml file and add the following lines:

```
<jaas_config>
   <useStandardJAASConfiguration>true</useStandardJAASConfiguration>
</jaas_config>
```

If you choose to use the standard JAAS configuration, then you must also do one of the following two things:

- Specify the appropriate JAAS command line options when the container is started

–or–

- Configure the java.security file in $JAVA_HOME/jre/lib/security for JAAS.

Command line options

The command line properties required for use of the standard file-based configuration are as follows:

```
java -classpath <list of jars> \
  -Djava.security.manager \
  -Djava.security.policy==java2.policy \
  -Djava.security.auth.policy==jaas.policy \
  -Djava.security.auth.login.config==jaas.config \
  <MyMainClass>
```

For <*list of jars*>, enter the list of jars used by your JAAS-enabled Java application.

For <*MyMainClass*>, enter the fully qualified class name of the Java main program class.

Windows NT Challenge/Response

Windows NT Challenge/Response is one of the ways Windows NT facilitates the authentication of users on a Web server. The process consists of a secure handshake between Internet Explorer (IE) and the Internet Information Server (IIS) Web server. The handshake lets the Web server know exactly who the user is, based on how they logged in to their workstation. This allows the Web server to restrict access to files or applications based on who the user is. Applications running on the Web server can use this information to identify users without requiring them to log in.

The Get-Resources uses Windows NT Challenge/Response as follows:

- The user logs in to a Windows XP/2000/NT workstation.
- The user starts the IE browser and navigates to the login.asp page.
- IE automatically sends user authentication information to IIS. The user's password is not transferred, but the Windows NT Challenge/Response handshake between IE and IIS is enough for the server to recognize the user.
- The Web application login automatically detects the user by using the Windows NT Challenge/Response/IIS server data.
- The user is logged in without requiring that a name and password be entered.

During this process, Archway authenticates and impersonates the NT user with each of its adapters.

The following circumstances are exceptions to the normal Windows NT Challenge/Response login process:

- The Windows NT user has already registered with an Archway adapter. When this occurs, the web application asks the user to register and enter profile information. The application then lets the user log in and stores this information for future login attempts.
- The Windows NT user name is already registered as an Administrator in the back-end system. When this occurs, the web application does not proceed with automatic login. The user is presented with another login screen and is asked to verify their password. This step is an added security measure to prevent a user from accidentally logging in with administrative rights.

Setting up Windows NT Challenge/Response

Setting up Get-Resources to use Windows NT Challenge/Response is a five step process:

Step 1 Set Web server properties for the **login.asp** file. See *Setting Web server properties for the login.asp file* on page 127.

Step 2 Set Web server properties for the **e_login_main_start.asp** file. See *Setting Web server properties for the e_login_main_start.asp file* on page 128.

Step 3 Set Web server properties for the **loginverify.asp** file. See *Setting Web server properties for the loginverify.asp file* on page 129.

Step 4 Define the **LogoutURL** from the Get-Resources administration page. See *Setting up the LogoutURL* on page 129.

Step 5 Set permissions on the Get-Resources presentation folder. See *Setting permissions for the presentation folder* on page 130.

The following procedures illustrate how to setup Windows NT Challenge/Response authentication using Windows NT as an example. If you are using Windows XP or 2000, the overall procedure is the same. However, in Windows XP and 2000, Challenge/Response is called Integrated Windows Authentication, and the IIS Management Console is called Internet Information Services.

Setting Web server properties for the login.asp file

1 Open the IIS Management Console (Start>Programs>Administrative Tools>Internet Services Manager).

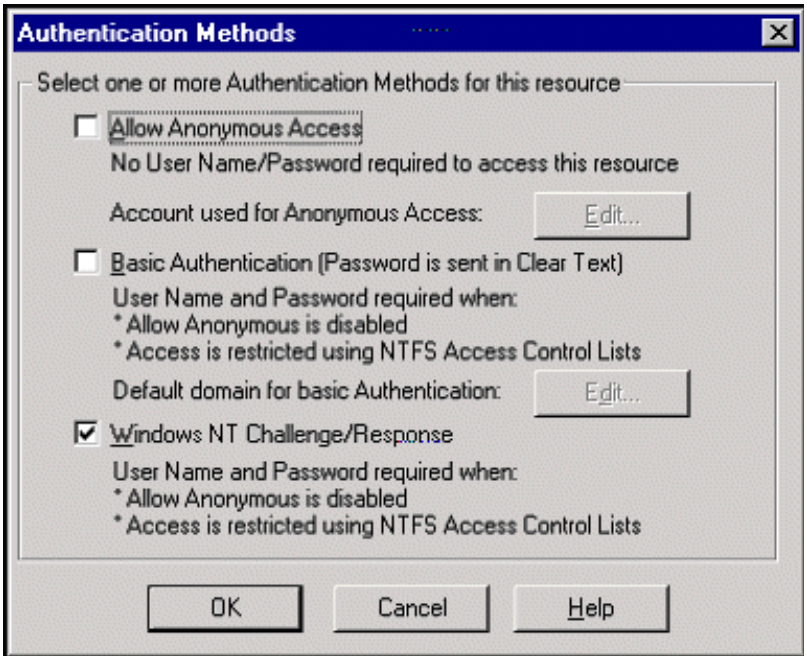
In Windows 2000 and XP, open Internet Information Services (Start>Programs>Administrative Tools>Internet Information Services)

2 Click on the oaa virtual directory.

3 Right-click on login.asp and select **Properties**.

4 Select the **File Security** tab.

5 Click **Edit** in the "Anonymous Access and Authentication Control" section.



6 Check **Windows NT Challenge/Response**. Make sure this is the only option checked.

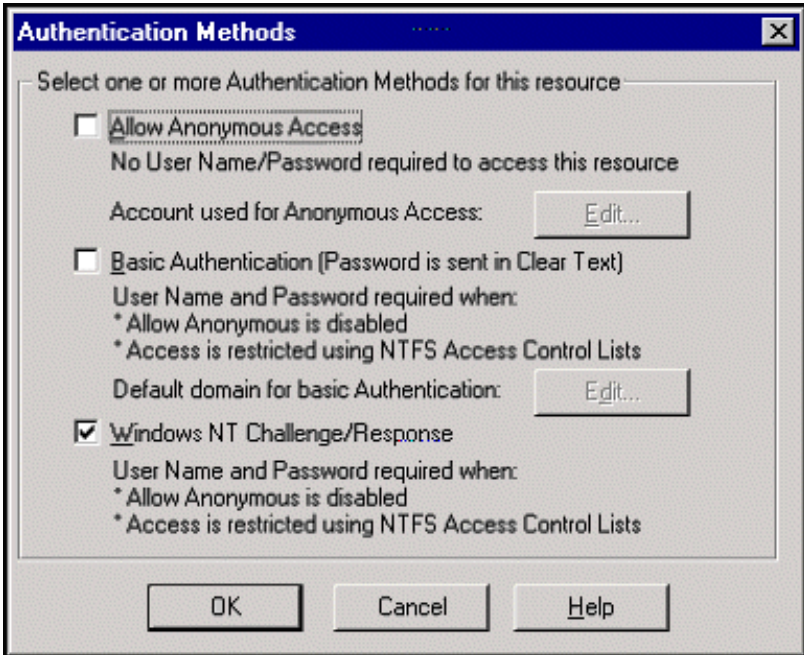
7 Click **OK** on all windows displayed until you return to the Microsoft Management Console.

Setting Web server properties for the e_login_main_start.asp file

1 From the IIS Management Console, right-click on e_login_main_start.asp and select **Properties**.

2 Select the **File Security** tab.

3 Click **Edit** in the "Anonymous Access and Authentication Control" section.



4 Check **Windows NT Challenge/Response**. Make sure this is the only option checked.

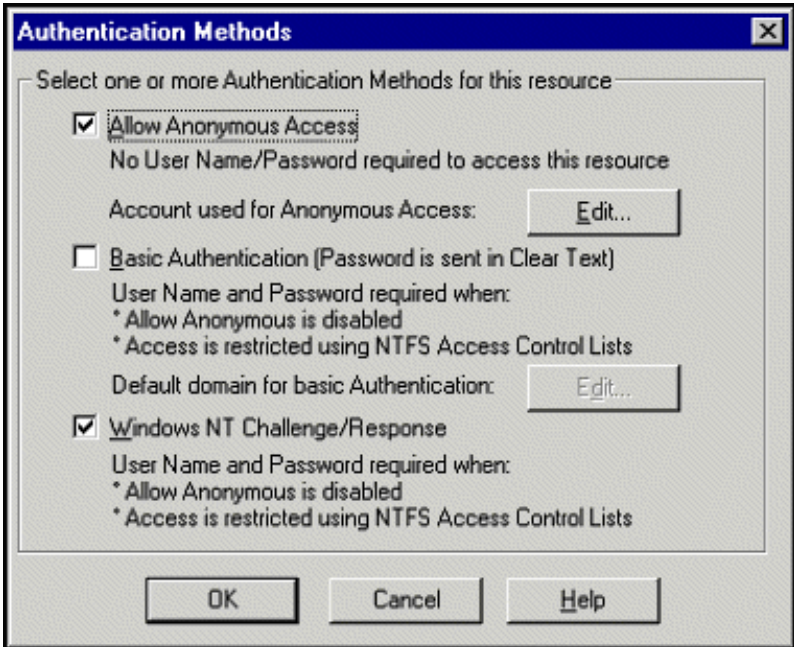
5 Click **OK** on all windows displayed until you return to the Microsoft Management Console.

Setting Web server properties for the loginverify.asp file

1 From the IIS Management Console, right-click on loginverify.asp and select **Properties**.

2 Select the **File Security** tab.

3 Click **Edit** in the "Anonymous Access and Authentication Control" section.



4 Click **OK** on all windows displayed until you return to the Microsoft Management Console.

5 Close the Management Console.

Setting up the LogoutURL

1 Open a Web browser.

2 Enter the following URL: http://webserver/oaa/admin.asp in the browser address field (where webserver is the name of your Web server and oaa is the name of the virtual directory created during installation).

3 Login using the administrator name and password.

4 Click **Settings**.

5 From the **Common** tab, set the **LogoutURL** setting to the URL you want users to go to if Windows NT Authentication fails or is not possible due to the user's current browser.

Setting permissions for the presentation folder

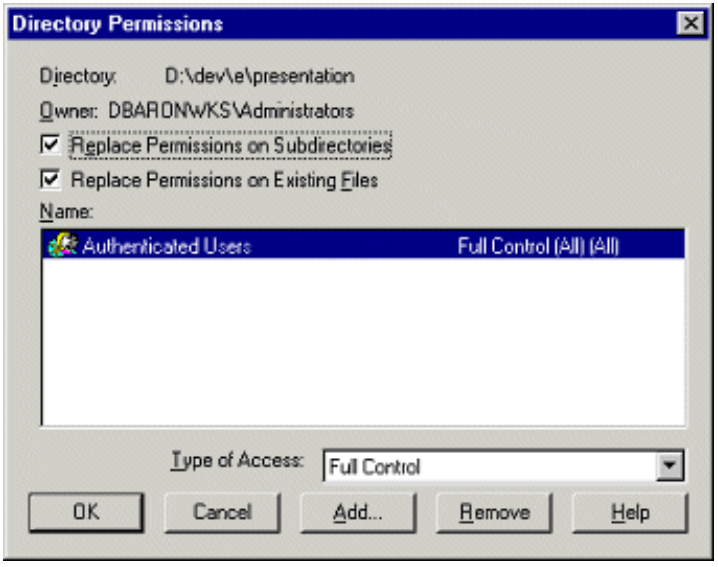
1 Use Windows NT Explorer to navigate to the ...oaa\presentation folder.

2 Right-click on presentation and select **Properties**.

3 On the **Security** tab, click **Permissions**.

Note: If you do not see a Security tab, verify that your Web application is installed on an NTFS partition.

4 Click **Add** to change the user groups that have permission to access the folder. Change the permission to a named authenticated group. For example, you could change permissions to all "Authenticated Users".



5 If the user group called "Everyone" has permissions, highlight the entry, and then click **Remove** so that only the group you selected in the previous step can access the ProductCoreAbbreviated.

6 Click **OK**. Close all remaining windows.

Testing the settings

Log in to your Peregrine Web application to make sure the access permissions are set correctly. The Windows NT Challenge/Response settings are activated when you log in through a special login page named login.asp. Accessing your applications through the standard login.jsp page results in the users needing to log on as usual.

To test the settings:

1. Open a Web browser.
2. Enter the following URL: http://webserver/oaa/login.asp in the browser address field (where webserver is the name of your Web server and oaa is the name of the virtual directory created during installation).
3. Verify that access to Get-Resources is what you expected based on the settings you chose for the login.asp and loginverify.asp files.

Creating an alternate login page

If you do not want to use the default Peregrine OAA login page, you can create your own login page that authenticates users and redirects them to the proper start page. Creating an alternate login page requires two basic steps:

Step 1 Create a login Web page with the necessary authentication parameters. See the following section, *Creating a login Web page*.

Step 2 Edit the archway.xml to specify the HTTP authentication method you want to use. See *Specifying an alternate authentication method* on page 133.

Creating a login Web page

Your custom login web page can be any HTML form that prompts for the following required parameters:

- Username
- Password

In addition, you can include optional login parameters such as:

- Display Language and Locale
- Time format
- Theme

A sample HTML login form, login_sample.html is in the OAA deployment folder of your application server:

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

Customize this sample HTML form using the following guidelines:

- Whatever custom login file you create becomes part of your login URL. For example, if you create a custom page called my_login.htm, then the login URL is http://<server>:<port>/oaa/my_login.htm
- You must specify the basicauth servlet in the form action. For example, action="http://<server>:<port>/oaa/servlet/basicauth"
- Users who have the *getit.portal* capability word see the e_portal_home_start.jsp home page if successfully authenticated
- Users who do *not* have the *getit.portal* capability word see the e_home_main_start.jsp home page if successfully authenticated
- Users who fail to be successfully authenticated see the page specified in the _failURL value
- The basicauth servlet does not encrypt usernames and passwords during login. You must enable HTTPS if you are concerned about password security on your intranet.
- If no URL is specified in the form action, then authenticated users are redirected to http://<server>:<port>/oaa/login.jsp where they see either e_portal_home_start.jsp or e_home_main_start.jsp depending on their login capability words.
- There are no specific Administration page settings needed to set up a custom login page. You must define all login parameters in your custom login page.
- If you want to display a specific OAA page at login, you can specify the page in the form action URL. For example, the value action="http://<server>:<port>/oaa/servlet/basicauth/e_home_main_start.jsp" displays the non-portal version of the Peregrine OAA home page.

- The following login parameters are available:

Login parameters	Description
loginuser	This is a required login parameter specifying the user name. You must specify a form input for this parameter.
loginpass	This is a required login parameter specifying the login password. You must specify a form input for this parameter.
_locale	This is an optional login parameter specifying the user's locale and regional display settings.
_timezone	This is an optional login parameter specifying the user's timezone.
_theme	This is an optional login parameter specifying which theme should be displayed in the Peregrine OAA Portal

Specifying an alternate authentication method

By default, Peregrine OAA uses HTTP basic authentication provided by the HttpBasicAuthenticationManager class. If you create a custom login page, you need to specify the alternate authentication method in the archway.xml file.

To specify an alternate HTTP authentication method:

1. Stop your application server.
2. Using a text editor, open the archway.xml file located at:

   <application server>\webapps\oaa\WEB-INF\default.
3. Edit the line containing:

   ```
   <httpauthclass ...>HttpBasicAuthenticationManager</httpauthclass>
   ```
4. Change the value HttpBasicAuthenticationManager to HttpAlternateAuthenticationManager.
5. Save the file.
6. Restart your application server.

Warning: Changing the HTTP authentication setting to the Alternate Authentication Manager exposes queries (including login names and passwords) in the URL. If you want to protect URL queries, then you must restrict access to this information through your Web server.

Index

A

Activity menu 53
adapter transactions, viewing 81
Admin module
  changing Settings 78
  Control Panel 74
  displaying message queues 79
  generating web archive files 81
  message queues 79
  script status 79
  Server Log 76
  Settings page 76
  showing queue status 80
  verifying script status 79
  viewing adapter transactions 81
Archway architecture
  building blocks 15
  clients 17
  diagram 16
  document manager 21
  executing queries against a system 21
  query string 19
  requests 19
  XML 17
AssetCenter 62
  calculated field 92
  catalog 31
  Certification field 91
  workflows 31
authentication
  overriding the login script 131
  user registration 107
  users 115
Automatic PO Generation workflow, disabling 90

B

book
  audience 7
  organization 10

C

calculated field 92
capability words 63
catalog, AssetCenter 31
Certification field in AssetCenter 91
changing passwords 85
changing the Peregrine Portal layout 57
changing themes 58
components
  adding Portal 54
  creating new 53
Control Panel 74
conventions, typographical 9
CSS files, editing 41
customer support 10

D

deploying themes 36
document manager 21
documentation, related 8

F

form details 84
form details, displaying 83
Form Info, displaying 82
form information, displaying 60
framesets, changing 45

G

getit.admin 109
getit.admin user rights 72
Get-Services
    overview 7

H

header graphic, changing 37

I

IBM Websphere portal 81
icons, personalizing 66
Info button 83
ISO character encoding. See character encoding

J

JAAS
    authentication 115
    login modules 116

L

labels, personalizing 67
layers, changing 43
layout, changing
    MSIE 57
    Netscape Navigator 58
LDAP 106
Lightweight Directory Access Protocol 106
local.xml file 72, 77
log, form details 84
logging user sessions 85
login authentication 115
login modules, JAAS 116
login script, overriding 131
login.asp 131

M

message queues 79
message queues, displaying 79
monitoring user sessions 85

O

overriding the login script 131

P

parameters
    ServiceCenter securepassword 110
parameters, defining 78
password security 110
password, changing 85
passwords
    protecting 106
Peregrine Portal
    adding components 54
    personalizing 54
Peregrine Portal, tailoring 35
Peregrine Systems customer support 10
personalization
    adding fields 66
    arranging field order 67
    interface description 63
    requirements 62
    settings 62
    user rights 62
personalizing
    adding a new section to the field layout 67
    changing field layout 67
    forms 63–70
    icons 66
    labels 67
    portal 54–59
personalizing the Peregrine Portal 54
PO generation, disabling workflow 90
Portal Components, creating new 53
preXSL, form details 84

Q

query string 19
queue status, displaying 80

R

registering new users 107
related documentation 8

resetting the server 74

S

script input, form details 84
script output, form details 84
script status 79
script status, verifying 79
Secure Sockets Layer 106
securepassword parameter 110
security
    alternate login authentication 131
    capability words 63
    user authentication 115
    user registration 107
    Windows NT Challenge/Response 125
self-registration 84
server log 76
ServiceCenter 62
Settings page 78
SSL 106
string files
    translating 47, 49

T

tailoring themes 35
    changing framesets 45
    changing layers 43
    changing the header graphic 37
    deploying themes 36
    editing CSS files 41
    modifying style sheets 39
technical support 10
terminology 8
themes
    deploying 36
    tailoring 35
themes, changing 58
themes, modifying 39
translating tailored modules 47
typographical conventions 9

U

user registration 84, 107
user rights 62
    getit.admin 72
user session 84
user sessions, logging 85
user.log file 86

W

web archive (war) files 81
Websphere portal 81
Windows NT Challenge/Response
    configuring 125
    security 106
workflow, disabling 90
workflows
    Automatic PO Generation 33
    Bundle Ordering 31
    Request Approval 32
    Request Status 34
    Routing Request 32

Peregrine
SYSTEMS