



Developer's Guide

Enterprise Portal

6.0

DOCUMENT ID: DC32752-01-0600-02

LAST REVISED: March 2004

Copyright © 2000-2004 by Sybase, Inc. All rights reserved.

This publication pertains to Sybase software and to any subsequent release until otherwise indicated in new editions or technical notes. Information in this document is subject to change without notice. The software described herein is furnished under a license agreement, and it may be used or copied only in accordance with the terms of that agreement.

To order additional documents, U.S. and Canadian customers should call Customer Fulfillment at (800) 685-8225, fax (617) 229-9845.

Customers in other countries with a U.S. license agreement may contact Customer Fulfillment via the above fax number. All other international customers should contact their Sybase subsidiary or local distributor. Upgrades are provided only at regularly scheduled software release dates. No part of this publication may be reproduced, transmitted, or translated in any form or by any means, electronic, mechanical, manual, optical, or otherwise, without the prior written permission of Sybase, Inc.

Sybase, the Sybase logo, AccelaTrade, ADA Workbench, Adaptable Windowing Environment, Adaptive Component Architecture, Adaptive Server, Adaptive Server Anywhere, Adaptive Server Enterprise, Adaptive Server Enterprise Monitor, Adaptive Server Enterprise Replication, Adaptive Server Everywhere, Adaptive Server IQ, Adaptive Warehouse, Anywhere Studio, Application Manager, AppModeler, APT Workbench, APT-Build, APT-Edit, APT-Execute, APT-Translator, APT-Library, AvantGo, AvantGo Application Alerts, AvantGo Mobile Delivery, AvantGo Mobile Document Viewer, AvantGo Mobile Inspection, AvantGo Mobile Marketing Channel, AvantGo Mobile Pharma, AvantGo Mobile Sales, AvantGo Pylon, AvantGo Pylon Application Server, AvantGo Pylon Conduit, AvantGo Pylon PIM Server, AvantGo Pylon Pro, Backup Server, BizTracker, ClearConnect, Client-Library, Client Services, Convoy/DM, Copernicus, Data Pipeline, Data Workbench, DataArchitect, Database Analyzer, DataExpress, DataServer, DataWindow, DB-Library, dbQueue, Developers Workbench, Direct Connect Anywhere, DirectConnect, Distribution Director, e-ADK, E-Anywhere, e-Biz Impact, e-Biz Integrator, E-Whatever, EC Gateway, ECMAP, ECRTP, eFulfillment Accelerator, Embedded SQL, EMS, Enterprise Application Studio, Enterprise Client/Server, Enterprise Connect, Enterprise Data Studio, Enterprise Manager, Enterprise SQL Server Manager, Enterprise Work Architecture, Enterprise Work Designer, Enterprise Work Modeler, eProcurement Accelerator, EWA, Financial Fusion, Financial Fusion Server, Gateway Manager, GlobalFIX, iAnywhere, ImpactNow, Industry Warehouse Studio, InfoMaker, Information Anywhere, InformationConnect, InternetBuilder, iScript, Jaguar CTS, jConnect for JDBC, Mail Anywhere Studio, MainframeConnect, Maintenance Express, Manage Anywhere Studio, M-Business Channel, M-Business Network, M-Business Server, MDI Access Server, MDI Database Gateway, media.splash, MetaWorks, My AvantGo, My AvantGo Media Channel, My AvantGo Mobile Marketing, MySupport, Net-Gateway, Net-Library, New Era of Networks, ObjectConnect, ObjectCycle, OmniConnect, OmniSQL Access Module, OmniSQL Toolkit, Open Biz, Open Client, Open ClientConnect, Open Client/Server, Open Client/Server Interfaces, Open Gateway, Open Server, Open ServerConnect, Open Solutions, Optima++, Orchestration Studio, PB-Gen, PC APT Execute, PC DB-Net, PC Net Library, PocketBuilder, Pocket PowerBuilder, Power++, power.stop, PowerAMC, PowerBuilder, PowerBuilder Foundation Class Library, PowerDesigner, PowerDimensions, PowerDynamo, PowerJ, PowerScript, PowerSite, PowerSocket, Powersoft, PowerStage, PowerStudio, PowerTips, Powersoft Portfolio, Powersoft Professional, PowerWare Desktop, PowerWare Enterprise, ProcessAnalyst, Rapport, RepConnector, Replication Agent, Replication Driver, Replication Server, Replication Server Manager, Replication Toolkit, Report-Execute, Report Workbench, Resource Manager, RW-DisplayLib, RW-Library, S-Designer, SDF, Secure SQL Server, Secure SQL Toolset, Security Guardian, SKILS, smart.partners, smart.parts, smart.script, SQL Advantage, SQL Anywhere, SQL Anywhere Studio, SQL Code Checker, SQL Debug, SQL Edit, SQL Edit/TPU, SQL Everywhere, SQL Modeler, SQL Remote, SQL Server, SQL Server Manager, SQL SMART, SQL Toolset, SQL Server/CFT, SQL Server/DBM, SQL Server SNMP SubAgent, SQL Station, SQLJ, STEP, SupportNow, s.W.I.F.T. Message Format Libraries, Sybase Central, Sybase Client/Server Interfaces, Sybase Financial Server, Sybase Gateways, Sybase MPP, Sybase SQL Desktop, Sybase SQL Lifecycle, Sybase SQL Workgroup, Sybase User Workbench, SybaseWare, Syber Financial, SyberAssist, SyBooks, System 10, System 11, System XI (logo), SystemTools, Tabular Data Stream, TotalFix, TradeForce, Transact-SQL, Translation Toolkit, UltraLite, UltraLite.NET, UNIBOM, Unilib, Uninull, Unisep, Unistring, URK Runtime Kit for UniCode, VisualWriter, VQL, WarehouseArchitect, Warehouse Control Center, Warehouse Studio, Warehouse WORKS, Watcom, Watcom SQL, Watcom SQL Server, Web Deployment Kit, Web.PB, Web.SQL, WebSights, WebViewer, WorkGroup SQL Server, XA-Library, XA-Server and XP Server are trademarks of Sybase, Inc. 02/04

Unicode and the Unicode Logo are registered trademarks of Unicode, Inc.

All other company and product names used herein may be trademarks or registered trademarks of their respective companies.

Use, duplication, or disclosure by the government is subject to the restrictions set forth in subparagraph (c)(1)(ii) of DFARS 52.227-7013 for the DOD and as set forth in FAR 52.227-19(a)-(d) for civilian agencies.

Sybase, Inc., One Sybase Drive, Dublin, CA 94568.

Contents

About This Book	xiii	
CHAPTER 1	Introduction	1
	What is Enterprise Portal?	1
	Developing enterprise portals	3
	Getting started.....	4
	Using Portal Studio.....	4
	Selecting a Portal Studio left pane menu option	6
CHAPTER 2	Managing Portal Studio Security	11
	Integrating Enterprise Security and Portal Studio	11
	Using J2EE role-based security	12
	Setting up role-based security in Portal Studio	15
	Configuring domains	16
	Administering organizations	17
	Using existing single sign-on mechanisms.....	23
	Customizing EP to using an existing single sign-on mechanism .	24
	Generating an single sign-on outbound token.....	25
CHAPTER 3	Auditing Enterprise Portal.....	27
	Introduction	27
	Setting up Enterprise Portal auditing.....	28
	Auditable portal actions	31
	Portal Interface	32
	Portal Studio	32
	Reading audit information	34
	Sample Enterprise Portal audit records.....	34
	Auditing rules	36
	Auditing a distributed installation.....	37
CHAPTER 4	Building Portlets	39

- Introduction 39
 - Portlet Manager..... 40
 - Portlet types 41
 - Portlet Builder..... 41
- Building aggregated portlets 48
- Creating Web elements..... 49
 - Capture formats..... 49
- Creating XML elements..... 52
- Creating database elements 54
- Creating JSP elements 56
 - Using a Web application..... 56
 - Including the Secure Web Proxy in a JSP portlet..... 59
 - Using a JSP template..... 60
 - Using a remote URL..... 61
 - JSP portlet single-sign-on check box 62
- Creating Web service elements 62
 - Web services standards 63
 - Examples..... 65
 - Using Enterprise Portal Web services..... 65
 - Creating Web service portlets 69
 - Creating new UDDI registry entries..... 73
- Creating HTML-based elements 74
- Creating document elements 75
 - Adding new content types 77
- Building secure portlets..... 79
 - Setting up Enterprise Portal to create secure portlets..... 79
 - Creating secure portlets 81
- Saving portlets 82
- Approving elements and portlets..... 84
- Viewing, editing, and deploying portlets..... 85
 - Editing portlets..... 85
- Finding and replacing portlets 91
 - Performing a portlet search 91
 - Finding and replacing portlets 92

CHAPTER 5 Advanced Portlet Creation..... 95

- Building messaging and application portlets 95
 - Building click-across messaging portlets..... 96
 - Building application portlets..... 106
 - Shared context 108
- Working with data-capable elements 132
 - Data-capable post-processing options 133
- Creating portlet drop-down lists for input fields 136
- Enabling portlets and pages to support multiple languages..... 137

	Web browsers and UTF-8	139
CHAPTER 6	Charting Portlets	141
	Introduction	141
	Creating portlet charts	142
	Creating a bar graph chart	143
	Creating a pie graph charting portlet	145
	Creating a line graph charting portlet	146
	Turning off chart caching.....	147
CHAPTER 7	Personalizing Portlet Content.....	149
	Introduction	149
	Adapter types	150
	Creating new personalization adapter keys	151
	Building portlets with personalization adapters	152
	Using the security proxy adapter.....	152
	Using the user name/password adapter.....	158
	Using the database table adapter	158
	Using the EJB adapter	159
	Entering user-specific values in Portal Studio.....	160
	Importing and exporting personalization keys.....	161
CHAPTER 8	Building Templates	163
	Introduction	163
	Template Manager	163
	Choosing a template type	165
	HTML templates	165
	HELP templates	167
	JSP templates	168
	XSL templates	169
	Creating and editing templates	170
	Changing the default template	172
	Customizing user XML element templates.....	173
	Enabling different templates for the same element.....	175
CHAPTER 9	Building Catalogs	179
	Introduction	179
	Catalog Manager.....	180
	Catalog Builder.....	181
	Creating and editing catalogs.....	182
	Updating Portal Interface with changes	186

CHAPTER 10	Building Pages	187
	Introduction	187
	Page Manager	187
	Page Builder	189
	Choosing the page type	190
	Selecting a page layout	191
	Creating and editing pages	194
	Updating Portal Interface with changes	198
CHAPTER 11	Building Page Groups	199
	Introduction	199
	Page Group Manager	200
	Page Group Builder	201
	Choosing a navigation style	203
	Sample navstyles.xml file	203
	Sample *View.xml file	204
	Page groups, page numbers, and <IFRAME>	205
	Creating and editing page groups	206
	Updating Portal Interface with changes	210
CHAPTER 12	Using Enterprise Portal Agents.....	211
	Introduction	211
	Using agents	213
	Changing an agent's status and running an agent once	213
	Schedule tab	214
	Rules tab	215
	Actions tab.....	221
	Error Management tab	224
	Saving a new agent.....	225
	Handling "permission denied" error messages.....	226
	Viewing or changing agent properties	226
	Using adapters	227
	Creating a file-system adapter	227
	Creating an e-mail adapter	228
	Creating a database adapter.....	230
	Editing an adapter or viewing the log	231
	Using agent servers	231
	Editing an agent server	232
	Viewing transaction logs	232
	Limiting the log size.....	233
CHAPTER 13	Importing, Exporting, and Deploying Portal Objects	235

	Introduction	235
	Importing files into Portal Studio	236
	Exporting objects from Portal Studio.....	238
	Deploying portal objects to a remote server.....	239
	Rules and restrictions.....	242
	Importing and exporting page groups.....	243
	Exporting applications	244
	Exporting agents	244
	Roles and security integration	244
CHAPTER 14	Creating Co-brands	247
	Introduction	247
	Creating co-brands.....	250
	Adding a resource	250
	Editing the cobrands.xml file	251
	Copying resource-specific files.....	252
	Editing resource-specific files.....	253
	Implementing a co-brand	253
CHAPTER 15	Changing the Portal's Look And Feel	255
	Introduction	255
	Editing styles.xml	257
	Editing css.css	258
	Editing messages.xml	258
	Editing global.properties.xml	259
APPENDIX A	Configuring Global Properties.....	261
	Introduction	261
	Viewing global.properties.xml contents	262
	Global property group.....	263
	Audit property group.....	272
	UWP property group.....	272
	Product configuration property group	274
	Docservers property group.....	275
	Portal property group.....	276
	Agents property group.....	277
	Nav property group.....	278
	Capture logging properties	280
	Database property group.....	280
	Database property-specific parameters	282
	Updating global properties	283
	Limiting access to global.properties.xml	286

APPENDIX B	Using the Search Services	289
	Introduction	289
	Caching	290
	Calling methods.....	290
	Search services Enterprise JavaBean	291
	com.sybase.ep.search.PortalSearchServices.	
	SearchHome.....	292
	create	292
	com.sybase.ep.search.PortalSearchServices.Search	293
	remove	295
	setTracing.....	296
	doQuery.....	296
	doNameSearchQuery.....	296
	suggestMore.....	297
	getNextRecord	298
	getDocContent	298
	getDocId	298
	getDocSummary.....	298
	getDocTitle	299
	getDocUrl	299
	getDocWeight.....	299
	getTotalDocs	299
	getDatabaseNames.....	300
	getDocField	300
	getLongDocField	300
	addRoleForChannel	300
	addRoleForUserAgent.....	301
	cleanupUserMemory	301
	copyUserAgent.....	301
	createAgent	302
	createChannel	302
	doQueryWithAgent	303
	editAgent	303
	getCountOfChannels.....	304
	getCountOfRolesForChannel	304
	getCountOfRolesForUserAgent	304
	getCountOfSubChannels	305
	getCountOfUserAgents	305
	getCurrentSubChannelAge	305
	getCurrentSubChannelMaxDocs.....	305
	getCurrentSubChannelRelevance.....	306
	getCurrentSubChannelTrainingText.....	306
	getCurrentUserAgentAge	306
	getCurrentUserAgentMaxDocs	306

getCurrentUserAgentRelevance	307
getCurrentUserAgentTrainingText	307
getNextChannelName	307
getNextRoleForChannel	307
getNextRoleForCurrentUserAgent	308
getNextSubChannelName	308
getNextUserAgentName	308
removeChannel	308
removeRoleFromChannel	309
removeRoleFromUserAgent	309
removeSubChannel	309
removeUserAgent	310
renameAgent	310
retrainSubChannelByResults	310
retrainUserAgentByResults	311
setupUserMemory	311
Search and SearchHome examples	311
Search example	311
SearchHome example	313
User agent and channel examples	313
Advanced search by proper name	314
Advanced search APIs	314
addDatabaseForUserAgent	315
removeDatabaseFromUserAgent	315
addDatabaseForChannel	315
removeDatabaseFromChannel	316
getCountOfDatabasesForUserAgent	316
getNextDatabaseForUserAgent	316
getCountOfDatabasesForChannel	317
getNextDatabaseForChannel	317
doAdvancedQuery	317
getCountOfDatabases	318
getNextDatabaseForDre	319
existsDatabase	319

APPENDIX C

Setting Up Automatic Indexing and Search Functionality	321
Overview	321
Performing channel, subchannel, and agent searches	323
Creating channels and subchannels	323
Editing channels and subchannels	326
Performing concept-based searches	326
Dynamic Reasoning Engine	327
Search options	328
Passing security information to the DRE	328

Sorting queries in the DRE	332
DRE administration tool	333
Managing the connection cache	334
Automatic indexing	334
Starting and stopping the automatic indexing utility	335
Restarting the indexing utility	336
Configuration file	337
Basic configuration parameters	338
Default parameters	339
Import parameters	341
[job_name] section of configuration file	353

APPENDIX D

Using the Enterprise Portal Samples.....	357
Using the single sign-on sample	357
Creating a secure Web application	358
Creating a deployable WAR file	360
Deploying the WAR file	360
Testing the single sign-on example as a standalone Web application	360
Creating a page with a portlet that executes the JSP	361
Creating proxy authentication information	363
Creating the single sign-on JSP	366
Creating single-sign-on portlets with Web applications.....	368
Creating a secured Web application	369
Deploying the secured Web application	371
Creating a single sign-on JSP portlet	372
Creating single-sign-on portlets	375
Creating a secured Web application	375
Deploying ProxyAuth.war	378
Testing ProxyAuth.jsp as a standalone Web application	378
Creating a catalog page with a portlet that executes ProxyAuth.jsp 378	
Entering portal authentication data.....	381
single-sign-on JSP programming	383
Viewing the ProxyAuth portlet	384
Using the sample search portlets	385
Dynamic Reasoning Engine	386
Search EJB	387
Search portlets	388
Testing search as a standalone Web application	389
Using the Content Explorer sample portlet	390
Users	391
Administrator functions	392
Using the Address Book sample	395

	Building and deploying the Address Book portlet.....	395
	Using the Address Book.....	401
	Application files	403
	Implementation details	405
APPENDIX E	Accessing Portal Interface Content with Pocket PC	409
	Accessing content using Pocket PC 2003	410
	Accessing Portal Interface content using the Pocket PC Emulator	410
APPENDIX F	Troubleshooting.....	413
	ActiveX and portlet creation on UNIX.....	413
	Agent log fills up the database	414
	Auditing	414
	Auto indexer does not work.....	414
	Chart text displays improperly.....	415
	Co-brands error.....	415
	Portal Security Officer password.....	416
	Portal Studio login	416
	Glossary	419
	Index	435

About This Book

Subject	This guide describes how to develop and deploy portlets, portals, and their associated components (pages, page groups, catalogs, and so on).
Audience	This book is for anyone developing portals and portlets for Sybase® Enterprise Portal.
How to use this book	<ul style="list-style-type: none">• Chapter 1, “Introduction,” introduces Enterprise Portal and explains how to get started using Portal Studio.• Chapter 2, “Managing Portal Studio Security,” describes how Enterprise Portal is integrated with Enterprise Security to implement role-based authentication in the portal.• Chapter 3, “Auditing Enterprise Portal,” explains how to set up auditing in Portal Studio for Portal Interface.• Chapter 4, “Building Portlets,” describes how to build and manage a variety of portlets.• Chapter 5, “Advanced Portlet Creation,” has advanced topics for portlet development, including creating messaging and application portlets.• Chapter 6, “Charting Portlets,” explains how to create bar, pie, and line charts from portlet contents.• Chapter 7, “Personalizing Portlet Content,” describes how to create input fields for portlet values and how users can personalize these values to fit the content they want to see.• Chapter 8, “Building Templates,” explains how to build and manage templates. Every portlet is placed on either a default HTML template, or a user-defined HTML template.• Chapter 9, “Building Catalogs,” explains how to build and manage catalogs that contain portlets. Portal Interface has one default display catalog from which users can select and add portlets to their user pages.• Chapter 10, “Building Pages,” describes how to build predefined pages, from which portal users can select.

-
- Chapter 11, “Building Page Groups,” explains how to add pages to page groups.
 - Chapter 12, “Using Enterprise Portal Agents,” describes how use agents to schedule or externally trigger automatic processing of a portlet’s content.
 - Chapter 13, “Importing, Exporting, and Deploying Portal Objects,” explains how to copy Portal Studio objects from one Portal Studio installation to another and how to update Portal Interface with updated Portal Studio pages, catalogs, and portlets.
 - Chapter 14, “Creating Co-brands,” explains how to create multiple portals that can be accessed from the same Web browser and are associated with a single Enterprise Portal installation. This chapter also explains how to customize colors, images, and text for each portal.
 - Chapter 15, “Changing the Portal’s Look And Feel,” describes the portal’s primary configuration files and explains how to alter the look and feel of your portal.
 - Appendix A, “Configuring Global Properties,” lists the properties in the portal master configuration file—*global.properties.xml*.
 - Appendix B, “Using the Search Services,” describes how to integrate the search services available with Enterprise Portal into client applications and your enterprise’s portal.
 - Appendix C, “Setting Up Automatic Indexing and Search Functionality,” explains how to set up automatic indexing and search functionality using the portal search engine.
 - Appendix D, “Using the Enterprise Portal Samples,” describes how to use the samples included with Enterprise Portal.
 - Appendix E, “Accessing Portal Interface Content with Pocket PC,” explains how to set up a Pocket PC to view Enterprise Portal content.
 - Appendix F, “Troubleshooting,” provides possible solutions to common portal problems.

Related documents

Enterprise Portal printed documentation The following Enterprise Portal documents are available on the *Getting Started with Enterprise Portal* CD:

- The Enterprise Portal installation guide for your platform explains how to install the Enterprise Portal software.

- The Enterprise Portal release bulletin for your platform contains last-minute information not documented elsewhere. You can also access the release bulletin from the Enterprise Portal installer.

Enterprise Portal online documentation The following Enterprise Portal documents are available in PDF and DynaText format on the *Enterprise Portal 6.0 Technical Library* CD:

- The *Enterprise Portal Developer's Guide* (this book) includes development topics for Enterprise Portal components, Portal Interface portlets, and Java Template Framework pages.
- The *Portal Interface User's Guide* describes the Portal Interface user interface and how to use Portal Interface to build and manage your enterprise's portal.
- The *Enterprise Security Administration Guide* explains Enterprise Portal security architecture and describes how to set up both basic and advanced security models in Enterprise Portal and Portal Interface.

Note The *Enterprise Portal Administration Guide* is not being released for EP 6.0. Many functional changes have occurred in EP 6.0, and much of the *Administration Guide's* content was either obsolete or has been incorporated into other books in the EP 6.0 documentation collection. An EP-specific administration guide may be released at a future date with new content (for example, performance and tuning information).

EAServer documentation EAServer is one of the applications servers into which you can install Enterprise Security, and it is included with Enterprise Portal. These EAServer documents are available in HTML format in your EAServer software installation, and in PDF and DynaText format on the *EAServer Technical Library* CD.

- *What's New in EAServer* summarizes new functionality in the latest version of EAServer.
- The *EAServer Feature Guide* explains application server concepts and architecture, such as components, transactions, and Web applications. This book also explains how to use the optional EAServer products such as Message Bridge for Java™, Web Services Toolkit, Application Integrators, and so on.
- The *EAServer Programmer's Guide* explains how to create, deploy, and configure component-based applications, Web applications, Java servlets, JavaServer Pages, and how to use CORBA and Java APIs.

-
- The *EAServer System Administration Guide* explains how to manage EAServer with the Jaguar Manager plug-in, create new application servers, monitor servers and application components, define connection caches, and so on.
 - The *EAServer Security Administration and Programming Guide* explains how to configure role-based security, configure SSL certificate based-security, implement custom security services for authentication, authorization and role-membership, and so on.
 - The *EAServer Cookbook* contains tutorials and explains how to use the sample applications included with your EAServer software.
 - The *EAServer API Reference Manual* contains reference pages for proprietary EAServer Java classes, ActiveX interfaces, and C routines. This document is available only online.

The *EAServer Installation Guide* for your platform explains how to install the EAServer software; it is available on the *Getting Started* CD.

The *EAServer Troubleshooting Guide* describes problems you may encounter running EAServer and possible solutions; it is available online—see the EAServer Troubleshooting Guide at <http://www.sybase.com/detail?id=1024509>.

jConnect™ for JDBC™ documents Enterprise Portal 6.0 includes the jConnect for JDBC driver to allow JDBC access to Sybase database servers and gateways. The *Programmer's Reference jConnect for JDBC* is included on the *Enterprise Portal Technical Library* CD.

Note See the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.

Other sources of information

Use the *Sybase Getting Started* CD, the *Sybase Technical Library* CD and the Technical Library Product Manuals Web site to learn more about your product:

- The *Getting Started* CD contains release bulletins and installation guides in PDF format, and may also contain other documents or updated information not included on the Technical Library CD. It is included with your software. To read or print documents on the Getting Started CD you need Adobe Acrobat Reader (downloadable at no charge from the Adobe Web site, using a link provided on the CD).
- The *Technical Library* CD contains product manuals and is included with your software. The DynaText reader (included on the Technical Library CD) allows you to access technical information about your product in an easy-to-use format.

Refer to the *Technical Library Installation Guide* in your documentation package for instructions on installing and starting the Technical Library.

- The Technical Library Product Manuals Web site is an HTML version of the Technical Library CD that you can access using a standard Web browser. In addition to product manuals, you will find links to EBFs/Updates, Technical Documents, Case Management, Solved Cases, newsgroups, and the Sybase Developer Network.

To access the Technical Library Product Manuals Web site, go to Product Manuals at <http://www.sybase.com/support/manuals/>.

Sybase certifications on the Web

Technical documentation at the Sybase Web site is updated frequently.

❖ Finding the latest information on product certifications

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Select Products from the navigation bar on the left.
- 3 Select a product name from the product list and click Go.
- 4 Select the Certification Report filter, specify a time frame, and click Go.
- 5 Click a Certification Report title to display the report.

❖ Creating a personalized view of the Sybase Web site (including support pages)

Set up a MySybase profile. MySybase is a free service that allows you to create a personalized view of Sybase Web pages.

- 1 Point your Web browser to Technical Documents at <http://www.sybase.com/support/techdocs/>.
- 2 Click MySybase and create a MySybase profile.

Sybase EBFs and software updates

❖ Finding the latest information on EBFs and software updates

- 1 Point your Web browser to the Sybase Support Page at <http://www.sybase.com/support>.
- 2 Select EBFs/Updates. Enter user name and password information, if prompted (for existing Web accounts) or create a new account (a free service).
- 3 Select a product.

- 4 Specify a time frame and click Go.
- 5 Click the Info icon to display the EBF/Update report, or click the product description to download the software.

Conventions

The formatting conventions used in this manual are:

Formatting example	To indicate
commands and methods	When used in descriptive text, this font indicates keywords such as: <ul style="list-style-type: none"> • Command names • C++ and Java method or class names
<i>variable, package, or component</i>	Italic font indicates: <ul style="list-style-type: none"> • Program variables, such as <i>myCounter</i> • Parts of input text that must be substituted, for example: <pre style="text-align: center;">Server.log</pre> • File names
File Save	Menu names and menu items are displayed in plain text. The vertical bar shows you how to navigate menu selections. For example, File Save indicates “select Save from the File menu.”
package 1	Monospace font indicates: <ul style="list-style-type: none"> • Information that you enter in Jaguar Manager, on a command line, or as program text • Example program fragments • Example output fragments

Variables

The variables used in this manual to represent software installation directories are:

Variable	Represents
<i>SYBASE</i>	The Sybase installation directory.
<i>JAGUAR</i>	The EAServer installation directory
<i>SECURITY</i>	The Enterprise Security installation directory

If you need help

Each Sybase installation that has purchased a support contract has one or more designated people who are authorized to contact Sybase Technical Support. If you cannot resolve a problem using the manuals or online help, please have the designated person contact Sybase Technical Support or the Sybase subsidiary in your area.

Introduction

Enterprise Portal (EP) is a ready-to-deploy, secure, and scalable portal solution that allows you to easily consolidate Web-accessible information sources and integrate existing applications into a personalized view.

Enterprise Portal provides your employees, partners, and customers immediate access to the information and services they need to perform their jobs. EP provides an intuitive user interface for advanced administration, comprehensive security, and support for enterprise application integration.

Enterprise Portal's browser-based components—Portal Studio and Portal Interface—translate the end user, developer, and administrator experience into a reduction in development and deployment time and portal project cost.

This guide provides instructions for the portal and portlet developer along with some system administration topics. See the *Enterprise Portal User's Guide* for end-user information.

Topic	Page
What is Enterprise Portal?	1
Developing enterprise portals	3
Getting started	4

What is Enterprise Portal?

Enterprise Portal is a portal framework that includes:

- Portal Interface – an enterprise portal with which end users interact to access predefined content or to create their own.
- Portal Studio – the application for portal development and administration integrated with Enterprise Security.

Developers create content in Portal Studio and deploy it to portlets in Portal Interface. Administrators use both Portal Studio and Portal Interface to perform portal administrative tasks.

Enterprise Portal is a J2EE Web application that uses Java servlets and JSP technology, which allows EP to be deployed and installed on any J2EE-based application server such as EAServer or Tomcat.

Portal objects, such as portlets, templates, catalogs, and pages are defined in Enterprise Portal with XML documents stored in a database repository. Developers use Portal Studio to create and modify portal objects by manipulating XML documents. Extensible Stylesheet Language Transformations (XSLTs) convert XML documents into the HTML presentation of Portal Studio. Portal Studio uses Java servlets to retrieve and upload the XML documents that define the portal objects. Because EP uses XML to define portal objects, Portal Studio can export a collection of these objects to an XML document, which can then be moved and imported into other EP installations. This simplifies consolidation of portal objects that have been developed separately in different locations.

Portal Interface uses Java servlets and JSP applications to:

- Construct the portal page framework (logos, icons, colors, and so on)
- Request the page and the portlet definitions for the portal page
- Retrieve portlet content from the corresponding sources

Pages and portlets within a Portal Interface user account are controlled by either an administrator or by the end user. Portlets can be arranged into pages that are automatically added to portal user accounts or made available through a catalog of pages. Alternatively, instead of selecting an entire page of portlets, portal users can add individual portlets to their own account from a portlet catalog. Lastly, users can define their own pages and portlets in Portal Interface. Portal Interface and Portal Studio are completely integrated. Changes made to portlets, catalogs, or pages from Portal Studio are updated to Portal Interface.

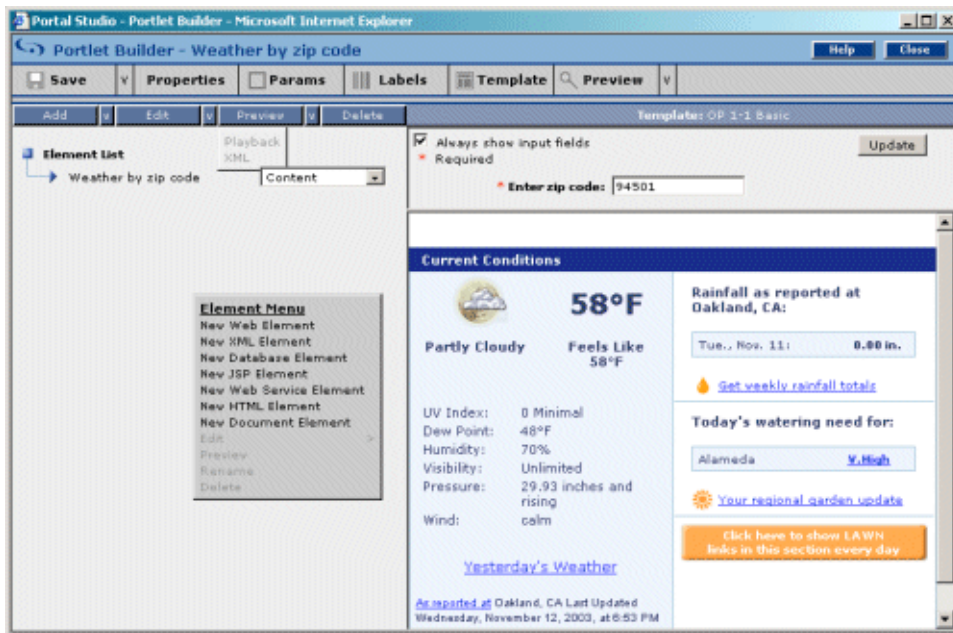
Portal users are assigned J2EE roles that are defined using Enterprise Security. These roles are also assigned to portlets and pages, which ensures that a portal user can select and view only those portlets or pages that are associated with their roles. Enterprise Security also associates login information with specific roles that enable single-sign-on capabilities. With single sign-on, users can view or access a portlet without providing specific authentication credentials. Additionally, you can integrate Enterprise Security with external security systems to synchronize existing users and roles. This allows Enterprise Portal to act as the security hub between different data systems and applications that are delivered through the portal.

Developing enterprise portals

Portal Studio allows developers to build a variety of portlets, templates, pages, and catalogs without having to learn any portlet application interfaces.

Developers can create HTML, JSP, XML, database, and Web content portlets either by using wizards or by using the Portal Studio's Portlet Builder.

Figure 1-1: Portlet Builder



Portal Studio also provides a collaborative environment for developers and administrators by offering versioning, a search filter, a detail view, and roles-based access.

Portal Studio provides developers with automated real-time deployment of individual portlets, XML files, portal applications, and Enterprise Archive (EAR) files. You can deploy to multiple portals, or deploy multiple portals in a single instance.

This guide includes complete instructions on using Portal Studio to develop your enterprise's portal content.

Getting started

This section describes the Portal Studio user interface and how to get started using Portal Studio.

Using Portal Studio

Portal Studio is a Web application, and you must log in using Internet Explorer.

❖ Logging in to the Portal Studio

- 1 Log in to Portal Studio by entering the following URL in your Internet Explorer browser:

```
http://HOSTNAME.PORTALDOMAIN:8080/onepage/index.html
```

For example, if your machine name is “tahiti”, and your portal domain is “enterprise.com,” enter:

```
http://tahiti.enterprise.com:8080/onepage/index.html
```

If you are running the Demo version of Enterprise Portal, enter:

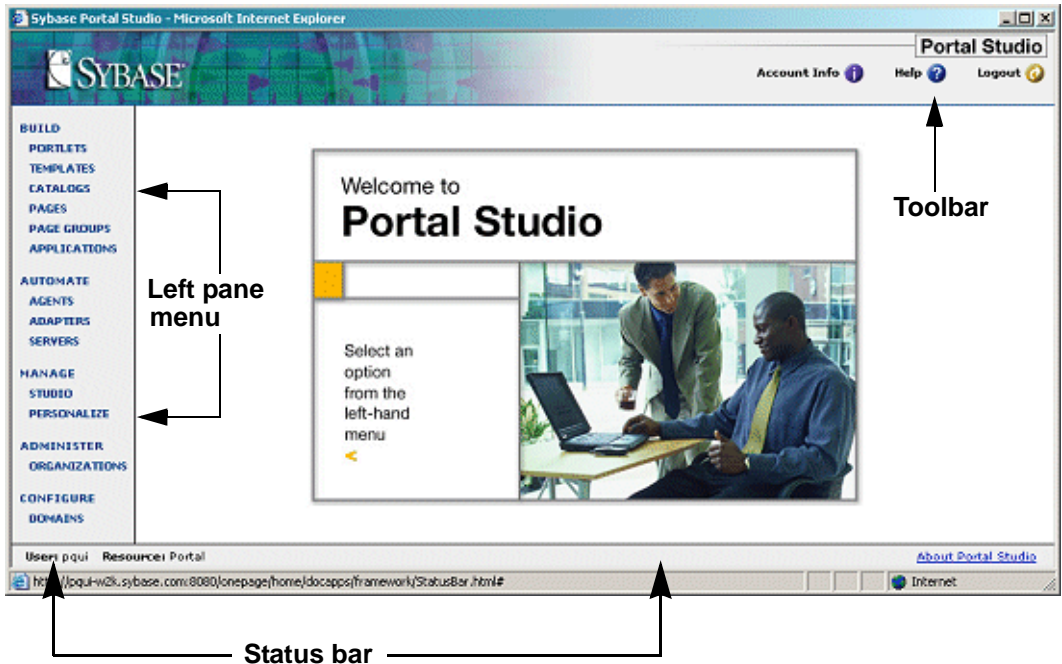
```
http://demo.sybase.com:4040/onepage/index.html
```

Note In a development environment, your port number may be different, and in a production system, the port number may not be necessary at all.

- 2 When the Portal Studio Login window displays, enter the user name and password for a user with portal administrator privileges and click Login.

You see the Portal Studio welcome window.

Figure 1-2: Portal Studio welcome window



The welcome window displays the:

- Left pane menus – select from five menus that allow you to:
 - Build – create, edit, and manage portlets, templates, catalogs, pages, page groups, and application portlets.
 - Automate – create agents to automatically process portlet content, or use adapters to write portlet content to an e-mail, database, or file system.
 - Manage – manage Portal Studio resources and create predefined portlet input fields or drop-down lists, which users can later to personalize with their own values.
 - Administer – access Enterprise Security to create Portal Studio user accounts, resources, and assets. This menu option requires the Portal Security Officer (PSO) role.

- **Configure** – configure the security domain assets. This menu option requires the PSO role.
- **Status bar** – always shows the user name of the person logged in to Portal Studio (User), the group of portal pages available to this user (Resource).
- **Toolbar** – this is the Portal Studio’s static toolbar. From this toolbar, you can view your account information, access online help, and log out of Portal Studio. Once you make a selection from the left pane, an option-specific toolbar displays, which is discussed in the next section.

Selecting a Portal Studio left pane menu option

Once you log in to Portal Studio, you can choose the Portal Studio objects with which you want to work and have permission to access:

Build menu

Objects you access from the Build menu include:

- **Portlets** – create, edit, delete, and manage portlets and their content. See Chapter 4, “Building Portlets,” and Chapter 5, “Advanced Portlet Creation.”
- **Templates** – define the organization of aggregate portlets (portlets with more than one element), where portlets are located, background HTML code, and so on. Most portal pages contain an aggregate or collection of portlets. You can use the templates you create in the Page Builder. See Chapter 8, “Building Templates.”
- **Catalogs** – create user-defined hierarchies of content for display in a portal. Catalogs allow you to choose how your content is organized. See Chapter 9, “Building Catalogs.”
- **Pages** – create, edit, delete, and manage the portal pages on which portlets display. See Chapter 10, “Building Pages.”
- **Page Groups** – create, edit, delete, and manage page groups to group and organize portal pages. See Chapter 11, “Building Page Groups.”
- **Applications** – create virtual Web applications using portlets. See Chapter 5, “Advanced Portlet Creation.”

Automate menu

Objects you access from the Automate menu include:

- **Agents** – create, edit, manage, start, stop, and view logs for agents that automatically process portlet content. See “Using agents” on page 213.

- Adapters – create, edit, manage, and view logs for adapters that write portlet content to e-mails, databases, or file systems. See “Using adapters” on page 227.
- Servers – create, manage, and view logs for servers on which agents run. See “Using agent servers” on page 231.

Manage menu

Objects that you access from the Manage menu include:

- Resources – create and manage portal resources, also referred to as co-brands. Different resources can represent departments, divisions, or organizations for which you want different content and a different portal look and feel. See Chapter 14, “Creating Co-brands.”
- Roles – create, edit, and manage the roles that define role-based access to portal objects. “Using J2EE role-based security” on page 12.
- Users – edit users that have logged in to the portal and manage the resource with which they are associated. See “Creating users” on page 18.

Administer menu

Objects you access from the Administer menu include:

- Users – create, edit, delete, and manage your organization’s user accounts. The information in each user’s account determines whether the user is allowed to log in to the portal. See the *Enterprise Security Administration Guide* or the Enterprise Security on-line help in Portal Studio.
- Groups – create and edit groups that organize users in a way that is meaningful to your enterprise. See the *Enterprise Security Administration Guide* or the Enterprise Security on-line help in Portal Studio.
- Roles – create and edit J2EE roles. Enterprise Security enforces a role-based access control policy for the portal, which enables you to maintain individual accountability. See the *Enterprise Security Administration Guide* or the Enterprise Security on-line help in Portal Studio.
- Assets – create and edit the assets that represent portal objects in your security system. See the *Enterprise Security Administration Guide* or the Enterprise Security on-line help in Portal Studio.

Configure menu

Object you access from the Configure menu include:

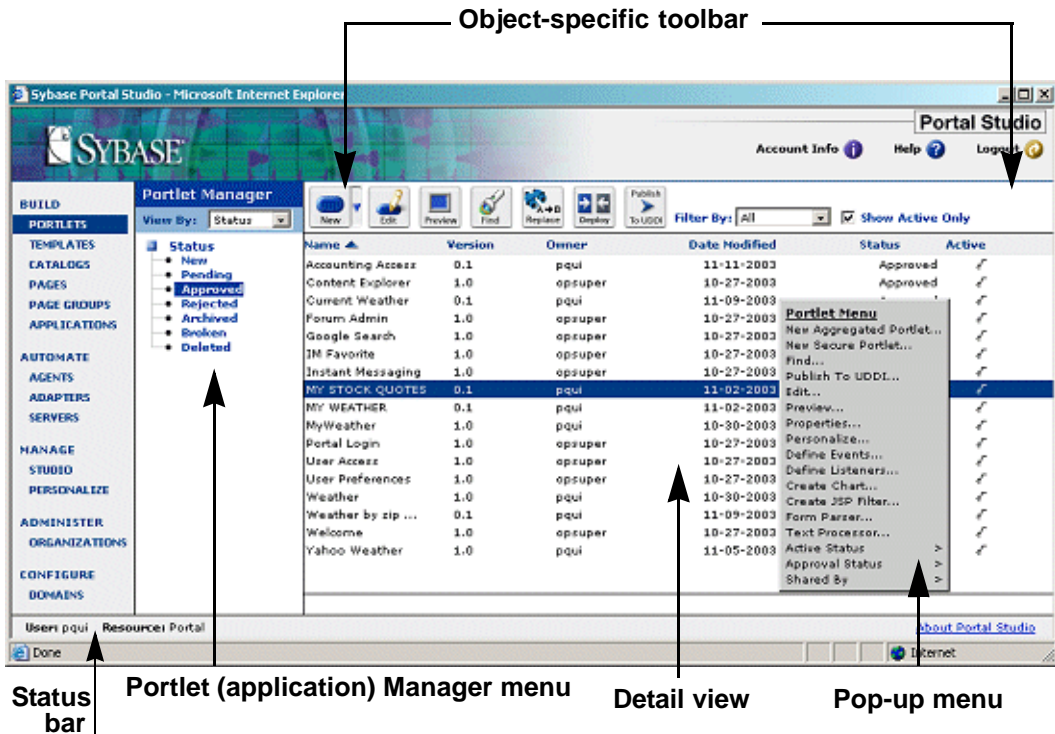
- Security Domain Roles – create and edit J2EE roles. Enterprise Security enforces a role-based access control policy for the portal, which enables you to maintain individual accountability. See the *Enterprise Security Administration Guide* or the Organizations | Roles on-line help in Portal Studio.

- Security Domain Assets – create and edit the assets that represent portal objects in your security system. See the *Enterprise Security Administration Guide* or the Organizations | Assets on-line help in Portal Studio.

Once you make a selection from the Portal Studio left pane, the Manager menu, detail window, and toolbar for the selected component display.

Note You must select from an object’s Manager menu for information to display in the detail window. Figure 1-3 displays the window you see when you choose Build | Portlets and select Approved from the Portlet Manager Status menu.

Figure 1-3: Portlet Manager



Object managers

All portal object managers (Portlet Manager, Template Manager, and so on) display similar object-specific user interface components:

- **Manager menus** – display View By options in the Portlet and Page Managers to change the grouping of portlets or pages. For portlets, the menu changes the grouping from Status to Category. For pages, the menu changes the grouping from Status to Type. The menus do not appear in the Template or Catalog Manager.
- **Detail window** – displays the type of items you selected from the Manager menu; for example, only new or approved portlets.
- **Toolbar** – lets you perform object-specific activities; for example, create a new object, edit, preview, and so on.
- **Pop-up menus** – let you perform object-specific activities on the selected item in the list.

Manager menus

Object managers allow you to display items based on your menu selection. You can display portlets and pages by Status or Category. Templates and catalogs can be displayed only by Status. Additional options on the toolbar allow you to filter the selected lists by additional criteria.

Detail window

The detail window displays items for the selected object based on your selections from:

- **View By** – view items by Status or Category. Templates and catalogs display only by Status.
- **Menu** – view items of a specific status or category. Templates and catalogs display items only by Status.
- **Filter By** – choose to see all items or only the items created by you.
- **Show Active Only** – choose to see only the selected items (portlets, templates, and pages) that are marked as active (available to be used).

If a list contains more items than can display in one window, numbers display directly below the detail window that allow you to navigate to the next group of items in the list.

Toolbars

Above the detail window are icons that allow you to perform a variety of activities on an item in the detail view. The icons that appear vary depending on the object manager with which you are working.

Pop-up menus

Once a list displays in the detail window, right-click a list item to display an object-specific pop-up menu showing activities you can perform on the selected item.

Note For instructions on getting started with Portal Interface, see the *Portal Interface User's Guide*.

Managing Portal Studio Security

Enterprise Portal Enterprise Edition is integrated with Sybase Enterprise Security to provide complete J2EE role-based container authentication for the portal. Portal users, roles, and assets are controlled through the Enterprise Security Manager, which is accessible through Portal Studio to users with the appropriate access rights.

This chapter describes how to set up and administer Portal Studio security. See the *Enterprise Security Administration Guide* for additional information.

Topic	Page
Integrating Enterprise Security and Portal Studio	11
Using J2EE role-based security	12
Setting up role-based security in Portal Studio	15
Using existing single sign-on mechanisms	23

Integrating Enterprise Security and Portal Studio

When you install EP Enterprise Edition 6.0 and later, you are prompted to choose an installation type:

- **Demo** – installs Enterprise Portal Studio and Portal Interface without Enterprise Security. The EP Demo installation uses database authentication and cannot be integrated with Enterprise Security.
- **Typical** – installs Enterprise Portal Studio and Portal Interface integrated with Enterprise Security. The EP Typical installation uses J2EE role-based container authentication by default.
- **Custom** – installs Enterprise Portal Studio and Portal Interface integrated with Enterprise Security and lets you select an existing or different data server or application server. The EP Custom installation uses J2EE role-based container authentication by default.

- Search only – installs only the Portal Search daemons and the dynamic reasoning engine utility, which allows you to distribute the DRE to one or more remote machines.

Note This document assumes that you installed Enterprise Portal with Enterprise Security and describes that configuration. Because this is the default configuration, you do not need to perform the procedures in this section—the installation performs the configuration for you. The information is provided here for reference.

Using J2EE role-based security

With the integration of Enterprise Security, Portal Studio allows users to see only those objects for which they have the necessary roles. For example, if Portlet_1 is protected by Role_A, and a user is not assigned Role_A, when the user selects Build | Portlets, Portlet_1 does not display in the Portlet Manager list of portlets.

This prevents Portal Studio users from bypassing the security policy for portal content by trying to preview a portlet they would not normally be allowed to see in Portal Interface.

Enterprise Portal's basic security infrastructure consists of:

- Users (accounts) – users are created by the Portal Security Officer (PSO) and administered in Enterprise Security, which allows EP to use container (application server) authentication.

To create a Portal Studio user, the PSO grants a new user a role that authorizes access to some of the Portal Studio assets. When that user logs in to Portal Studio, he or she is prompted to select a resource (by default, Portal or Japan, Inc.). After the user selects a resource, he or she becomes a Studio user, then appears in the Manage | Studio | Users list.

- Resources – users are grouped under resources. Each resource corresponds to an EP co-brand. Co-brands can represent companies, divisions, departments, and so on. The first time a user logs in to Portal Studio, he or she is prompted to choose a resource.

- Assets – any object to which you want to restrict access. There are security domain assets and portal assets. Security domain assets (also known as controlling assets) control access to the default organization’s object (user, groups, roles, and so on). Portal assets control access to the portal’s objects (pages, portlets, catalogs, and so on).
- Roles (permissions) – roles are created and administered in Enterprise Security. Portal Studio objects (pages, portlets, catalogs, and so on) are mapped to corresponding assets in Enterprise Security. Each asset has its own set of access permissions, mapped to the operations (create, edit, manage, and so on) a Portal Studio user can perform on a Portal Studio object. Assign role access permissions on specific assets, then assign these roles to a user.
- Company – the concept of “companies” is supported only in the Enterprise Portal version 6.0 Demo installation. EP Enterprise Edition comes with a default company, but the company does not appear in the Portal Interface or Portal Studio user interfaces.

To set up role-based security in the portal, an administrator logs in to Portal Studio using the PSO account and creates the appropriate assets, roles, and users in the security organization. For example, the PSO might create the StudioDeveloper, StudioDesigner, and PortalUser roles, and give the StudioDeveloper role “create” permissions on the Enterprise Security “portlet” asset.

Once the identity of a user has been verified, the security system controls the information the user is allowed to see, modify, or the applications this user is allowed to execute. Controlling access to a data source is called access control; assigning permissions to users or groups of users to access secured assets is called authorization.

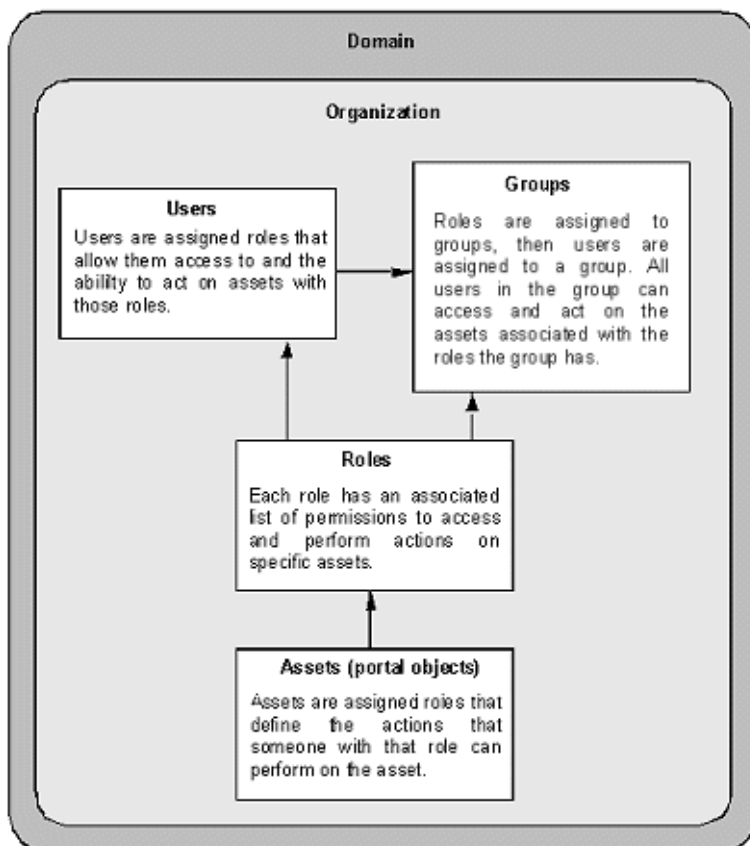
The PSO creates the user accounts and assigns each user the appropriate role. For example, the administrator assigns the StudioDeveloper and PortalInterfaceUser roles to a user named “Smith.” User Smith logs in to Portal Studio and creates a portlet, which is allowed because he is assigned the StudioDeveloper role that has that access right.

When Smith creates the portlet, he assigns the portlet the roles that a user must have to access that portlet. Smith sees all of the roles defined in Portal Studio, including those roles that he does not have. This allows him to assign roles he does not have to protect the portlet. Smith then assigns All users that have the roles assigned to the new portlet can access that portlet.

Note Although Smith sees all J2EE roles when creating an object, he sees only he sees only the Portal Studio objects that the StudioDeveloper and PortalUser roles have permission to see.

In summary, roles are assigned permissions to view and act on specified assets. User access to assets is controlled by the roles the user is assigned.

Figure 2-1: Portal Security role-based security



Suppose you are a Studio developer for an executive portal and your assignment is to create a new portlet that shows projected financial results for the next quarter. You know that only one group of portal users (financial officers) are allowed to see this information. Additionally, you are not allowed to see the portlet content.

First, build a new portlet based upon some dummy data of the right general format (for example, a PDF document). Actual content replaces this data when the portlet is deployed to the production system.

Next, assign the correct role to protect this portlet. You do not need to know the EP role that will be used on the production portal. In Portal Studio, select *Manage | Studio | Roles* to see if any existing roles meet your needs. If not, create a new role (for example, CFO). Using Jaguar Manager, create a new role mapping from CFO to PortalUser (or any other role that you have). Next time you log in to Portal Studio, you have that role.

Finally, create the portlet with the dummy data, assign the CFO role to it, and save it. At this point, if you have not created the CFO role mapping to a role you do have, the portlet disappears from your view in Portal Studio. Assuming you do have the appropriate role, you continue with testing, debugging, and so on, and eventually export the content and send it to the administrator to import into the production portal.

When the portlet is imported to the production portal, the administrator creates the correct mapping of the J2EE role reference CFO to the actual FinancialOfficer security role.

Setting up role-based security in Portal Studio

When you perform a typical or custom installation, Enterprise Portal and Enterprise Security have a default configuration that allows you to log in to Portal Studio and work with the Studio and its functionality.

Configuring domains

Enterprise Portal supports the creation of multiple security domains, which enables system administrators to define multiple domains within an enterprise, each with its own security policy. Domain managers can configure a set of properties (for example, to enable auditing or password validation) independent of other domains. Domain-specific properties are stored in the security Access Control Database (ACDB).

You can modify domain values using the integrated Portal Studio–Enterprise Security interface. Other security properties can be configured for your entire security system, rather than for a specific domain; these global properties are stored in the *security.properties* file.

The Enterprise Security installation creates a default domain—which contains the root organization—and an associated security policy. If you create suborganizations, you can choose whether to create them in the default domain or in a new domain.

For more information about setting up additional domains, see the *Enterprise Security Administration Guide*, Chapter 4, “Delegated Administration.”

❖ Configuring domains

- 1 To access Portal Studio, enter the following URL in your Internet Explorer browser:

```
http://HOSTNAME.PORTALDOMAIN:httplistenerport/onepage/index.html
```

For example, if your machine name is “tahiti”, your portal domain is “sybase.com,” and you are using the default HTTP listener port, enter:

```
http://tahiti.sybase.com:8080/onepage/index.html
```

- 2 When the Portal Studio Login window displays, enter the PSO user name (pso) and password (123qwe).
- 3 Select Configure | Domains. The default domain contains only one domain role for the Domain Security Officer—by default, the PSO user in the default organization—who has all the initial access rights assigned to manage the security domain and organization.

Enterprise Security also creates the domain assets listed in Table 2-1:

Table 2-1: Security domain assets

Predefined domain assets	Controls access to
SYBDOMAIN_DefaultDomain_AccessAssetTypeCtrlAsset	Access types and asset types

Predefined domain assets	Controls access to
SYBDOMAIN_DefaultDomain_AssetCtrlAsset	Assets
SYBDOMAIN_DefaultDomain_DomainCtrlAsset	The Domain
SYBDOMAIN_DefaultDomain_GroupCtrlAsset	Groups
SYBDOMAIN_DefaultDomain_OrgCtrlAsset	Organizations
SYBDOMAIN_DefaultDomain_RoleCtrlAsset	Roles
SYBDOMAIN_DefaultDomain_SubjectCtrlAsset	Subjects

- Right-click each asset in the detail pane and select Manage Access Permission. When the Manage Access Permission on Asset window appears, you can see at the bottom that the PSO has all rights for each domain asset, which allows the PSO to set up the initial security for your enterprise.

Note The PSO can work only with the security domain and organization. To use the Portal Studio's Build, Automate, and Manage options, you must create a new user and assign him the StudioAdmin role. See the *Enterprise Security Administration Guide*.

Administering organizations

The root organization is defined when you install and configure Enterprise Portal and Enterprise Security. You can create suborganizations to mirror the organizational structure of your enterprise. Enterprise Portal does not limit the number of suborganizations or the number of organizational levels. However, there can be only one root organization.

The root organization is where you create and administer your enterprise's users, groups, roles, and portal assets. An asset can be a document, database information, another computer system, an application, or any other object within the enterprise's computer systems.

The process of implementing access control involves selecting an asset, selecting a role with access permissions to perform operations on that asset, then granting the role to specific users or groups to accomplish authorization.

Typical access permissions include read, write, update, create, and delete. A permission can also give a user the right to start or stop an application or to access a back-end system.

Creating users

The Enterprise Portal installation automatically creates these security users:

- PortalSecOfficer (PSO) – automatically defined when you install and configure Enterprise Security. By default, the PSO user has the PortalSecOfficer and StudioAdmin roles, which makes him a superuser in the default Security domain and in the root portal organization.

Warning! Do not remove the PortalSecOfficer role from or add the PortalGuest role to the PSO user. Either of these actions may disable Portal Studio.

By default, the PSO has complete access to Portal Studio operations, which are the same privileges that the OPSUPER user had in pre-6.0 versions of EP.

Generally, the PSO sets up initial security for the domain and the organization, including creating the portal's users, but PSO can perform all other Portal Studio functions as long as they have the StudioAdmin role.

Note Once the PSO has set up the portal's users, including the Portal Studio administrator, you can remove the StudioAdmin role from the PSO to prevent that user from interacting with portal objects. See "Administering roles" on page 21 for instructions.

When the PSO creates a new user in the security organization, that user gets mapped to a Portal Studio user when they have been granted a Portal Studio role. The user name also displays in the Manage | Studio | Users list after the first time the user logs into Portal Studio and selects a resource.

- PortalAdmin – the portal administrator. If you want this user to represent your portal administrator, grant this user the StudioAdmin role.

Enterprise Portal lets you enable the deploy functionality for Portal Studio users with the PortalAdmin role, as described in the *global.properties.xml* portal configuration file:

```
<Property name="PortalAdministrationRole"
value="PortalAdmin" description="The J2EE role
required to administer the Portal performing
```

```
export/import and update operations."  
menugroup="-1" />
```

Warning! This property is designed for a single role and not a list of roles. There is no parsing of the string to look for multiple roles.

- **PortalGuest** – a role for the EP guest account. After you install and configure Enterprise Security, the guest account allows users to log in to your portal. The login name for the guest account is “guest”; the password is also “guest”.

Warning! Do not delete the guest account. It is required for Enterprise Portal.

Enterprise Security automatically creates the PortalGuest role and grants the role to the guest account. The sole purpose of the PortalGuest role is to enable self-registration in Portal Interface. The guest account does not have permission to access any assets. The system can have only one guest role. The guest role is valid for an indefinite period.

- **PortalUser** – a role for Portal Interface users. Users with this role can work with any Portal Interface object, but cannot access Portal Studio objects.
- **ReadAllListAll** – reserved for future use.
- **PortalWebPlugin** – related to the security Web server plug-in that protects assets stored in the ACDB from unauthorized access. See the *Enterprise Security Administration Guide*, Chapter 10, “Configuring the Web Server Plug-in.”

To create the organizational hierarchy for your enterprise, see “Managing Organizations and Suborganizations” in Chapter 2, “Setting Up Security” in the *Enterprise Security Administration Guide*.

For detailed instructions on creating users, see “Creating and Managing User Accounts” in Chapter 2, “Setting Up Security” in the *Enterprise Security Administration Guide*.

❖ **Creating a new user to access Studio assets and security**

Note There are two security assets—EPStudio.Studio and EPStudio.Personalize—with their own set of permissions. Each user must have permissions for one of these assets to see the Portal Studio Manage | Studio and Manage | Personalize menu options.

- 1 Log in to Portal Studio as the Portal Security Officer. Enter the user name “pso” and the password “123qwe”.
- 2 Select Administer | Organization from the menu in the left pane.
- 3 Select Users and click New on the Organization Manager toolbar.
- 4 When the Create New User window displays, complete the fields. Click Help to see a description of each field.

Note Do not use “opsuper” for the Login Name. This user does not exist in EP version 6.0 and later.

- 5 Click OK.
- 6 Grant the new user roles to have all permissions.
 - a Right-click the user you just created and select List Access Permissions. There are no permissions listed for the new user. Click OK.
 - b Right-click the user you just created and select Edit Roles.
 - c When the Edit User Roles window displays, select StudioAdmin, and click Add. The installer creates this role with all permissions.
 - d Click OK.
 - e Right-click the new user and select List Access Permissions. The permissions that display are based on the roles you granted the user.
 - f Click OK.
- 7 Log out.
- 8 Log in as the new user.

- 9 When prompted, select the “Portal” resource.

Note Users are grouped under resources. Each resource corresponds to an EP co-brand. Co-brands can represent companies, divisions, departments, and so on. The first time a user logs in to Portal Studio, he or she is prompted to choose a resource ID (RID).

- 10 Create some portlets and pages to use the permissions that were granted to this user.

❖ **Creating groups**

Groups are collections of users. If you grant a role to a group, each user in the group has that role.

Enterprise Security supports only one self-registration group, which is, by default, installed into the root organization. This allows users to self-registering in the root organization or any of the suborganizations. To use Portal Interface and Portal Studio, grant the PortalUser role to the self-registration group.

- 1 In Portal Studio, select Administer | Organizations.
- 2 Select Groups. The Enterprise Security installation automatically creates one group—SelfRegGroup.

New users log in to Portal Interface. From Portal Interface, they register with the portal and become members of the Enterprise Security self-registration group. By default, the self-registration group is granted the PortalUser role, which allows Enterprise Security to perform portal access checks.

To create additional groups and populate the groups with users, see “Creating and Managing Groups” in Chapter 2, “Setting Up Security” in the *Enterprise Security Administration Guide*.

❖ **Administering roles**

Roles are sets of permissions to access assets. The permissions assigned to a role define what a user with that role can do in the secured system. Each role can have multiple permissions assigned to it. Examples of permissions are create, update, administration, security, and management.

- 1 Select Administer | Organizations.

2 Select Roles. Enterprise Security has one default role—PortalSecOfficer. Portal Studio has these default roles—PortalAdmin, PortalGuest, PortalUser, PortalWebPlugin, ReadAllListAll, and StudioAdmin.

3 Select New to create a new role.

See “Creating and Managing Roles” in Chapter 2, “Setting Up Security” in the *Enterprise Security Administration Guide*.

❖ **Administering assets**

Assets are any portal object to which you want to restrict access. An asset can be a page, a portlet, a catalog, and so on.

1 Select Administer | Organizations.

2 Select Assets.

See “Creating and Managing Assets” in Chapter 2, “Setting Up Security” in the *Enterprise Security Administration Guide*.

When you install Enterprise Portal and Enterprise Security, the predefined assets and permissions shown in Table 2-2 are created under the root organization. In Portal Studio, the permission names are actually preceded by “EPStudio.”.

Table 2-2: Predefined assets and permissions

Assets	Permissions
EPStudio.Adapters	Create, Delete, Edit, Find, Manage, ViewLog
EPStudio.Agents	Broken, Create, Edit, Deleted, Find, Manage, Ready, Run, Stopped, ViewLog
EPStudio.Applications	Approved, Archived, Broken, Create, Deleted, Display, Edit, Find, Manage, New, Pending, Properties, Refresh, Rejected, Shared, SkipWorkflow
EPStudio.Catalogs	Approved, Archived, Broken, Create, Deleted, Display, Edit, Find, GetShared, Manage, New, Pending, Preview, Properties, Rejected, SelectCompany, Shared, SkipWorkflow
EPStudio.PageGroups	Approved, Archived, Broken, Create, Deleted, Display, Edit, Find, Manage, New, Pending, Properties, Refresh, Rejected, Shared, SkipWorkflow,

Assets	Permissions
EPStudio.Pages	Approved, Archived, Broken, Create, Deleted, Display, Edit, Find, Manage, New, Pending, Properties, Refresh, Rejected, Shared, SkipWorkflow
EPStudio.Personalize	Create, Delete, Edit, Manage
EPStudio.Portal	Deploy, Export, Import, Manage, UpdateCatalog, UpdatePage
EPStudio.Portlets	Active, Approved, Archived, Broken, Create, Deleted, Display, Edit, Element, Find, GetShared, Manage, New, Pending, Preview, Properties, Publish, Rejected, Replace *, SelectCompany, Shared, SkipWorkflow
	<p>Note * The Replace permission is not used. To have the Replace button display for a user in Portal Studio, the user must be in the RoleBaseDisplaySeeAllRoles security role as specified in <i>global.properties.xml</i>.</p>
EPStudio.Resources	Create, Delete, Edit, Manage, Undelete
EPStudio.Roles	Create, Delete, Edit, Manage, Undelete
EPStudio.Servers	Broken, Create, Delete, Deleted, Edit, Find, Manage, Ready, Stopped, ViewLog
EPStudio.Studio	Manage
EPStudio.Templates	Active, Approved, Archived, Broken, Create, Deleted, Edit, Manage, New, Pending, Preview, Rejected, Shared, SkipWorkflow
EPStudio.Users	Edit, Delete, Manage

Using existing single sign-on mechanisms

If your enterprise already has a single sign-on mechanism in place for its users, you can customize Enterprise Portal to:

- Use the existing single sign-on mechanism to guide an authenticated user seamlessly into the portal without having to log-in again, and

- Modify EP so that when a previously anonymous user is authenticated by the portal, your single sign-on token is generated, which allows this user single sign-on access to other services in your enterprise.

Note The instructions in this section assume that the developer fully understands the existing single sign-on system and its Java-based programming interface. This knowledge is necessary to modify portal JSPs to use that Java API to participate in an existing single sign-on or to generate a new single sign-on token for a user.

Customizing EP to using an existing single sign-on mechanism

To customize the portal to use an existing single sign-on mechanism, modify the *custom_declarations.jsp* and *custom_sso.jsp* files—which do nothing by default—to include the single sign-on detection logic.

These files are located in

\$SYBASE/EAServer/Repository/WebApplications/onepage/ on UNIX and in *%SYBASE%\EAServer\Repository\WebApplications\onepage* on Windows.

When *custom_sso.jsp* detects a single sign-on token in the request, it sets the user name ID and password variables appropriately to allow the user to log in automatically. Use the *custom_declarations.jsp* as necessary to add imports, declare class functions or members, initialize statics, and so on.

If you modify the *custom_*.jsp* files to support single sign-on into the portal using your own SSO mechanism, start your Portal Interface sessions using:

```
http://HOSTNAME.DOMAIN.PORT/onepage/ep_sso.jsp
```

not:

```
http://HOSTNAME.DOMAIN.PORT/onepage/index.jsp
```

.

Generating an single sign-on outbound token

To customize the portal to generate a single sign-on outbound token, modify the *customize_protected_page.jsp* file as necessary. This JSP also does nothing by default, but has commented out sample code that sets a cookie.

This file is located in

\$\$SYBASE/EAServer/Repository/WebApplications/onepage/fw/baseApps/fwlogin on UNIX and in

%SYBASE%\EAServer\Repository\WebApplications\onepage\fw\baseApps\fwlogin on Windows.

Auditing Enterprise Portal

Topic	Page
Introduction	27
Auditable portal actions	31
Reading audit information	34
Auditing rules	36
Auditing a distributed installation	37

Introduction

EP Enterprise Edition integrates Enterprise Security with Enterprise Portal and includes an auditing feature that allows you to monitor user actions.

You can configure auditing to:

- Send notifications of audited events to a JMS message topic.
- Direct the audit output to a file, a JDBC-compliant database, or a custom application.
- Use a database connection cache to improve performance and scalability by allowing client connections to share open connections.
- Refuse authorization and authentication requests if errors occur when an audit message is logged, such as when the file system is full.
- Disable auditing in the case of an error, allowing users to continue to log in to the system.

In the portal, auditing is supported in both Portal Interface and Portal Studio and provides audit records for actions performed on protected resources, such as portlets and pages.

Portal audit asset types and access types are added into the Enterprise Security access control database (ACDB) during EP installation, so that Portal Studio users with the appropriate role can administer portal audit events.

Note Enterprise Portal uses the public Audit EJB component—a stateless EJB similar to AuditSPI used in Enterprise Security—in the portal framework to provide control on what to audit.

You set up auditing in several places—in Enterprise Security, in Portal Studio, and in the portal *global.properties.xml* configuration file. This section describes in detail the changes you make to *global.properties.xml* to enable auditing in Portal Interface and Portal Studio. The procedures you perform in the Enterprise Security Manager are mentioned here for reference, and covered in detail in the *Enterprise Security Administration Guide*, Chapter 7, “Auditing.”

Setting up Enterprise Portal auditing

To audit Enterprise Portal activities, you enable the auditing feature, define where to store the audit information, and specify which actions to audit.

Note To configure auditing for a domain, you must have update permission on the domain’s controlling asset.

To enable auditing in Enterprise Portal, you perform these steps:

- 1 Edit the *global.properties.xml* portal configuration file to set audit-specific properties.
- 2 Set up auditing in Enterprise Security.
- 3 Enable auditing in the EP security domain.
- 4 Specify which actions to audit.

This section contains a procedure for each step.

❖ Editing *global.properties.xml*

- 1 Open *global.properties.xml* in WordPad on Windows and in any text editor on UNIX. This file is located in:

- `%JAGUAR%Repository\WebApplication\onepage\config` on Windows
- `$JAGUAR/Repository/WebApplication/onepage/config` on UNIX

Note If you are using an application server other than EAServer with EP, enter that application server's path to the `/onepage/config` directory.

- 2 Search for “AuditGroup” and set or change these property values:
 - `auditEnabled` (true/false) – set to “true” to enable auditing in Portal Interface and Portal Studio. Set to “false” to disable auditing.
 - `auditPortletPlayback` (true/false) – set to “true” to audit portlet playbacks.
 - `auditRoleFilter` (true/false) – set to “true” to audit role filtering on auditable resources.
- 3 Search for “portal.epSecurity” and set the value to “true.”
- 4 Search for “*epsecurity.xml*” and set the value to “true.” This property allows the Enterprise Security menu to display on the left panel in Portal Studio.
- 5 Save the file and exit the text editor.
- 6 Stop and restart the application server to implement the changes.

❖ **Setting up auditing in Enterprise Security**

- 1 Enable auditing in Enterprise Security using either `securetool` or the Enterprise Security Manager. See the procedure “Enabling auditing” in the *Enterprise Security Administration Guide* section, “Setting up auditing.”
- 2 Establish where to store the audit information. See the procedure “Specifying where to store audit information” in the *Enterprise Security Administration Guide* section “Setting up auditing.”
- 3 Set up the audit database. See the procedure “Setting up the audit database” in the *Enterprise Security Administration Guide* section “Setting up auditing.”

❖ **Enabling auditing for the Enterprise Portal security domain**

Enable auditing for the security domain (usually on the default domain) and specify which portal resources and actions to audit.

- 1 To access Portal Studio, enter the following URL in your Internet Explorer browser:

`http://HOSTNAME.PORTALDOMAIN:httplistenerport/onepage/index.html`

For example, if your machine name is “tahiti”, your portal domain is “sybase.com,” and you are using the default HTTP listener port, enter:

`http://tahiti.sybase.com:8080/onepage/index.html`

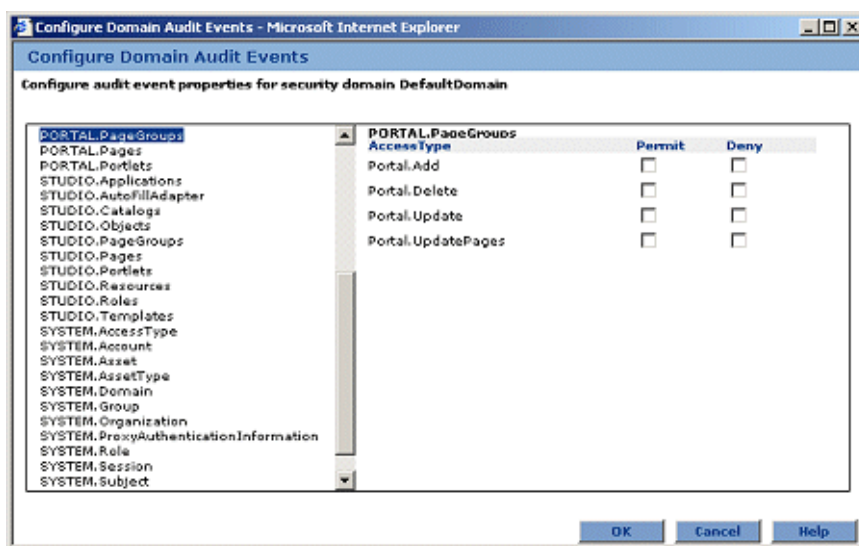
- 2 When the Portal Studio Login window displays, enter the user name and password for a user with portal administrator privileges.
- 3 Select Configure | Domains in the Portal Studio left pane.
- 4 Click “Default Domain” (or another domain if your organization is not in the default domain) to select that option in the center pane, then right-click in the left pane and select Configure General Properties.
- 5 In the Configure | Domain General Properties window, select:
 - Domain Property Refresh Time Interval – to specify how often (in seconds) the system refreshes the domain-specific properties by reading their values from the security access control database (ACDB).
 - Enable Auditing – to enable auditing for the domain. To specify which events to audit, see Defining which events to audit to turn on auditing for the selected security domain.
 - Suspend Auditing When Unable to Log Audit Message – to turn off auditing when a system problem prevents logging the auditing information. Selecting this property prevents a failure in the auditing module itself from causing a transaction to roll back.
 - Include User’s DN in Audit Records – to include the subject DN in the audit record’s subject ID.

- Notify Audit Events – to send notifications of audited events to a JMS message topic.

Warning! Do not select Notify Audit Events until you have set up the message service and created the message topic in the application server; otherwise, you will not be able to log in to Portal Interface or Portal Studio. See the procedure “Setting up JMS auditing notifications” in the *Enterprise Security Administration Guide* section “Setting up auditing.”

Once you enable notifications, a message is sent to a JMS message topic in the application server for each audited event.

- 6 Click OK.
- 7 Configure Audit Events from the pop-up You see the Configure | Domain Audit Events window.



Auditable portal actions

This section lists the Portal Studio and Portal Interface assets that you can audit. For a list of auditable system assets, see also system assets that you can audit

Portal Interface

Table 3-1 lists the actions you can audit in Portal Interface.

Table 3-1: Portal Interface auditable assets actions

Asset types	Action name	Action description
PORTAL.PageGroups	Portal.Add	Adding a page group.
	Portal.Delete	Deleting a page group.
	Portal.Update	Changing a page group's name or page order.
	Portal.UpdatePages	Updating a user's page groups and pages after the user logs in. This happens when the global property deferupdate is turned on and a Portal Studio user initiated the update operation.
PORTAL.Pages	Portal.Add	Adding a page.
	Portal.Delete	Deleting a page.
	Portal.List	Listing pages that are available to add.
	Portal.Share	Sharing a page with another user.
	Portal.Update	Changing a page's layout or name.
	Portal.UpdatePages	Updating a user's page groups and pages after the user logs in. This happens when the global property deferupdate is turned on and a Portal Studio user initiated the update operation.
PORTAL.Portlets	Portal.Add	Adding Portal Studio-created portlet from a catalog
	Portal.Create	Creating a one-click capture portlet in Portal Interface using Create Portlet
	Portal.Delete	Deleting a portlet.
	Portal.PortalPlayback	Playing back a portlet on a page. For <DIV> tag portlets, if the portlet has already been played back once and cached, subsequent playbacks are not audited. However, refreshing a portlet audits portlet playback again.
	Portal.Update	Editing a portlet.

Portal Studio

Table 3-2 lists the actions you can audit in Portal Studio.

Table 3-2: Portal Studio auditable assets actions

Asset types	Action name	Action description
STUDIO.Applications	Studio.Create	Creating a new application. Triggers page and page group add and update actions, which occur immediately, bypassing the deferupdate global setting only for the current user.

Asset types	Action name	Action description
STUDIO.AutoFillAdapter	Studio.Create	When an autofill adapter is created using Manage Personalize Adapter, the adapterType attribute in the audit records shows the adapter type.
	Studio.Delete	Deleting a personalization adapter.
	Studio.Update	Clicking Edit in Adapter Builder.
STUDIO.Catalogs	Studio.Create	Creating a new catalog or saving an existing catalog. If the “requested catalog id” is zero, the catalog was newly created. Does not audit the portlets added to the catalog
	Studio.Update	Clicking Update Catalog.
STUDIO.Objects	Studio.Deploy	Clicking Deploy and performing deploy operation. All Studio objects deployed (saved to .ear file) are audited as attributes
	Studio.Export	Clicking Deploy and performing export operation. All the objects exported (saved to .xml file) are audited as attributes. The log content is listed as an attribute.
	Studio.Import	Clicking Deploy and performing import operation. All the objects imported (from the .xml file) are audited as attributes, which may trigger the page and page group update actions.
STUDIO.PageGroups	Studio.Create	Creating a new page group or saving an existing page group. If the “requested page group id” is zero in the audit record, the page group was newly created. Does not audit pages and portlets added to the page group.
	Studio.Update	Clicking Update. If the global property deferupdate is turned on, the update happens when the user logs in; otherwise, the update occurs immediately. Either way, the audit record is generated. When the update happens, it triggers page and page group deletions and portlet additions.
STUDIO.Pages	Studio.Create	Creating a new page or saving an existing page. If the “requested page id” is zero in the audit record, the page was newly created. Does not audit portlets added to the page.
	Studio.Update	Clicking Update. If the global property deferupdate is turned on, the update happens when the user logs in; otherwise, the update occurs immediately. Either way, the audit record is generated. When the update happens, it triggers page and page group deletions and portlet additions.
STUDIO.Portlets	Studio.Create	Creating a new portlet or saving an existing portlet.

Asset types	Action name	Action description
	Studio.Preview	Clicking Preview in Portlet Builder. The portlet preview is audited when a valid portlet ID is passed to <code>UWPWindowApp.GetWindowContent</code> .
	Studio.Search Replace (one word)	Performing a search and replace action on a portlet. The pages and catalogs affected are listed as attributes in the audit record. If Update Pages is selected, this action also triggers the page and page group update action and audit.
STUDIO.Resources	Studio.Create	Creating a new resource ID.
	Studio.Delete	Deleting or undeleting a resource. Deleting a resource ID does not physically delete the resource; the resource is just marked for deletion from the database.
	Studio.Update	Clicking Edit in the Resource Builder.
STUDIO.Roles	Studio.Create	Creating a J2EE role reference.
	Studio.Delete	Deleting a J2EE role reference.
	Studio.Update	Clicking Edit in the Role Builder. If you “undelete” a J2EE role, an update action is generated.
STUDIO.Templates	Studio.Create	Creating a new template or saving an existing template.
	Studio.Preview	Clicking Preview in Template Builder.

Reading audit information

To read information from the audit database, use a SQL statement to retrieve the data. For example, to retrieve the information in the `auditData` field for all events that occurred after 1 September 2003, use this statement:

```
select auditData from Audit where timeStamp >
(select convert(datetime, "9/1/2003"))
```

Sample Enterprise Portal audit records

This section contains sample audit records for various Portal Interface AssetType/AccessType combinations.

SYSTEM.Session/SYSTEM.Login

```
<?xml version="1.0" encoding="UTF-8"?>
```

```

<audit:AuditLog xmlns:audit="http://www.sybase.com/ep/xsd/audit/6.0">
<AuditRecord Action="SYSTEM.Login" Decision="permit"
Subject="90810200000001001" When="2003-08-12T23:41:02"
SessionID="SYBSECPK28670A1655304788d5f6f888a4198000"
ID="0A165530:4788d5:f6f888a419:-7fff">
<Resource Class="SYSTEM.Session"
ID="SYBSECPK28670A1655304788d5f6f888a4198000"/>
<Attribute Name="AuthenticationType" Value="AUTH_BY_CREDENTIAL"/>
<Attribute Name="SubjectUID" Value="hwong1"/>
</AuditRecord>

```

SYSTEM.Session/SYSTEM.Logout

```

<AuditRecord Action="SYSTEM.Logout" Decision="permit"
Subject="90810200000001001" When="2003-08-12T23:46:05"
SessionID="SYBSECPK28670A1655304788d5f6f888a4198000"
ID="0A165530:4788d5:f6f888a419:-7ff6">
<Resource Class="SYSTEM.Session"
ID="SYBSECPK28670A1655304788d5f6f888a4198000"/>
<Attribute Name="LogoutTime" Value="2003-08-12T23:46:05"/>
<Attribute Name="LogoutType" Value="Terminated"/>
</AuditRecord>

```

PORTAL.Page Groups/Update

```

<AuditRecord Action="Portal.Update" Decision="permit"
Subject="90810200000001001" When="2003-08-12T23:43:36"
SessionID="SYBSECPK28670A1655304788d5f6f888a4198000"
ID="0A165530:4788d5:f6f888a419:-7ff9">
<Resource Class="PORTAL.Page Groups" ID="MyPage"/>
<Attribute Name="tabsetName" Value="MyPage"/>
<Attribute Name="taborder" Value="2"/>
<Attribute Name="uid" Value="221"/>
<Attribute Name="rid" Value="1"/>
<Attribute Name="tabset" Value="3"/>

```

PORTAL.Pages/Add

```

<AuditRecord Action="Portal.Add" Decision="permit"
Subject="90810200000001001"
When="2003-08-12T23:42:46"
SessionID="SYBSECPK28670A1655304788d5f6f888a4198000"
ID="0A165530:4788d5:f6f888a419:-7ffa">
<Resource Class="PORTAL.Pages" ID="page1"/>
<Attribute Name="addTabName" Value="page1"/>

```

```
<Attribute Name="taborder" Value="1"/>
<Attribute Name="uid" Value="221"/>
<Attribute Name="rid" Value="1"/>
<Attribute Name="catalogName"/>
<Attribute Name="tabset" Value="3"/>
```

PORTAL.Pages/Delete

```
<AuditRecord Action="Portal.Delete" Decision="permit"
Subject="90810200000001001" When="2003-08-12T23:45:03"
SessionID="SYBSECPK28670A1655304788d5f6f888a4198000"
ID="0A165530:4788d5:f6f888a419:-7ff7">
<Resource Class="PORTAL.Pages" ID="{taborder=1, uid=221, rid=1, tabset=3}"/>
<Attribute Name="taborder" Value="1"/>
<Attribute Name="uid" Value="221"/>
<Attribute Name="rid" Value="1"/>
<Attribute Name="tabset" Value="3"/>
```

PORTAL.Portlets/Playback

```
<AuditRecord Action="Portal.Playback" Decision="permit"
Subject="90810200000001001" When="2003-08-12T23:41:16"
SessionID="SYBSECPK28670A1655304788d5f6f888a4198000"
ID="0A165530:4788d5:f6f888a419:-7ffe">
<Resource Class="PORTAL.Portlets" ID="231"/>
<Attribute Name="taborder" Value="1"/>
<Attribute Name="rid" Value="1"/>
<Attribute Name="tabset" Value="1"/>
</AuditRecord>
```

Auditing rules

The following rules apply when Portal Interface and Portal Studio auditing are enabled:

- When you add a page, portlets on the new page that you do not have the role to access are removed from the page. The removed portlets are not audited as deleted.

- When you delete a page group, the pages within the page group and the portlets within the pages are not audited as deleted.

Note You cannot delete or modify the Default Page Group.

- When you delete a page, the portlets within the page are not audited as deleted.

Note You cannot delete or modify default pages.

- Portal Interface and Portal Studio do not have fine-grain access control on all assets; that is, the portal does not use Enterprise Security's `PortalSession.getAuthorization()` to check the authorization during the execution of an action on a asset. Therefore, the audit records that capture those actions show only "permit" records.
- Objects that user's do not have access to are filtered out.

Auditing a distributed installation

To enable auditing in a distributed Enterprise Portal installation, Enterprise Security with the Audit EJB must be installed on each EAServer or other application server. If audits are stored in a database, all audit entries from all EAServers are there. File-based audits are stored on each machine.

Building Portlets

Portlets are the primary objects that developers work with in Portal Studio. Portlets comprise one or more content elements and portal pages can contain one or more portlets. This chapter explains how to create and manage portlets.

Topic	Page
Introduction	39
Building aggregated portlets	48
Creating Web elements	49
Creating XML elements	52
Creating database elements	54
Creating JSP elements	56
Creating Web service elements	62
Creating HTML-based elements	74
Creating document elements	75
Building secure portlets	79
Saving portlets	82
Approving elements and portlets	84
Viewing, editing, and deploying portlets	85
Finding and replacing portlets	91

Introduction

Portlets are dynamic, reusable components that capture and deliver information to end users of portal or wireless devices. Portlets consist of one or more content elements arranged within a template. These elements can display Web content, XML feeds, database query results, JavaServer Pages, or HTML. Portal Studio allows you to create and manage portlets.

Portlet Manager

The Portlet Manager allows you to:

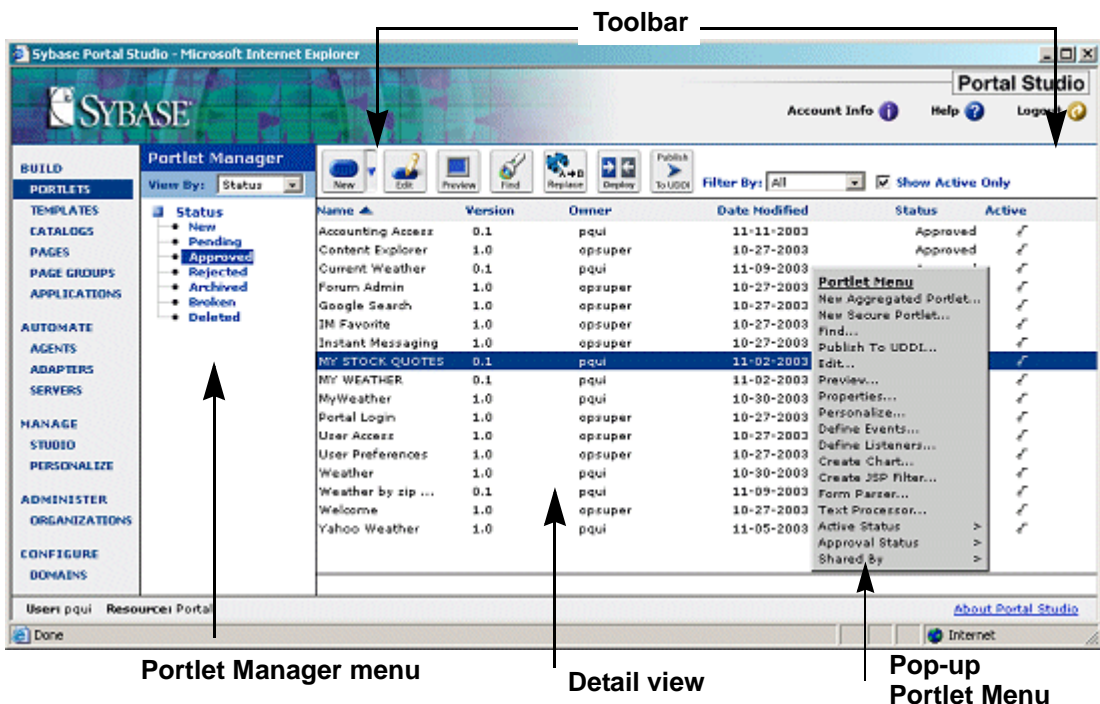
- Create new portlets by acquiring and displaying content.
- Manage existing portlets and display the content acquired from various sources.

Use the Portlet Manager to perform management functions for existing portlets. Select New or Edit from the Portlet Manager toolbar to create or edit portlets. The portlet attributes that displays as columns in the detail view are Name, Version, Owner, Date Modified, Status, and Active.

❖ Launching the Portlet Manager

- 1 To launch the Portlet Manager, log in to Portal Studio (see “Using Portal Studio” on page 4).
- 2 Select Build | Portlets from the Portal Studio left pane.

Figure 4-1: Portlet Manager



New objects created in Portal Studio are associated with the resource that displays on the status bar.

Portlet types

Portlets are composed of “elements”—pieces of information. The Portlet Manager supports two basic types of portlets:

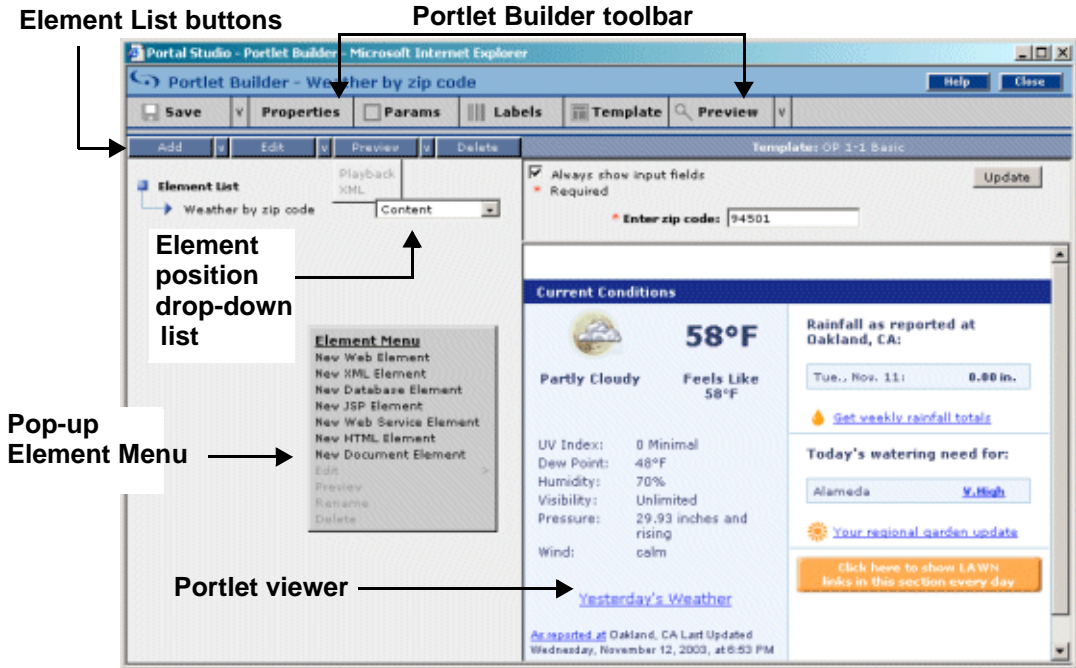
- Aggregated portlet – use the Portlet Builder to select multiple elements, modify the template selection and other portlet properties.
- Secure portlet – content captured from a secure Web site. Use the Portlet Builder to create and edit secure portlets.

Portlet Builder

Use the Portlet Builder to create or edit portlets. To launch the Portlet Builder:

- Click New on the Portlet Manager toolbar to display the Portlet Builder; or click the New icon down arrow on the toolbar and select Aggregated Portlet or Secure Portlet, or
- Right-click in the Portlet Manager detail view and select New Aggregated Portlet or New Secure Portlet from the pop-up.

Figure 4-2: Portlet Builder



Element list

The Element List displays the names and position drop-downs for each element within a portlet. Selecting an option from the position drop-down assigns the corresponding element to a template cell.

Note A template contains HTML code that defines the position and layout of elements within a portlet. See Chapter 8, “Building Templates.”

Table 3-1 describes the functionality of the Element List buttons.

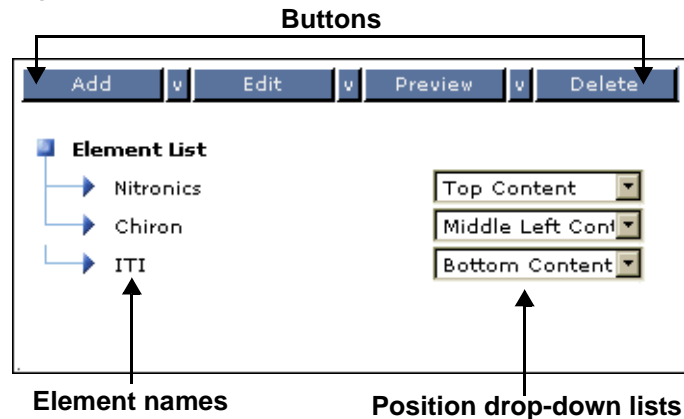
Table 4-1: Element list buttons

Button	Description
Add	Click the down arrow beside the Add button to add an element of type: Web, XML, Database, JSP, Web Service, HTML, or Document. Create portlets with multiple elements from multiple sources.
Edit	Edit the element definition for XML, database, JSP, and document elements.

Button	Description
Preview	Launch a preview in a separate window: <ul style="list-style-type: none"> HTML – displays the HTML output of the selected element. XML – displays the XML output of the selected element.
Delete	Remove the selected element.

Figure 3-3 shows the Element List in the Portlet Builder.

Figure 4-3: Element List

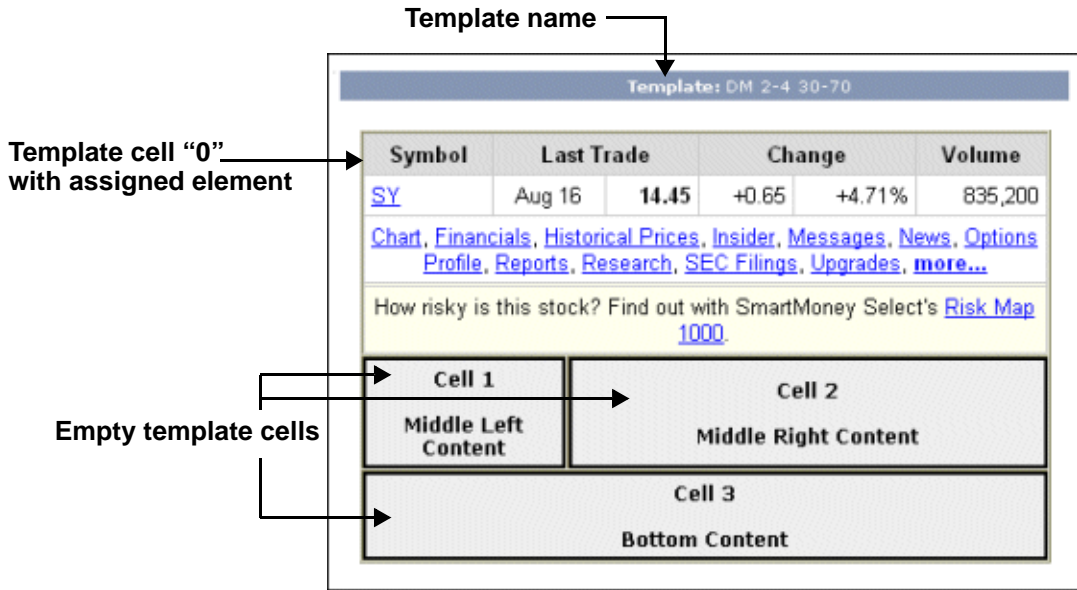


The position drop-down lists let you specify the area of the portlet—the template cell—where the element should display.

Portlet viewer

The portlet viewer (the right pane of the Portlet Builder) displays the element layout within the portlet.

Figure 4-4: Portlet viewer



Template cells are sequentially numbered beginning with "0" (zero). Template cells that do not have elements assigned to them display the cell number and the cell name. The portlet viewer also displays the name of the template assigned to the portlet.

Portlet Builder toolbar

The Portlet Builder buttons allow you to edit and view portlet properties and parameters, create and edit labels and templates, and preview portlet content. Table 3-2 describes these buttons.



Table 4-2: Portlet Builder toolbar options

Option	Description
Save	<p>Choose from:</p> <ul style="list-style-type: none"> • Save – save the portlet under the same name. • Save As – save a new portlet or save an existing portlet under a new name. <p>See "Saving portlets" on page 82.</p>

Option	Description
Properties	View and edit the properties that were specified when the portlet was created. Refer to the section in this chapter on each element type.
Params	View and configure portlet parameters. See “CGI parameters” on page 87.
Labels	Launch the edit window, which allows you to specify portlet labels. See “Portlet labels” on page 90.
Template	Select or create a different template. See Chapter 8, “Building Templates,” for more information about templates. Warning! Do not create new templates when you create secure portlets—you can receive a JavaScript error.
Preview	Preview the portlet as follows: <ul style="list-style-type: none"> • Playback – display the HTML preview. • XML – display only data-capable elements. See “XML output definition” on page 45. • Edit Screen – display the window that portal users see if they select Edit Portlet from their portal environment. Warning! If you choose HTML preview for an XML portlet, the portlet does not playback correctly.

XML output definition

If you choose to preview a portlet’s tabular data output as XML, EP uses the following XML definition to format the data:

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- edited with XMLSPY v5 rel. 3 U (http://www.xmlspy.com) by XMLSPY 5
Enterprise Edition, Installed Multi-User (Web Services Architecture WG) -->
<!--W3C Schema generated by XMLSPY v5 rel. 3 U (http://www.xmlspy.com) -->
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
elementFormDefault="qualified">
  <xs:element name="ContentDoc">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="ApplicationDef" minOccurs="0"/>
        <xs:element ref="CGIParameters" minOccurs="0"/>
        <xs:element ref="Element"/>
        <xs:any processContents="lax" minOccurs="0"
maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```



```

    </xs:complexType>
  <xs:element name="SubField">
</xs:element>
  <xs:element name="Record">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Field" maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="PostProcessingDef">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Field" maxOccurs="unbounded"/>
        <xs:any processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="apply2xml" type="xs:boolean"/>
      <xs:attribute name="name" type="xs:string" use="required"/>
      <xs:attribute name="order" type="xs:string" use="required"/>
    </xs:complexType>
  </xs:element>
  <xs:element name="Url">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Parameter" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Parameter">
    <xs:complexType>
      <xs:sequence>
        <xs:any processContents="lax" minOccurs="0"
          maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="auto_fill_adapter" type="xs:string"/>
      <xs:attribute name="auto_fill_key" type="xs:string"/>
      <xs:attribute name="cgi_mapped" type="xs:boolean"/>
      <xs:attribute name="default_value" type="xs:string"/>
      <xs:attribute name="display_name" type="xs:string"/>
      <xs:attribute name="name" type="xs:string"/>
      <xs:attribute name="required" type="xs:boolean"/>
      <xs:attribute name="sequence" type="xs:boolean"/>
      <xs:attribute name="type" type="xs:string"/>
      <xs:attribute name="variable" type="xs:boolean"/>
    </xs:complexType>
  </xs:element>

```

```
</xs:complexType>
  </xs:element>
</xs:schema>
```

Building aggregated portlets

An aggregated portlet contains one or several elements or portlets that display in one window. Regardless of the type of portlet you create, the steps to begin the procedure and the steps to save the element or portlet at the end of the procedure are the same. These common steps are provided in this section, followed by instructions on how to create each portlet type.

❖ Creating multi-element portlets

- 1 Open the Portlet Builder:
 - Click New on the Portlet Manager toolbar to display the Portlet Builder; or click the New icon down arrow on the toolbar and select Aggregated Portlet or Secure Portlet, or
 - Right-click in the Portlet Manager detail view and select New Aggregated Portlet from the pop-up.
- 2 A default template appears in the right frame. To choose a different template, click Template. The Find Template window displays.
- 3 Enter a template's name or partial name using an asterisk (*) as a wildcard if you are searching for a specific template. Leave the Name field blank to see all available templates. Click Search.
- 4 A list of Approved templates displays in the Results pane, click:
 - New – to launch the Template Builder and create a new template.
 - Edit – to edit the selected template.
 - Select – to choose the selected template. When you do this, the template that displayed in the Portlet Builder is replaced by the template you selected.
 - Preview – to view the element layout of the selected template.

Note See Chapter 8, “Building Templates,” for instructions on creating new templates and editing existing templates.

- 5 Click the Add icon down arrow and select an element type.

Note Regardless of the element type, the procedure to save the element and portlet is the same. See “Saving portlets” on page 82.

Creating Web elements

Web elements allow you to capture Web content and display it in a portlet. The captured content is updated dynamically as the source Web content changes.

The New Web Element wizard steps you through creating a Web element and provides different capture strategies, which represent the different HTML objects that you can capture from Web pages.

Warning! The content of many Web sites is protected by the Copyright Law of the United States, related laws contained in Title 17 of the United States Code, and international treaties. Failure to comply with the provisions of such law may result in the imposition of fines and/or damages. Each developer and user is responsible for compliance with applicable law in the use and reproduction of copyrighted content.

Capture formats

The capture strategies listed in the Format drop-down of the Web element wizard represent the different HTML objects that Portal Studio can capture. These include: Applets, Area, Buttons, Grids, Images, Image Maps, Lists, and Forms. The capture option window displays all the instances of the particular HTML object found on a Web page.

One-click

Enterprise Portal uses a one-click content capture technology that allows users to create portlets from Web accessible sources.

Using this technology, you use links to navigate to the page that has the information from which you want to create a portlet. The content capture technology allows you to select only those pieces of information that you want use in a portlet.

Because of the numerous technologies that are used, you cannot always navigate to every possible page and successfully create portlets from all Web accessible sources. You may not be able to navigate to some sites or pages within sites. You may also not be able to create a portlet using the exact piece of information that is viewable to the user on a Web page. Especially challenging are Web sites that use JavaScript to generate some or all of an HTML page, and sites that use frame-based solutions.

Once you create a portlet using a Web accessible source, the portlet gets the current information from the source. If the layout of the information on the original page changes, the underlying technology attempts to locate the correct piece of information that represents the portlet's content. However, the portlet can lose its content or obtain an incorrect piece of information because of changes in the source site. In such a case, the portlet is broken and must be re-created.

The one-click capture strategy allows you to target a specific area on a Web page from which to capture content. Click the left mouse button in a suitable area, indicated by a shadow-box that trails the mouse arrow. The capture option window displays the "best guess" capture from the specified area.

Web elements do not store static HTML, but instead are defined by a CCL statement that dynamically retrieves the content from its source. The CCL statement consists of:

- The location of the Web page, and
- The feature tag of the captured content. This feature tag is an electronic signature of the content and consists of a feature list that describes the content.

When Portal Studio executes the CCL statement, the capture technology parses the Web page and returns the piece of content that matches the feature tag. If an exact match cannot be determined, the content with the closest matching feature tag is returned. This allows the Portal Studio to render the correct Web element even though the content, location, or formatting of the source may have changed.

❖ Creating a Web element

- 1 Select Web Element. You see the Web Element Definition window. Enter a beginning uniform resource locator (URL)—the Internet address—of the Web content you want to capture in the Location text field, then click Find.
- 2 When the location you entered displays, navigate to the page that contains the desired content, then click Next.
- 3 Select the capture strategy from the Format drop-down list. This example uses the one-click capture format, so you select the content you want to capture. If you chose one of the other formats, only those elements that match the chosen format display.
- 4 If necessary, select one, both, or neither (the default) of the two options to the left of the Back button. These options (CapAll and ACX) are not exclusive; they are separate, independent options.
 - CapAll – captures all the GETs and POSTs. Using the default settings during navigation and capture, EP remembers the final URL of the page that you want to capture and play back, along with any intermediate URLs that are generated by HTTP “POST” requests. These latter URLs typically result from submitting form data. During playback, these URLs are submitted in order and the resulting page content returned to the client browser.

However, for some sites this is not sufficient to guarantee that enough context is provided to the Web server to return the desired page. In this situation, it may be necessary to include other intermediate HTTP requests (that is, HTTP “GET” requests) so that the correct client state can be established and the desired page returned. If you suspect this situation is occurring, select this option to request that during the capture phase, EP should record all significant HTTP requests against the Web server from which the page is being captured.

- ACX – passes HTML and JavaScript through the ActiveX component. Problems can occur when you capture and play back content that is partially or completely generated by client-side JavaScript. If you suspect that this is occurring on a site with content you want to capture, select ACX to specify that the capture technology use ActiveX controls to force the rendition of any embedded JavaScript and generate the entire HTML content. When using this option, all JavaScript is removed from the page content returned to the client browser, and the resulting HTML is static.

To capture all GETs and POSTs, and also pass the final HTML or JavaScript through the ActiveX component, select both CapAll and ACX.

Note The CapAll and ACX options are only available in Portal Studio from Portlet Builder. These options are not selectable from within Portal Interface. Once you select CapAll or ACX for a portlet, the options cannot be unselected, and a portlet rendered with these options will always play back in Portal Interface with these options used.

Using CapAll and/or ACX can cause a significant performance hit during portlet play back, as more interaction is required between the EP server and the Web server from which the content is being captured.

- 5 Click Next.

Note When you create a portlet using one-click capture, and the portlet content contains numerous links, some of the links may not work properly due to underlying JavaScript or redirect code in the link content.

- 6 Click the left mouse button in a suitable area, indicated by a shadow-box that trails the mouse arrow and says “Select it”.
- 7 The capture option window displays the “best guess” capture from the specified area.

Click the light blue button to the left of the option you want to select. The button changes to a darker blue when you select it. Click Next.

- 8 Enter a name for the element, then click Finish. You return to the Portal Builder where the new element displays.

Note Regardless of the element type, the procedure to save the element and portlet is the same. See “Saving portlets” on page 82.

Creating XML elements

An XML portlet is an XML-based rather than HTML-based. This type of portlet is generally used by portal administrators who are familiar with XML to XSLT transformations.

❖ **Creating a new XML portlet**

- 1 Select XML Element. You see the XML Element Definition window.
- 2 Enter:
 - XML URL – the base URL of the XML source; for example, the following URL, entered as one line, creates a new Common Gateway Interface (CGI) parameter named “Category” with a default value of “Biotech.” CGI Parameters are discussed in “Using portlet parameters” on page 166.

```
http://p.moreover.com/cgi-local/page?c=Biotech%20news&o=xml
```

- Content XSLT – click Select to search for and choose the extended XSL template (XSLT) that you want applied to this portlet’s content; for example:


```
moreover_content_xsl
```
 - UI XSLT – click Select to search for and choose the XSL template (XSLT) that you want applied to the user interface; for example:


```
pink_ui_xslt
```
- 3 When you are satisfied with your entries, click Next. You see a window that allows you to split the columns or rows in the table. See “Adding split rules” on page 175.
 - 4 Click Next. Define how you want the records in the table to display. See “Defining record layout” on page 176.
 - 5 Click Next. Specify which rows and columns you want to filter out of the display. See “Using filter rules” on page 177.
 - 6 Click Next, then enter a name for the element.
 - 7 Click Finish. You return to the Portal Builder.

Regardless of the element type, the procedure to save the element and portlet is the same. See “Saving portlets” on page 82 for instructions.

Note See Chapter 10, “Building Pages,” for instructions on adding portlets to pages.

Creating database elements

An database element allows you to query a back-end database and display the results in a portlet.

❖ Creating a database element

- 1 Log in to Portal Studio, select Build | Portlets, and right-click in the detail view and select New Aggregated Portlet.
- 2 When the Portlet Builder appears, right-click in either frame and select New Database Element from the Element Menu. You can also click the down arrow next to the Add icon and select Database Element.
- 3 Complete the Database Element Definition.

Note This example uses an Adaptive Server Anywhere database.

- Username – enter the user name used to access the database; for example, to access Adaptive Server Anywhere, the default user name is:

dba

- Password – enter the password used to access the database; for example, the default password for the Adaptive Server Anywhere database is:

SQL

- JDBC Connect URL – enter the connection string to access the database; for example:

```
jdbc:sybase:Tds:localhost:port/portaldatabase
```

where *localhost* is the name of installed database server, *port* is the port used to connect to the specified database, and *portaldatabase* is the name of the database to which you want to connect.

For example, if the database server is installed on “tahiti” and the port used to connect to Adaptive Server Anywhere is 6100, you would enter:

```
jdbc:sybase:Tds:tahiti:6100/ASANY8
```

Note If the database server is installed on the same machine as EP, you do not have to change the “localhost” value.

- JDBC Driver – enter the JDBC driver used to access the database; for example:

```
com.sybase.jdbc2.jdbc.SybDriver
```

- SQL query string – enter the SQL statement to retrieve the requested data from the database. For example, the simple query shown below displays everything.

```
select *
```

- UI XSLT – (optional) enter the complete URL of the XSLT to be applied to the content returned by the database or click Select to choose from existing XSL templates.

- 4 Click Preview to see the query results.
- 5 Click Next in the upper-right corner to display the Split window. Split columns or rows using delimiters. See “Adding split rules” on page 133.
- 6 Click Next to display the Define window. Define the record layout and specify a record header. See “Defining record layout” on page 134.
- 7 Click Next to display the Filter window. Specify filter rules to select only the relevant data from the element. See “Adding filter rules” on page 134.
- 8 Click Next. The next page displays the element’s CGI parameters. See “CGI parameters” on page 87 and “Using the database table adapter” on page 158.
- 9 Click Next to display the Element Name input field.
- 10 Enter a descriptive name.
- 11 Click Finish to close the window and display the Portlet Builder with the new database element.

Regardless of the element type, the procedure to save the element and portlet is the same. See “Saving portlets” on page 82.

Note See Chapter 10, “Building Pages,” for instructions on adding portlets to pages.

Creating JSP elements

JSP elements allow you to capture Web applications and display them in a portlet. See Appendix D, “Using the Enterprise Portal Samples,” for examples of using JSP portlets.

Note JSP portlets must be single-element portlets even though you build JSP portlets by selecting Aggregated Portlet.

❖ Creating a JSP element

- 1 In the Portlet Builder, click the down arrow beside the Add button and select JSP Element. You see the JSP Element Definition window.
- 2 Select which option to use for the JSP element:
 - Use Web Application – if the references a Web application archive (WAR) file that is on the application server.
Go to “Using a Web application” on page 56 to complete the procedure.
 - Use Remote URL – if the JSP is not part of a Web application in the same Web container as the portal.
Go to “Using a remote URL” on page 61 to complete the procedure.
 - Use JSP Template – if the JSP element does not reference a WAR file.
Go to “Using a JSP template” on page 60 to complete the procedure.

Using a Web application

- 1 After you select Use Web Application, click Browse to the right of the WAR File to Upload field and navigate to select a local WAR file to upload to the application server.
- 2 When you locate the WAR file, select the file and click Open in the Choose File dialog box.
- 3 When the path to the WAR file displays in the JSP Element Definition window, click Upload. When the confirmation that the WAR file was uploaded successfully displays, click OK. The WAR File and Web App Display Name fields are automatically filled in.

- 4 Click Deploy. When the confirmation that the WAR file was deployed successfully displays, click OK.
- 5 Enter the Initial Resource—the initial JSP page to display.
- 6 Complete these options:
 - Web App Qualified URLs – select this option to enter an URL with parameters to call a Web application from the JSP portlet. See “Using a remote URL” on page 61.
 - Single Sign-On Required – select if single sign-on is required. See “JSP portlet single-sign-on check box” on page 62. When you select this option, three more options become available:
 - JSP Include – use the `javax.servlet.RequestDispatcher.include(...)` method to enable the target JSP to retrieve the JSP’s contents. This method returns the target JSP’s output to the portal and allows you to manipulate, filter, and integrate that output with other elements in a portlet with multiple elements—an aggregate portlet. Aggregate portlets use either an `<IFRAME>` or `<DIV>` tag style.

The target JSP gets a `contextPath` in the `ServletRequest` object that matches the application name under which the JSP is installed.

JSPs using the `.include()` method have a `<BASE HREF=“targetJSP path”>` element inserted into the HTML `<HEAD>` element. This ensures that relative HREFs or IMG links generated from that JSP are loaded correctly.

- JSP Forward – use the `javax.servlet.RequestDispatcher.forward(...)` method to enable the target JSP to retrieve the JSP’s contents. This method does not return the target JSP’s output back to the portal.

This portlet style must run in an `<IFRAME>` tag and cannot be included in an aggregate portlet (portlet with multiple elements). The `forward` method ensures that the target JSP receives the `contextPath` with which it was installed. There is no opportunity for the portal to insert the `<BASE HREF=...>` element into the returned HTML, so if the `<BASE HREF=...>` information is not provided by the portlet, relative HREFs or IMG links may not work.

- 302 Redirect – this option generates a HTTP Status 302 redirect back to the browser. When the browser receives a 302 redirect status, it executes a get method to the URL contained in that 302 message. This portlet style must run in an <IFRAME> tag, otherwise the portlet overlays the entire screen.

One benefit of this portlet style, is that the declarative J2EE security constraints that are part of the target Web application are enforced by the container. The J2EE standard does not require the container to apply declarative security constraints for RequestDispatcher include or forward.

See Appendix D, “Using the Enterprise Portal Samples,” for an example of creating a single-sign-on portlets.

Note Web applications in J2EE containers can have security descriptors that are mentioned in *web.xml*. This section of the XML puts a constraint on all the resources of the Web application. Access is granted only to users who are in the manager role. Users who try to access the Web application from the browser (or other client program) must be authenticated. If the type of authentication required is basic authentication, the browser prompts the user for a user name and password.

If the Web application is added as a portlet to a page, users need not reauthenticate with the application. The portal uses container authentication to authenticate users when they log in to the portal. The same session context is used to authenticate the user with the Web application that is used to authenticate the user with a portlet. If the user roles are same across the Web application, the user is authenticated into the Web application and the contents display in the portlet.

See the *Enterprise Security Administration Guide* for details.

- 7 Input Parameters – enter the input parameter names as defined by the JSP code.

- a You must use a name–value pattern like this:

```
name1=value1&name2=value2&name3=value3
```

- b The name must be exactly the same as your JSP code’s input parameter’s name. For example, if the JSP page’s code is:

```
EPLogin:<input type="text" name="epuser">
EPPswd:<input type="password" name="eppassword">
```

```

HostName:<input type="text" name="hostname">
AssetDN:<input type="text" name="asset">
PAI URL:<input type="text" name="paiURL">

```

then the Input Parameters must be:

```

epuser=tester&eppassword=test123&hostname=drillbit2k&asset=a1=pubs,
dc=sybase,dc=com&paiURL=jdbc:sybase:Tds:sapphire:5000/pubs2

```

Including the Secure Web Proxy in a JSP portlet

The Secure Web Proxy has been enhanced to allow integration into Portal Interface as a portlet. The Secure Web Proxy should be deployed and configured as a standalone Web application on EAServer, leaving the application server connection information blank within *swp.properties*. Once you verify that the Secure Web Proxy is configured and functioning standalone, you can create a portlet using Portal Studio that includes a JSP element including the desired Secure Web Proxy content. To create this type of portlet, set the following parameters in the JSP Element Definition window:

- 1 In the JSP Element Definition window, select Use Web Application, click Browse to the right of the WAR File to Upload field and navigate to select *%SYBASE%\Security\lib\webproxy.war* on Windows and *\$SYBASE/Security/lib/webproxy.war* on UNIX.
- 2 When you locate the WAR file, select the file and click Open in the Choose File dialog box.
- 3 When the path to the WAR file displays in the JSP Element Definition window, click Upload. When the confirmation that the WAR file was uploaded successfully displays, click OK. The WAR File and Web App Display Name fields are automatically filled in.
- 4 In the Initial Resource field, enter:

```
proxy/[application name]/
```

Replace *[application name]* with the configured application's name from *swp.properties* that you want to have appear in the portlet.

- 5 Select Single Sign On Required and 302 Redirect.
- 6 When you are satisfied with your entries, click Next.

Once this configuration is complete, you should see the appropriate content appear in the preview window.

- 7 Enter an element name, then click Finish. You return to the Portlet Builder.

Regardless of the element type, the procedure to save the element and portlet is the same. See “Saving portlets” on page 82 for instructions.

See *security.log* and the various Jaguar logs for error and diagnostic information. See also the *Enterprise Security Administration Guide*.

Using a JSP template

- 1 Use JSP Template – select if the JSP element does not reference a WAR file.
 - JSP Template – select a JSP template using the Find Template window.
- 2 Input Parameters – enter the input parameter names as defined by the JSP code.

a You must use a name–value pattern like this:

```
name1=value1&name2=value2&name3=value3
```

b The name must be exactly the same as your JSP code’s input parameter’s name. For example, if the JSP page’s code is:

```
EPLogin:<input type="text" name="epuser">  
EPPswd:<input type="password" name="eppassword">  
HostName:<input type="text" name="hostname">  
AssetDN:<input type="text" name="asset">  
PAI URL:<input type="text" name="paiURL">
```

then the Input Parameters must be:

```
epuser=tester&eppassword=test123&hostname=drillbit2k&asset=a1=pubs,  
dc=sybase,dc=com&paiURL=jdbc:sybase:Tds:sapphire:5000/pubs2
```

Using a remote URL

- 1 Use Remote URL – select this option if the JSP is not part of a Web application in the same Web container as the portal.

Note Since that JSP is not deployed in the portal Web container, you do not need to fill in the fields associated with Use Web Application option. You also cannot select Single Sign-on.

- Remote URL – specify the fully-qualified URL to the remote JSP, starting with HTTP(S).
- 2 Input Parameters – enter the input parameter names as defined by the JSP code.
 - a You must use a name–value pattern like this:

name1=value1&name2=value2&name3=value3

- b The name must be exactly the same as your JSP code’s input parameter’s name. For example, if the JSP page’s code is:

```
<BR>EPLogin:<input type = "text" name="epuser">
<BR>EPPswd:<input type="password" name="eppassword">
<BR>HostName:<input type="text" name="hostname">
<BR>AssetDN:<input type = "text" name="asset">
<BR>PAI URL:<input type="text" name="paiURL">
```

then the Input Parameters must be:

```
epuser=tester&eppassword=test123&hostname=drillbit2k&asset=a1=pubs,
dc=sybase,dc=com&paiURL=jdbc:sybase:Tds:sapphire:5000/pubs2
```

- 3 When you are satisfied with your entries, click Next.

Note The Preview button does not work for JSP portlets.

- 4 Enter an element name, then click Finish. You return to the Portlet Builder. Regardless of the element type, the procedure to save the element and portlet is the same. See “Saving portlets” on page 82 for instructions.

JSP portlet single-sign-on check box

When you create a JSP portlet from a WAR file in Enterprise Portal version 5.1 and later – Enterprise Edition, there is a check box labeled Single Sign On (SSO). If you do not select this option, the implications are:

- The JSP portlet is invoked without any credentials passed to it.
- If the JSP portlet has the directive `<% page session="true" %>` in the JSP page, a new HTTP session is created whenever the JSP portlet is invoked.
- A cookie is sent back to the portal, but the cookie is not remembered. Therefore, any state stored within the HTTP session is lost.
- Since the HTTP session does not go away quickly—the default is 30 minutes in EAServer—repeated use of the JSP portlet can create a severe memory drain to the system.

Creating Web service elements

A Web service is software that is available over the Internet and uses a standardized XML messaging system to invoke the service and supply the response. Web services are a useful way to provide data to a variety of consumers; for example, traffic reports, stock quotes, and travel services. This section provides basic introductory material about Web services and describes how to implement Web services in your portal. This section has these topics:

- “Using Enterprise Portal Web services” on page 65
- “Creating Web service portlets” on page 69
- “Creating new UDDI registry entries” on page 73

Note This section provides an overview of Web services. It does not provide the detailed information necessary to fully understand and use Web services and Web service protocols. You can find detailed information at <http://searchwebservices.techtarget.com/> and at <http://www.w3.org/2002/ws/>.

Web services are software components delivered over the Internet or intranet networks. Unlike traditional client/server models, such as a Web server/Web page system, Web services do not provide the end user with a graphical user interface (GUI). Web services instead share business logic, data, and processes through a programmatic interface across a network. Applications interface with each other, not the users. Developers can add a Web service, such as a Web page or an executable program, to a GUI to offer specific functionality to users.

Web services simplify and standardize application interoperability within and across companies. Web services allow applications to communicate with each other in a platform- and programming language-independent manner.

Each service is independent, and you can stack services like building blocks. You can, therefore, create a single page that displays multiple services where the end result looks like a stripped-down version of my.yahoo or my.excite.

Web service architecture has three primary functions—discovery, description, and transport. For each of these functions, there is an accompanying standard based on XML.

Web services are discovered (published and found) via Universal Description, Discovery and Integration (UDDI), described by Web Services Description Language (WSDL), and transported using Simple Object Access Protocol (SOAP). Web service transactions run over HTTP and TCP/IP networks.

Web services standards

A Web service describes a collection of operations that can be accessed over the Internet using standardized XML messaging. It uses protocols based on XML to describe an operation to execute or data to exchange with another Web service.

Web services use standards—groups of related protocols—to describe, deliver, and interact with other Web services. You should understand the restrictions of each standard, specifically SOAP, WSDL, and UDDI.

- Simple Object Access Protocol (SOAP) – allows communication between Web services. Encodes messages so they can be delivered over the network using a transport protocol such as HTTP, IIOP, SMTP, and so on.

At a basic level, SOAP is the equivalent of an empty envelope that allows you to put an electronic business document inside the SOAP envelope to create a packet that delivers an electronic message.

The SOAP envelope has two parts—a header, which contains information such as an electronic address, and a body that contains the business document.

SOAP does not:

- Provide address standards for what an electronic address should look like.
- Provide message security or authentication to determine who sent the message.
- Support long transactions. SOAP only allows you to define the way you interact with a Web service as a request or a response.
- Offer reliable delivery that ensures the message was delivered.
- Web Services Description Language (WSDL) – a series of XML statements that form the definition of each Web service interface. WSDL is the standard language for describing Web service interfaces. WSDL by itself only describes the pattern of communication between the parties involved and must be extended with Web service type and binding information. The type information describes exactly what must be communicated, and the binding information details exactly how it must be communicated.

WSDL does not allow you to combine Web services or explain how to combine multiple services with their individual requests and responses into a more complex composite application.

- Universal Description, Discovery, and Integration (UDDI) – defines a directory of businesses and the Web services they provide. Users can access the directory, discover services that exist, and download and use the WSDL definition for the Web services they require.

Some UDDI directories are public; that is, anyone can review and use the Web services they find there. Other UDDI directories are private; for example, Web services for a specific enterprise.

Warning! Public UDDIs may contain inaccurate or incorrect data.

Examples

A list of publicly available Web services can be found at <http://www.xmethods.com/>, including a stock-quote service, a traffic-report service, and a weather service.

Each service is independent, reusable, and performs a distinct task. These services can be linked together across or within enterprises using XML. you can stack services like toy blocks, which lets you build applications from Web services quickly, borrowing blocks from UDDI registries when you need them.

For public Web services, the WSDL file is typically available on the Web site of the organization that publishes the service. For private Web services, contact the organization that supports the Web service to obtain the WSDL file.

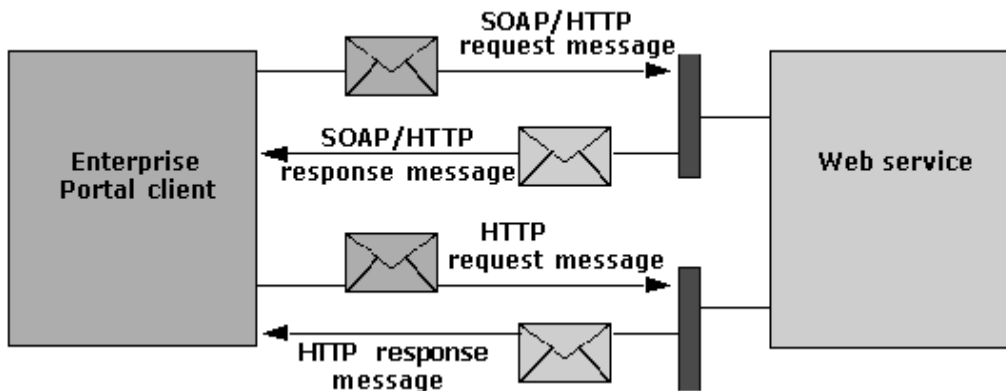
You can also find WSDL files through both public and private UDDI registries. To learn more about UDDI, see the information at <http://www.uddi.org>.

Note Sybase cannot guarantee that publicly-available Web services will remain static. If you try to use a public Web service and it no longer functions, contact the Web service provider to verify that the service has not changed.

Using Enterprise Portal Web services

Enterprise Portal has implemented Web service support using Apache SOAP 2.3.1, and has tested it with the client side of the same version. EP has also successfully performed compatibility tests with Apache Axis.

Figure 4-5: Web services illustration



Enterprise Portal allows you to create Web services elements and incorporate those elements into portlets. An example of a Web service might be weather conditions that are returned based on postal code, or stock quote results based on company symbols.

You create Web service elements that use standard WSDL files or service keys (globally unique identifiers that refer to services published in a UDDI Business Registry), then add the elements to portlets that execute local or remote Web services to extract and display portlet content.

Enterprise Portal also lets you create Web services portlets and publish the portlets as Web services to any of these UDDI registries—IBM test, IBM public, TrySybase, and Microsoft test.

Requirements

To use Web services with Enterprise Portal:

- You must use the version of *global.properties.xml* installed with Enterprise Portal version 6.0 and later that includes the `WebServiceRoot` property in the `UWP` property group.
- Enterprise Portal comes preconfigured to use Web services. To verify or change Web services configuration settings:
 - a Beneath your Enterprise Portal installation directory, go to the onpage Web application configuration directory
 - `%SYBASE%\EAServer\Repository\WebApplication\onpage\config` on Windows

- `$$SYBASE/EAServer/Repository/WebApplication/onepage/config` on UNIX

Open `webservices.properties.xml` in Microsoft WordPad or the text editor of your choice on other platforms.

- b The `ArraySize1`, `WSCache`, and `DefaultExpireLength` properties are set on installation, but you can change them if necessary.

```
<Properties name="Properties">
<Property name="ArraySize1" value="2"
description="default array size"/>

<Property name="WSCache" value="true"
description="(true/false)' true to turn on
caching for processed complexTypes and
definitions of wsdl"/>
```

```
<Property name="DefaultExpireLength"
value="259200000" description="default expire
length for cached WS element, e.g., 3 days"/>
```

- c If you change any of the configuration properties, restart the application server to put your changes into effect.
- If you get a Secure Sockets Layer (SSL) certificate error while trying to publish a Web service to a registry using HTTPS, use one of these options to resolve this:
 - Install a new version of Java (for example, JDK 1.4.1) and run Enterprise Portal with that version. The root certificate on the registry site should already be installed in your default keystore file—`cacerts`.
 - Use your browser to download a root certificate (issued from Verisign), then use `keytool` to import this certificate to your default keystore (`cacerts`) or a newly created keystore. If you import the certificate to a new keystore, you must also point the client keystore property to the new keystore:

```
[System.setProperty("javax.net.ssl.keyStore", "...")]
```

- Register accounts with the UDDI public registries, then publish your Web service.

Note The TrySybase registry does not provide a site to register an account; use `guest` (user name) and `guest` (password) as the account login to publish to the Sybase registry.

Complex Web service types

WSDL documents can use the `<types>` element as a container for datatype definitions that are relevant for exchanged Web services messages. For maximum interoperability and platform neutrality, WSDL uses the XML Schema Definition (XSD) language for datatype definitions. For example:

```
<definitions .... >
  <types>
    <xsd:schema .... />*
  </types>
</definitions>
```

Before Enterprise Portal version 6.0, when you created Web service portlets or elements, the portal could not process WSDL files that used the `complexType` datatype format in the schema definition, resulting in the unsuccessful invocation of the Web service.

Enterprise Portal 6.0 and later can parse all WSDL files that use the `complexType` data type and other compound datatypes using the format that complies with WC3 XML Schema specification.

Note Parsing of Web services WSDL takes longer the first time the WSDL is accessed.

You can define these datatypes within the `<types>` element to represent all or part of input or output messages. When you create Web services portlets and elements, you must provide inputs to these fields to generate input messages for the Web service's invocation.

Note Verify that you can execute the Java compiler command (`javac`) at the command line from anywhere on the local machine.

Creating Web service portlets

When you create a Web service portlet, you must have access to where that service lives on the Internet. EP allows you to identify Web services that you want to use in a portlet either of two ways—using a service key or using WSDL.

- Service key – is a globally unique identifier that refers to a service published in a UDDI Business Registry.
- WSDL – a normalized description of disparate software that allows users to access the software in a manner independent of protocol or location.

❖ Creating a Web service element using WSDL

Use this procedure to create a Portal Studio element that uses the standard Web Services Description Language (WSDL).

- 1 Specify the location in which the Web service files will be generated and from where they are exposed.
 - a Open *global.properties.xml* in WordPad on Windows or in any text editor on UNIX.
 - b Locate the `WebServiceRoot` property in the UWP property group and modify the default value if necessary. There are two ways to specify where to put the generated Web service files:
 - Enter the absolute location; for example, if you enter *d:\work\service\ws* on Windows or */work0/service/ws* on UNIX, the system adds files directly to the specified location. Ensure that the location can be accessed from a URL, such as *http://localhost:8085/onepage/ws*.
 - Enter the relative location. For example, if you enter *service\ws* on Windows or *service/ws* on UNIX, the system adds files to the location relative to the current Web application; specifically, *\work0\tomcat\webapps\onepage\service\ws* on Windows or */work0/tomcat/webapps/onepage/service/ws* on UNIX. Ensure that the location can be accessed from a URL, such as *http://localhost:8085/onepage/service/ws*.
- 2 Log in to Portal Studio and select Build | Portlets from the left pane.
- 3 Click the down arrow beside the New icon and select New Aggregated Portlet.
- 4 When the Portlet Builder displays, click the down arrow beside the Add button, and select Web Service Element.

You see the Web Service Element Definition window.

5 Create a Web service element using WSDL URLs:

- UDDI Operator – accept the default. This field is not used when you are using WSDL.
- Services – select WSDL, then enter a Web service WSDL location in the provided text field and press Enter; for example:

```
http://www.xmethods.net/wsdl/query.wsdl
```

Note This is a sample Web service that provides a SOAP remote procedure call (RPC) interface to the xmethods.net Web site for query operations. xmethods.net provides a listing of publicly available Web services.

6 Click GetMethod. The system retrieves the WSDL XML. The list of defined services is extracted from the WSDL file and populates the Method drop-down list.

Select a method from the drop-down list.

7 Click Preview. The system lists the input parameters of the selected Web service method in the preview pane.

8 Click Next and enter an element name.

9 Click Finish.

❖ **Creating a Web service element using a service key**

This procedure lets you use a globally unique identifier (service key) to reference a Web service in the UDDI Registry. Service keys are hexadecimal strings generated when a Web service is registered with a UDDI registry. For example, when you search the Web service provider at www.bindingpoint.com for UDDI Web services registries, you see the service key in addition to other information about the service.

```
Search Rated UDDI
```

```
Description: Search UDDI using rated and validated
information only. This service looks at all UDDI entries
and allocates a weighting to each record depending on
validity and completeness. You can then search UDDI
registry and retrieve only valid records.
```

```
Service Key: 5ad60ec4-846a-4517-b391-724bcdbdbe26
Provider: Lucin
```


Date Added: Aug 12, 2002
Average response time: 0.32 seconds
Price (per transaction): Free

- 1 Log in to Portal Studio.
 - a Open Internet Explorer, enter the Portal Studio URL in the Address entry field, and press Enter.
 - b When the Portal Studio Login window displays, enter your user name and password, then click Login.

You see the Portal Studio Welcome window.

- 2 Select Build | Portlets from the Portal Studio left pane.
- 3 Select New from the Portlet manager toolbar to create an Aggregated Web services element.
- 4 When the Choose Portlet Type window displays, select Web services, then click Next.
- 5 At the Web Services Element Definition window, create a Web service element using a service key.
 - UDDI Registry – from the drop-down list, select the UDDI registry that specifies where to look for the key definition.
 - Services – select Key, then enter the Web services key in the provided text field and press Enter; for example:

```
8BD03750-BBE2-11D6-B8B8-000629DC0A7B
```

You obtain this key from the UDDI Registry where the associated Web service is published.

- 6 Click GetMethod. The system retrieves the WSDL XML. From the WSDL, the list of defined services is extracted and populates the Method drop-down list.

Select a method from the drop-down list.
- 7 Click Preview. The system extracts the input parameters of the selected Web service method and lists them in the preview pane.

❖ **Publishing a portlet to a UDDI registry as a Web service**

- 1 Log in to Portal Studio and select Build | Portlets.
- 2 In the detail window, select the Approved status, then select the portlet that you want to publish as a Web service.

3 Click Publish to UDDI on the Portlet Manager toolbar. You see the Publish to UDDI window.

4 Complete the fields in the Publish to UDDI window:

- Service Name – enter the name by which this portlet is to be published; for example, “Welcome.”
- Provider Name – select or define a new Web service provider under whose name the portlet Web service is to be published. To update or create a new provider, click Details.

When you define a new provider, you see the window where you can enter or update the information about the provider.

a Enter or update the information on the Provider Details window:

- Provider Name – the provider of the Web service.
- Person Name – the contact person.
- E-mail Address – the provider’s e-mail address.
- Provider Description – optional description of the provider.

b Click OK.

5 Complete the remaining fields on the Publish to UDDI window:

- Category – select a category from the drop-down list for the Web service.
- UDDI Registry – select from a list of preconfigured UDDI Registries. See “Creating new UDDI registry entries” on page 49 for instructions on adding new entries to the registry list.
- User name – enter the user name used to access the UDDI registry.
- Password – enter the password used to access the UDDI registry.
- Respond with HTML – select this option to receive a response in HTML rather than an XML format.
- Remember Settings – select this option to have the system remember the settings on this page for this session.

6 Click OK.

The system generates the Web service definition interface, the implementation WSDL files, and the deployment file in the directory specified for the WebServiceRoot property entry in *global.properties.xml*. You may have to wait for several seconds before the system responds.

Creating new UDDI registry entries

Note The current Web service implementation in Enterprise Portal version 6.0 uses Apache SOAP 2.3 and UDDI4J version 2.0.

When you create a new Web services element and select a UDDI registry from which to select a service, or enter information to publish a Web service to a UDDI registry, you see a drop-down list. The entries on these lists are extracted from the *webservices.properties.xml* file in *EAServer\Repository\WebApplication\onepage/config* on Windows and in *EAServer\Repository\WebApplication\onepage/config* on UNIX. An example of this file follows.

- To add a new UDDI registry from which to extract Web services to use in new Web service elements, use a text editor to add a new `<UDDIRegistry>` section to *webservices.properties.xml*.
- To update the default publishing information with which you want to publish the Web service, use a text editor to update the value for the corresponding `<Property>` to the `<DefaultPublishInfo>` section of *webservices.properties.xml*.

```
<WebServiceProps>
<UDDIRegistry name="IBMTTestRegistry">
  <Property name="displayname" value="IBM Test Registry"/>
  <Property name="id" value="0" />
  <Property name="inquiryURL"
    value="http://uddi.ibm.com/testregistry/inquiryapi"/>
  <Property name="publishURL"
    value="https://uddi.ibm.com/testregistry/publishapi"/>
</UDDIRegistry>
<UDDIRegistry name="IBM">
  <Property name="displayname" value="IBM Public Registry"/>
  <Property name="id" value="1" />
  <Property name="inquiryURL" value="http://uddi.ibm.com/ubr/inquiryapi"/>
  <Property name="publishURL"
    value="https://uddi.ibm.com/ubr/publishapi"/>
</UDDIRegistry>
<UDDIRegistry name="MicrosoftTestRegistry">
  <Property name="displayname" value="Microsoft Test Registry"/>
  <Property name="id" value="2"/>
  <Property name="inquiryURL"
    value="http://test.uddi.microsoft.com/inquire"/>
  <Property name="publishURL"
    value="https://test.uddi.microsoft.com/publish"/>
</UDDIRegistry>
```

```
</UDDIRegistry>
<UDDIRegistry name="Sybase">
  <Property name="displayname" value="TrySybase"/>
  <Property name="id" value="3"/>
  <Property name="inquiryURL"
    value="http://uddi.trysybase.com/juddi/inquiry"/>
  <Property name="publishURL"
    value="http://uddi.trysybase.com/juddi/publish"/>
</UDDIRegistry>
<UDDIRegistry name="QA">
  <Property name="displayname" value="QA Test Registry"/>
  <Property name="id" value="4"/>
  <Property name="inquiryURL"
    value="http://epstnt9.sybase.com/juddi/inquiry"/>
  <Property name="publishURL"
    value="http://epstnt9.sybase.com/juddi/publish"/>
</UDDIRegistry>

<DefaultPublishInfo name="DEFAULT">
  <Property name="provider_name" value="Sybase EP Services"/>
  <Property name="person_name" value="John Smith"/>
  <Property name="email_address" value="js@sybase.com"/>
  <Property name="provider_desc" value="Sybase EP Web Services"/>
  <Property name="username" value="sybase00"/>
  <Property name="password" value="sybase"/>
  <Property name="respond_with_html" value="false"/>
  <Property name="remember_settings" value="false"/>
</DefaultPublishInfo>
</WebServiceProps>
```

Creating HTML-based elements

HTML elements are pieces of HTML code that you can arrange within a portlet with other element types.

❖ Creating an HTML element

- 7 Select HTML Element. You see the New HTML Element window.
- 8 Complete these options:
 - Name – enter a name for the element.
 - HTML – enter the HTML code; for example:

```
<IFRAME STYLE="width:100%; height:552;"  
SRC="http://investor.stockpoint.com/marketindex.asp">  
</IFRAME>
```

- Preview – previews the rendered HTML.
- 9 When you are satisfied with your entries, click Save. You return to the Portlet Builder.

Note Regardless of the element type, the procedure to save the element and portlet is the same. See “Saving portlets” on page 82.

Creating document elements

The document element allows you to load an application document (Microsoft Word, Microsoft Excel, PDF, and so on) into a portlet.

When you create document elements:

- Portlets built with the document element may require that the Web browser plug-in for the application in which the document was authored be installed on the machine from which the user is browsing before they can view the portlet’s content. For example, to view a portlet with PDF content, the Adobe Acrobat Reader plug-in for your Web browser must be installed on the machine from which you are browsing. Refer to your Web browser’s Internet site for more information.
- If you upload a document with the same name as a previously uploaded document, the second document overwrites the first, unless you create a subdirectory below the upload directory and upload the second document to the subdirectory.
- Document elements must be saved with the IFRAME attribute.
- You can use only the default template except for the content types text/HTML and text/plain.
- You cannot import or export a document portlet across servers.
- You can create a document portlet with a new content type that is not included in the drop-down menu. In this case, it may or may not work, depending on whether there is an application supports the plug-in.

- You must set the content type of document element portlets to the correct type.
- Portlets that use the document element (except for the content types text/HTML and text/plain) can contain only one element.

❖ **Creating a document element**

- 1 Select Document Element. You see the Document Element Definition window.
- 2 In the Document to Upload field, enter the full path to the document you want to upload, or click Browse to search for and select the document.
- 3 In the Destination Location field, accept the default folder name or append the default name with an additional folder name. Enter only the folder name; that is, do not enter the complete path. This is the location to which the document is uploaded.

Note The default upload folder name is configurable in *global.properties.xml*. See Appendix A, “Configuring Global Properties.” Also, the parent location of the document root folder must have “write” permissions.

Portal Studio creates a document root level directory on the drive of your EP installation and uses the current user’s login name as the default folder. For example, if the default for the Destination Location displays “Smith”, and the EP installation drive is D:, the uploaded document is saved to *D:\tmp\PortalDocs\Smith*.

To create a different folder using this structure, enter another file name after the name that displays in the Destination Location field. For example, if the field displays “Smith” and you enter “\Smith,” the file is uploaded to *D:\tmp\PortalDocs\Smith\Smith*.

- 4 Click Upload. When a message displays confirming that the upload was successful, click OK.

Note The length of time before the confirmation message displays depends on the document’s size.

- 5 Document Name – displays the document’s name.
- 6 Content Type – from the drop-down list, select the type of content the document contains.

If you do not see the content type you want, select Add New from the drop-down list to add a new content type.

- 7 New Content Type – displays only when you select Add New from the Content Type drop-down list. Enter the new type in this format:

```
[application | image | text]/plug-in name
```

Note Whether the new type works depends on whether there is an application and the plug-in installed for the content type specified. See “Adding new content types” on page 57.

- 8 Click Next.
- 9 Enter a name for the element and click Finish. You return to the Portlet Builder.

Note Regardless of the element type, the procedure to save the element and portlet is the same. See “Saving portlets” on page 82.

Adding new content types

When you create a document element and add a new content type, you must enter the name and type of the plug-in that allows you to view the document in a Web browser. The plug-in must also be registered with the Web browser. Registering plug-ins or helper applications is done very differently for Internet Explorer and Netscape.

❖ Registering new content types with Internet Explorer

- 1 Go to <http://support.microsoft.com>.
- 2 Enter 306790 in the Search the Knowledge Base field, and click the Go arrow.

When a link to Microsoft Knowledge Base Article - 306790 “Description of Internet Explorer Support for Netscape-Style Plug-ins,” click the link.

This document describes how to register content types with Internet Explorer and how to specify the application associated with the document you want to view in the Web browser.

❖ **Adding a new plug-in in Netscape**

- 1 In Netscape, select Edit | Preferences.
- 2 When the Preferences window displays, select Navigator | Applications if you are using Netscape 4.76. If you are using Netscape 7.0, select Navigator | Helper Applications.
- 3 To add a new helper application (plug-in) to the list, click New Type (New on UNIX).
- 4 When the New Type window displays:
 - a Enter a description of the type of files the application works with; for instance, “movies” or “sound files.”
 - b Enter the file extension (suffix) used by this file type. If this type of file uses more than one suffix, enter them all, separated by spaces. Examples: text documents often have the suffix “txt”; some image files have the suffix “gif”; and HTML documents have the suffix “htm” or “html.”
 - c Enter the MIME type. MIME types consist of two words separated by a slash (/). For instance, a TIFF image file’s MIME type is “image/tiff”. If you do not know the MIME type, leave the field blank.
 - d On Windows, click Browse and select an application that handles files of this type. On UNIX, click Application, then click Choose.
 - e Click OK.
- 5 You can specify how Navigator should handle downloaded files. Select a file type from the list and click Edit.
 - a Choose a “Handled By” option to open downloaded files in a Navigator window, save them on your hard disk, or open them in a application that you specify. Click Browse to specify the application.
 - b Check “Ask me before opening downloaded files of this type” if you want to be notified before downloading files that are handled by applications other than Navigator.
 - c Click OK.

Building secure portlets

Use the Secure Portlet Builder to create and edit Web elements captured from secure sites (URLs that begin with `https://`). Select New Secure Portlet from the New drop-down menu or the right-click menu in the Portlet Manager to launch the Secure Portlet Builder.

The primary type of secure implementation on the Internet is SSL version 3. The portal handles SSL3 using the Java Secure Sockets Extension (JSSE) implementation, which is included in Enterprise Portal. EP does not work with SSL2 sites.

Note The portal handles secure portlets in a catalog completely differently. See Chapter 9, “Building Catalogs,” for more information.

For more information about enterprise-wide security, see the *Enterprise Security Administration Guide*.

Setting up Enterprise Portal to create secure portlets

This section describes the steps required to set up Secure Sockets Layer (SSL) and to turn on portal security parameters that enable the creation of secure portlets.

❖ Setting up SSL for secure portlet creation

- 1 Copy `jnet.jar`, `jcrt.jar`, and `jsse.jar` from `$$SYBASE/EAServer/Webservices/java` to `$$SYBASE/EAServer/java/lib` on UNIX and from `%SYBASE%\EAServer\Webservices\java` to `%SYBASE%\EAServer\java\lib` on Windows.
- 2 Modify `%JAGUAR%\bin\serverstart.bat` (Windows) or `$JAGUAR/bin/serverstart.sh` (UNIX) as follows:
 - a Add these lines at the beginning of Set `EAS_CLASSPATH_P1`.

Windows:

```
set
EAS_CLASSPATH_P1=%EAS_CLASSPATH_P1%;%JAGUAR%\java\lib\jsse.jar

set
EAS_CLASSPATH_P1=%EAS_CLASSPATH_P1%;%JAGUAR%\java\lib\jnet.jar

set
```

```
EAS_CLASSPATH_P1=%EAS_CLASSPATH_P1%;%JAGUAR%\java\lib\jcert.jar
```

UNIX:

```
set
EAS_CLASSPATH_P1=$EAS_CLASSPATH_P1;$JAGUAR/java/lib/jsse.jar

set
EAS_CLASSPATH_P1=$EAS_CLASSPATH_P1;$JAGUAR/java/lib/jnet.jar

set
EAS_CLASSPATH_P1=$EAS_CLASSPATH_P1;$JAGUAR/java/lib/jcert.jar
```

- b Add these lines at the beginning of Set EAS_BOOTCLASSPATH_P1:

Windows:

```
set EAS_BOOTCLASSPATH_P1=%EAS_BOOTCLASSPATH_P1%;
    %JAGUAR%\java\lib\jsse.jar

set EAS_BOOTCLASSPATH_P1=%EAS_BOOTCLASSPATH_P1%;
    %JAGUAR%\java\lib\jnet.jar

set EAS_BOOTCLASSPATH_P1=%EAS_BOOTCLASSPATH_P1%;
    %JAGUAR%\java\lib\jcert.jar
```

UNIX:

```
set EAS_BOOTCLASSPATH_P1=$EAS_BOOTCLASSPATH_P1;
    $JAGUAR/java/lib/jsse.jar

set EAS_BOOTCLASSPATH_P1=$EAS_BOOTCLASSPATH_P1;
    $JAGUAR/java/lib/jnet.jar

set EAS_BOOTCLASSPATH_P1=$EAS_BOOTCLASSPATH_P1;
    $JAGUAR/java/lib/jcert.jar
```

- 3 In a text editor, open *java.security*, which is located in *%SYBASE%\Shared\jdk1.3.1_06\jre\lib\security* on Windows and in *\$SYBASE/Shared/jdk1.3.1_06/jre/lib/security* on UNIX, and add this line:

```
security.provider.2=com.sun.net.ssl.internal.ssl.Provider
```

❖ **Enabling portal security**

The final step in setting up the portal to create secure portlets is to enable the portal's security parameters in the *global.properties.xml* master configuration file.

- 1 In a text editor, open *global.properties.xml*, which is located in the *Repository/WebApplication/onepage/config* subdirectory of your EAServer installation.

- 2 In the Global Properties group, locate the “secure” property and set the value to “on”:

```
<Property name="secure" value="on" description="(on/off). on if HTTPS is enabled for the web-container. HTTPS will be used for login and registration" menugroup="-1" />
```

- 3 Optionally, you can also set the “turnOffSecurePassword” property to “true” so you are not always prompted for another password.

```
<Property name="turnOffSecurePassword" value="@SECURE_PASSSWD_OFF@" description="Turn on/off the usage of the secure password prompting by PI when using http portlets created from Studio. Default is on" menugroup="-1" />
```

- 4 Save your changes in *global.properties.xml* and exit the text editor.
- 5 Restart or refresh EAServer to implement the changes.

Creating secure portlets

❖ Adding a secure portlet

- 1 Select Build | Portlets from the Portal Studio left pane.
- 2 Right-click in the Portlet Manager detail view, and select New Secure Portlet to invoke the Secure Portlet Builder.
- 3 Click the Add icon down arrow and select Web Element.
- 4 When the New Web Element window appears, enter the URL for a secure Web site (for example, *https://www.delta.com*), then click Next.
- 5 Continue with the usual creation of the Web portlet, as described in “Creating an aggregated portlet” on page 22.

Note Unlike other portlets, when you save a secure portlet, it does not automatically appear in the Portlet Manager detail view when you select the New status due to protocol restraints. To view the portlet in the detail view, select a different status, then reselect New. The new secure portlet listing should be visible now.

If you try to share a page with secure content in Portal Interface, you see a message similar to this:

To protect your personal information, Enterprise Portal allows you to only send pages that do not contain secure

portlets. Please share one of your other pages, or remove any secure portlets from this page and try "Share Page" again.

Secure content portlets cannot be shared in Portal Interface.

Note Sybase recommends that you do not mix insecure and secure portlets on the same page.

Saving portlets

Regardless of the element type, the procedure to save a portlet is the same.

❖ Saving portlets

- 1 When you finish creating an element and return to the Portlet Builder, click the down arrow next to the Save button to save the portlet for the first time. You see the New Element Finish window.
- 2 Complete the new element's properties:
 - Portlet Name – enter a unique name for the portlet that will contain this element.
 - Active – this option is selected by default. Deselect the option if you do not want this portlet to be active. Only portlets with an approved status and marked as active are available to portal users.
 - Iframe – select this option to display the portlet in an inline frame that can be placed inside of another HTML document or placed directly on a page with other elements flowing around it.

Note Unlike frames created using <FRAMESET> and <FRAME>, <IFRAME> creates a frame that sits in the middle of a regular nonframed Web page. <IFRAME> works like , only instead of putting a picture on the page, it puts another Web page on the page.

- No Popup – (available only when you select iFrame) select this option to enable links in this portlet to display inline; that is, links will not display in a separate pop-up window.

- Editable – whether the portlet can be edited by the users with the access roles to view the portlet.
- Secure – whether this is a secure portlet.
- Version – select this option to create a new version of the portlet each time you save it. Subsequent versions have new version numbers.
- Parameter – used to create an invisible parameter to which you can assign click-across events.

Note This property does not display in the Portlet Properties window that appears when you click Properties in the Portlet Builder.

- In Context – indicates that the click-thru session should remain in context (that is, continue to use the internal HTTP client to retrieve destination pages pointed to by links on the page), rather than opening a separate browser connection to each link’s target.

Note In Context helps address certain capture problems where there is a session state associated with the played-back page, and opening a new browser connection on link click (that is, not remaining “in context”) results in this state being lost. It may be necessary to experiment with this option, especially if the result of clicking a link on a played-back page fails to return the expected result.

- Title URL – shows the URL of the content you captured.
- Help URL – enter the location of the online help file to associate with this portlet, or click Select to browse to and select the file.
- Content Type – select the type of content this portlet will display.
- Charset – select the character set to use in displaying this portlet’s content.
- Category/Subcategory – select a category and subcategory that best fit this portlet.
- Portlet Refresh Interval – select the interval at which you want this portlet’s content to refresh. The default is five minutes. The browser automatically makes a request for the portlet content when this interval runs down to zero.
 - The default is Daily, which translates into 1440 minutes.

- Specify custom values by selecting Add New in the drop-down list. Specify this value minutes or seconds.
 - Default Size – select the default size at which you want this portlet to display.
 - Content Cache Interval – the interval at which you want the content of a portlet to be refreshed for the defined source. If this value is not Real Time, the content is held by the UWP cache for the duration of the specified interval. All requests made during this interval receive the cached content.
 - The default is Real Time, which means retrieve the portlet content from its source on every playback request.
 - Specify custom values by selecting Add New in the drop-down list. Select the minute/second drop-down list and enter a value in the text box.
 - Show Last Refresh – display the latest refresh image when the portlet displays. By default, this option is turned “on.”
 - Roles – select roles from the Available Roles list and add them to the Assigned Roles list to define what roles can access this portlet. See the *Enterprise Security Administration Guide* for more information on roles.
-
- Note** User accounts with the roles in the Assigned Roles list are the only user accounts that can access this portlet.
-
- Description/Notes – add an optional description or note.
- 3 Click Finish. After a message displays that the portlet was saved successfully, click OK to return the Portlet Builder.

Approving elements and portlets

When you finish creating and saving an element or portlet, Portal Studio automatically gives it a status of New and marked Active. Only portlets that have been marked as approved and active can be assigned to pages and deployed to Portal Interface.

- 1 To change a portlet’s status, select Build | Portlets.

- 2 Select the status of the portlet you want to reassign.
- 3 Right-click the portlet in the detail view, select Status, then select the new status that you want to assign the portlet.

Viewing, editing, and deploying portlets

Portlet management functions appear in windows that you launch from the Portlet Manager toolbar. Management functions include:

- Edit – brings up the Portlet Builder where you can edit existing portlets and their properties and perform a variety of other functions. See “Editing portlets” on page 85.
- Preview – preview any portlet in a separate browser window.
- Find – search for specific portlets or view all active approved portlets. See “Performing a portlet search” on page 91.
- Replace – search for broken portlets and replace them. See “Finding and replacing portlets” on page 92.
- Deploy –
 - Deploy an enterprise archive (EAR) file that contains a WAR file for portlet Web applications and the XML file that contains metadata.
 - Export catalogs, pages, and J2EE roles to Portal Interface on the same file system.
 - Import catalogs and pages from another source.

See Chapter 13, “Importing, Exporting, and Deploying Portal Objects.”

- Publish to UDDI – publish a Web services portlet to a public registry. See “Creating Web service elements” on page 62.

Editing portlets

To edit a portlet:

- 1 Select Build | Portlets from the Portal Studio left pane.
- 2 Select the status of the portlet you want to edit from the Portlet Manager Status menu.

- 3 In the detail view:
 - Right-click the portlet you want to edit and select Edit, or
 - Double-click the portlet you want to edit, or
 - Select the portlet you want to edit and click Edit on the Portlet Manager toolbar.

The toolbar lets you edit portlets, create or change templates and parameters.

Portlet properties

The Properties option lets you view and edit the properties that were specified when the portlet was created. See “Saving portlets” on page 82 for definitions of each field or option.

❖ Viewing or changing portlet properties

- 1 Click Properties on the Portlet Builder toolbar. The Properties Editor window displays, which is similar for all element types.
- 2 You cannot change the Name or Default Size, but you can change any of the other properties and options that display.
- 3 When you finish, click OK.

Portlet parameters

The parameters identified for an element can be configured during portlet creation in one of the element definition windows or after portlet creation using the Portlet Builder Configure Parameter wizard. You launch the wizard by clicking the Params button on the Portlet Builder toolbar.

Portal Studio allows you to identify or create portlet parameters that control the portlet content. These can be CGI script parameters for Web elements or custom @OP parameters for XML and database elements. You can specify these parameters as variables, which causes the portlet to display input fields during playback. End users can submit their own parameter values into these input fields to return personalized content.

CGI parameters

A CGI program, or script, is a way users interact dynamically with Web pages; for example, submitting a string to a search form. When parameters are submitted to a CGI script to navigate to the Web page, the program automatically detects these parameters and their values. You can designate one or more CGI parameters as variables. Portlets with variable parameters display a window containing input fields for these parameters before portlet content is rendered. This allows portal users to submit their own parameter values to return personalized content.

When the element contains data-capable parameters, a window displays during the element creation process that displays all parameters that you can designate as variables.

In addition to specifying variables for an element, you can also select a personalization adapter to use with the parameters. For example, Figure 4-6 shows four parameters for a database element. You can select the username and password parameters as variables, then select the personalization username/password login adapter. The Key represents the value that a user needs to enter.

Figure 4-6: Portlet parameters

Variable	Name	Display Name	Default Value	Type	Personalize	Key	Required
<input type="checkbox"/>	connection	connection	jdbc:zybase:Tds:pqui+ v2k:6100/ASANY8				
<input type="checkbox"/>	adapter	adapter	com.zybase.jdbc2.jdbc.SybDriver				
<input checked="" type="checkbox"/>	password	password		Text Field	username/p...	Login Name	<input checked="" type="checkbox"/>
<input checked="" type="checkbox"/>	username	username		Password	username/p...	Password	<input checked="" type="checkbox"/>

Users can use the My Info link in Portal Interface to personalize the values for these input fields so that they can always log in using their own user name and password. See “Using the user name/password adapter” on page 158 for more information.

Whether you access the parameter configuration windows during portlet creation or by clicking Params, you can edit the following parameter properties.

Note The parameter options display in a different order and on more than one screen when you click Params to access the Configure Parameters window. The order shown here is how the parameter options display during portlet creation.

- Position – use the up and down arrows to organize the parameters in the order you want them to appear in the portlet.
- Variable – select the parameter that you want to configure as a variable. This option is labeled “Configure” in the Configure Parameters wizard.
- Name – displays the parameter name used in the CGI script.
- Display Name – enter an alternate name to use as the input field label.

Note You can assign two or more parameters the same display name to combine input parameters. When you are prompted to enter input for the parameter, that input is used to fill in all the parameters with that display name, saving retyping.

- Default Value – enter a default value to automatically populate the input field. The user can override this value by typing in a different value.
- (Display) Type – select the input field type: Text Field, Password, Text Area, and None. When using the username/password personalization adapter, you must select Password (rather than Text Field) for this option.
- Personalize – personalization adapters retrieve values from external systems or databases and automatically submit them to the portlet. They must be registered with Portal Studio before they appear in the Personalize drop-down. Once a Personalize adapter is selected, the adapter’s available methods display in the Key drop-down.

Key – select a method to specify the parameter within the personalization adapter.

- Required – specify whether the parameter is required.
- Enter Setup Description HTML – this option appears only in the Configure Parameters wizard.

@OP parameters

You can insert an @OP parameter into the SQL query of a database. This dynamically modifies the query string submitted to the database.

❖ **Inserting @OP parameters**

- 1 Create a new database element:
 - Right-click the mouse in the Portlet Manager detail view and select New Aggregated Portlet from the pop-up.
 - Click the down arrow beside the New icon on the Portlet Manager toolbar and select Aggregated Portlet from the drop-down menu.
- 2 When the Portlet Builder appears, right-click in either frame and select New Database Element from the pop-up. You can also click the down arrow next to the Add icon and select Database Element.
- 3 Complete the Database Element Definition:

Note This example uses an Adaptive Server database Enterprise.

- Username – enter the user name to access the database; for example:

```
sa
```

- Password – enter the password to access the database. Leave this field blank for this example.
- JDBC Connect URL – enter the connection string to access the database; for example:

```
jdbc:sybase:Tds:<ASE_server_name>:5000
```

- JDBC Driver – enter the JDBC driver used to access the database; for example:

```
com.sybase.jdbc2.jdbc.SybDriver
```

- SQL Query String – enter the SQL statement to retrieve the requested data from the database; for example:

```
select * from tiles where header like '@OP["Header"="%CNN%"]'
```

To determine other header values, use a wildcard search '%'.

- UI XSLT – enter the complete URL of the XSLT to be applied to the content returned by the database or click Select to select from existing XSL templates.

- 4 Select the Variable option next to the Header parameter and enter “%CNN%” into the Default Value input field.
- 5 Click Next in the upper-right corner.
- 6 Enter an element name such as “DB Search” and click Finish.
- 7 In the Portlet Builder, click Save to display the Portlet Save window.
- 8 Complete the fields and click OK.
- 9 Click Finish in the upper-right corner. You should see a pop-up that displays “Portlet saved successfully.”
- 10 Click OK to close the window, then click Close in the upper-right corner.

Portlet labels

Portlet labels allow you to create custom field labels for data-capable elements. Portlet labels let you combine multiple data-capable elements into a single table, sort records by a particular field, and modify the column order. You can map labels to table columns or rows. A mapped element is created when you map a grid element that has one or more columns mapped to portlet labels.

See “Working with data-capable elements” on page 132.

❖ Creating portlet labels

- 1 In the Portlet Manager, select a portlet listing from the detail view for which you want to add labels and click Edit.
- 2 When the Portlet Builder displays, click Labels on the toolbar.
- 3 Enter the text you want as a label for one of the columns, then click the plus sign to add that label. Continue this process until you have created all the labels you need.
- 4 For the Sort Records By option, select:
 - Sort Records By – to sort the table records by one of the labels listed in the drop-down list.
 - Ascending/Descending – to sort the records in ascending or descending order.
- 5 Click OK to save your entries and close the window.

❖ Assigning portlet labels

- 1 Click the name of a data-capable element in the Element List. This displays the element field labels below the element name and the associated Portlet Label drop-down lists.
- 2 Select a portlet label from the drop-down list associated with an element field. The element is added to a new mapped element within the Element List. The Mapped Element has a cell position drop-down list. You must assign a position to the Mapped Element for it to become visible in the portlet.
- 3 Select a location from the cell position drop-down list for the mapped element and preview the results in the portlet viewer pane of the Portlet Builder.
- 4 Click Save to save the portlet with the new labels.

The mapped elements and grid element columns along with the resulting labels previews in the portlet viewer.

Finding and replacing portlets

This section describes how to search for portlets and how to search for and replace broken portlets.

Performing a portlet search

Find Portlet lets you search for a portlet based on search parameters. The Find Portlet window displays the available search parameters in the left pane and the search results in the right pane.

❖ Performing a portlet search

- 1 Click Find on the Portlet Manager toolbar.
- 2 Enter search parameters. Select Advanced Search to display additional search parameters. Click Clear to remove the current search parameters. If you do not enter parameters, the search displays all portlets with an active state and an approved status.

- 3 Click Search. The Search Results pane displays portlets matching the search criteria you specified. To sort the results, click the heading of the Search Results column you want to sort.

The search results display in sets of 41 records. If there is more than one set of result records, the record sets are numbered and you can navigate to a specific record set by clicking the record set number that displays below the search results.

- 4 Select a portlet from the search results and click either Open or Preview. Open displays the portlet in the Portlet Builder and Preview displays the portlet content in a new browser window.

Finding and replacing portlets

As portlets change, you may need to replace existing portlets with newer versions or different portlet implementations. Use the Find/Replace Portlet window to perform this update.

This functionality is particularly useful to replace broken portlets in your production environment.

Note Using Find/Replace for any portlet with CGI parameters is limited to finding the same portlet with a different version. For example, if portlet A has CGI parameters, you can find or replace another portlet with the same name, portlet A, but the portlet must have a different version to find it using Find/Replace.

❖ Enabling find and replace portlet functionality

Enterprise Portal lets you enable the find and replace portlets functionality for Portal Studio users with the PortalAdmin role, as described in the *global.properties.xml* portal configuration file:

```
<Property name="PortalAdministrationRole"
value="PortalAdmin" description="The J2EE role
required to administer the Portal performing
export/import and update operations." menugroup="-1" />
```

- 1 To grant a Portal Studio user the Portal Admin role, you must be the PSO or have the StudioAdmin role.
- 2 Log in to Portal Studio and select Administer | Organizations from the left pane.

- 3 Select the organization in which the user resides to whom you want to give find and replace access.
- 4 Select Users in the Organization Manager menu.
- 5 Right-click the user in the detail view and select Edit Roles.
- 6 When the Edit User Roles window displays, grant the user the PortalAdmin role, then click OK.

❖ **Finding and replacing portlets**

Users that have the PortalAdmin role and want to find and replace portlets, should follow this procedure.

- 1 Enter this URL in an Internet Explorer browser to access Portal Studio:

```
http://HOSTNAME.PORTALDOMAIN:8080/onepage/index.html
```

For example, if your machine name is “tahiti”, and your portal domain is “sybase.com,” enter:

```
http://tahiti.sybase.com:8080/onepage/index.html
```

If you are running the EP Demo version, enter:

```
http://demo.sybase.com:4040/onepage/index.html
```

- 2 At the Login page, enter your user name and password, then click Login.
- 3 When the Portal Studio main window displays, select Build | Portlets from the left pane on the left, then select Replace from the toolbar above the detail pane.
- 4 Complete these fields:
 - Find Name / Version / ID – select the portlet you want to replace and click Add. You return to the Find / Replace Portlet window with the information filled into the text areas.
 - Replace with Name / Version / ID – select the replacement portlet and click Add. You return to the Find / Replace Portlet window with the information filled into the text areas.

Note Before executing the find and replace operation, the portal checks whether the portlets you specified are using CGI parameters. If the portlets use CGI parameters, both the find and replace portlet must have the same name.

- (From) Active Catalogs – select this option to find and replace the specified portlets in all active catalogs. Active catalogs in Portal Interface display portlets that users can select and add to their own personal pages. If any active catalogs are modified with a new portlet, Portal Interface users catalogs are immediately updated.
 - (From) Non-Active Catalogs – select this option to find and replace the specified portlets in all non-active catalogs.
 - (From) Portal Pages – select this option to find and replace a portlet for all page types—guest, default, and catalog.
 - Update Portal User Pages – when you select the previous option (Portal Pages), select this option to have Portal Interface user page groups automatically updated for all changed portal pages.
 - Agent’s Rule – agent rules evaluate a portlet’s content and determine if the agent should continue. Select this option to find and replace the portlet specified in the agent’s rule. Agents can use one portlet in the processing rule. If a criteria is selected for the rule, no portlet is included in the agent and no update is needed. Chapter 12, “Using Enterprise Portal Agents,” for more information.
 - Agent’s Action – select this option to find and replace the specified portlet in the agent’s delivery action. Agents can deliver or write portlets to a location. These portlets are listed in the agent’s actions.
- 5 To view a log of the changes that would occur as a result of your selections, click Preview.
- Click OK.
- 6 Once you click OK, the Details button is enabled. Click Details to see a detailed log of the changes that would occur:
- 7 Once you preview the changes, click Replace to permanently implement the changes.
- You see an alert message similar to the one that displays when you click Preview.

This chapter discusses advanced functionality that you can implement in the elements and portlets that you develop in Portal Studio.

Topic	Page
Building messaging and application portlets	95
Working with data-capable elements	132
Creating portlet drop-down lists for input fields	136
Enabling portlets and pages to support multiple languages	137

Building messaging and application portlets

This section describes two advanced portlet implementations—click-across messaging portlets that share information with other messaging portlets and application portlets that combine to form comprehensive interactive business applications.

A click-across messaging portlet shares information between itself and other messaging portlets on the same portal page. When you define portlets as messaging portlets, they act together to establish a new form of Web application. See “Building click-across messaging portlets” on page 96 for instructions.

Enterprise Portal lets you build interactive business application portlets to solve multi-staged business problems. Application portlets use information from one portlet for continued processing in another portlet. When these portlets are combined to interact and create collaborative functionality, the portlets become a virtual application. See “Building application portlets” on page 106 for instructions.

Building click-across messaging portlets

You create portlets in Portal Studio from a variety of content sources, including Web, XML, database, JSP, and Web services elements. Portal Interface supports portlets communicating with each other when they are placed on the same Portal Interface page or different pages.

Note EP also supports portlets communicating with each other when they are placed on different pages within the same page group. See “Building application portlets” on page 106 for details.

Portal Studio includes a wizard-driven graphical user interface that allows you to build messaging portlets without programming.

Messaging portlets use a “click-across” mechanism, which uses the publish-and-subscribe model. Portlets that communicate through a publish and subscribe paradigm require the sending portlet (publisher) to publish messages without explicitly specifying recipients or having knowledge of intended recipients. Similarly, a receiving portlet (subscriber) must receive only those messages in which the subscriber has registered an interest.

Note Not all of portlet types can be publisher or subscriber portlets. JSP portlets can only be publisher portlets and Web service portlets can only be subscriber portlets.

A portlet publishes notifications, which are received by a listening portlet subscriber. The publishing entity does not have to be aware of who or what has subscribed to its broadcasts. This allows for a flexible system where a portlet publishes multiple notices and a subscriber subscribes to multiple notices. Additionally, publishers can also be subscribers and subscribers can be publishers. A portlet can listen for an event while broadcasting another event.

When a user clicks an enabled link in a portlet with a defined event, the portlet broadcasts the event and passes along some data, such as the text on which the user clicked or the URL associated with the link. The parent page passes the event notification to portlets on the same or other pages that have registered as listeners for that event.

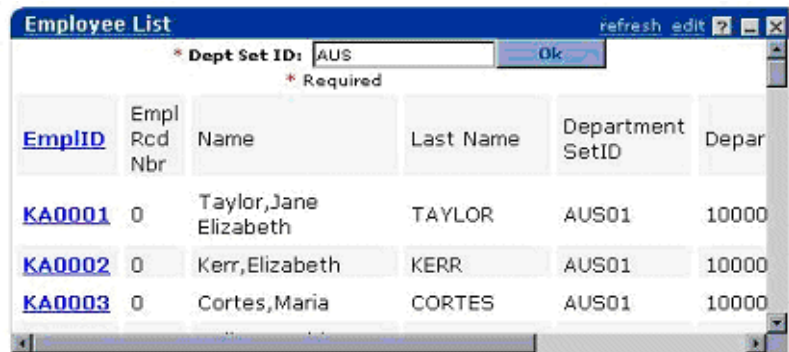
For example, some portlets represent a series of navigation steps that are recorded during portlet creation and executed as a series of HTTP GETs and POSTs during playback. If a portlet contains CGI parameters from any GETs and POSTs, a portlet's creator can override the CGI parameters to place input boxes above the portlet when it is played back. These input boxes allow the person viewing the portlet to enter their own data values in place of the CGI parameters.

If a portlet's content contains tabular data, you can specify input parameters (commonly referred to as CGI parameters) during the portlet creation process.

A portlet's creator can override the CGI parameters to place input boxes above the portlet when it is played back. These input boxes allow the person viewing the portlet to enter their own data values in place of the CGI parameters.

Figure 5-1 shows a portlet with the CGI parameter that corresponds to the Department Set ID. The value has been overridden and set to "AUS".

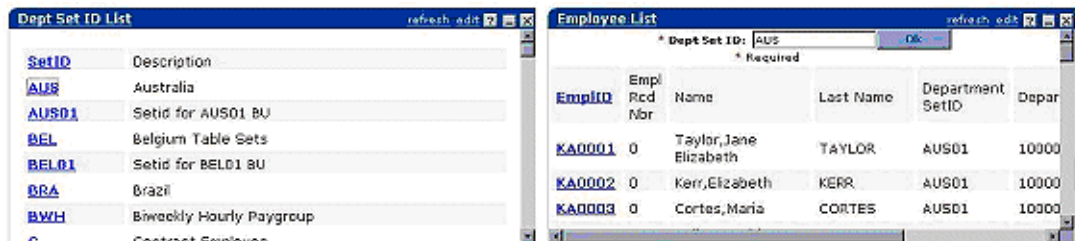
Figure 5-1: Portlet input parameter



EmplID	Empl Rcd Nbr	Name	Last Name	Department SetID	Deapar
KA0001	0	Taylor, Jane Elizabeth	TAYLOR	AUS01	10000
KA0002	0	Kerr, Elizabeth	KERR	AUS01	10000
KA0003	0	Cortes, Maria	CORTES	AUS01	10000

When used in conjunction with click-across messaging portlets, the data passed to the portlet from the input boxes can be populated automatically from data in another portlet on the same page or a different page.

Figure 5-2: Portlets exchanging data



SetID	Description
AUS	Australia
AUS01	Setid for AUS01 BU
BEL	Belgium Table Sets
BEL01	Setid for BEL01 BU
BRA	Brazil
BWH	Biweekly Hourly Paygroup
C	Contract Employees

The left portlet displays a list of Department Set IDs. The right portlet lists the employees for the given Department Set ID. When a user clicks the Department IDs in the left portlet, the selected ID is sent to the listening portlet on the right. Once the data is received, the listening portlet re-executes using the new data, and captures the resulting content.

Note Whether the listening portlet re-executes automatically depends on how you set the Auto Submit property in the Listener Editor when you define the portlet listener. If you selected Auto Submit when you define the listener, the listening portlet re-executes immediately. If Auto submit is not selected, the listening portlet does not re-execute instantly; click Refresh on the title bar to manually re-execute the listening portlet.

For example, if the user clicks the Belgium Department ID “BEL,” the string “BEL” is sent to the right portlet as input for the Department Set ID. The right portlet replays its capture sequence using the new query value of “BEL” and returns the result—the employee list for the Belgium department.

When developing click-across messaging portlets, you are not limited to one parameter. As a portlet’s creator, you can override any number of parameters and set any number of those parameters to accept data from another portlet on the same page. Additionally, you can set any number of message events within a portlet, as explained later in this section.

Creating messaging portlets

Note This discussion of click-across messaging portlets uses the Web element portlet type for examples. However, you can create click-across messaging portlets from any portlet type that has tabular data, a grid, or table structured output, and allows you to enter input parameters—JSP, XML, database, and Web services.

This section walks you through creating a click-across messaging portlet:

- 1 Creating a subscriber portlet
- 2 Creating a publisher portlet
- 3 Defining broadcast events
- 4 Assigning events to listeners
- 5 Adding messaging portlets to a page

6 Adding the click-across messaging portlets page to a page group

7 Testing the messaging portlets

❖ **Creating a subscriber portlet**

1 Select Build | Portlets from the Portal Studio left pane.

2 Click New to open the Portlet Builder.

3 When the Portlet Builder opens, click Add to open the New Web Element window.

4 In the Location text box, enter `http://investor.stockpoint.com` and click Find.

5 In the Web page Quote field, enter “SY” and click Go to submit your search query.

6 Click Next in the upper-right corner.

7 On the next page, move the mouse around until the cursor is positioned over some text in the first column of the stock quote table. Click one of the row titles—such as Last or Open—once.

You see a preview with several selections. Each selection contains more data from the captured table.

8 Click Select next to the capture option you want to use, then click Next. Click Next until you bypass the Split Rules, Define, and Filter windows. Stop when you see the window where you define or override CGI parameters.

9 Click the Variable option next to the Symbol parameter and complete these options:

- Symbol – select this option to create parameters for the stock quote.
- Default Value – enter “SY”.
- Type – select “Text Field.”
- Required – select this option.

Click Next to see a preview of the captured data.

Note Verify that you have entered the “SY” default value for the Symbol parameter. Parameters without default values may not work.

10 In the Element Name field, enter a name for the captured data; for example, “Stock Quote.” Click Finish.

- 11 In Portlet Builder, you see a working preview of the portlet.
- 12 To test overriding the CGI parameter, type a different stock symbol into the Symbol text box and click Update. Some example stock symbols to use are:
 - Sybase: SY
 - Microsoft: MSFT
 - Oracle: ORCL
 - Cisco: CSCO
 - PeopleSoft: PSFT
- 13 Click Save. You see the New Element Finish window. Complete these options:
 - Portlet Name – enter “My Stock Quote”.
 - Active, Iframe, No Popup, Editable – select all of these options. Leave the remaining options unselected.

Note Messaging portlets must have the Iframe option selected.

- Content Type – select “text/html.”
- Assigned Roles – click Add All to move all roles from the Available Roles list to the Assigned Roles list.

Accept the defaults for the remaining options.

- 14 Click Finish to save the portlet.

❖ **Creating a publisher portlet**

- 1 Select Build | Portlets from the Portal Studio left pane.
- 2 Click New to open the Portlet Builder.
- 3 When the Portlet Builder opens, click Add to open the New Web Element window.
- 4 In the Location text box, enter `http://finance.yahoo.com` and click Find.
- 5 In the middle column of the Yahoo Finance Web page, beneath the Top Stories heading, click the “Most Actives” link.

- 6 Click Next, then move the mouse around until the cursor is positioned over the Name column of the table. Click on the “Name” heading once to capture the table.
- 7 Click Select next to the top captured table option, then click Next.
- 8 On the Split page, click Next.
- 9 On the Define page, click Next.
- 10 On the Filter page, click Next.
- 11 On the page where you define or override CGI parameters, do nothing. Click Next to continue.
- 12 In the preview window, give the element a good name like “Most Actives” and click Finish.
- 13 Click Save in Portlet Builder.
- 14 In the Finish window, verify your save options and settings:
 - Portlet Name – enter “Most Actives”.
 - Active, Iframe, No Popup, Editable – select all of these options.

Note Messaging portlets do not work unless Iframe is selected.

- Content Types – select “text/html.”
- Assigned Roles – click Add All to move all roles from the Available Roles list to the Assigned Roles list.

Accept the defaults for the remaining options.

Click Finish.

- 15 Click Close to exit the Portlet Builder and return to the Portal Studio main window.

❖ Defining broadcast events

Note Not all portlet types can have events defined; the portlet must have tabular data, a grid, or table structured output. If you try to create an event and nothing seems to happen, try using a different portlet type. Portal Studio does not always issue a warning if the portlet is not capable of having an event defined.

- 1 Select Build | Portlets from the Portal Studio left pane.

- 2 On the Portlet Manager Status menu, click New to display, in the detail pane, new portlets to which you have access.
- 3 Locate the two newly saved portlets (publisher and subscriber).

Note By default, a portlet has the “New” status until you right-click the portlet in the detail pane, select Approval Status from the pop-up and select a different status.

- 4 Right-click the Most Actives portlet and select Define Events from the pop-up.

The Define Events wizard displays in a new window. Defining events is a three-step process. The first step shows you all grid, table, or list (structured) objects contained in the current portlet. You can define events only on structured (tables and lists) content.

- 5 In the first wizard window, click Select to choose the top table listed and click Next. The next window allows you to define or edit an event.

You can define an event for a row, a column, or any subset of rows and columns. You can also use the row and column number to identify a specific cell to define an event.

Enter the keyword “all” for Row and Column to indicate all values in a row or column for an event.

Event Name is a logical name that must be unique across the entire portal and you must have separate names for distinct events. You can reuse the name for different cells to have the same event generated.

Warning! Defining multiple events on the same data causes invalid HTML. In other words, column 1 can not have more than one event defined on it.

The differences between the types of data that can be sent in an event are:

- Dynamic – sends the text you click on as the event data.
- Static – sends a constant value you specify in the text box to the right of the drop-down list as the event data.
- HREF – if the text for which you are defining the event is a link, the underlying URL is sent as the event data.

- 6 Enter these values:

- Row – the word “all”.
 - Column – “1” (the number one).
 - Event Name – “Stock_Click” (note the underscore).
 - Select “Dynamic Value” from the drop-down list
- 7 Click Add to add the event to your events list. After a short time, the first column should turn blue, which indicates there is an event defined for that column.
 - 8 Click Next to save the new event.
 - 9 When the Preview page displays, click Finish.
 - 10 Back in the Portal Studio, right-click the Most Actives portlet and select Approval Status | Approved.

❖ **Defining listeners**

This step lets you define a listener for the event to which a portlet is going to subscribe and listen. You can register a portlet to listen for one or more events. Each event to which a portlet listens must be assigned a portlet input parameter. You must remember the name of the event to which you want a portlet to listen.

- 1 On the Status Menu, click New, then right-click the Stock Quote portlet and select Define Listeners.
- 2 In the Event Name field, type “Stock_Click”—the name of the event you defined for the My Actives portlet—then make sure that Listener Param is set to “Symbol.”
- 3 Select Auto Submit to have the publisher of the event call this listener and have the portlet re-execute immediately.
- 4 Click the plus sign to the right of Listener Param to add the listener event to the portlet. Once the plus turns into a check mark, click OK to save your changes. If you make an error, click the X icon to delete the Listener Param association, and repeat steps 1 and 2.
- 5 In the Portal Studio, right-click Stock Quote portlet and select Approval Status | Approved.

❖ **Adding messaging portlets to a page**

To test the event you defined, you must add the new portlets to a new page that can be viewed in Portal Interface.

- 1 In Portal Studio, select Build | Pages from the left pane.

- 2 Click New on the Page Manager toolbar.
- 3 In the Page Builder window, click Add.
- 4 In the Search window that displays, click Search.

Note You must have already given the portlets the Approved status; otherwise, the portlets do not display in the search results.

- 5 In the Results pane, hold down the Ctrl key and select the Most Actives and Stock Quote portlets. When both are selected, click Add.
- 6 Click and drag the boxes in the right preview until the Most Actives portlet is on the left and the Stock Quote portlet is on the right.

The options for each portlet in the left pane are:

- Invisible – lets you hide a specific portlet from the end user. For example, you could set the display status of an portlet to invisible because the portlet is performing a server-side lookup operation; for example, transforming city names to postal codes.
 - Active – when this option is selected, which it is by default, the associated portlet is marked as active. Only portlets marked active and approved display in Portal Interface.
- 7 Click Save As to save the page. Complete these fields:
 - Name – enter “My Stock Quotes”.
 - Type – select Default.
 - Active – this option is selected by default.
 - Roles – click Add All to assign all the available roles to the page.

Click OK to save the page.

- 8 After a dialog box tells you the page has been saved, click OK, then click Close to exit the Page Builder.
- 9 On the Page Manager Status menu, click New to locate the new page.
- 10 Right-click the My Stock Quotes page in the detail pane, and select Status | Approved.

❖ **Adding the click-across portlets page to a page group**

Add the new page that contains the click-across messaging portlets to a page group.

- 1 In Portal Studio, select Build | Page Groups.
- 2 Click the New button on the Page Groups Manager toolbar.
- 3 When the Page Group Builder displays, click Add.
- 4 When the Search window appears, click Search.
- 5 In the Results pane, select the My Stock Quotes page and click Add.
- 6 In the Page Group Builder, click Save As.
- 7 When the Save Page Group window appears, complete the fields as follows:
 - Name – enter “My Stock Quotes”.
 - Type – select Default.

Note Pages must be of the same type as the page group to which you add them. For example, if you create a “default” type page group, you can only add pages to that page group that are also the “default” type.

- Navigation Style – accept the default.
 - Active – verify that this option is selected.
 - Assigned Roles – hold down the Ctrl key, select PortalAdmin and PortalUser, then click Add to move those roles to the Assigned Roles list.
- 8 Click OK to save the page group, and click OK again when a prompt verifies that the page group has been saved.
 - 9 Click Close to exit the Page Group Builder.
 - 10 Select New from the Page Group Manager Status menu in the Portal Studio main window.
 - 11 Right-click the new page group, and select Status | Approved from the pop-up. Click OK when the page group saved confirmation displays.
 - 12 Select Approved from the Status menu.
 - 13 Right-click the My Stock Quotes page group select Update Users.
 - 14 When you see the message that the operation succeeded, click OK. Because the My Stock Quotes page group is the “default” type, all registered Portal Interface users can access this page group after the update operation.

15 Click Logout to exit Portal Studio.

❖ **Testing the messaging portlets**

1 Start Portal Interface by entering:

`http://HOSTNAME.PORTALDOMAIN:8080/onepage/index.jsp`

If you are using the EP Demo version, enter:

`http://demo.sybase.com:4040/onepage/index.jsp`

- 2 Log in using an existing account or create a new account using the Join Now link.
- 3 Select My Stock Quotes from the page group drop-down list on the far right of the Portal Interface window. You see the messaging portlets.
- 4 To test the click-across messaging portlets, select one of the stock symbols in the Most Actives portlet. The My Stock Quote Symbol input field updates with the stock symbol that you clicked on and updated stock information for the new symbol appears.

Building application portlets

You create and manage virtual applications using the Application Manager and Application Builder. You use the Application Builder to assemble portlets and pages to create an interactive application, and you use the Application Manager to manage content.

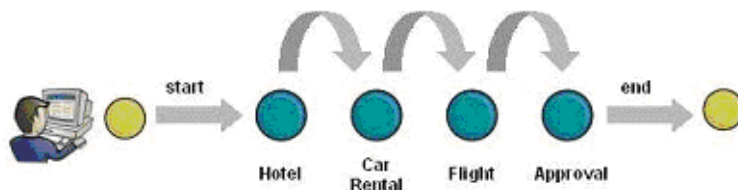
Portlet navigation is used to display and populate the portlets that form a logical sequence of user actions. The portlet sequence usually defines individual points within a larger user-based task or process. You can position sequenced portlets on the same page (portlet flow) or on different pages (page flow).

Application portlets bind existing portlets into a common operation, sharing data between the portlets and allowing them to operate as a virtual application. Application Manager and Application Builder let you create application portlets in Portal Studio.

Application portlets let you create a shared context in which individual portlets interact. Application portlets can include individual portlets on one page (a *portlet flow*) or serialized portlets on multiple pages (a *page flow*).

To arrange business travel, an employee must often interact with several Web sites that are independent of each other and maintain no shared context. For example, when arranging a trip, an employee must reenter travel dates into each individual Web site.

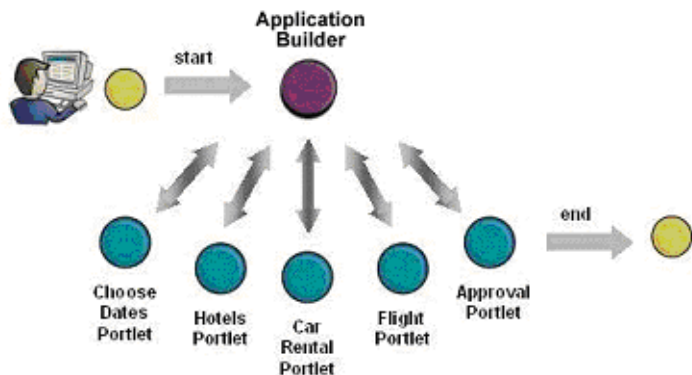
Figure 5-3: Arranging travel at unrelated Web sites



The employee follows a preferred order in which each travel element is arranged. For example, the employee reserves a hotel room, reserves a rental car, then reserves a flight. An application portlet lets you combine these disparate tasks into a single application available at the portal. Each task in the process, such as reserving a hotel, is provided by an individual portlet.

The portal allows a user to personalize the presentation of the portlets. A personalized presentation could include arranging commonly-used portlets on one page and others on alternate pages. The Application Builder lets you enter the information relevant to each task.

Figure 5-4: Arranging travel through an application portlet



Each portlet is separate from the other portlets in the application, but the Application Builder displays them in a shared context.

Shared context

To minimize the number of entries an employee has to make, a common view of individual tasks allows each task to share the same contextual view as the other tasks.

In this example, the travel dates are common to each portlet. The hotels portlet uses dates to check for the availability of rooms. The car rental portlet uses dates to check the availability of vehicles. The flights portlet uses the dates to check preferred airlines schedules. In each case, the user can override the dates if necessary.

In addition to the common dates, each portlet also has its own requirements. For example, the Hotel portlet requires the destination city, the Car Rental portlet requires the rental duration, and the Flights portlet requires the destination airport code. The user can enter each value directly into the portlet, automatically setting the context of the portlet request for each.

When constructing shared context, you identify the key fields that each of the component portlets has in common, which allows the fields to be linked. For example, the application designer uses form processing in the Hotel portlet to examine the hotel reservation form. The designer is presented with a list of form fields with the relevant fields selected. The next step is to define events for the common relevant fields. After defining the events, the designer uses the Application Builder to assign events, as needed, to the fields of the Car Rental and Flight portlets. The end result is a link between the common input fields of the forms in the application's component portlets. When published to Portal Interface, the link between the form fields is used to automatically fill in the specified common fields of the other portlets used in the application.

Application Manager

The Application Manager allows you to manage and run the application objects through a workflow approval process. Select **Build | Applications** from the Portal Studio menu in the left pane.

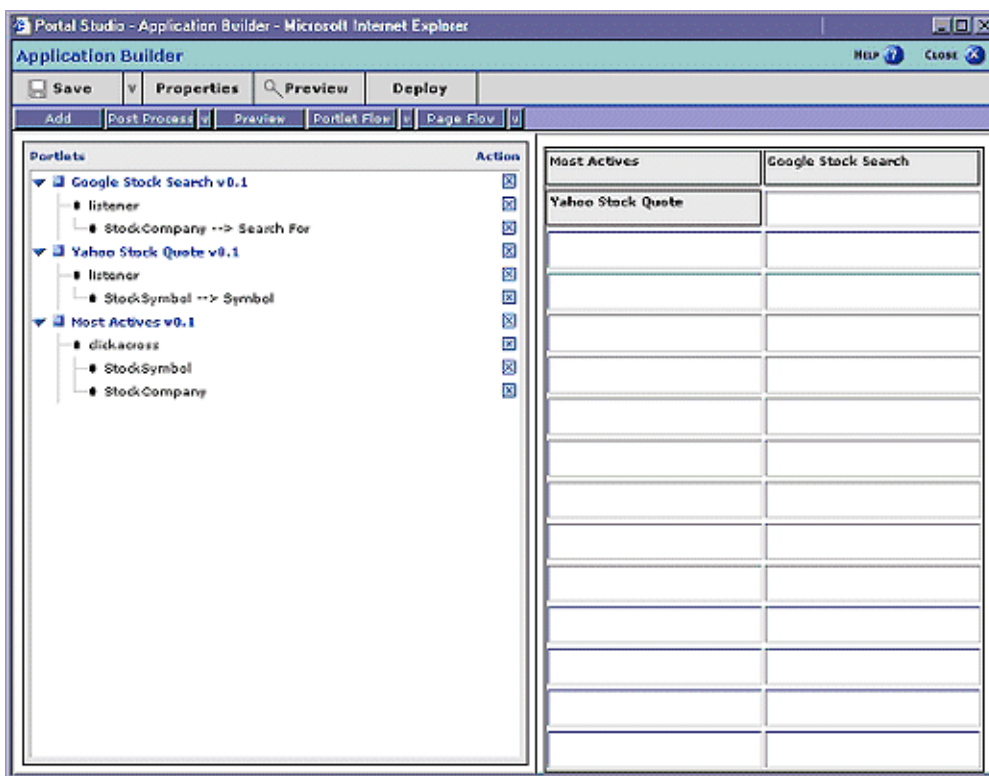
The middle pane of the Application Manager displays the Status menu. The right pane shows applications for each status to which the logged in user has access. For example, select **Status | Approved** to see all approved applications. The display of applications is controlled by role-based authentication.

Application Builder

The Application Builder lets you produce multi-portlet applications that are connected to one another using events or messaging. You lay out an application's portlets in a "portlet flow" or "page flow."

An application laid out with portlet flow displays all of its portlets on one page. An application that uses page flow displays each row of portlets on a different page. Figure 5-5 shows the Application Builder with an existing sample application.

Figure 5-5: Application Builder



The left pane displays the portlets and associated events that make up the application. The right panel displays a drag-and-drop workspace where you lay out and order the portlets.

The top row buttons are:

- Save – displays the Application Properties window if the application has not been previously saved. If the application has been saved, this option saves the application without opening the properties window. Click the down arrow to the right of the Save option to select Save As and save the application under a different name.
- Properties – displays the Application Properties window where you can change the application’s flow type, assign roles, status, and specify whether or not the application is active.
- Preview – opens a new window that lets you preview the application. You must save the application before you can preview it.
- Deploy – sends the application to Portal Interface.

The second row buttons are:

- Add – launches the Find dialog box, where you can select one or more portlets to use in the application. The portlets shown in the Results pane are approved portlets. The portlets you add display in the left pane of the Application Builder.
- Post Process – provides a list of post processing options that you can add to portlets. See “Portlet post-processing options” on page 111.
- Preview – previews the portlet selected in the left pane portlet list.
- Portlet Flow – select one of these configurations in which to lay out application portlets on one page:
 - 30% / 70% Layout
 - 50% / 50% Layout
 - 25% / 50% / 25% Layout
 - Full LayoutSee “Selecting a page layout” on page 191 for more information on the layout options.
- Page Flow – select one of these configurations in which to lay out the application portlets on different pages in the same page group.
 - 30% / 70% Layout
 - 50% / 50% Layout
 - 25% / 50% / 25% Layout

- Full Layout

See “Selecting a page layout” on page 191 for more information on the layout options.

Portlet post-processing options

The key to building virtual applications is the addition of post-processing functions or filters to the application’s portlets. Post-processing functions can either transform a portlet’s content (for example, render the content as a chart), or create events. For example, you can transform a portlet’s content into a chart or you can create events that are triggered when a user accesses a portlet’s content. You can add more than one post-processing function to handle a portlet’s content and configure the order in which the postprocessing occurs. When you use multiple postprocessing functions, each function operates on the output of the previous function—similar to a Pipes and Filter design.

Note Pipes and Filters is a design architecture where the components are known as filters and the connectors are the pipes. Filters read input data, perform some computation or transformation on the data, and produce output data. Pipes serve as conduits for the data streams, passing outputs from one filter to the inputs of another filter.

The post-processing options are available by right-clicking a portlet in the Application Builder left pane:

- Define Events – launches an Event Definition wizard. See “Defining broadcast events” on page 101 and “Adding events” on page 112.
- Define Listeners – launches a Listener Definition wizard. See “Adding listeners” on page 113.
- Create Chart – lets you chart a portlet’s structured content. See “Adding charts” on page 114.
- Form Parser – lets you process form input and define which form parameters should be event default values or listeners for events from other sources. See “Adding form processing” on page 114.
- Create JSP Filter – lets you write custom code to process portlet content. See “Adding a JSP filter” on page 116.
- Text Processor – launches a text processor wizard that lets you define an event to occur when the user clicks on selected text. See “Processing text to trigger events” on page 117.

- Order Post Processing – lets you sequence a portlet’s postprocessing options at anytime. See “Ordering filters” on page 118.
- Invisible Status – lets you hide a specific portlet from the end user. See “Invisible status” on page 118.

All available post-processing modules are configured in the *uwp.xml* configuration file, which is located in *%JAGUAR%Repository\WebApplication\onepage\fw\properties* on Windows and in *\$JAGUAR/Repository/WebApplication/onepage/fw/properties* on UNIX.

To add additional post-processing modules, use this interface:

```
Public interface PortletProcessingFilter {  
    String Process(string content, UWPRequest request,  
        UWPResponse response, UWPDataAccess access);  
}
```

The following five post-processing filters will be offered as a part of the current implementation.

```
<PostProcessingFilters>  
    <FilterDef name="Events" class="com.onepage.fw.uwp.filters.events"/>  
    <FilterDef name="Listeners" class="com.onepage.fw.uwp.filters.listeners"/>  
    <FilterDef name="Forms" class="com.onepage.fw.uwp.filters.forms"/>  
    <FilterDef name="Charts" class="com.onepage.fw.uwp.filters.charts"/>  
    <FilterDef name="jsp" class="com.onepage.fw.uwp.filters.jsp"/>  
</PostProcessingFilters>
```

There is a method in the *UWPManager* *getPostProcessingFilters* that returns a list of configured filters.

Adding events

Click the down arrow beside the Post Process button and select Define Events to launch the Event Definition wizard. When you define an event on a portlet, the portlet is saved under the same version, irrespective of what the versioning property is set to. See “Defining broadcast events” on page 101 for instructions on creating events.

Once an event has been defined on a portlet, all defined events are listed under the portlet name in a hierarchy.

Event-naming convention

Events can have a three-part name:

`[page_group_name] . [page_name] . [event_name]`

However, a three-part name is not mandatory. You can also create a name with just `[event_name]`. The three-part name allows greater flexibility and registered listener name mapping.

The event name is matched with listeners that have their `[page_group_name]`, `[page_name]` registered along with the subscribed-to `[event_name]`. If an event name has only one part, all registered listeners are matched with that event name. If an event name includes a `[page_name]`, that event is matched to listeners with the same page name. The same rules apply to `[page_group_name]`.

Note Do not using spaces in event names.

Adding listeners

You can add listeners only to portlets with parameters. Portlets with listeners must be saved with the `Iframe` attribute.

The Listener Editor lets you assign multiple parameters to event names. The portlet is saved under its current version number, regardless of whether the global versioning property is set to “on”.

❖ Creating a listener

- 1 Click the down arrow beside the Post Process button and select Define Listeners to launch the Listener Editor.
- 2 When the Listener Editor displays, complete these options:
 - List of Events – select the event to which you want this listener assigned.
 - Event Name – select an event from the List of Events drop-down list, or enter a name manually in this field.

Note Do not using spaces in event names.

- Listener Param – select the overridden CGI parameter that you want to associate with the selected event.

- Auto Submit – when Auto Submit is enabled, a listener portlet re-executes when it receives an event. If Auto Submit is disabled, the portlet does not re-execute immediately.
- 3 Click the plus sign to add the association with the selected event or the X sign to remove any associations that have already been made.

Once you assign one or more parameters to event, parameter-to-event mapping is listed under the portlet name in the Application Builder in a hierarchy.

Adding charts

Click the down arrow beside the Post Process button and select Create Chart to launch the Chart Definition wizard that is launched by right-clicking a portlet in the Portlet Manager. See Chapter 6, “Charting Portlets,” for detailed information on charting portlets.

When you define charts for a portlet, the portlet is saved as the same version, regardless of whether the global versioning property is turned “on.” Once you define a chart, the chart is listed under the portlet name in a hierarchy.

Adding form processing

Form processing allows you to preview a form, identify input fields, assign or create an event name, and assign default autofill values to the form’s fields.

❖ Parsing a form for post-processing

- 1 Log in to Portal Studio and select Build | Applications from the menu in the left pane.
- 2 In the detail pane, right-click the application to which you want to add post-processing, and select Edit from the pop-up.
- 3 When the Application Builder displays, right-click the portlet on which you want to parse a form and select Form Parser from the pop-up.

The Form Parser window appears with a preview of the portlet.

- 4 Click Next. The Form Input Definitions window displays.
- 5 Click the Variable column for each form variable you want to be an event, default value, auto-filled, or listener for events from other sources.

Once you select a variable, fields and drop-down lists display for the that variable.

Variable	Name	Event Name	Default Value	Type	Personalize	Key
<input type="checkbox"/>	mycapp		FWLogin	hidden		
<input type="checkbox"/>	b			hidden		
<input type="checkbox"/>	rid			hidden		
<input type="checkbox"/>	command		login	hidden		
<input type="checkbox"/>	clientBaseURL			hidden		
<input checked="" type="checkbox"/>	userid	<input type="text"/>	<input type="text"/>	text	username/p...	Login Name
<input checked="" type="checkbox"/>	password	<input type="text"/>	<input type="text"/>	password	username/p...	Password

6 Complete these options for each selected variable:

- Event Name – this field lets you define an event on-the-fly. Type in the name of the event for which you want this field to be default value.

Note Do not using spaces in event names.

- Default Value – enter the event’s default value.
- Type – this column displays the field type. To change the field type, select a different type from the drop-down list.
- Personalize – if the field can be filled in using an auto-fill adapter, select the adapter from the drop-down list.
- Key – if the field has a key values, select the appropriate key from the drop-down list.
- Notify Now – selection this option to send any event immediately.
- Notify Across Page – select this option to send the event across pages if there is a listener on another page; otherwise, look for events on the current page. for this example, do not select this option.

7 Click Finish.

Adding a JSP filter

JSP filters allow you to write custom code to post-process portlet content using programming. When you create JSP filters, you look at a portlet's content, enter the Java code to transform the content in any way you want, then save the code as a JSP template, which you can later select for post-processing.

❖ Selecting a JSP filter

- 1 Log in to Portal Studio and select Build | Applications from the menu in the left pane.
- 2 In the detail pane, right-click the application to which you want to add post-processing and select Edit from the pop-up.
- 3 When the Application Builder displays, right-click the portlet to which you want to add a JSP Filter and select Create JSP Filter from the pop-up. The JSP Editor appears with a preview of the portlet.
- 4 Complete these options:
 - JSP Template – enter the name of an existing JSP template you want to use or click Select to search for a template using the Find JSP Template dialog box.

Click Search to display existing approved JSP templates. When the list appears in the Results pane, select:

- New – to create a new JSP template.
- Edit – to modify the template selected in the Results pane.
- Select – to choose the template selected in the Results pane.
- Preview – to preview the template selected in the Results pane.

- Apply JSP Filter to XML Playback – select this option to apply the filter only when the portlet’s content is returned as XML. When this option is not selected, which is the default, the filter is applied only when the portlet’s content is returned as HTML.

Note When portlets are played back, they return their content in either HTML or XML. Whether HTML or XML is returned is determined by the entity making the request.

For example, when Portal Interface requests that portlet A be played back, the output is requested as HTML. When an agent requests that sample portlet A be played back, the output is requested as XML. When you create a JSP filter and attach the filter to a portlet for post-processing, the developer uses this option on a case-by-case basis to apply or not apply the JSP filter during portlet playback.

- 5 Click Finish to associate the JSP Filter with the selected portlet.

You can add more than one JSP filter as post-processing elements on a portlet and you can apply the same JSP filters to numerous portlets. When the JSP filters have been added, the portlet is saved under the same version, regardless of how the global versioning property is set. Once you assign a JSP filter for portlet post-processing, the filter is listed in the Application Builder under the portlet name.

Processing text to trigger events

The Text Processor wizard is similar to the Event Definition wizard and allows you to create events that are triggered when a user clicks on the text you associate with the event.

❖ Post-processing text to trigger events

- 1 Right-click any portlet in the Application Builder Portlets list, click Post Processing, then select Text Processor.
- 2 When the Text Processor wizard displays, you see a preview of the selected portlet.
Click Next. You see the window where you define click-across events.
- 3 Complete the options as follows:
 - Regular Expression – enter a regular expression to place an event on text that matches the expression.

- Event Name – the name of the event to generate when the matching text is clicked. The event must have already been created.

Note Do not using spaces in event names.

- Value drop-down list – select the type of data to send with the event:
 - Dynamic Value - sends the actual text you click on as the event data.
 - Static Value - sends a constant value as the event data. You specify this value in the text box to the right of the value drop-down list.
 - Matching Scheme – select the matching scheme to use when finding matches for your regular expressions. Select:
 - Match All Occurrences – to place the event on every match to your regular expression that is found.
 - Match First Occurrence Only – to place the event on only the first occurrence of a match to your regular expression.
 - Notify Now – send the event immediately. For this example, select this option.
 - Notify Across Page – send the event across pages if there is a listener on another page; otherwise, look for events on the current page.
- 4 When you are satisfied with your entries, click Next.
 - 5 When you see a final preview of the portlet, click Finish. You return to the Application Builder.

Ordering filters

Click the down arrow beside the Post Process button and select Order Post Processing to launch a wizard that lets you order and re-order a portlet's post-processing filters.

Invisible status

Click the down arrow beside the Post Process button and select Invisible Status to make the selected portlet Visible or Invisible. The default is Visible. When you make a portlet Invisible, the end user will not see it on the page.

For example, you could set the display status of an portlet to invisible because the portlet is performing a server-side look up operation that the end user does not need to see, such as transformation (from city name to postal code, currency conversion), look ups, polling queues or a table for availability of data.

Application runtime entities

This section discusses the runtime entities that must be generated to execute the portlet application.

Portlet flow

If the application is laid out for portlet flow, the application's portlets are positioned on one page at portal runtime.

Click Portlet Flow to lay out an application's portlets on one page. Select one of the four page layouts that are available. Typically, you assign portlets to the layout cells from left -to-right and top-to-bottom, which reflects the natural flow of information. Once the layout is done, click Preview to view how the portlets will display on a Portal Interface page.

Page flow

If the application is laid out for page flow, the application's portlets are positioned on different pages at portal runtime. To lay out portlets on a page, click Page Flow. You assign portlets in the cells of the page-flow layout from top-to-bottom reflecting the natural flow of information. In Portal Interface, the pages show from left-to-right. Once the layout is done, click Preview to see how the pages will display in a Portal Interface personal page group.

Note To use Page Flow, the <page.group> property in the *global.properties.xml* file must be set to "on."

Application properties

A new application's associated pages and page group status is always saved as Approved. Although you can set the status to a different state, you can deploy an application only when it has the Approved status.

When you click Properties in the Application Builder and are saving an application for the first time, you see the Application Properties window. Once you save the application and click Properties, the roles you have assigned appear in the Assigned Roles list and the system-assigned ID displays.

Note You can change an application’s status only from the Application Properties window.

If you change the Type to Page Flow, an additional option—Page Naming—displays. The two naming options are Use Application Name and Use Portlet on Page. When the pages are created for Page Flow, the pages names are determined by this setting.

- Use Application Name – names each page in the format:

`_[Application Name]_[Page number]`

For example, `_StockApp_01`, `_StockApp_02`, and so on.

- Use Portlet on Page – uses the portlet name from the page in the format:

`_[Portlet Name]_`

Warning! Saving applications with the Page Flow type and the Page Naming option set to Use Portlet Name can overwrite a previous application’s page if the names are the same.

Creating application portlets

The examples in this section guide you through creating a small application that uses three portlets. The portlets created in this tutorial include:

- A list of the 25 most active stocks of the day
- A stock quote portlet
- A Google.com search of a company's name

The sample application that you build will use “portlet flow,” meaning all of the portlets are on the same page. See “Portlet flow” on page 119.

You will use the Application Builder to define click-across events and listeners in the portlets so data can be sent between them. You will define two click-across events on the 25 most active stocks portlet. The first event will send the stock symbol to the listening stock quote portlet and the second event will send the company name to the listening Google.com search portlet, which searches Google.com for the name of the company.

❖ **Creating the Most Actives portlet**

- 1 Log in to Portal Studio and select Build | Portlets from the menu in the left pane.
- 2 Click New on the Portlet Manager toolbar. You see the Portlet Builder.
- 3 Click the down arrow beside the Add button and select Web Element from the pop-up.
- 4 When the Portal Studio – New Web Element window displays, enter `http://finance.yahoo.com/` in the Location field and click Find. The Yahoo Finance Web site displays in the window.
- 5 Click the link in the middle of the page titled “Most Actives.” The Web site changes to display NASDAQ Volume Leaders.
- 6 Click Next. When the page reloads, move the mouse around until the cursor is positioned over some text in the stock table and click once to perform a one-click capture. You see a preview with several selections. Each selection contains more data from the captured table.
- 7 Click Select next to the capture option you want to use, then click Next.
- 8 In the Split window, click Next.
- 9 In the Define window, click Next.
- 10 In the Filter window under Add Filter Rule, set a filter rule to remove extraneous columns from the table.
 - a In the first drop-down list, select “Include field(s)”.
 - b In the second drop-down list, select “number”.
 - c In the text box, enter:

1 - 2 , 4 - 6

This tells the filter to include columns where the column number is in the ranges of 1 through 2 and 4 through 6.
 - d Click Add and the grid should highlight the selected columns.

- 11 Click Next.
- 12 On the CGI parameter override screen, click Next.
- 13 On the final screen, give the Web element the name “Most Actives” and click Finish to close the window.

Back in Portlet Builder, you see a preview of the Most Actives element.

- 14 Click Save to save the portlet.
- 15 In the Finish window, complete these options:

- Portlet Name – enter “Most Actives”.
- Active, Iframe, Editable – select these options and leave the No Popup, Secure, and Versioning options unselected.

Warning! Application messaging portlets will not work unless Iframe is selected.

- Content Type – select “text/html”.
- Charset – select “iso-8859-8”.
- Content Cache Interval – select “Real-Time”.
- Default Size – select “Medium”.
- Assigned Roles – select PortalUser and click Add to move that role from the Available Roles list to the Assigned Roles list.

Accept the defaults for the remaining options.

- 16 Click Finish. When you see the confirmation that the portlet was saved successfully, click OK.
- 17 Click Close to exit the Portlet Builder.
- 18 On the Portlet Manager Status menu, click New to display the new portlet in the right pane detail window.

Note By default, a portlet has the “New” status until you right-click the portlet in the detail pane, select Approval Status from the pop-up and select a different status.

- 19 Right-click on the Most Actives portlet and select Approval Status | Approved, then click Approved on the Status menu and verify that the Most Actives portlet displays in the Approved list.

❖ Creating the Stock Quote portlet

- 1 Log in to Portal Studio and select Build | Portlets from the menu in the left pane.
- 2 Click New on the Portlet Manager toolbar. You see the Portlet Builder.
- 3 Click the down arrow beside the Add button and select Web Element from the pop-up.
- 4 When the Portal Studio – New Web Element window displays, enter `http://finance.yahoo.com/` in the Location field and click Find. The Yahoo Finance Web site displays in the window.
- 5 In the text box with the title “Enter symbol(s)” enter “SY”, select Detailed from the drop-down list to the right of the text box, then click Get.

Note This screen may vary depending on your browser’s version. On some browsers, the drop-down list may not display and the Get button is labeled Go. The most important thing is to enter the SY symbol and click the button that means “go get it.”

You see the detailed stock quote for the stock symbol SY at Yahoo Finance.

- 6 Click Next. When the page reloads, move the mouse around until the cursor is positioned over some text in the stock table and click once to perform a one-click capture. You see a preview with several selections. Each selection contains more data from the captured table.
- 7 On the capture selection page, select the capture that displays the stock table, then click Next.
- 8 On the CGI parameter override window:
 - a Select the Variable option next to the variable named “s”.
 - b Change the Display Name to “Symbol”.
 - c Enter “SY” for the Default Value.
 - d Leave Type as “Text Field”.
 - e Verify that the Required option is selected.
 - f Click Next.
- 9 In the final window, enter “Yahoo Stock Quote” and click Finish.

Back in Portlet Builder, you see a preview of the Yahoo Stock Quote element.

- 10 Click Save to save the portlet.
 - 11 In the Finish window, complete these options:
 - Portlet Name – enter “Yahoo Stock Quote”.
 - Active, Iframe, Editable – select these options and leave the No Popup, Secure, and Versioning options unselected.
-
- Warning!** Application messaging portlets will not work unless Iframe is selected.
-
- Content Type – select “text/html”.
 - Charset – select “iso-8859-8”.
 - Refresh Interval – select “Real-Time”.
 - Default Size – select “Medium”.
 - Assigned Roles – select PortalUser and click Add to move that role from the Available Roles list to the Assigned Roles list.
- Accept the defaults for the remaining options.
- 12 Click Finish. When you see the confirmation that the portlet was saved successfully, click OK.
 - 13 Click Close to exit the Portlet Builder.
 - 14 On the Portlet Manager Status menu, click New to display the new portlet in the right pane detail window.

Note By default, a portlet has the “New” status until you right-click the portlet in the detail pane, select Approval Status from the pop-up and select a different status.

- 15 Right-click on the Yahoo Stock Quote portlet and select Approval Status | Approved, then click Approved on the Status menu and verify that the portlet displays in the Approved list.

❖ **Creating the Google Stock Search portlet**

- 1 Log in to Portal Studio and select Build | Portlets from the menu in the left pane.

- 2 Click New on the Portlet Manager toolbar. You see the Portlet Builder.
- 3 Click the down arrow beside the Add button and select Web Element from the pop-up.
- 4 When the Portal Studio – New Web Element window displays, enter `http://www.google.com/` in the Location field and click Find. The Google search Web site displays in the window.
- 5 Click Google Search. Google.com accepts the empty search string to create the required CGI request. The subsequent window shows the empty Google search result.
- 6 Select Page from the Format drop-down list and click Next. This option allows you to capture the entire results page instead of just a portion of it.
- 7 On the capture selection page, there should be only one option. Select the capture, then click Next.
- 8 On the CGI parameter override window:
 - a Select the Variable option next to the variable named “q”.
 - b Change the Display Name to “Search for”.
 - c Select a Type of “Text”.
 - d Verify that the Required option is not selected.
 - e Click Next.
- 9 In the final window, enter “Google Stock Search” and click Finish.
Back in Portlet Builder, you see a preview of the Google Stock Search element.
- 10 Click Save to save the portlet.
- 11 In the Finish window, complete these options:
 - Portlet Name – enter “Google Stock Search”.
 - Active, Iframe, Editable – select these options and leave the No Popup, Secure, and Versioning options unselected.

Warning! Application messaging portlets will not work unless Iframe is selected.

- Content Type – select “text/html”.
- Charset – select “iso-8859-8”.

- Refresh Interval – select “Real-Time”.
- Default Size – select “Medium”.
- Assigned Roles – select PortalUser and click Add to move that role from the Available Roles list to the Assigned Roles list.

Accept the defaults for the remaining options.

- 12 Click Finish. When you see the confirmation that the portlet was saved successfully, click OK.
- 13 Click Close to exit the Portlet Builder.
- 14 On the Portlet Manager Status menu, click New to display the new portlet in the right pane detail window.

Note By default, a portlet has the “New” status until you right-click the portlet in the detail pane, select Approval Status from the pop-up and select a different status.

- 15 Right-click on the Google Stock Search portlet and select Approval Status | Approved, then click Approved on the Status menu and verify that the portlet displays in the Approved list.

❖ **Creating the application**

- 1 Log in to Portal Studio and select Build | Applications from the menu in the left pane.
- 2 Click New on the Portlet Manager toolbar. You see the Application Builder.
- 3 Click Add to add existing portlets to the application using the Search dialog box.
- 4 In the Search dialog box, click Search to retrieve a list of all approved portlets.
- 5 In the Results pane, hold down the Ctrl key and click the Google Stock Search, Most Actives, and Yahoo Stock Quote portlets that you created, then click Add to add them to the Application.
- 6 To adjust the portlet placement, drag and drop the portlets around the right side of the Application Builder window.

❖ Most Actives portlet post processing

The Most Actives portlet will be used to publish click-across events. You can either define an event on the portlet outside of the Application Builder or within the Application Builder.

Note Any post processing added to a portlet in the Application Builder also effects the portlet outside of the Application Builder. See the notes at the end of this tutorial for more details.

- 1 Right-click the Most Actives portlet in the Application Builder's Portlets pane and select Define Events. This starts the Define Event wizard.

Defining events is a three-step process. The first step shows you all grid, table, or list (structured) objects contained in the current portlet. You can define events only on structured (tables and lists) content. The first screen shows you a list of tables in the selected portlet.

- 2 In the first wizard window, click Select to choose table of stock quotes and click Next. The next window allows you to define or edit an event.

You can define an event for a row, a column, or any subset of rows and columns. You can also use the row and column number to identify a specific cell to define an event.

- 3 In the Assign Event section, complete these fields:

- Row – enter the keyword “all”.
- Column – enter “1”.
- Event Name – enter “Stock_Symbol”. The event name is a logical name that must be unique across the entire portal. You must have separate names for distinct events. You can reuse the name for different cells to have the same event generated.

Note Do not use spaces in event names.

- With – select “dynamic value” from the drop-down list. The value that gets transmitted on an event can be Dynamic, Static, or HREF. The differences between the types of data that can be sent in an event are:
 - Dynamic – sends the text you click on as the event data.
 - Static – sends a constant value you specify in the text box to the right of the drop-down list as the event data. If you select this option, you must enter a value.

- HREF – if the text for which you are defining the event is a link, the underlying URL is sent as the event data.
 - Notify Now – send the event immediately.
 - Notify Across Page – send the event across pages if there is a listener on another page; otherwise, look for events on the current page. for this example, do not select this option.
- 4 Click Add to add the event. After a few moments, the first column turns blue. This means the event has been defined for the column.
- 5 Define an event on the second column:
- Row – enter the keyword “all”.
 - Column – enter “2”.
 - Event Name – enter “Stock_Company”.

Note Do not using spaces in event names.

- With – select “dynamic value” from the drop-down list. The value that gets transmitted on an event can be Dynamic, Static, or HREF. The differences between the types of data that can be sent in an event are:
 - Dynamic – sends the text you click on as the event data.
 - Static – sends a constant value you specify in the text box to the right of the drop-down list as the event data.
 - HREF – if the text for which you are defining the event is a link, the underlying URL is sent as the event data.
 - Notify Now – send the event immediately.
 - Notify Across Page – send the event across pages if there is a listener on another page; otherwise, look for events on the current page. for this example, do not select this option.
- 6 Click Add to add the event. After a few moments, the second column turns blue. This means the event has been defined for the second column.
- For more details on click-across and defining events, see “Building click-across messaging portlets” on page 96.
- 7 Click Next to continue and see a preview of the table with the defined events.

- 8 Click Finish to close the Define Events wizard. The Application Builder displays the two defined events beneath the Most Actives portlet in the Portlets list.

❖ **Google Stock Search portlet post processing**

The Google Stock Search portlet will be used to listen for certain click-across events and perform a search for the data it receives with the event. You can define a listener for the portlet outside of the Application Builder or within the Application Builder.

Note Any post processing added to a portlet in the Application Builder also effects the portlet outside of the Application Builder. See the notes at the end of this tutorial for more details.

- 1 In the Application Builder, right-click the Google Stock Search portlet and select Define Listeners to open the Listener Editor.
- 2 Select these options:
 - List of Events – “StockCompany”.
 - Listener param – “Search for”.
 - Auto Submit – select, then click the plus sign to enable the listener.

Note If Auto Submit is enabled, a listener portlet re-executes when it receives an event. If Auto Submit is disabled, the portlet does not re-execute right away.

- 3 Click OK to save the settings. The listener you defined displays beneath the Google Stock Search portlet in the Application Builder Portlets list.

Note To remove a listener from the portlet, click the associated X in the Action column.

❖ **Yahoo Stock Quote portlet post processing**

The Yahoo Stock Quote portlet will be used to listen for certain click-across events and performs a stock symbol search at <http://finance.yahoo.com> for the data it receives with the event. You can define a listener on the portlet outside of the Application Builder or within the Application Builder.

Note Any post processing added to a portlet in the Application Builder also effects the portlet outside of the Application Builder. See the notes at the end of this tutorial for more details.

1 In the Application Builder, right-click the Yahoo Stock Quote portlet and select Define Listeners to open the Listener Editor.

2 Select these options:

- List of Events – “Stock_Symbol”.

Note With most Listeners, you can select an event from the list or enter a name manually into the Event Name field.

- Listener param – “Symbol”.
- Auto Submit – select, then click the plus sign to enable the listener.

Note If Auto Submit is enabled, a listener portlet re-executes when it receives an event. If Auto Submit is disabled, the portlet does not re-execute right away.

3 Click OK to save the settings. The listener you defined displays beneath the Google Stock Search portlet in the Application Builder Portlets list.

Note To remove a listener from the portlet, click the associated X in the Action column.

❖ **Saving the application**

1 In the Application Builder, click Save to save the application you built. This opens the Application Properties window.

2 Complete these fields:

- Name – enter “StockApp”.

- Type – select Portlet Flow.

Note When you select Page Flow, you are prompted to specify the number of frames. Set the number of frames to the number of pages in the application and select Auto Initialize so the listener self registers.

- Status – select Approved.
 - Roles – select PortalUser and click Add to move that role from the Available Roles list to the Assigned Roles list.
- 3 Click OK to save your entries.

❖ **Previewing the application**

- To preview your new application, click Preview in the Application Builder. This opens the application in a new window that lets you test-drive a preview of your application.

❖ **Editing and deleting applications**

- 1 To edit an application, select Build | Applications.
- 2 Select the status of the application you want to edit.
- 3 In the detail view, select the application and click Edit.
- 4 To disable or inactivate an application:
- 5 Select the application in the detail view and click Edit.
 - a When the Application Builder displays, select Properties, change the Status to Deleted, then click OK.
 - b Click Save to implement your change, then click Close to exit the Application Builder.
 - c Select Build | Page Groups and change the status of the Page Group associated with the application to Deleted or mark the Page Group as Inactive.
 - d Click Update on the Page Group Manager toolbar to update Portal Interface with the changes.

Working with data-capable elements

Data-capable elements conform to the Portal Studio internal data format. The element types that can be data-capable are Web, XML, and database. The wizards for data-capable elements allow you to add split, define, and filter rules on the element content. The output of data-capable elements enables additional capabilities, such as assignment of portlet labels to element fields and the formatting of HTML output with cascading style sheet class definitions.

Database elements are data-capable by default. Web elements are data-capable when the visible text can be assigned a tabular structure. Data-capable options appear in the Web Element wizard as unformatted tables with a grey background. If the data-capable option does not appear, select an option and click Gridify. This forces the visible text in the element into a table that is then data-capable.

❖ **Creating data-capable Web element content**

- 1 Select Build | Portlets from the Portal Studio left pane.
- 2 Click the New icon on the Portlet Manager toolbar.
- 3 When the Portlet Builder displays, right-click in the left pane and select New Web Element.
- 4 Enter `www.yahoo.com` in the Location field and click Find.
- 5 Click Next.
- 6 Click any list.
- 7 When the captured options display, select the second Select option and click Gridify. You see your selection displayed as a table.

XML elements are data-capable if the XML format conforms to the internal Portal Studio Document Type Definition (DTD). If the XML does not conform to the Portal Studio DTD, a content XSLT can be used to perform an XML to XML transformation that converts the source XML to the correct XML format. Data-capable XML elements have a default HTML presentation and do not require an UI XSLT. You can override the default HTML presentation by specifying an UI XSLT, alone or in combination with a content XSLT.

❖ **Creating data-capable XML element content**

- 1 Select Build | Portlets from the Portal Studio left pane.
- 2 Click the New icon on the Portlet Manager toolbar.

- 3 When the Portlet Builder displays, right-click in the left pane and select New XML Element.
- 4 When the XML Element Definition window appears, enter the following input as an example:
 - XML URL –
`http://p.moreover.com/cgi-local/page?c=Biotech%20news&o=xml`
 - Content XSLT – click Select. When the Search window displays, click Search, select any XSL template you have created from the Results pane, and click Select.
 - UI XSLT – `pink_ui_xslt`
- 5 Click Next, then click Finish.
- 6 When you return to the Portlet Builder, click the down arrow beside the Save button to save the new element. See “Saving portlets” on page 82 for specific instructions.

Data-capable post-processing options

When you create any data-capable element except an HTML element, options appear during portlet creation that allow you to isolate specific pieces of element content on which you can perform these post-processing options:

- Adding split rules – allows you to split columns or rows using commas, spaces, and so on.
- Defining record layout – allows you to define the record layout and specify a record header.
- Applying newly created filter rules – allows you to add rules to filter the records so only the specified data displays.

Adding split rules

Split rules allow you to add parameters by which to split the columns or rows in the table. The Add Split Rules options are:

- Split – split the table by All Rows, a specific Row No. (enter the row number in the text box after the Split drop-down list), All Columns, or a specific Column No. (enter the column number in the text box).

- Split by Delimiter – split the rows or columns by Line Feed, Space, Comma, or Other. When you select Other, specify the delimiter to use in the text box to the right of the delimiter drop-down list.
- Personalization – enabled when you select the Variable option. Personalization adapters retrieve values from external systems or databases and automatically submit them to the portlet. Personalization adapters must be registered with Portal Studio before they appear in the Personalization drop-down. Once you select a Personalization adapter, the available methods within the adapter display in the drop-down. Select a method to specify the parameter within the Personalization adapter to submit to the portlet. See

To remove a split rule, select the rule in the Current Split Rules section and click Remove. Portal Studio removes the highlighted rule. To remove multiple rules, press the Ctrl key while you select rules and click Remove.

Defining record layout

Record layout allows you to define the format of grid elements. When you have captured an object that you have formatted as a grid (table), click Next from the Add Split Rule window.

The Define Record Layout options include:

- Record Layout – specify record orientation; column- or row-based.
- Records Contain Labels – select if the records do or will contain field labels.
- Labels Are Displayed in Record – displays when you select Records Contain Labels. Enter the record number in which the labels display.

Click Preview to see the result of defining the record layout.

Adding filter rules

Click Next on the Define Record Layout window to the Add Filter Rules window, which allows you to create basic filters to include or exclude records and fields.

The Add Filter Rule options are:

- Include/Exclude – select the type of Filter Rule to create:
 - Include Records

- Exclude Records
- Include Fields
- Exclude Fields
- Position/Condition (record rules only) – select:
 - Number – to specify the position of the record to filter. Enter the numbers or ranges of numbers for the filter rule.
 - Where – to specify a condition for the filter rule. Select the field where the condition should be met.
- Select the Boolean argument for the filter rule.
- Enter the value for the filter rule in the last input field.

Click Add to add the filter rule. The preview highlights any records or fields that match the filter rules.

- Position/Label (field rules only) – select:
 - Number – to specify the position of the field.
 - Where – to specify a condition for the filter rule.
 - Label – to include or exclude a field specifically by the contents of its label.

❖ **Applying newly created filter rules**

Continue to use the data-capable Web element example used in the previous section.

- 1 Select Include Records from the Include/Exclude drop-down.
- 2 Select Number from the Number/position drop-down.
- 3 Enter “1– 4” in the Number field. When you click Preview, records 1– 4 are highlighted in the preview.
- 4 Click Next. The element preview displays only those records that are defined by the filter rules.
- 5 Enter the name under which you want to save the element in the Element Name field, and click Finish.
- 6 When you return to the Portlet Builder, click the down arrow beside the Save button to save the new element. See “Saving portlets” on page 82 for specific instructions.

Creating portlet drop-down lists for input fields

The drop-down /select list functionality enables you to predefine input values for portlet input parameters. When a portlet requires that a user enter a value before receiving a result, the user can select one of several predefined values from a drop-down list.

There are four types of portlet input fields that you can convert to drop-down lists—text fields, password, select lists, and text area.

❖ Creating a portlet drop-down/select list

This example creates a stock quote portlet with a drop down list that lets the user select from several stock symbols.

- 1 Start Portal Studio by entering this URL in Internet Explorer 5.0 or later:

```
http://HOSTNAME.DOMAIN:8080/onepage/index.html
```

- 2 Enter your user name and password to log in.

- 3 Select Build | Portlets from the Portal Studio left pane.

Click New on the Portlet Manager toolbar.

- 4 When the Portlet Builder displays, right-click anywhere in the window and select New Web Element. You see the Web Element Definition window.

- 5 Enter a beginning uniform resource locator (URL)—the Internet address—of the Web content you want to capture in the Location text field, then click Find. For this example, enter

```
http://finance.lycos.com/qc/default.aspx
```

- 6 When the Quote.com Web page displays, enter SY in the Symbol Lookup box and click Go. Stock quote information displays for Sybase.

- 7 Click Next in the New Web Element window.

- 8 Click the left mouse button on Sybase, Inc. below Quote.com, indicated by a shadow-box that trails the mouse arrow and says “Select it” and click Next.

- 9 Click the light blue button to the left of the first capture option. The button changes to a darker blue when you select it. Click Next.

- 10 You see the CGI parameters window. Complete these fields:

- Variable – select this option next to the “symbols” name.
- Display Name – enter `Stock Symbol`

- **Default Value** – a comma-separated list of the options you want to display in the drop-down list. For this example, enter:

```
SY, MSFT, ADSK, GE
```

- **Type** – select Drop Down
- **Required** – select this option.

Click Next.

- 11 Enter “Stock Quotes” for the element name, then click Finish. You return to the Portal Builder where the new element displays.

Click the down arrow beside the Preview button and select Playback. You see a preview of the portlet. Select the drop-down list to see the options you created.

- 12 Click Save to save the new portlet.

❖ **Changing items in a drop-down list**

- 1 Select Build | Portlets from the Portal Studio left pane.
- 2 Select the status of the portlet you want to edit, select the portlet in the detail view, and click Edit.
- 3 When the Portlet Builder displays, click Params. You see the first of several screens that allow you to edit the portlets input parameters. Follow the instructions on the screen.

Enabling portlets and pages to support multiple languages

To enable EP portlets and pages to support multiple languages simultaneously, use UTF-8 encoding. UTF-8 allows users to place portlets from different languages on the same page and also allows portlet names in different languages.

EP portlets are preconfigured with ISO encoding, which typically supports only one or more closely-related language sets.

To enable the portal for UTF-8 support:

- 1 Open *global.properties.xml* in WordPad on Windows and in any text editor on UNIX. This file is located in *%JAGUAR%\Repository\WebApplication\onepage\config* on Windows and in *\$JAGUAR/Repository/WebApplication/onepage/config* on UNIX.
- 2 Search for the “use_utf8” property and set the value to “true”, which sets the portal to use UTF-8 globally. For example, change this:

```
<Property name="use_utf8" value="false" description="(true/false). true to enable global to set the Portal to use UTF-8" menugroup="-1"/>
```

to this:

```
<Property name="use_utf8" value="true" description="(true/false). true to enable global to set the Portal to use UTF-8" menugroup="-1"/>
```

- 3 Save *global.properties.xml* and close the text editor.
- 4 Set the charset property and emailCharset property to UTF-8 for each co-brand you have listed in *cobrand.xml*.

Note The *cobrand.xml* file contains a definition for each co-brand (portal). See Chapter 14, “Creating Co-brands,” for more information.

- a Open *cobrand.xml* in MS WordPad on Windows and in any text editor on UNIX.

The default *cobrand.xml* is located in *%JAGUAR%\Repository\WebApplication\onepage\config* on Windows and in *\$JAGUAR/Repository/WebApplication/onepage/config* on UNIX.

- b Set the charset property to “UTF-8”.
- c In *cobrand.xml*, set the e-mailCharset property to “ISO-xxxx-x”.

Note E-mail clients typically cannot interpret UTF-8 encoded e-mail messages. However, if your e-mail client can interpret UTF-8 encoded e-mail messages, you can set this value to UTF-8 also.

For example, change this:

```
<CobrandDef rid='1' name='Sybase'  
  dir='/fw/cobrand/onepage-1'  
  charset='ISO-8859-1'  
  emailCharset='ISO-8859-1'  
  emailerSetting='1' >  
</CobrandDef>
```

to this:

```
<CobrandDef rid='1' name='Sybase'
  dir='/fw/cobrands/onepage-1'
  charset='UTF-8'
  emailCharset='ISO-8859-1'
  emailerSetting='1' >
</CobrandDef>
```

- 5 Set the correct character set in *styles.xml* for each co-brand you have created. The default *styles.xml* is located *%JAGUAR%\Repository\WebApplication\onepage\fw\baseapps* on Windows and in *\$JAGUAR/Repository/WebApplication/onepage/fw/baseapps* on UNIX.

If you have more than one co-brand, you must change the character set in the *styles.xml* file for each one. The *styles.xml* for each co-brand is stored within that portal's co-brands subdirectory; for example, on Windows *%JAGUAR%\Repository\WebApplication\onepage\fw\cobrands\japanese-11*.

An example of this step is to change this:

```
<Styles
  charset="charset=iso-8859-1"
  charset_ie="charset=ISO-8859-1"
  charset_nn="charset=ISO-8859-1"
  customerCareEmail="customercare@Sybase.com"
  documentDomain="sybase.com" />
```

to this:

```
<Styles
  charset="charset=UTF-8"
  charset_ie="charset=UTF-8"
  charset_nn="charset=UTF-8"
  customerCareEmail="customercare@Sybase.com"
  documentDomain="sybase.com" />
```

Web browsers and UTF-8

- UTF-8 encoding works successfully in Internet Explorer versions 5.0, 5.5, and 6.0. The Internet Explorer client machine must have the language font installed.
- If you use Netscape and have set the server to use UTF-8 encoding, you must install UTF-8 fonts. MS Windows fonts typically do not work.

Charting Portlets

Enterprise Portal allows you to create graphical charts using data present in existing portlets. When you chart a portlet, EP creates an image from the parameters you specify, then displays the data in a graph that is easy to read and analyze.

Note Charting portlets is available only in Portal Studio.

Chapter	Page
Introduction	141
Creating portlet charts	142
Turning off chart caching	147

Introduction

The charting portlet wizard lets you chart portlets without programming. Charting portlets lets you create graphical representations of tabular data. Data can display in one of three chart formats—bar, pie, or line. You can create a chart for portlets containing any element type that has input parameters and a grid or table with structured output.

You can chart any number of variables, as long as the data is contained in a tabular form (a grid or table).

Note You cannot remove a chart definition once it has been created.

Creating portlet charts

After you create a portlet that contains tabular data, you can create a chart using the Charting Portlet wizard. The charting wizard takes you through steps to select which data to use, the chart type to display, and which rows or columns to interpret into the specified chart format. When the wizard completes, you can see a preview of the chart, after which you can save the chart itself.

Warning! If you are running Enterprise Portal on Solaris and before you start EAServer, you must open an X-window and set the DISPLAY variable for portlet charts to display correctly. See your operating system documentation for instructions.

❖ Creating a portlet with tabular data

Before you create a chart, you must create a portlet from which to create the chart. This procedure describes how to create a portlet containing tabular data. It uses `http://bonds.yahoo.com` as an example. Your own portlets can use any site containing tabular data.

- 1 Log in to Portal Studio as a user with the StudioAdmin roles.

Note The user `opsuper` cannot create portlet charts. To create a new user with the StudioAdmin role, log in as the user `ps0` (default password `123qwe`) and create a new user with that role assigned.

- 2 Select Build | Portlets in the Portal Studio left pane. When the right pane finishes loading, click New to open the Portlet Builder.
- 3 When the Portlet Builder displays, click Add to open the New Web Element window.
- 4 Enter `http://bonds.yahoo.com` in the Location text box, then click Find.
- 5 Select Grid from the Format drop-down list, then click Next.
- 6 The next window displays several grids from which you can select. Select the table showing U.S. Treasury Bonds quotes, then click Next.

The remaining windows of the wizard allow you to set rules to filter the data. Because the bonds chart already contains information in an acceptable format, no additional rules are necessary.

- 7 Click Next until the final wizard window displays a preview of the captured portlet.
- 8 Enter an element name and click Finish. This example uses the name “Yahoo-Bonds.”
- 9 When you return to the Portlet Builder, click Save.
- 10 Complete the options in the Finish window. Make sure that the iFrame and No Popup options are selected, and that you add any required roles to the Assigned Roles list. See “Saving portlets” on page 82 for definitions of the fields in the Finish window.
- 11 Click Finish to save the portlet.
- 12 Click Close to exit the Portlet Builder.

Creating a bar graph chart

These steps reference the Yahoo-Bonds portlet created as an example in the previous section, but you can use any portlet that contains tabular data.

❖ **Creating a bar graph of the Yahoo-Bonds portlet**

- 1 In Portal Studio, select Build | Portlets from the left pane.
- 2 In the detail view, select Status | New to locate your portlet. Right-click the portlet you just created and select Create Chart.

The Charting Portlet wizard opens in a new window and displays all available structured grid, table, or list objects included in the current portlet’s content.

- 3 Click Select to choose a grid, table, or list object and click Next.
- 4 Complete the these fields:
 - Chart Title – enter a title.
 - Chart Height – accept the default height, which is in pixels, or enter a different value.
 - Chart Width – accept the default width, which is in pixels, or enter a different value.

- Font Name – accept the default font—Sans Serif—or enter a different font name. The font you specify must be a True Type font and the font must be installed on your local machine.

Note TrueType fonts are device-independent fonts stored as outlines. You can size the fonts to any height, and they print exactly as they appear on the screen.

- Chart Legend – select where you want the chart legend positioned—None (no legend), Top (above the chart), Bottom (below the chart), Left (to the left of the chart), or Right (to the right of the chart).
- Chart Type – select whether the chart should be Bar, Line, or Graph. For this example, select Bar.
- Create an Image Map – select this option to create a clickable image map of the chart from which users can access the originating data.

When you select this option, you are required to complete these additional fields:

- Category Event Name – the name for the event
 - Notify Now – send the event immediately.
 - Notify Across Page – send the event across pages if there is a listener on another page; otherwise, look for events on the current page. for this example, do not select this option.
 - Series Event Name – the name of the series.
 - Notify Now – send the event immediately.
 - Notify Across Page – send the event across pages if there is a listener on another page; otherwise, look for events on the current page. for this example, do not select this option.
- 5 Click Next when you finish with your entries. The Define Chart window appears.

The Preview section displays the original data from which the chart is built. The Chart Parameters section contains the fields where you define the chart's appearance.

- 6 Complete the Chart Parameters:
- X-Axis Label – enter labels for the X-axis and Y-axis of the chart. Our example labels the X-axis “Maturity” and the Y-axis “Yield”.

- **Category Labels** – drop-down list lets you choose which data set to use as a label for the bar graphs. This example uses “Column 1”, for bond maturity rates.
- **Show Values** – check box displays the literal number of the data above each bar; leaving it unchecked hides this information.
- **Series** – Each Series represents one bar color of information. Choose meaningful names for display in the legend, and assign each data set a unique color for easy identification. This example uses columns 2, 3, 4, and 5 as series information, naming them “yield”, “yesterday”, “last week”, and “last month” respectively.

Note The Category Labels drop-down list determines which data you can select in Series. If the category label is “None,” choose any row or column for data. If the category label is a column, choose only columns. If the category label is a row, choose only rows.

- 7 Click Next for a preview of your chart. To save the results, click Finish.

A confirmation appears, indicating that the portlet has been saved and updated. Click OK.

Creating a pie graph charting portlet

These steps reference the Yahoo-Bonds portlet created as an example in “Creating a portlet with tabular data” on page 142, and assume that the portlet is newly created. You can use any portlet containing tabular data.

The steps for creating a pie chart are identical to those for creating a bar graph, with the following exceptions:

- 1 After you enter a title in the Chart Title field, select Pie Chart as the format.
- 2 When the Define Pie Chart window appears, complete the three drop-down lists.
 - **Data** – select which row or column to display as data. For this example select “Column 2,” the Yield column.
 - **Labels** – select which row or column to use as labels in the legend. For this example, select “Column 1,” the Maturity column.

- Format – add predefined labels to each section. For this example, select “Name and Percent” as labels to display the names of the categories, as well as the size each individual slice occupies in the pie chart.
- 3 Click Next for a preview of your chart. To save the results, click Finish.
A confirmation appears, indicating that the portlet has been saved and updated. Click OK.

Creating a line graph charting portlet

These steps reference the Yahoo-Bonds portlet created as an example in “Creating a portlet with tabular data” on page 142, and assume that the portlet is newly created. You can use any portlet containing tabular data.

The steps for creating a line graph are identical to those for creating a bar graph, with the following exceptions:

- 1 After you enter a title in the Chart Title field, select Line Graph as the chart format.
- 2 When the Define Chart window appears, complete these options in the Chart Parameters section:
 - X-Axis Label –
 - Y-Axis Label –
 - Category Labels –
 - Show Values –
 - Series – each series represents one line color of information. The data can be any column or row in your table. Choose meaningful names for display in the legend, and assign each data set a unique color for easy identification. This example uses columns 2, 3, 4, and 5 as series information, naming them “yield”, “yesterday”, “last week”, and “last month” respectively.
 - Click Next for a preview of your chart. To save the results, click Finish.

A confirmation appears, indicating that the portlet has been saved and updated. Click OK.

Turning off chart caching

Because EAServer caches static images, text may not display correctly for your chart. If your chart text displays incorrectly, use this procedure to resolve the problem.

1 Start Jaguar Manager, select Tools | Connect | Jaguar Manager, then enter:

- User Name – jagadmin
- Password – leave blank.
- Host Name – the name of your local machine.
- Port Number – 9000

Click Connect.

2 Expand the Servers folder, right-click the Jaguar tree node, and select Server Properties.

3 Select the Static Page Caching tab and do either of the following:

- Deselect the Enable Static Page Caching option so no static pages or items are cached; or
- Prevent caching of items stored in */onepage/jspfilter* by entering the following into the Exclude WebApp Files text box:

```
(onepage/jspfilter, *.*)
```

4 Select File | Refresh Static Cache.

5 Exit Jaguar Manager, then restart EAServer.

From there on, charts are no longer be cached.

Personalizing Portlet Content

Personalization allows you to configure portlet parameter input values to be filled in from adapters that extract values from other sources; for example, values that have been stored in the back-end database. When you configure a portlet input field to use a personalization adapter, the adapter is invoked at runtime to provide values for preconfigured key fields. The input fields of portlet instances that belong to different users can receive input values based on each user, providing customization of a portlet's content and its execution on a per-user basis.

Topic	Page
Introduction	149
Creating new personalization adapter keys	151
Building portlets with personalization adapters	152
Entering user-specific values in Portal Studio	160
Importing and exporting personalization keys	161

Note Content personalization is also available to Portal Interface users when they select MyInfo. See the *Portal Interface User's Guide* for details.

Introduction

To use personalization adapters, first you create new key values for the adapter you want to use, then you create a portlet that provides input values using that key. Later, users can personalize the key by entering their own default value for the portlet's input field. For example, you have a portlet where you enter the postal code to see the weather for the associated area. It uses a database table adapter to provide the postal code input field. Each user can edit the key and enter their own personal postal code so the local weather displays each time he or she accesses the weather portlet in Portal Interface.

To use the personalization feature, select the following from the Portal Studio main window:

- **Manage | Personalize** – select from the left pane to create, edit, manage, or delete personalization keys for each adapter type.
- **Build | Portlets** – select from the left pane to create portlets that pull input values from personalization keys.
- **Account Info** – to manage and customize your personalization values.

Adapter types

Enterprise Portal provides four adapters:

- **Security proxy** – key values (for example, user name, password, URL) allow users to log in to back-end applications using their ProxyAuthenticationInfo (PAI), which uses the key values as input parameters to portlets to enable single sign-on to back-end systems. This option is useful when the user wants to log in to an application through a portal that requires a user name and password different from their portal login. This adapter is available only in the Enterprise Edition.
- **User name/password** – the keys shown for this adapter are automatically created when a user logs in to Portal Interface or Portal Studio. Users cannot create, edit, or delete keys. The key values are stored in memory and maintained only for the user's current login session. The values are reset each time a user logs in or changes their password. The key values can be used when a user logs in to Portal Studio or Portal Interface, then wants to log in to a portlet during the same session that requires a login with the same user name and password.
- **Database table** – uses the back-end database to store keys and user values. You can create an input field with no default values, a single default value, or a drop-down list of default values from which a user can select.
- **EJB** – uses the Sybase Personalization Web service to store and retrieve personal key values. The EJB adapter uses only basic features of the Sybase Personalization service. The service itself has features and functionality well beyond the other personalization adapters. The EJB adapter uses a subset of those features. This adapter is available only in the Enterprise Edition.

Note The Sybase Personalization service is useful across a wide range of applications—not just the portal. It is a complete EJB application. Sybase software other than EP uses this service to store personal information of any nature. For example, the iAnywhere Wireless server uses this service to store user’s mobile device characteristics (phone number, phone type, notification schedule preferences, and so on).

Creating new personalization adapter keys

Select Manage | Personalize from the Portal Studio left pane.

The center pane lists personalization adapters extracted from the *uwp.xml* portal configuration file. The right pane lists the keys configured for the selected adapter.

The security proxy, database table, and EJB adapters do not have default keys installed. To use these adapters, you must create a new key.

❖ Creating new keys

- 1 Log in to Portal Studio. To access Portal Studio, enter the following URL in your Internet Explorer browser:

`http://HOSTNAME.PORTALDOMAIN:8080/onepage/index.html`

For example, if your machine name is “tahiti”, and your portal domain is “sybase.com,” enter:

`http://tahiti.sybase.com:8080/onepage/index.html`

Note If you are logging in to a production system, the port number may be different or may not be necessary at all in the login address.

If you are running the Demo version of Enterprise Portal, enter:

`http://demo.sybase.com:4040/onepage/index.html`

- 2 When the Portal Studio Login window displays, enter the user name and password for a user with portal administrator privileges.
- 3 Select Manage | Personalize from the Portal Studio left pane.
- 4 Select the adapter type for which you want to create a new key.

- 5 Click New on the Personalize Manager toolbar.
- 6 When the Create New Key window displays, select or complete these options:
 - Name – enter a unique key name. The name used by portlets along with the user ID to locate personalized input values for different users.
 - Type – select the type of input required for the value.
 - Text – the input field requires a text string.
 - Int – the input field requires whole numbers.
 - Select – the input field allows users to select a value from a drop-down list of predefined values.
 - Password – the input field requires a password. The value is shown as ***** when typed in the user’s personalized section.
 - Permission – select whether the key can be edited or is read-only. Read-only is appropriate for any user name and password keys.
 - Description – enter a text description of the key.
- 7 Click OK to save your entries.

Building portlets with personalization adapters

When a user adds a portlet to a personal page, they can personalize the input field to always use a specified value.

You select which personalization adapter and key to use during the element definition process for Web, database, JSP, and Web services portlets.

The remainder of this section describes how to use each adapter type.

Using the security proxy adapter

This section illustrates how to use the security proxy personalization adapter by creating single-sign-on access to a back-end accounting system.

Create an asset in Enterprise Security that logically represents the back-end application to which you want to construct single sign-on access, then create a personalization key for the security proxy adapter.

❖ Creating a new security asset

- 1 Log in to Portal Studio as a user with LIST, READ, and UPDATE permissions on the AssetCtrlAsset; for example, log in as the PSO with the “pso” user name and the “123qwe” password.
- 2 When the Portal Studio main window displays, select Administer | Organizations.
- 3 Select Assets in the Organization Manager menu, then click New.
- 4 When the Create New Asset window displays, enter or select:
 - Asset Name – a name for the asset. For example, “Accounting”. You could also enter the actual name of the back-end accounting system.
 - Asset Type – select either Unspecified or URL.

Note Access controls for ProxyAuthInfo attach to assets checks if the user has LIST, READ, and UPDATE permissions on the asset. To grant those permissions, you must choose an asset type with at least these three permissions associated with it, which the Unspecified and URL types have. Do not use the assets with names prepended with “EP”, which do not have those permissions.

- Description – a text description for the asset.
- 5 Click OK.

❖ Granting permissions on the new asset

Before anybody can associate their own personal ProxyAuthenticationInfo objects with an asset, they must have LIST, READ, and UPDATE permission on that asset.

- 1 In Portal Studio, select Administer | Organizations if it is not already selected.

Select Assets from the Organization Manager menu, then right-click the newly created asset and select Manage Access Permission.
- 2 When the Manage Access Permission on Asset window displays, grant:
 - LIST and UPDATE permission to studio administrators who need to create asset-level proxy authentication autofill adapter keys on the asset.

- READ permission to everyone who should have SSO access into the target (accounting) system.

Note You can grant the READ permission through one or more roles. Before any user can create their own ProxyAuthenticationInfo for this asset in Portal Interface (using MyInfo | Personalize), they must have a role with READ permission.

3 Click OK.

❖ **Copying the asset's distinguished name**

Copy the asset's distinguished name (DN) so you can paste it into a dialog box when you create the autofill keys.

- 1 Click the new asset in the Organization Manager right pane and click Edit. The full Asset DN is shown in the New Asset dialog box.
- 2 Highlight the Asset DN and press Ctrl+C on your keyboard or right-click and select Copy. You must know the full DN when you create autofill keys for this asset—the key names include the full DN.
- 3 You can change the Asset DN to anything you want as long as it is a unique asset name across all assets in the Enterprise Security Access Control Database (ACDB). For example, the original DN might have been:

```
a1=Accounting,dc=sybase,dc=com
```

which you can change to a simpler:

```
Accounting
```

Note Sybase recommends that you shorten the DN so administrators and portal users can more easily understand the key's purpose.

❖ **Creating the personalization key**

The security proxy adapter administrative implementation is based on asset-level ProxyAuthenticationInfo definitions. This insures that any Portal Studio user can view ProxyAuthenticationInfo personalization keys as they configure portlet parameters; that is, anyone with READ permission on an asset can access the asset-level ProxyAuthenticationInfo.

- 1 In Portal Studio, select Manage | Personalize from the left pane.
- 2 Select Security Proxy from the Personalize Manager Adapters menu, then click New.

- 3 When the Create New Key window displays, enter or select:
 - Name – the name of the asset for which you are creating the ProxyAuthenticationInfo key. If the asset’s common name is unique within Enterprise Security, that is sufficient. If the asset’s common name is not unique, you must enter the asset’s complete distinguished name. For our example, put your cursor in the field and press Ctrl+V to paste in the full DN that you copied from the Edit Asset dialog box.
 - Type, Permission, Description – these values are ignored.
- 4 Click OK. The personalization adapter creates an asset-level ProxyAuthenticationInfo key on the asset.
- 5 When the message displays that the personalization key was saved successfully, click OK.

When you create a ProxyAuthenticationInfo key, the portal generates three personalization keys—user name, password, and URL.

In the Key Name column you see three new keys created for this adapter; for example:

```
password/a1=Accounting,dc=sybase,dc=com
URL/a1=Accounting,dc=sybase,dc=com
userid/a1=Accounting,dc=sybase,dc=com
```

There are three keys generated from the one asset name you entered in the Create New Key window. These keys correspond to the three attributes of the PAI. These keys are marked READ ONLY because you cannot modify the key name—the name is tightly associated with the asset to which the ProxyAuthenticationInfo key is attached.

Warning! If you have UPDATE permission, do not select one of these keys, and select Delete. This action will destroy the ProxyAuthenticationInfo key along with the three associated keys.

Note You cannot use Manage | Personalize to edit key values. However, you can use SMAPI ProxyAuthenticationInfoManagement beans to modify the values.

❖ Using the ProxyAuthenticationInfo personalization keys

- 1 In Portal Studio, select Build | Portlets.

- 2 Right-click anywhere in the Portlet Manager detail view and select New Aggregated Portlet.
- 3 When the Portlet Builder displays, click the down arrow to the right of the Add button and select JSP Element.

Note This example uses the JSP element, but any element type that uses CGI parameters (Web, database, Web service) can be used with the personalization autofill keys.

- 4 The JSP Element Definition window displays. For example, select Use Web Application and fill in the name of your WAR file. Add a Web App Display Name and the Initial Resource (the JSP page that initially displays).

- 5 Enter input parameters using this format:

```
parametername=X&parametername=Y&parametername=Z
```

where *parametername* is the actual name to display for each parameter. For example:

```
URL=X&USERNAME=Y&PASSWORD=Z
```

The X, Y, and Z are default input values and can be anything. The “&” separate each parameter definition.

- 6 Click Next when you are satisfied with your entries.
- 7 On the Finish window, enter the Element name, then click Finish.
- 8 Select the Configure option for each parameter that you want to configure.
- 9 Click Next. You see the Setup Configuration window.

You can reorder parameters, enter defaults, and perform other functions.

- Enter Setup Description HTML – enter a description to appear above the parameters in the portlet.
- Position – use the up and down arrows to reorder the parameters.
- Display Name – the name entered for the parameter. Set the Display Type to None to hide parameters.
- Default Value – the default value entered when you created the parameters during portlet creation. You can change these values here to any value to automatically populate the input field. If you leave the field blank, “Default” displays here.

- Display Type – select the input field type: Text Field, Password, Text Area, None.

Note If the display type is not Password or a password parameter, the actual password displays in preview windows, and so on. When you select the Password display type, you see only ***** displayed.

Also, password autofill keys are available from the Key drop-down list only when the Display Type for a parameter is set to Password. For example, the password/accounting key would not be available in the Key list for URL or user name because their Display Type is not Password.

- Personalize – select the adapter you want to use. For example, Security Proxy. Once you select an adapter, the available keys for that adapter display in the Key drop-down list.
 - Key – select a method to specify the parameter within the Auto-fill adapter to submit to the portlet.
 - Required – select whether the user must provide input for this parameter.
- 10 Click Next. You see the Setup Preview window with a preview of the portlet and parameters.
 - 11 Click Save. You return to the Portlet Builder.
 - 12 Click Save to save the portlet.
 - 13 When the Finish window displays, complete the options, then click Finish.
 - 14 Click Close to exit the Portlet Builder.
 - 15 When you return to the Portlet Manager, select New from the Status menu, right-click the new portlet, and select Approval Status | Approved.

Now you can add the portlet to a page and page group and update Portal Interface.

Using the user name/password adapter

Warning! This adapter does not work if you are using certificate-based authentication to access the portal (browser certificates, smart cards, biometric devices) because there is no user password. If your portal supports certificate-based user authentication, developers should use the security proxy adapter instead of the user name/password adapter.

Use the single sign-on adapter when you have a back-end application that shares the same user name and password as Enterprise Security, but is running on a separate machine, and is not just another application running within the same Web container.

For example, if your portal is configured to use the LDAP Authentication delegate instead of the Access Control Database (ACDB), and the LDAP delegate is pointed at your enterprise's LDAP server, you could use your PC user name/password to log in to the portal. Then you could use the same user name/password adapter key to create a portlet that gets you into another enterprise application that is also using the enterprise's LDAP server to authenticate users.

Using the database table adapter

The default database personalization adapter uses the back-end database to store values.

The database adapter is build on top of two tables—`personalization_keys` holds the key values; `personalization_values` reference the `personalization_keys` and holds the value for each user.

❖ Creating a database table adapter key

- 1 Log in to Portal Studio by entering the following URL in your Internet Explorer browser:

`http://HOSTNAME.PORTALDOMAIN:8080/onepage/index.html`

For example, if your machine name is “tahiti”, and your portal domain is “sybase.com,” enter:

`http://tahiti.sybase.com:8080/onepage/index.html`

Note If you are logging in to a production system, the port number may be different or may not be necessary at all in the login address.

If you are running the Demo version of Enterprise Portal, enter:

`http://demo.sybase.com:4040/onepage/index.html`

- 2 When the Portal Studio Login window displays, enter the user name and password for a user with portal administrator privileges.
- 3 Select Manage | Personalize.
- 4 Select Database Table from the Adapter menu and click New on the toolbar.

When the Create New Key window displays, complete these options:

- Name – a unique key name; for example, “Weather by postal code”.
 - Type – the type of input required for the value. Select “Int”.
 - Permission – whether the key can be edited or is read-only. Select “Editable”.
 - Description – a text description of the key; for example “Weather for selected postal code areas”.
- 5 Click OK.

Using the EJB adapter

If a postal code is something you want to use more widely than just inside the portal, the EJB adapter may be a better place to maintain the information.

Imagine that your payroll processing system runs a batch job (outside the portal) to automatically print and mail paychecks to all employees every two weeks. The part of that payroll application that prints the address needs to look up the employee’s address, including the postal code. It would be difficult to write the payroll application to use the postal code information from the Database adapter.

However, you could write such an application using the SybPersonalization EJBs:

```
InitialContext ctx = new InitialContext();
Personalization pers = (Personalization)
    ctx.lookup("SybPersonalization/Personalization");
Item[] zipcode = pers.getItems("pqui", "Zipcode");
```

Entering user-specific values in Portal Studio

This section explains how a user can personalize the values for any portlet that uses personalization adapter input fields.

❖ Personalizing input values

When a user is logged in to Portal Studio and has access to the newly created portlet, then can add their own personal default values for the input fields.

- 1 Log in to Portal Studio.
- 2 Click Account Info at the top of the window.
- 3 When the Account Info window displays, click Personalize. You see the Personalization window.

Each line represents personalization adapter keys for any portlets with personalization input fields which you can enter a default value. When you access portlets in Portal Interface that use these adapter keys, your personal default value displays in the input field.

Note For users that are not allowed to access Portal Studio, this information can also be entered, changed, or deleted in Portal Interface by clicking My Info and clicking Personalize.

To remove a key, select the X box to the right of the key's description. You cannot remove the Portal Login Name and Password keys.

To add a key, click Add to select from a list of available keys that you can personalize.

- 4 When you finish, click OK.

Importing and exporting personalization keys

When you use Portal Studio to export portlets that use personalization keys, any keys associated with the database adapter are exported along with the portlets. When you subsequently import those portlets to another portal, the database adapter keys are automatically created, and become available to personalize.

Keys for the other adapters are not automatically propagated in this way. If you import a portlet that uses security proxy or EJB keys from another portal, you must manually create those keys on the importing portal before you can use those portlets.

Building Templates

Templates allow you to predefine the position and layout for portlets that contain one or more elements. This chapter describes how to use the Portal Studio Template Manager, which is the GUI administrative tool you use to change template information.

Topic	Page
Introduction	163
Choosing a template type	165
Creating and editing templates	170
Changing the default template	172
Customizing user XML element templates	173

Introduction

Portlets that contain only one element use a system-assigned template specified in the *global.properties.xml* master configuration file. See “Changing the default template” on page 172 for more information.

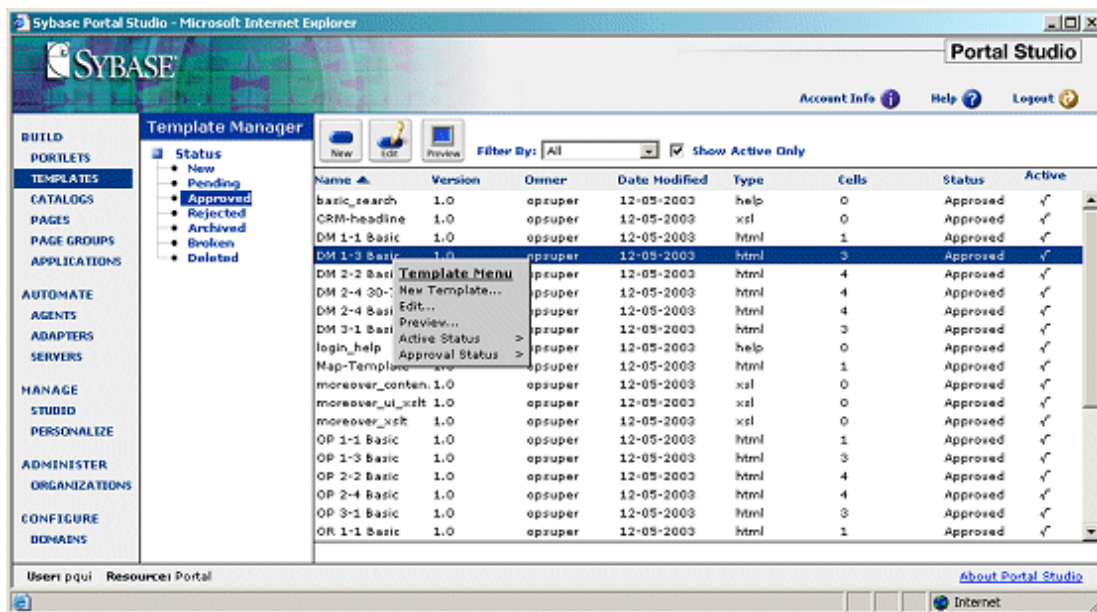
Aggregated portlets contain more than one element, and are defined using templates. A template is code (HTML, HELP, JSP, or XSL) that defines the position and layout of elements within a portlet.

Template Manager

Use the Template Manager to create and manage templates.

Select Build | Templates from the Portal Studio left pane to display the Template Manager.

Figure 8-1: Template Manager



The Template Manager interface includes:

- Template Manager Status menu – select a status to see templates for the selected workflow status: New, Pending, Approved, Rejected, Archived, Broken, or Deleted.
- Toolbar – displays New and Edit buttons that let you add, view, edit, or delete templates, and the Preview button that lets you preview a selected HTML template.
- Status bar – shows the current user and resource.
- Detail view – displays templates based on your selection from the Status menu, and the Filter By and Show Active Only options.
- Pop-up Template Menu – the options on this menu let you add and edit templates, and allow you to approve and activate specific templates. Only templates that are active and have an approved status can be applied to portlets.

Template Upload window

The Template Upload window is where you create and edit templates.

Click the New icon on the Template Manager toolbar, or right-click in the detail view and select New.

Portal Studio allows you to create these types of templates:

- HTML – HTML template.
- HELP – online help template.
- JSP – JavaServer Page (JSP) template.
- XSL – Extensible Stylesheet Language (XSL) template. XSL allows you to apply one style document to multiple portlets. XSL also lets you dictate the manner in which Web content prints, and lets you transfer XML documents across different applications.

Choosing a template type

This section describes the types of templates and how to create each type.

HTML templates

All portlets are wrapped in an HTML template to display in a Web browser. Other template types are available only when you are creating that type of portlet; that is, you can access JSP templates only when you are creating a JSP portlet.

When you create an HTML template that includes a table, the code must include the `<OPContent>` tag in each HTML table row description. The `<OPContent>` creates cells in which to add content. The `<OPContent>` tag uses this syntax:

```
<OPContent id="id_number" name="position_descriptor_name" />
```

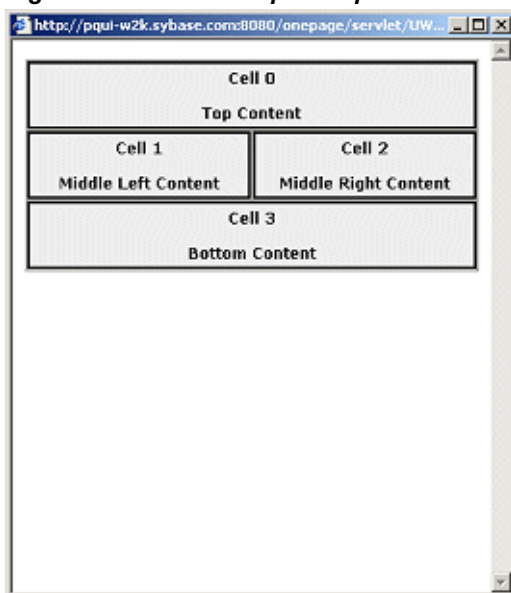
- *id_number* – a sequential number, starting with 0 (zero), which identifies table cells. The *id_number* of the first `<OPContent>` tag must be `id=0` and subsequent `<OPContent>` tags must be sequential.
- *position_descriptor_name* – a descriptive name that specifies the cell's position within the table.

The *id_number* and *position_descriptor_name* for each <OPContent> tag must be unique. Only content that would normally occur between the opening and closing <body> tags should be included in the HTML template.

This code represents a template that looks like Figure 8-2. The code between the <STYLE> uses the Cascading Style Sheet (CSS) classes that define the formatting of data-capable elements.

```
<STYLE TYPE="text/css">
.gridRowHeader{background:#DEDEB9;padding:4px;padding-bottom:0px;border:1px
solid #FFF;font-size:13px;font-family:verdana;color:#336699;font-weight:bold;}
.gridRowA{background:#EEF9FF;padding:4px;padding-bottom:0px;border:1px solid
#FFF;font-size:13px;font-family:verdana;}
.gridRowB{background:#FFFFFF;padding:4px;padding-bottom:0px;border:1px solid
#FFF;font-size:13px;font-family:verdana;}</STYLE>
<table width="100%" border="1" cellspacing="0" cellpadding="0">
<tr align="center" valign="middle">
<td height="10" colspan="2"><OPContent id="0" name="Top Content"/></td></tr>
<tr align="center" valign="middle">
<td height="10" width="50%"><OPContent id="1" name="Middle Left Content"/></td>
<td height="10" width="50%"><OPContent id="2" name="Middle Right Content"/>
</td></tr>
<tr align="center" valign="middle">
<td height="10" colspan="2"><OPContent id="3" name="Bottom Content"/>
</td></tr></table>
```

Figure 8-2: HTML example template



HELP templates

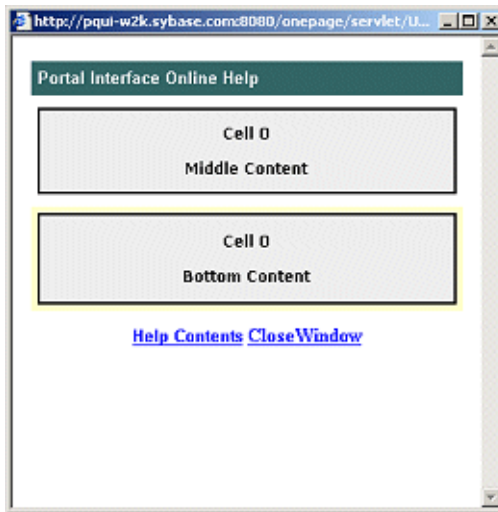
Help templates are HTML files that you save to the database and display when a user clicks the help icon on a portlet's title bar in Portal Interface.

When you save a new element, one of the fields asks for the Help URL. You can either enter the URL of a Web page or select a Help template from the database.

The HTML code shown next represents a template that looks like Figure 8-3.

```
<script language="JavaScript">
<!--
function openWindow(URL)
{
window.open(URL,"PopUp","scrollbars=yes,resizable=yes,width=400,height=400";
}
function Winload()
{
}
//-->
</script>
<body bgcolor="#FFFFFF" text="#000000">
<table width="100%" border="0" cellspacing="5" cellpadding="5" height="190">
<tr>
<td colspan="2" height="27" bgcolor="#336666">
<p><font color="#FFFFFF" size="2" >
<b><font face="Arial, Helvetica, sans-serif">Portal Interface Online
Help</font></b></font></td></tr>
<tr height="16">
<td bgcolor="#FFFFFF" colspan="2" height="20">
<OPContent id="0" name="Middle Content"/>
<b><font face="Arial, Helvetica, sans-serif" size="2"></font></b>
<hr noshade width="100%" size="1"></td></tr>
<tr>
<td colspan="2" bgcolor="#FFFFCC"><OPContent id="0" name="Bottom Content"/>
<p><font face="Arial, Helvetica, sans-serif" size="2"></font></p></td></tr>
<tr bgcolor="#FFFFCC"><td width="100%" bgcolor="#FFFFFF">
<p align="center"><b>
<a href="JavaScript:openWindow('index.html')">
<font face="Arial, Helvetica, sans-serif" size="2">Help Contents</a></font></b>
<imgsrc="images/spacer.gif" width="122" height="6">
<a href="javascript:window.close();"><b>CloseWindow</b></a></font></td></tr>
</table>
```

Figure 8-3: HELP template



JSP templates

You can create Web applications and deploy them into the portal as a portlet. Alternatively, you can use JSP files for small reusable pieces of functionality. The JSP approach requires you to create a *.jsp* file with embedded Java and HTML code. Servlets or other JSPs can invoke the JSP code by referring to the file.

JSP-based templates provide a way to manage JSP code across portlets. You can create templates without managing individual files in the file system.

A JSP template has a name and a definition, which are saved and managed by Portal Studio. Once you create a template, you can populate the template with portlets. Those portlets can be included in other portlets. When you change a template definition, those changes are applied universally to all the associated portlets of that template.

XSL templates

Portal Studio allows you to create an XSL template (XSLT) that you can use for XML portlets and elements. Code the XSLs for data-capable elements to use the Enterprise Portal internal document type definition (DTD).

Warning! The only XSL stylesheet supported to create XSL templates is:

```
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"
  version="1.0">
```

If you use a later stylesheet, a parsing error is generated because of the version of Xerces used.

The following XSL code is a sample of an XSL template for a data-capable user interface XSLT.

```
<?xml version="1.0"?>
<xsl:stylesheet
  xmlns:xsl="http://www.w3.org/1999/XSL/Transform"version="1.0">
  <xsl:template match="/">
    <TABLE WIDTH="100%" BORDER="1" BORDERCOLOR="#000000">
      <xsl:for-each select="//Record">
        <xsl:text> </xsl:text>
        <TR><xsl:for-each select="Field"><xsl:text></xsl:text>
          <TD>
            <xsl:for-each select="./text()">
              <xsl:value-of select="."/>
            </xsl:for-each>
          </TD>
        </xsl:for-each>
        <xsl:text> </xsl:text>
      </TR>
    </xsl:for-each>
  </TABLE>
</xsl:template>
</xsl:stylesheet>
```

Creating and editing templates

This section describes how to create, edit, and delete templates. There are also procedures for changing a template's status.

Note You can also create templates on the fly when you create and edit aggregated portlets. Do not create templates in this way when you create secure portlets—you may receive a JavaScript error. See Chapter 4, “Building Portlets,” for details.

❖ Creating a template

- 1 Select Build | Templates from the Portal Studio left pane.
- 2 Click New on the Template Manager toolbar to launch the Template Upload window.
- 3 Complete these fields:
 - Name – enter a name for the new template. For example, you might want to name a template “2-4 Basic” for a portlet with two columns and four elements.
 - Roles – assign the appropriate roles to the template. When you assign a role to a template, all users assigned that role can access the template in Portal Studio.

Note Once you add a portlet to a page, the roles assigned to the page and portlets supersede the roles assigned to the underlying template. A template's roles pertain only to which roles can access templates in Portal Studio.

- Type – select from HTML, HELP, JSP, or XSL.
 - Status – select the template's status. You can only add portlets and elements to approved templates.
 - Code – enter the code that defines this template. Templates are stripped of header tags, so include only the content that normally resides between the opening and closing <BODY> tags.
- 4 If this is an HTML or HELP template, click Preview to see the template preview in a separate window.
 - 5 Click Save to save the new template definition.
 - 6 Click Close to exit the window.

❖ Editing a template

- 1 In the Template Manager, select the status of the template you want to modify.
- 2 Double-click the template listing in the detail view to open the Template Editor window.
- 3 Change the Status, Type, Assigned Roles, or Code, and activate or deactivate the template.
- 4 When you are satisfied with your changes, click Preview to review your modifications if this is an HTML or HELP template.
- 5 Click Save to save your changes under the same name, or click Save As to save the template under a different name.
- 6 Click Close to exit the Template Editor.

❖ Changing workflow status

If you are using the Portal Studio's workflow feature, you can change a template's approval status as it moves through the workflow process.

- 1 Right-click a template in the Template Manager's detail view, select Approval Status from the pop-up Template Menu, then select the appropriate status; or
- 2 Select a template in the Template Manager's detail view, click Edit on the toolbar, select a new status from the Status drop-down list in the Template Editor window, then click Save.

❖ Changing a template's active status

You cannot add a portlet to a template unless the template is approved and active.

- 1 Right-click a template in the Template Manager's detail view, select Active Status from the pop-up Template Menu, then select the appropriate status; or
- 2 Select a template in the Template Manager's detail view, click Edit on the toolbar, select or deselect Active, then click Save.

❖ Deleting a template

When you delete a template, it remains in the database, and displays in the detail view with a status of "Deleted." To mark a template as deleted:

- 1 Right-click a template in the Template Manager's detail view, select Approval Status from the pop-up Template Menu, then select Deleted; or

- 2 Select a template in the Template Manager's detail view, click Edit on the toolbar, select Deleted from the Status drop-down list box in the Template Editor window, then click Save.

To change a template's deleted status, select Deleted from the Template Manager's Status menu, right-click the template in the detail view, select Approval Status, then select a new status.

Changing the default template

Template1 is the default template automatically assigned to portlets (see Chapter 4, "Building Portlets"). This system-assigned template is the value specified for the DefaultTemplateName property in the *global.properties.xml* portal configuration file, which is located in `%SYBASE%\EAServer\Repository\WebApplication\onepage\config\` on Windows and in `$$SYBASE/EAServer/Repository/WebApplication/onepage/config/` on UNIX.

You can change the default to any template you want. For example, if you want every portlet to use the same header logo or footer, you can create a template with the logo, and enter that template's name in *global.properties.xml*.

Note When you change the template name in *global.properties.xml*, the new default template is used by all resource IDs in Portal Studio.

❖ Changing the default portlet template

- 1 Create and save a new portlet template using the instructions in "Choosing a template type" on page 165.
- 2 Use any text editor to open *global.properties.xml*.
- 3 Search for DefaultTemplateName and replace the value "Template1" with the name of the aggregated portlet template you created.
- 4 Save the file.
- 5 Restart the application server.

Customizing user XML element templates

Enterprise Portal allows you to customize XSL and XSLT templates on a per-user basis.

When you create XML portlets, you specify an Extensible Style Language Transformation (XSLT) template for content transformation and a user interface (UI) template for presentation.

The XSLT template contains the language used in XSL style sheets to transform XML documents into other XML documents. The UI template specifies how to present the portlet; that is, shading, borders, and so on.

In earlier versions of Enterprise Portal, the XSL and UI templates were selected and defined in the Portal Studio during portlet design and were fixed for all users of the portlet. With Enterprise Portal 6.0, you can customize these templates on a per-user basis for XML portlets.

❖ Creating an XML sample portlet

- 1 Log in to Portal Studio.
- 2 Select Build | Portlets from the Portal Studio left pane and select New from the Portlet Manager Status menu.
- 3 Right-click in the detail pane and select New Aggregated Portlet.
- 4 When the Portlet Builder displays, click the down arrow beside the Add button and select XML Element from the menu.

You see the XML Element Definition window.

- 5 Enter the following sample input:

XML URL –

```
http://p.moreover.com/  
cgi-local/page?c=Biotech%20news&o=xml
```

Content XSLT – `moreover_content_xsl`

UI XSLT – `pink_ui_xslt`

- 6 Click Preview.

The content has a pink background, as specified in the UI XSLT. Any user that accesses an XML portlet that used the specified UI XSLT sees a pink background.

- 7 Click Next until you reach the window where you enter the Element Name.

- 8 Enter “XML_sample” and click Finish to save the new element.
- 9 When the Portlet Builder reappears, click Save to save a portlet that uses the XML_sample element.
- 10 On the Finish window, enter a portlet name and select the J2EE roles:
 - Portlet Name – enter “XML_sample.”
 - J2EE Roles – click Add All.
- 11 Click Finish.
- 12 When you see the prompt “Portlet successfully saved,” click OK, then click Close to exit the Portlet Builder.
- 13 In the Portlet Manager, select New from the Status menu. You see the XML_sample portlet.
- 14 Approve and Activate the portlet. Right-click XML_sample portlet and select:
 - Active Status – Active.
 - Approval Status – Approved.

The portlet now displays under the Approved status.

❖ **Adding the XML_sample portlet to a page**

You have created an XML_sample portlet that uses the XML_sample element and incorporates the pink_ui_xslt user interface. Now add the portlet to a page.

- 1 Select Build | Pages from the Portal Studio main window and click New on the Page Builder menu bar.
- 2 When the Page Builder appears, click Add.
- 3 When the Search window appears, click Search.
- 4 Click XML_sample in the Results pane and click Add.
- 5 In the Page Builder, click Save As.
- 6 In the Save Page window:
 - a Enter “XML_sample” for Name.
 - b Click Add All for J2EE Roles.
 - c Click OK.
- 7 When the “Page saved” prompt appears, click OK, then click Close to exit the Page Builder.

- 8 At the Portal Studio main window, select New from the Manager Status menu.
 - 9 Right-click XML_sample in the detail view and select Status | Approved from the pop-up. When the “Page saved” prompt appears, click OK.
 - 10 Select Approved from the Page Manager Status menu.
 - 11 Click XML_sample in the detail view and click Update from the Page Manager menu bar.
 - 12 Enter the appropriate user name and password and click Update.
- Portal Interface now has the page available for users.

❖ Creating a user-specific UI XSLT template

- 1 Select Build | Templates and select Approved from the Template Manager Status menu.
- 2 Right-click the pink_ui_xslt template in the detail view and select Edit from the pop-up.
- 3 When the Template Editor appears, locate this line in the Code window:


```
<TD BGCOLOR="#ffccff" >
```
- 4 Change the value from “#ffccff” to:


```
#ccffcc
```
- 5 Click Save As.
- 6 When the Save Template As dialog box displays, enter:


```
green_ui_xslt
```

 for the name, then click OK.
- 7 When you see the prompt “Template successfully saved,” click OK, then click Close to exit the Template Editor.

Enabling different templates for the same element

To apply different templates to the same element, create the *xsltinfo.properties* file and populate it with new template and user information.

This file should have the following content and structure:

```
<userid>.<portlet id>.ui=<xslt file>
<userid>.<portlet id>.content=<xsl file>
```

where:

- *userid* – is the login name of the user.
- *portlet id* – is the ID of the portlet for which you want to specify a user-specific UI or content XSLT. To determine a portlet's ID:
 - a In Portal Studio, select Build | Portlets, then select Approved from the Portlet Manager Status Menu. Right-click the XML_sample portlet in the detail view and select Properties.
 - b When the Portlet Properties window displays, write down the ID value, then click Close.
- *content* or *UI* file – enter the user-specific template for the portlet.

For example, to set up the file for the XML_sample portlet, enter this information:

```
bob.231.ui=pink_ui_xslt
cindy.231.ui=green_ui_xslt
```

where *pink_ui_xslt* is the name of the presentation file for Bob, and *green_ui_xslt* is the name of the presentation file for Cindy.

Place the *xsltinfo.properties* file in the `%%SYBASE%\EAServer\Repository\WebApplication\onepage\config` directory on Windows and in `$$SYBASE/EAServer/Repository/WebApplication/onepage/config` on UNIX, then restart EAServer.

When each user that is specified in *xsltinfo.properties* adds the XML_sample page to their portal view, Bob will see the XML_sample portlet with a pink background and Cindy will see the XML_sample portlet with a green background.

To create user-specific content, follow the same procedures in this section, but create a new content template and specify the two different content files in the *xsltinfo.properties* file; for example:

```
bob.231.content=moreover_content_xsl
cindy.231.content=new_content_xslt
```

You can also provide different presentation and different content for different users in this file.

```
bob.231.ui=pink_ui_xslt
bob.231.content=moreover_content_xsl
cindy.231.ui=green_ui_xslt
cindy.231.content=new_content_xslt
```

The first time you create *xslt.properties*, you must shut down and restart EAServer to initialize the user-specific settings. When you make additions, changes, or deletions to the file the portal is updated automatically with the new information; that is, you do not need to restart the application server

Catalogs allow you to group existing portlets. This chapter describes how to work with Portal Studio and Portal Interface catalogs.

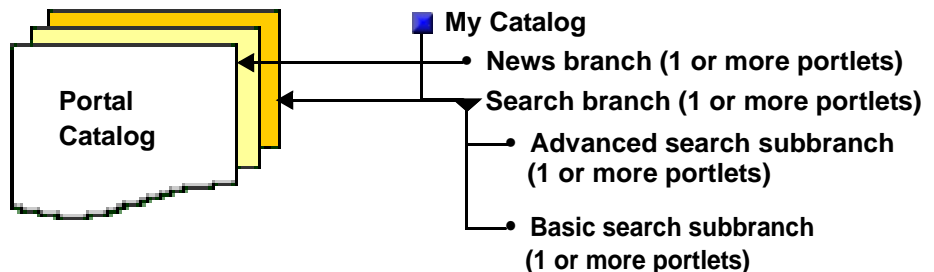
Topic	Page
Introduction	179
Creating and editing catalogs	182
Updating Portal Interface with changes	186

Introduction

You create catalogs in Portal Studio. A catalog contains portlets grouped in user-specified branches. You can arrange portlets into different branches and subbranches within a catalog for sorting and organization.

Although you can create an unlimited number of catalogs, you can display only one catalog per resource in Portal Interface. The catalog that is downloaded to Portal Interface—the display catalog—is the only catalog in Portal Studio with an approved status and marked as active.

Figure 9-1: Catalog branch and subbranches illustration



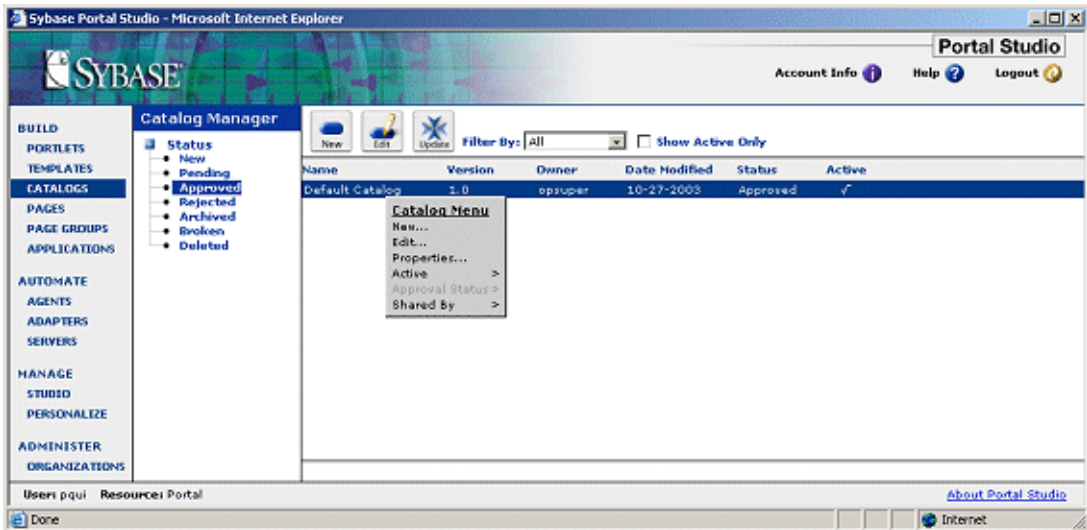
When portal users select Add Portlets in Portal Interface, they can browse through the catalog and add portlets of their choosing to their personal pages and page groups. Users cannot add portlets to predefined or default pages or page groups.

A catalog can contain portlets associated with any role. In Portal Interface, a user sees only the portlets that have access roles assigned to the user.

Catalog Manager

Use the Portal Studio Catalog Manager to create new catalogs and perform management functions for existing catalogs. To access the Catalog Manager, select Build | Catalogs from the Portal Studio left pane.

Figure 9-2: Catalog Manager



The Catalog Manager interface includes:

- Status menu – select a status to see catalogs with that status displayed in the Catalog Manager detail view.
- Toolbar – displays the New and Edit buttons that let you add, view, and edit catalogs. If you change the display catalog (that is, mark the current approved display catalog as inactive and mark another approved catalog as active), Update lets you update the portal with a new display catalog, which can be the only approved catalog that is marked active. In the detail view, you can also filter whether to display only your catalogs of a selected status, all catalogs of a selected status, or only the active catalog.
- Status bar – shows the current user and resource.

- Detail view – displays catalogs based on your selections from the Status menu, and the Filter By and Show Active Only options.
- Right-click Catalog Menu – right-click on the detail view to display the Catalog Menu. The options let you add and edit catalogs, view catalog properties, mark an approved catalog as active, and change a catalog's approval status.

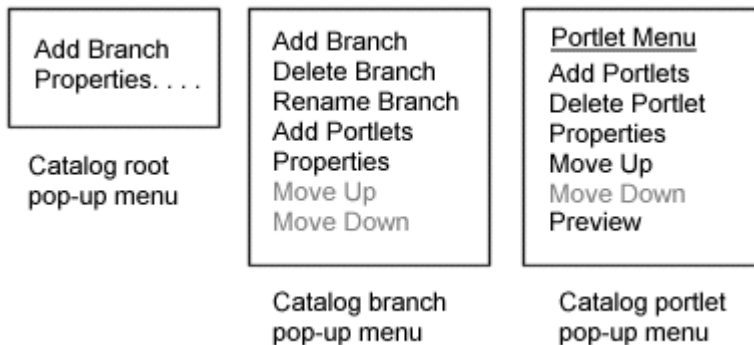
Catalog Builder

To launch Catalog Builder, click New on the Catalog Manager toolbar or right-click in the detail view and select New from the pop-up.

The Catalog Builder interface includes:

- Catalog tree – displays the catalog branch and subbranch order. You can add, delete, rename, and move branches, subbranches, and portlets, which display in the portlet detail view.
- Toolbar – save an existing catalog with the same name and assigned J2EE roles, or use Save As to save a new or existing catalog with a new name and new J2EE roles. You can also use the options on the toolbar to add branches and portlets, edit or view portlet properties, and preview portlets in a separate browser window.
- Portlet detail view – displays the portlets of the branch selected in the catalog tree.
- Pop-up menus – right-click the top of the catalog tree, a branch in the tree, or on a portlet in the detail view to access the functionality of each menu.

Figure 9-3: Catalog Builder pop-up menus



Creating and editing catalogs

Note When you add, modify, or delete portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes. See “Updating Portal Interface with changes” on page 186.

To create a new catalog, do one of the following:

- Right-click in the Catalog Manager detail view and select New from the Catalog Menu. The Catalog Builder window opens.
- Click New on the Catalog Manager toolbar. The Catalog Builder launches.

The catalog tree shows only an untitled catalog. The portlet detail view is empty because no portlets or branches have been created for the new catalog.

Use a branch to organize portlets within a catalog. As you create new branches, keep these rules in mind:

- You must create a branch for a new catalog before you can add portlets.
- A branch can contain other branches or portlets.
- You can have both portlets and branches beneath a single branch.
- You can create a new branch beneath the top level of a catalog, or you can create a branch beneath other branches.

- You cannot create branches beneath portlets.

❖ Adding a new branch

- 1 In the Catalog Builder, click the catalog top level to add a branch, or an existing branch to add a subbranch. If this is a new catalog with no branches, there is only one level to select—Untitled Catalog.

Note When the Catalog Builder is launched for a new catalog, the catalog tree shows only an untitled catalog. The portlet detail view is empty because no portlets or branches have been created off the new catalog.

- 2 Click the down arrow next to the Add icon and select Add Branch, or right-click the catalog's top level and select Add Branch from the pop-up menu.
- 3 Enter a name in the text field that appears. The default name is New Branch.
- 4 Click OK.

❖ Adding a portlet to a catalog

Portlets can only be added to branches of a catalog. Only portlets that are approved and marked as active can be added to a catalog.

The portlet list displays the portlets that are assigned to a branch. You can add, delete, or move portlets in the portlet list. The columns displayed in the Portlet List are Name, Version, Owner, and Date Modified.

- 1 In the catalog tree, select an existing branch to which you want to add a portlet.
- 2 Click Add on the Catalog Builder toolbar to launch the Find Portlet window. You can also:
 - Right-click the branch and select Add Portlets from the pop-up menu; or
 - Select the branch, click the down arrow next to the Add button, and select Add Portlets; or
 - Select the branch, right-click in the portlet detail view, and select Add Portlets from the Portlet Menu.
- 3 Fill in the criteria by which you want to search for existing portlets, then click Search. Portlets that meet the search criteria display in the Search Results pane.

Select Advanced Search to display additional search criteria:

- Modified By – search for portlets modified by a particular user.
 - Status – because only Approved portlets can be added to a catalog, you cannot change this option.
 - Title URL – search for portlets with a specific title URL.
- 4 In the Search Results pane, click a listing to select a portlet. To select several portlets, hold down the Ctrl key and click each listing you want to select.
 - 5 Click Add to move the selected portlets to the portlet list in the detail view of the Catalog Builder.

❖ **Saving a new catalog**

- 1 Once you are satisfied with a new catalog's contents, click the down arrow to the right of the Save button and select Save As.
- 2 Enter a unique name for the catalog.
- 3 Select Versioning to save the catalog as a new version.
- 4 Enter an optional description and notes.
- 5 Click a role in the Available Roles list to select that role, then click Add to add the role to the Assigned Roles list. Click Add All to move all of the available roles to the Assigned Roles list.

Note Only the users assigned the roles in the Assigned Roles list can access this catalog.

To allow every portal user to access the Add Portlet functionality, add the PortalUser role to the Assigned Roles list. Every portal user is guaranteed to have the PortalUser role.

See the *Enterprise Security Administration Guide* for information on setting up Enterprise Security to work with the portal.

- 6 When you are satisfied with your entries, click OK.
- 7 When the confirmation prompt confirms that the catalog was saved successfully, click OK.
- 8 Return to the Catalog Manager and select New from the Catalog Manager Status menu. You see the new catalog in the detail view. When you create a catalog in Portal Studio, the catalog is automatically saved with a status of New.

❖ Viewing or changing a catalog's properties

- 1 To view or edit a catalog's properties, right-click a catalog in the Catalog Manager detail view and select Properties from the pop-up.

Change any of the following options:

- Status – to display a catalog in Portal Interface, change the status to approved.
- Shared – this option is reserved for a future release.
- Active – make this the active catalog that displays in Portal Interface. Only one catalog can be the display catalog and it must have an approved status and be marked active.
- Description/Notes – enter a note or description.

- 2 Click OK to save your changes.

❖ Changing a catalog's status

To display a catalog in Portal Interface, it must have a status of approved and be marked active.

- 1 In the Catalog Manager, select New from the Status menu.
- 2 In the detail view, right-click the catalog and select Approval Status | Approved from the pop-up.
- 3 Select Approved from the Status menu. The catalog displays in the approved detail view.
- 4 To make this catalog display in Portal Interface, right-click the catalog in the detail view and select Active | Yes.
- 5 To update Portal Interface with the new catalog contents, click Update on the Catalog Manager toolbar.

❖ Moving an existing branch

When a catalog displays in Portal Interface, the portlets display in the order in which they appear beneath a branch in the catalog tree. Move branches or portlets up and down according to how you want the branches and their associated portlets to appear in Portal Interface.

- 1 In the Catalog Manager, select the status of the catalog you want to edit.
- 2 In the detail view, right-click the catalog you want to change and select Edit from the Catalog Menu. The catalog displays in the Catalog Builder.
- 3 In the Catalog tree, select the branch you want to move.

- 4 Right-click and select Move Up or Move Down from the pop-up.

❖ **Deleting an existing branch**

- 1 Select the branch you want to delete.
- 2 Right-click and select Delete Branch from the pop-up.

❖ **Renaming an existing branch**

- 1 Select the branch you want to rename.
- 2 Right-click and select Rename Branch from the pop-up.
- 3 Type the new name in the text field that appears.
- 4 Click OK.

❖ **Deleting a portlet from a catalog**

- 1 To remove a portlet, select the branch with that portlet in the catalog tree.
- 2 In the portlet list detail view, right-click the portlet to remove and select Delete Portlet from the pop-up.

Updating Portal Interface with changes

When you add or change portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes.

Each of these objects has an Update icon on the Manager toolbar; pages and page groups also have an Update Users option on the right-click menu that functions the same as the Update icon.

❖ **Updating changes to Portal Interface**

- 1 Select Build | Catalogs in the left pane.
- 2 When the Catalog Manager displays, select Update from the toolbar.
- 3 When a dialog box confirms that the update was successful, click OK.

Building Pages

This chapter describes how to use the Page Manager to create, edit, and populate portal pages.

Topic	Page
Introduction	187
Choosing the page type	190
Selecting a page layout	191
Creating and editing pages	194
Updating Portal Interface with changes	198

Introduction

A page is a collection of portlets that displays in a portal environment, such as Portal Interface. In Portal Studio, you use the Page Manager and Page Builder to create and manage portal pages.

In Portal Interface, pages only exist as part of a page group, although you can update Portal Interface with both individual pages and page groups.

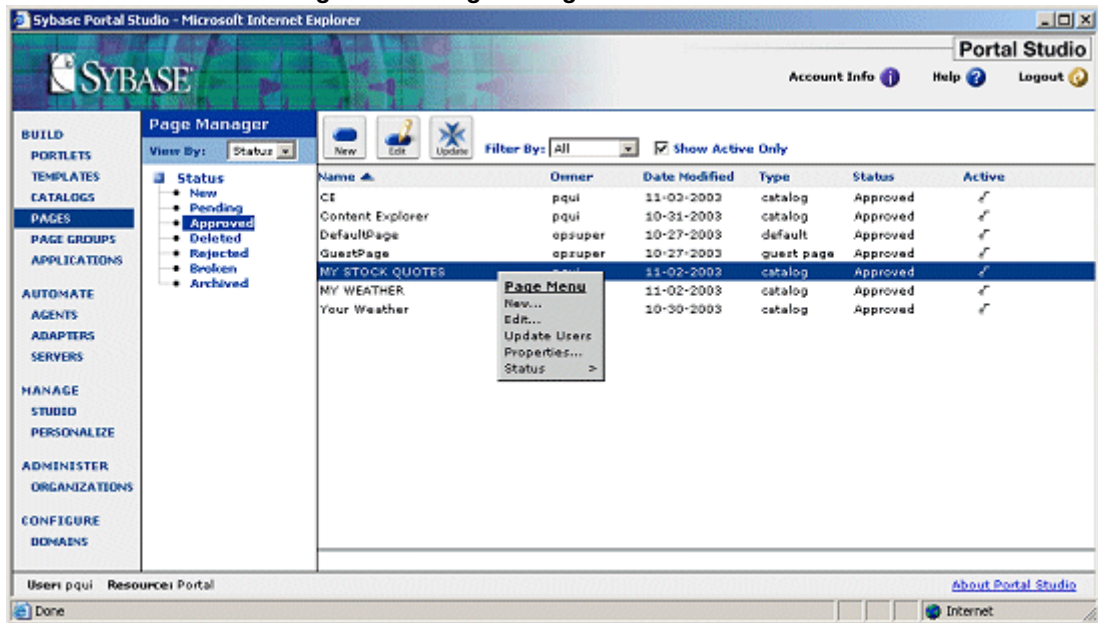
To create page groups and add a page to a page group, see Chapter 11, “Building Page Groups.”

Page Manager

Use the Page Manager to perform management functions for portal pages. Depending on your access rights, you can create, and edit pages and populate pages with portlets.

To display the Page Manager, select Build | Pages on the Portal Studio left pane.

Figure 10-1: Page Manager



The Page Manager interface includes:

- View By drop-down list – displays pages in the detail view by Status or Type.
- Page Manager menu – the options vary according to your selection in the View By drop-down list.
 - Status – see pages for the selected workflow status: New, Pending, Approved, Deleted, Rejected, Broken, or Archived.
 - Type – see pages for Default, Catalog, or Guest pages.
- Toolbar – displays the New and Edit buttons that let you add, view, and edit pages. Update lets you update the portal with active, approved new or changed pages. You can also filter whether to display only your pages, all pages, or only active pages in the detail view.
- Status bar – shows the current user and resource. Activities that you perform are associated with the displayed resource.
- Detail view – displays pages based on your selections in the View By drop-down list, the Page Manager menu, the Filter By, and Show Active Only options.

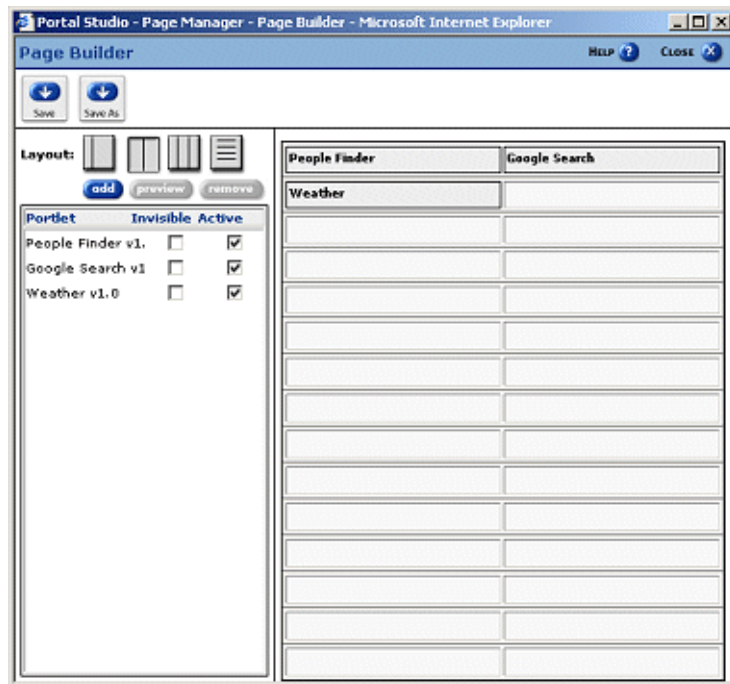
- Pop-up Page Menu – these options let you add and edit pages, update pages in Portal Interface, view page properties, and change a page’s status.

Page Builder

Use the Page Builder to create and edit portal pages. To access the Page Builder, select Build | Pages from the Portal Studio left pane, then click the New icon on the Page Manager toolbar, or right-click in the detail view and select New from the pop-up.

To access the Page Builder for an existing page, select the page from the detail pane, then click Edit.

Figure 10-2: Page Builder with portlets on page



The Page Builder interface includes:

- Toolbar – click Save to save an existing page with the same name. Click Save As to save a new page or save an existing page under a different name.

- Layout – select any of the four layout options (30/70, 50/50, 25/50/25, or Full) to change the page’s column layout. See “Selecting a page layout” on page 191.

Note Enterprise Portal 6.0 allows you to design pages with three columns in Portal Studio and Portal Interface. For Portal Studio instructions, see “Selecting a page layout” on page 191. For Portal Interface instructions, see the *Portal Interface User’s Guide*.

- Buttons – Add, Preview, and Remove portlets.
- Portlet list – the large box in the left pane lists the names of portlets you have added to this page.
 - Invisible – lets you hide a specific portlet from the end user. For example, you could set the display status of an portlet to invisible because the portlet is performing a server-side lookup operation.
 - Active – shows whether each portlet is active.
- The right pane displays a representation of the page. As you add portlets, smaller boxes that represent the portlets display according to the layout you have selected. You can drag and drop content from one cell to another and right-click any cell to select a custom cell width. See “Selecting a page layout” on page 191.

Choosing the page type

Portal Studio supports several page types: default pages, catalog pages, guest pages, and personal pages.

Note Default pages, catalog pages, and guest pages are created in Portal Studio. Personal pages are created in Portal Interface.

- Default – active and approved default pages are automatically added to new Portal Interface accounts. Portal users can use one or more default pages based on their roles.

- Catalog – predefined page available for users to add to their personal page groups by selecting Add Page in Portal Interface.

Note Do not confuse catalog pages with catalogs that group portlets. The default display catalog allows users to add portlets to their personal pages when they select Add Portlet in Portal Interface.

- Guest – guest pages display when you access the Portal Interface URL. These pages are visible to new users before they log in or register for an account.
- Personal pages – created by a user in Portal Interface. Personal pages do not display in Portal Studio, although a user can share their personal pages with other Portal Interface users. See the Portal Interface online help for instructions.

Selecting a page layout

Portal Studio lets you control the layout of a page, offering a variety of column widths and the ability to customize individual cell widths.

Note Portal Interface users can change the layout only of their personal pages; they cannot change the layout of system or catalog pages created in Portal Studio.

A portal page can have these layouts:

- 100 – the content occupies 100% of the Portal Interface browser window.
- 30/70 – 2 columns, with one column occupying 30% of the browser window, and the other column occupying 70%.
- 25/50/25 – 3 columns, with one column occupying 25% of the browser window, the center column occupying 50%, and the last column occupying 25%.

- 50/50 – 2 columns with each column occupying 50% of the browser window.

Warning! The page layout can display incorrectly if a user has not been assigned a role to view some of that page's portlets. The user may see only some of the pages portlets and cannot change the layout. You may want to keep this in mind when choosing a layout for pages that contain portlets that have been assigned different access permissions.

❖ Changing a page's layout

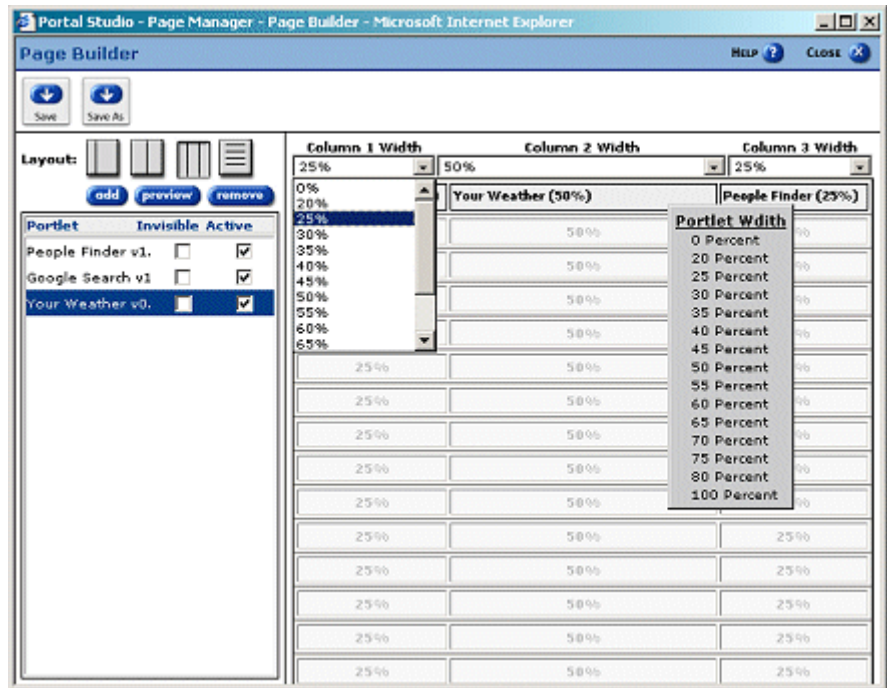
- 1 To change a page's layout, click a different layout choice. The selected layout displays in the page representation on the right side of the Page Builder.
- 2 In a 30/70, 25/50/25, or 50/50 layout, click the portlet box in the page representation and drag and drop the portlet from one side of the layout to the other.
- 3 Click Save.
- 4 Click Close to exit the Page Builder.

❖ Creating a three-column page

- 1 Log in to Portal Studio, select Build | Pages, then click New.
The default page layout of two equally wide columns displays.
- 2 Click the three-column Layout button.
The page switches to a three-column layout, with a default column percentage width of 25/50/25.
- 3 Click Add to begin adding portlets.

Note A three-column layout supports a maximum of 15 rows, which lets you add a maximum of 45 portlets to a page.

- 4 After you finish adding portlets, change the default column and portlet widths to suit your needs.



To change the width of a column, click the Column Width drop-down list above each column. To change the width of one portlet, right-click the portlet and select the width from the pop-up.

Note Once you set a custom cell width, the Column Width drop-down lists disappear. To redisplay the drop-down lists, click the three-column layout icon again.

Both column widths and portlet widths are specified in percentages of the page width. For example, the default width of 25% means that the column or portlet occupies 25 percent of the width of the page.

If you change the page layout from 3 columns to 2 columns on a page with 30 or fewer portlets, or move to one column from 2 or 3 columns on a page with 15 or fewer portlets, the portlets are randomly rearranged.

For example, if your page layout is like this:

A	B	C
---	---	---

D	E	F
G	H	I

and you remove the third column, the resulting two-column page looks like:

A	B
D	E
G	H
C	F
I	

The total percentage of cell widths in a row must equal 100%.

You can also set a column width to zero. This allows you to simulate having a row with only one or two columns, or lets you create two-column layouts with column widths other than 50/50 or 30/70.

When a user logs in to Portal Interface, they see the portlets laid out in the selected configuration.

Creating and editing pages

Note When you add, modify, or delete portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes. See “Updating Portal Interface with changes” on page 198.

Portal Interface users can add portlets only to their own user-created pages and page groups.

Note Do not mix insecure and secure portlets on the same page.

❖ Creating a page

- 1 Select Build | Pages from the Portal Studio left pane.
- 2 Click New on the Page Manager toolbar.
- 3 Click the icon that represents the layout you want for this page.

- 4 Click Add to populate the new page with portlets.
- 5 When the standard Search window displays, complete the Search criteria to display a list of active approved portlets in the Results window. Leave the search criteria blank to see all active approved portlets.
- 6 Click Search.
- 7 When the search results display, select a portlet in the Results pane. To select several portlets, hold down the Ctrl key while you click the portlets you want to add in the Results pane.
- 8 Click Preview to display the portlet in a separate browser window.
- 9 Click Add to add the selected portlets to the page you are building. You return to the Page Builder, which now displays the name of the portlets you added. You also see a representation of the portlets in the selected layout page representation on the right side of the window.
- 10 Select the layout. For more information, see “Selecting a page layout” on page 191.
- 11 In the portlet list, select or deselect these options:
 - Invisible – lets you hide a specific portlet from the end user. For example, you could set the display status of a portlet to invisible because the portlet is performing a server-side lookup operation.
 - Active – shows whether each portlet is active. Deselect Active to remove this portlet from the active list. The portlet will not display on the page until you reselect this option to check the box.
 - Select a portlet in the list and click Preview to view the portlet’s contents in a separate browser window.
 - Select a portlet in the list and click Remove to remove the portlet from the page.
- 12 When you are satisfied with your choices, click Save.
- 13 Complete or select these options:
 - Name – enter a unique page name.
 - Type – select Default, Catalog, or Guest Page from the drop-down list.
 - Active – select this option to mark this page active. When you mark a page as active, give the page the Approved status, and update pages to Portal Interface, this page is available to portal users, depending on their access rights.

- Roles – select roles from the Available Roles list and add them to the Assigned Roles list to define what roles can access this page. See the *Enterprise Security Administration Guide* for more information on security roles.

Note User accounts with the roles in the Assigned Roles list are the only user accounts that can access this page.

- 14 Complete or select these options. When you select any of these options, they apply to all portlets on the selected page.
- Enable Page Bar – displays the Change Layout, Add Portlet, Create Portlet, Share Page, Delete Page, and Refresh Page options to the left of the page group drop-down list.
 - Enable Refresh Page – puts the “Refresh Page” option on the page bar.
 - Enable Portlets Title Bar – adds a title bar to the portlets on this page.
 - Enable Portlets Title Bar’s Widget – adds widgets (graphic icons) that let the user maximize or minimize a portlet.
 - Enable Portlets Border – places a border around the portlets.
 - Portlets Border Color – accept the default color or enter a different hexadecimal code for the border color you want. You can find a list of Web-safe hexadecimal color codes at <http://html-color-codes.com/> or at <http://www.visibone.com/colorlab/>.
 - Enable Portlets Scroller – places a scroll bar on the portlet if the content is larger than the specified portlet size.

Note Although Portal Interface users can change the layout of their personal pages, they cannot change the page properties listed.

- Vertical Spacing – the percentage of empty page space to put between the portlet columns.
- Horizontal Spacing – the percent of empty page space above and below each portlet row. Horizontal spacing does not add space to the top of the page.
- Left Margin Spacing – the percent of empty page space to place on the left side of the page.

- Right Margin Spacing – the percent of empty page space to place on the right side of the page.

- 15 Click OK to save your entries.
- 16 Click OK again to save your entries on the Save Page window.
- 17 Click Close to exit the Page Builder.

❖ **Editing a page**

- 1 Double-click a page listing in the detail view, or
- 2 Right-click a page listing in the detail view and select Edit, or
- 3 Click to select a page in the detail view and click Edit on the toolbar.

❖ **Removing portlets from a page**

- 1 In the Page Builder, select the portlet you want to remove.
- 2 Click Remove to remove the selected portlet.
- 3 Click Save.
- 4 Click Close to exit the Page Builder.

❖ **Viewing or editing page properties**

- 1 To view or edit a page's properties, right-click the page listing in the Page Manager detail view and select Properties from the pop-up.
- 2 Edit any of the fields.
- 3 Click Properties to view and edit additional page parameters.
- 4 Click OK to save your changes.
- 5 Click OK again to save the changes from the first window of properties.
- 6 Click Close to exit the Page Builder.

❖ **Deleting a page**

- 1 Select the page's current status from the Page Manager Status menu to display the page listing in the detail view.
- 2 Right-click the page listing in the detail view, select Status, then select Deleted from the Page Menu.

Note The page listing moves to the Deleted status in the detail view. The page is not removed from the database.

- 3 To change the page's status once it has been deleted, select Deleted from the Page Manager Status menu.

Updating Portal Interface with changes

When you add, or modify portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes.

Each of these objects has an Update icon on the Manager toolbar. Pages and page groups also have an Update Users option on the right-click menu you access in the detail view.

❖ Updating page changes to Portal Interface

- 1 Select Build | Pages in the left pane.
- 2 When the Page Manager displays, right-click in the detail view and select Update Users.
- 3 When a dialog box confirms that the update was successful, click OK.

Building Page Groups

This chapter discusses how to create page groups.

Topic	Page
Introduction	199
Choosing a navigation style	203
Page groups, page numbers, and <IFRAME>	205
Creating and editing page groups	206
Updating Portal Interface with changes	210

Introduction

A page group is a set of pages that are grouped together. You can assign different roles to different page groups to restrict access to only the page groups relevant to specific users.

Page groups created by developers in Portal Studio are called *system page groups* (SPG), as opposed to page groups created by Portal Interface users, which are *personal page groups*. SPGs represent default or guest page groups.

For a system page group to get updated to Portal Interface, the “page.group” property must be set to “on” in *global.properties.xml*. When the page.group property is set to “off,” system page groups are still accessible in Portal Interface, but the Portal Studio update operation is disabled. See Appendix A, “Configuring Global Properties,” for more information on this file.

Warning! Do not set the page.group property to “off,” then back to “on.” These actions corrupt the Enterprise Portal onepage resource.

Pages can only be added to page groups. When the page.group property in the *global.properties.xml* file is set to “on”—the default, users see the page groups for which they have access rights.

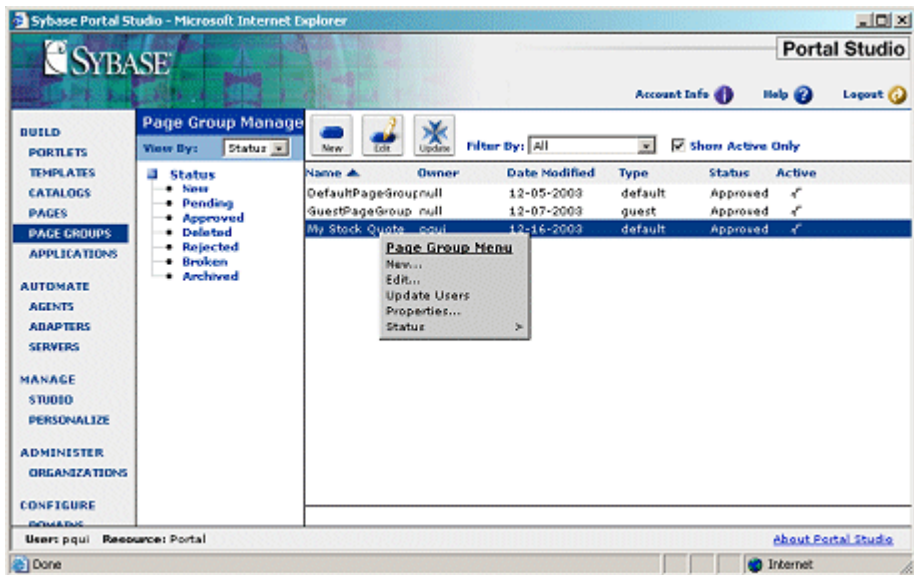
In addition, Sybase recommends that portal administrators set the “deferupdate” property in the *global.properties.xml* file to “on” to ensure that all users receive a refreshed view when they log in to Portal Interface. When deferupdate is set to “off”, the portal is refreshed based on only the current content and recent updates may not display immediately.

For more information on access rights and *global.properties.xml*, see Appendix A, “Configuring Global Properties.”

Page Group Manager

The Enterprise Portal Page Group Manager allows you to create a page group in Portal Studio, then deploy it to Portal Interface.

Figure 11-1: Page Group Manager



The Page Group Manager interface includes:

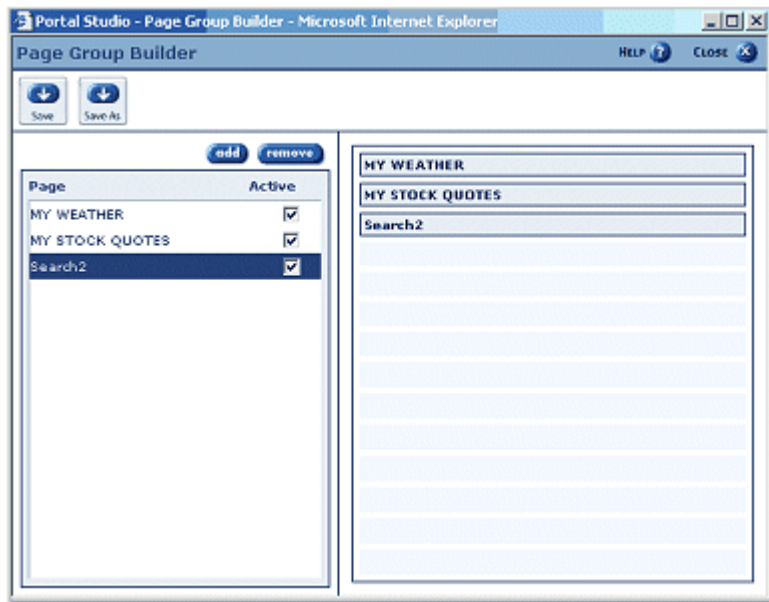
- View By drop-down list – displays page groups in the detail view by Status or Type.
- Page Group menu – the options vary according to your selection in the View By drop-down list.
 - Status – display page groups for the selected workflow status: New, Pending, Approved, Deleted, Rejected, Broken, or Archived.

- Type – display page groups for Default or Guest page groups.
- Toolbar – displays the New and Edit buttons that let you add, view, edit, or delete page groups. Update lets you update the portal with active, approved new or changed page groups. You can also filter whether to display only your page groups, all page groups, or only active page groups in the detail view.
- Status bar – shows the current user and resource. Activities that you perform are associated with the displayed resource.
- Detail view – displays page groups based on your selections in the View By drop-down list, the Page Group Manager menu, the Filter By, and Show Active Only options. You can filter whether to display only your page groups, all page groups, or only active page groups.
- Pop-up Page Group Menu – these options let you add and edit page groups, update page groups to Portal Interface, view page group properties, and change a page group's status.

Page Group Builder

Use the Page Group Builder to create and edit portal page groups. To access the Page Group Builder, select Build Page Groups from the Portal Studio left pane, then click the New icon on the Page Group Manager toolbar or right-click in the detail view and select New from the pop-up.

To access the Page Group Builder for an existing page group, select the page group from the detail pane, then click Edit.

Figure 11-2: Page Group Builder

The Page Group Builder interface includes:

- **Toolbar** – click Save to save an existing page group with the same name. Click Save As to save a new page group or save an existing page group under a different name.
- **Buttons** – Add and Remove pages.
- **Page list** – the large box in the left pane lists the names of pages you have added to this page group.
 - **Active** – shows whether each page is active.
- **The right pane** displays a representation of the page group. As you add pages, additional boxes that represent each page display according to the order in which each page is added. The box at the top represents the page that displays first in Portal Interface when that page group is selected. You can drag and drop page boxes up and down to change the order in which the pages display.

Choosing a navigation style

Enterprise Portal provides *navigation styles*, which let you specify how users pages display based on the user's operating system and browser. You specify a navigation style when you create a page group.

Enterprise Portal refers to the combination of a particular browser and operating system as a *user agent*. A user agent consists of only one combination of a browser and operating system; for example, Internet Explorer on Windows CE (a typical PocketPC configuration). A navigation style specifies one or more agents for use with a page group.

See for instructions on setting up

Every page group has the navigation style "Default." You can add others by editing the *navstyles.xml* file located in *\$JAGUAR/Repository/WebApplication/onepage/fw/properties/* on UNIX and in *%JAGUAR%\Repository\WebApplication\onepage\fw\properties* on Windows.

Sample *navstyles.xml* file

The code fragment below shows a typical *navstyles.xml* file. The example is from the FWPageServer Web application.

```
<?xml version="1.0"?>
<NavigationStylesConfigDoc>
<!--Default supports Netscape, IE browsers - Mozilla user agents -->
<!--Will be this the generic one for all devices/clients-->
  <NavStyle name="Default" clientpattern="Mozilla">
    <AppDefinition name="FWTopMenu">
      <Target default="topmenu_mop_ie" alternate="topmenu_mop_ie" />
    </AppDefinition>
    <AppDefinition name="FWPageServer">
      <Target default="main_IE" alternate="main_IE" />
    </AppDefinition>
  </NavStyle>
<!-- support IE browsers on PocketPC devices running on Windows CE -->
  <NavStyle name="PocketPC" clientpattern="MSIE" platform="Windows CE">
    <AppDefinition name="FWGuestPage">
      <Target default="guest" alternate="ppclogin" />
    </AppDefinition>
    <AppDefinition name="FWLogin">
      <Target default="default" alternate="default_ppc" />
    </AppDefinition>
  </NavStyle>
</NavigationStylesConfigDoc>
```

```

<AppDefinition name="FWMyOnePage">
  <Target default="myonepage_ie" alternate="myonepage_ppc" />
</AppDefinition>
<AppDefinition name="FWTopMenu">
  <Target default="topmenu_mop_ie" alternate="topmenu_ppc_ie" />
</AppDefinition>
<AppDefinition name="FWPageServer">
  <Target default="main_IE" alternate="main_ppc_ie" />
  <Target default="emptypage" alternate="main_ppc_ie" />
</AppDefinition>
<AppDefinition name="FWTileServer">
  <Target default="standard_tile" alternate="standard_tile_ppc_ie" />
</AppDefinition>
</NavStyle>
</NavigationStylesConfigDoc>

```

The `NavStyle` element specifies the navigation style name. The `AppDefinition` element specifies the Model View Controller (MVC) application. The default attribute in the `Target` element specifies the default logical name returned by the MVC application command, and the `alternate` attribute specifies the alternate logical name to use for the style. These logical names must exist in the applications View XML file (see “Sample *View.xml file” on page 204). For example, the guest page for the PocketPC style shown in the preceding code has the default logical name of `guest` and the alternate logical name of `ppclogin`.

Sample *View.xml file

The code fragment below shows a typical `*View.xml` file. This example uses `$JAGUAR/Repository/WebApplication/onepage/fw/baseApps/fwpageserver/FWPageServerView.xml` on UNIX and `%JAGUAR%\Repository\WebApplication\onepage\fw\baseApps\fwpageserver\FWPageServerView.xml` on Windows for the `FWPageServer` Web application.

```

<?xml version="1.0" encoding="UTF-8"?>
<!--
  User Interface Definition of FWPageServer
-->
<AppView xmlns="x-schema:http://localhost/fw/schema/framework-view.xml">
  <AppName>FWPageServer</AppName>
  <AppPage>
    <Target logicalname='main_IE'>
      <Resource realname='/fwdisplaystaticfile/htmlHeader.jsp' />
      <Resource realname='/fwdisplaystaticfile/style/css.jsp' />
    </Target>
  </AppPage>
</AppView>

```

```

        <Resource realname='/fwpageserver/pageserver_top.jsp' />
        <Resource realname='/fwpageserver/pageserver_body_ie_div.jsp' />
        <Resource
realname='/fwpageserver/pageserver_body_ie_iframes.jsp' />
        <Resource

realname='/fwpageserver/pageserver_body_ie_mixed_divIFrame.jsp' />
        <Resource realname='/fwpageserver/pageserver_runscript_ie.jsp' />
        <Resource

realname='/fwpageserver/pageserver_tabsetDropDown.jsp' />
        <Resource realname='/fwdisplaystaticfile/footer.jsp' />
    </Target>
</AppPage>
<AppPage>
    <Target logicalname='pocketpc_main_IE'>
        <Resource realname='/fwdisplaystaticfile/htmlHeader.jsp' />
        <Resource realname='/fwdisplaystaticfile/style/css.jsp' />
        <Resource realname='/fwpageserver/pageserver_top.jsp' />
        <Resource realname='/fwpageserver/pageserver_body_ie_mac.jsp' />
        <Resource

realname='/fwpageserver/pageserver_runscript_ie_mac.jsp' />
        <Resource

realname='/fwpageserver/pageserver_tabsetDropDown.jsp' />
        <Resource realname='/fwdisplaystaticfile/footer.jsp' />
    </Target>
</AppPage>
</AppView>

```

Page groups, page numbers, and <IFRAME>

When you build a page group, each page you add is wrapped in an <IFRAME> tag, which allows the page to display on a page group tab and not pop-up in a separate browser window.

Note <IFRAME> creates a frame that sits in the middle of a regular nonframed Web page. <IFRAME> works like , only instead of putting a picture on the page, it puts another Web page on the page

Versions of Netscape prior to 6.0 do not support the <IFRAME> tag.

The number of IFRAMEs (pages) contained in a page group is saved in the portal database column of the page group (tabsets) table. When multiple page groups exist with different values for the number of pages, the portal uses the maximum number of pages specified and uses that number for that log in session. If the maximum number of pages is more than the number of pages in a particular page group, some pages are left unused. However, if the maximum number of pages is less than the number of pages in a particular page group, it is possible that more than one page will share an IFRAME and switching between those pages will cause pages to reload.

Creating and editing page groups

Note When you add, modify, or delete portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes. See “Updating Portal Interface with changes” on page 210.

Portal Interface users can add portlets only to their own user-created pages and page groups.

❖ Creating a page group

- 1 To access Portal Studio, enter the following URL in your Internet Explorer browser:

```
http://HOSTNAME.PORTALDOMAIN:8080/onepage/index.html
```

For example, if your machine name is “tahiti”, and your portal domain is “sybase.com,” enter:

```
http://tahiti.sybase.com:8080/onepage/index.html
```

If you are running the Demo version of Enterprise Portal, enter:

```
http://demo.sybase.com:4040/onepage/index.html
```

- 2 When the Portal Studio Login window displays, enter the user name and password for a user with portal administrator privileges.
- 3 Select Build | Page Groups.

The Page Group Manager opens. The Status menu allows you to navigate page groups by their status. To see all page groups of a specific status, deselect the Show Active Only check box.

- 4 Click New. The Page Group Builder opens.
- 5 Click Add to begin adding pages to the page group. To eliminate a page from the page group, select the page and click Remove.

A new Search window opens. You can search for pages by page name, page owner, and date of last modification. To locate a page, enter search criteria and click Search.

- 6 Select the page you want to add, then click Add.

Note You can select multiple pages by holding down the Ctrl key while selecting page names.

The Search window closes, and the pages you selected appear in the Page Group Builder.

- 7 Click Save As.
- 8 When the Save Page Group window appears, complete these fields:
 - Name – enter a name for the page group.
 - Resource – displays the resource associated with the user.
 - Type – select the page group type. Page groups have two types: Default and Guest. Default is for registered portal users; Guest is for guests.

Note You can add pages only to page groups of an identical type; for example, you can add guest pages only to a Guest page group.

Guest and Default pages cannot be updated in the Page Manager when the *global.properties.xml* page.group property is set to “on,” and must be updated by changing the guest and default page groups.

- Active – indicate whether the page group is active. Selecting the Active check box makes the page group active for the roles you select, after you approve the page group.
- Specific Number of Frames – specify the number of frames that should be used to render the pages of the page group. The default value is the number of pages that you added to the page group. For more information, see “Page groups, page numbers, and <IFRAME>” on page 205.

- Initial on Load – select this option to load all the portlets on the different <IFRAMES> the first time a user logs in. This causes the listeners to register and receive events immediately. See “Page groups, page numbers, and <IFRAME>” on page 205 for more information.
- Navigation Style – select a navigation style. Navigation styles lets you assign a custom navigation type to the page group, such as navigation settings for a PocketPC. For more information about navigation settings, see “Choosing a navigation style” on page 203.
- Roles – assign roles to the page group. When you assign a user role to a page group, all users assigned that role can see the page group.

Note If a page in a page group is not assigned a role that is assigned to the group, the page does not display to users without the assigned role.

9 Click OK to save the page group.

10 Select Status | New in the Page Group Manager.

Right-click the page group you created and select Status | Approved to make the page group available.

❖ **Editing page group contents and properties**

You can add or remove pages from a page group, or modify a group’s properties after you create it.

- 1 Log in to Portal Studio as a user with administrator privileges.
- 2 Select Build | Page Groups. The Page Group Manager appears.
- 3 Navigate the Status menu to find the page group whose contents you want to edit, select the page group in the detail view and click Edit.
- 4 When the Page Group Builder displays, click Add to begin adding pages to the page group. A Search window opens. You can search for pages by page name, page owner, and date of last modification. To locate a page, enter search criteria and click Search.
- 5 Select the page you want to add, then click Add.

Note You can select multiple pages by holding down the Ctrl key while selecting page names.

The Search window closes, and the pages you selected appear in the Page Group Builder.

- 6 To eliminate a page from the page group, select the page in the Page Group Builder and click Remove. Click Save to save your changes.
- 7 To edit a page group's properties, right-click the page group in the detail view and select Properties. The Page Group Properties window appears. Follow the instructions in "Creating a page group" on page 206 to change the page group properties. Click OK when to finish and save your changes.

❖ **Disabling a page group**

To disable a page group and prevent it from showing in Portal Interface:

- 1 Log in to Portal Studio as a user with administrator privileges.
- 2 Select Build | Page Groups. The Page Group Manager appears.
- 3 Navigate the Status menu to find the page group that you want to disable.
- 4 Right-click the page group in the detail view and select Properties.
- 5 When the Page Group Properties window displays, deselect the Active option, then click OK.
- 6 When the confirmation that the page group was saved displays, click OK.

❖ **Marking a page group as deleted**

- 1 Select the page group's current status from the Page Group Manager Status menu to display the listing in the detail view.
- 2 Right-click the listing in the detail view, select Status from the pop-up, then select Deleted.

Note The page group listing moves to the Deleted status in the detail view; however the page is not removed from the database.

- 3 To change the page group's status once it has been marked as deleted, select Deleted from the Page Group Manager Status menu.
- 4 Right-click the page group listing in the detail view, select Status from the pop-up, then select any status that displays to recover the page group. The page group moves to the detail view of the status you selected.

Updating Portal Interface with changes

When you add, modify, disable, or delete portal objects—catalogs, pages, and pages groups—you must update Portal Interface with those changes.

Each of these objects has an Update icon on the Manager toolbar and an Update Users option on the right-click menu you access in the detail view.

❖ **Updating page changes to Portal Interface**

- 1 Select Build | Page Groups in the left pane.
- 2 When the Page Group Manager displays, right-click in the detail view and select Update Users.
- 3 When a dialog box confirms that the update was successful, click OK.

Using Enterprise Portal Agents

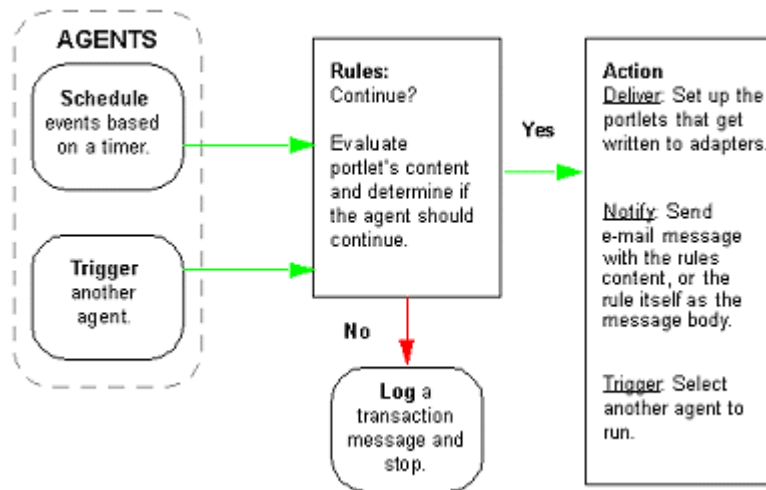
This chapter describes how to use Enterprise Portal's agent technology to automate the processing of portlet content.

Topic	Page
Introduction	211
Using agents	213
Using adapters	227
Using agent servers	231
Viewing transaction logs	232

Introduction

Agents are software components that act on behalf of users or other agents to gather or process information in the background. Typically, an agent is given a small, well-defined task.

Enterprise Portal agents allow you to schedule or externally trigger automatic processing of a portlet's content. You can also create agent rules that evaluate a portlet's content based on specified criteria. If a portlet's content matches the criteria, certain actions take place, such as writing the portlet's content to e-mail message, to a database, or to disk, which is accomplished using an adapter. If the portlet's content does not match the criteria, a transaction message is written to a log.

Figure 12-1: Agents process flow

Agents are executed by:

- A timer triggering the agent's execution based on the agent's schedule
- One agent triggering another agent's execution
- Selecting the Run option to execute any agent once

When an agent runs, the system checks the agent's rules—if there are any—to determine if the agent should continue running. If the criteria specified in the rules has been met, the agent's actions execute.

To write a portlet's content to a file, an e-mail message, or a database table, you create an adapter. When the agent is triggered, the adapter interface writes the associated portlet's content to a specified destination.

You can also assign an agent to a specific agent server to run that agent. Although you can create new agent servers, a default agent server is installed with Enterprise Portal. If an agent is unassigned and there is more than one agent server, the first available agent server runs the agent. If only the default agent server exists, all agents run on that server.

Using agents

The Portal Studio left pane lists three agent-related components from which to choose:

- Agents – create and edit agents.
- Adapters – write an agent's data to an adapter.
- Servers – monitor the transactions that have been processed and completed via transaction logging for the agent running on the agent server.

The list in the detail pane displays the agent's name, what the agent does, the owner, the last time the agent ran, the result of the agent's last execution, and the next time the agent is scheduled to run.

The Details column codes indicate that the agent:

- S – has a schedule.
- R – contains a rule.
- D – contains a delivery action.
- N – contains a notify action.
- T – contains a trigger action.
- E – uses error management.

Changing an agent's status and running an agent once

Agents can exist in any of three states:

- Ready – the running state for a scheduled agent. When you create a schedule for an agent and change that agent's status to Ready, the agent automatically runs based on the schedule you specified.
- Stopped – the state for agents that do not run on a schedule. New agents are always created in a Stopped state. Select Run to run an agent with a Stopped status once.
- Deleted – agents that have been deleted. Until you dump the database, the agent record still exists in the portal database table.

❖ Modifying agent status and executing an agent once

- 1 Select the agent's current state from the Agent Manager Status menu.
- 2 Select the agent in the detail pane.

- 3 From the Agent Manager toolbar, select:
 - Run – to run any agent with any status once.
 - Start – to change an agent’s status from Stopped or Deleted to Ready.
 - Stop – to change an agent’s status from Ready to Stopped.

Note You can also right-click on an agent, select Status, and choose the status from the pop-up that appears.

To run an agent repeatedly, create a schedule for the agent and change the agent’s status to Ready. See “Schedule tab” on page 214 for details.

❖ **Creating new agents**

When you create an agent, you create a schedule, create rules, and specify actions and error management.

- 1 Select Automate | Agents from the Portal Studio left pane.
- 2 Select New above the detail pane. You see the Agent Builder window.

The Agent Builder has four tabs. For details, see:

- “Schedule tab” on page 214
- “Rules tab” on page 215
- “Actions tab” on page 221
- “Error Management tab” on page 224

Schedule tab

The Schedules tab allows you to automate an agent’s recurring execution based on intervals that you specify. For example, you can create a schedule that runs an agent’s action every hour, once a day, once a week, or once a month.

❖ **Setting up an agent to run on a schedule**

- 1 Log in to Portal Studio, select Automate | Agents from the left pane, then click New. The Agent Builder appears with the Schedule tab displayed.
- 2 Select “Trigger Agent Based on a Schedule.” When you select this option, the remainder of the schedule options display.
- 3 Enter the following information for the schedule options:

- Time – the time of day (AM or PM) you want the agent to run.
- Recurrence – the frequency at which you want the agent to run, entered in minutes, hours, days, weeks, or months.
- Range – the date you want the agent to begin and end running, in month/day/year format. You can also click the small calendar to the right of the Start Date and End By to select the date. If you want the agent to run indefinitely, select No End Date.

Rules tab

The Rules tab lets you specify rules that the portlet's content must meet before the action associated with the agent can be executed. For example, you may want an agent to notify certain people of a stock price only when that price reaches a specified amount.

❖ **Setting up agent rules**

- 1 Log in to Portal Studio, select Automate | Agents from the left pane, then click New.
- 2 When the Agent Builder appears, select the Rules tab.
- 3 Select Execute Agent Based on a Rule and click Edit. The Rule Editor window displays where you select the source of the rule's criteria and create the actual rules.
- 4 Base Rule On – select:
 - Portlet – to base the rule on a selected portlet's content. See the next procedure, "Basing rules on a portlet's content."
 - Criteria – to base the rule on criteria that you specify. See the procedure, "Basing rules on criteria."

❖ **Basing rules on a portlet's content**

- 1 In the Rule Editor, select Portlet and click Find to select a portlet using the search dialog box. When you select the portlet and click Add, the portlet title appears in the Rule Editor's Name drop-down list and the portlet's content displays beneath Preview Source.
- 2 Select Playback to display the portlet as it previews. Select XML to display the portlet as XML.

The Rules section allows you to group rules together based on the Boolean operators “AND” or “OR.” See “Using rule operators” on page 217 for a description of each operator.

Additional options let you select how to construct the rules based on:

- **Label** – uses column labels in the data. When table fields reorder, the rule still picks up the correct field.
- **Position** – an index on positions in the data.
- **XPath** – if you choose to view the content as XML, this option allows all matches to be done via XPath expressions.

Note XPath is a set of syntax rules for defining parts of an XML document.

When the portlet displays as XML, you see this code:

```
<?xml version="1.0" encoding="UTF-8" ?>
- <ContentDoc>
  - <Element name="bonds">
    - <Record>
      <Field op_label="Maturity" op_position="1">Maturity</Field>
      <Field op_label="Yield" op_position="2">Yield</Field>
      <Field op_label="Yester- day" op_position="3">Yesterday</Field>
      <Field op_label="Last Week" op_position="4">Last<BR>Week</Field>
      <Field op_label="Last Month" op_position="5">Last<BR>Month
        </Field>
    </Record>
  - <Record>
    <Field op_label="Maturity" op_position="1">3 Month</Field>
    <Field op_label="Yield" op_position="2">0.99</Field>
    <Field op_label="Yester- day" op_position="3">1.00</Field>
    <Field op_label="Last Week" op_position="4">0.97</Field>
    <Field op_label="Last Month" op_position="5">1.04</Field>
  </Record>
  ...More XML records follow...
```

“Label” matches the XML `op_label` attribute and “Position” matches the XML `op_position` attribute.

XPath uses path expressions to locate nodes within XML documents. XML documents can be represented as a tree view of nodes.

Any XPath expression that uses partial paths starts at each “record” node and is applied. Unstructured data is in one record, so the pop-up selections for Label and Position contain only “content.”

Note XPath and the comparable options are valuable rule options. You can use the comparable options of “found” and “not found” to check if a specific node is found in an XML document regardless of the text content. This is useful to see if a portlet is broken, or has been changed in a way that requires attention.

To learn more about XPath, see the OASIS specification at <http://www.w3.org/TR/xpath>.

Additional examples to help you understand how to use XPath are available at <http://www.w3schools.com/xpath/default.asp> and at <http://www.zvon.org/xxl/XPathTutorial/General/examples.html>.

- 3 At the bottom of the Rules panel, in the three unlabeled selection boxes after the term “Execute agent when,” select:
 - 1st box – Maturity
 - 2nd box – equals (text)
 - 3rd box – 3 months
- 4 Click the plus sign next to the rule to save the rule. Once you save the rule, the entry appears in the Rules panel.
- 5 To edit a rule, click the check mark. To delete a rule, click the X.
When you edit an entry, the check mark changes to plus sign. Click the plus sign to save your changes.
- 6 When you finish, click OK to exit the Rule Editor and return to the Agent Builder. Click OK to save your entries.

Using rule operators

The Rules section of the Rule Editor lets you create and group rules together based on the Boolean operators AND or OR that tell the system which keywords the portlet’s content must include or exclude before the portlet’s agent executes.

- AND or OR– use these operators to specify what terms must appear in the selected portlet’s content. When you click the plus sign to save the rule, then click OK, the rule is summarized on the Rule tab. For example,

- If you set the rule as:

The rule will execute this agent when 'Maturity' equals (text)
'3 Month' AND label 'Yield' is greater than '0.5'

Then the bond portlet meets the rule, and the actions can be executed.

- If you set the rule as:

The rule will execute this agent when 'Maturity' equals (text)
'3 Month' AND label 'Yield' is greater than '1'

Then this bond portlet does not meet the rule, and the actions will not be executed.

- If you set the rule as:

The rule will execute this agent when 'Maturity' equals (text)
'3 Month' AND '6 Month'

Then this bond portlet does not meet the rule, and the actions will not be executed.

- If you set the rule as:

The rule will execute this agent when 'Maturity' equals (text)
'3 Month' OR 'Maturity' equals (text)'6 Month'

Then this bond portlet meets the rule, and the actions can be executed.

The relational operators that appear in the 2nd rule box are:

- equals (number) – converts the portlet data into floating point numbers for the equals comparison. If the data is not numeric, the comparison fails.
- is greater than – converts the portlet data into floating point numbers for the greater than comparison. If the data is not numeric, the expression returns false. The portlet data must be greater than the value supplied in the rule's text field.
- is less than – converts the portlet data into floating point numbers for the less than comparison. If the data is not numeric, the expression returns false. The portlet data must be less than the value supplied in the rule's text field.
- equals (text) – compares the portlet data to the text supplied in the rule's text field. If they are equal, the expression returns true.
- contains – returns true if the portlet data contains the text string supplied in the rule's text field.

- starts with – returns true if the portlet data starts with the text string supplied in the rule’s text field.
- ends with – returns true if the portlet data ends with the text string supplied in the rule’s text field.
- matches RegExp – returns true if the portlet data matches the regular expression supplied in the rule’s text field. To understand the format of regular expressions, see the documentation at <http://jakarta.apache.org/regexp/apidocs/org/apache/regexp/RE.html>.
- contains data – checks the portlet to see if any data has a size greater than zero. The rule’s text field is disabled for this option.
- contains no data – is the opposite of “contains data.” Checks the portlet to see if any data has a size equal to zero. The rule’s text field is disabled for this option.
- is found – used with XPath. Returns true if the XPath expression resolves to a XML node or attribute.
- is not found – used with XPath. This operator is the opposite of “is found.” Returns true if the XPath expression does not resolve to a XML node or attribute.

❖ **Basing rules on criteria**

Basing a rule on criteria allows you to execute a rule on a selection of portlets that meet user-specified parameters; for example, to execute the rule on portlets owned only by a particular user.

This procedure describes how to set up an agent to find portlets in which one-click content capture has failed and the portlets are broken.

- 1 In the Rule Editor, select Criteria and click Select. You see the Add Content by Criteria window.
- 2 To complete the Criteria Parameters, select:
 - Category – find portlets only in the selected category.
 - Subcategory – find portlets that are only in the selected subcategory.
 - Status – find portlets that only have the selected status.
 - Catalog – find portlets that are only in the catalog you select.
 - Page – find portlets that are only on the page you select.
 - Owner – find portlets that only belong to the user whose name you enter.

- Modified By – find portlets that only belong to the user whose name you enter.
 - Roles – find portlets that are assigned only those roles you select in the Available Roles list and move to the Assigned Roles list.
- 3 Click OK to save the criteria and return to the Rule Editor, which populates the Name drop-down list for Preview Source with all items that meet the selected criteria at that moment.
 - 4 In the Rules section, select xpath and enter this execution statement:

```
\'//ContentDoc/Element/text( )\' equals (text)
\'ERROR\'
```

and click OK.

The rule has been set up to check for a particular type of error by looking at the XML returned from the portlet when it fails. The XML for this portlet displays as:

```
<?xml version="1.0" encoding="UTF-8 ?>
-<ContentDoc>
<Element name="bonds2">ERROR</Element>
</ContentDoc>
```

Rule processing

The criteria you select allows a list of portlets to be processed by the rule you create. The criteria is processed when the rule is executed.

Criteria-based rule processing uses these directives:

- The criteria you select are processed as “AND” options.
- When you select a catalog or a page, the portal processes all portlets in that catalog or page by the rule.
- Selecting options other than catalog or page indicates that only the portlets that match all the selected criteria are processed. For instance, only portlets that have an Approved status are processed.
- The role selection takes advantage of the *global.properties.xml* property “RoleBaseDisplaySeeAllRoles.” In the example, this property is set to “superuser,” which means the example criteria returns all approved portlets created by everyone.

Actions tab

The Actions tab lets you set up actions for an agent to execute.

❖ Creating agent actions

- 1 Log in to Portal Studio, select Automate | Agents from the left pane, then click New. The Agent Builder appears.
- 2 Select the Actions tab.
- 3 Click Add and select:
 - Deliver – to write a portlet’s content to an adapter that writes the content to a file system, an e-mail message, or a database table.
 - Notify – to send an e-mail message that contains the rules content or the actual rules to the users.
 - Trigger – to select another agent to execute when this agent runs.

❖ Creating a delivery action

The Deliver action lets you select portlets that have content you want written to an adapter.

- 1 Select Add | Deliver from the Actions tab. You see the Action Details window.
- 2 Complete these fields:
 - Description – enter a description of this action.
 - Destination Viewer – specifies the format in which to write the content. Currently, only HTML is supported.
 - Destination Set – click Add to select an adapter into which a portlet’s content is written. See “Using adapters” on page 227 for details on defining adapters for agent use.

The content of each portlet in the Portlet Set is written to each defined adapter that you select for the Destination Set. To select multiple adapters, hold down the Ctrl key as you select each adapter, then click Add.

After you add an adapter to the Destination Set, select the adapter and click Properties to display a description of the adapter.

- Portlet Set – click Add to select the portlets from which content is written to an adapter. There are four options:
 - Portlet – select an individual portlet.

- Criteria – select a portlet based on criteria.
- Rule Portlet – pass the triggered rule portlet to this action. This option is enabled only if a rule is set in the agent.
- Matching Rule Record – pass the matching record from the rule as XML content to the destination set. This option is enabled only if a rule is set for the agent.

When you select Criteria, you see the Add Content by Criteria window.

3 Click OK to save the delivery action.

❖ **Creating a notify action**

This option lets you enter an e-mail address to send portlet content and an optional message without creating an e-mail adapter.

1 Select Add | Notify on the Actions tab.

2 Complete these options:

- E-mail To – enter the complete e-mail address of the recipient.
- Subject – enter the text that should appear as the subject for the e-mail's message.
- Prepend Subject with Rule's Portlet Name – adds the name of the portlets that were executed and passed through the Rules section to the beginning of the Subject line.

Note This option is for each portlet executed by a rule; that is, if a rule is a criteria, when a portlet triggered, a message gets sent for each portlet.

If, for example, you are searching for broken portlets, you can use this option to be notified of the portlet that is no longer working.

- Include Message – lets you write a message that displays in the message body. When you select this option, a text window displays where you type in the message.
- Include Rule's Content – select this option to display additional options that pertain to the executed rule. This option is enabled only if a rule is selected.
 - Send the Entire XML Rule Source – sends the XML content of the rule that was triggered.

- **Send Only the Record(s) From the Rule Source That Meet the Rule Criteria** – sends the section of the rule portlet that was triggered. For example, the e-mail content might look similar to this:

```
Date: Mon, 21 Jul 2003 11:35:39 -0700 (PDT)
From: agentserver@sybase.com
Reply-To: agentserver@sybase.com
Subject: bonds portlet matched
X-Mailer: Sybase-EP
```

Rule executed on portlet=bonds(id=271)v0.1

Included Matching Records:

```
<?xml version="1.0" encoding="UTF-8"?>
<MatchingRecords><Match><Record> <Field op_label="Maturity"
op_position="1">3 Month </Field> <Field op_label="Yield"
op_position="2">0.79 </Field> <Field op_label="Yester- day"
op_position="3">0.78 </Field> <Field op_label="Last Week"
op_position="4">0.77 </Field> <Field op_label="Last Month"
op_position="5">0.71 </Field>
</Record></Match></MatchingRecords>
```

- **Send the Portlet That Met the Rule Criteria** – sends the rule portlet that triggered the rule. The portlet is sent in the default playback format, which was assigned during portlet creation and set the content-type; specifically, it could be an XML portlet.
- **Send the Portlet as XML That Met the Rule Criteria** – sends the rule portlet that triggered the rule as the exact XML that matched the criteria.

3 Click OK to save the action.

❖ **Creating an agent action that triggers another agent**

Triggers allow one agent to process data and determine if another agent should be run.

- 1 Select Add | Trigger on the Actions tab. You see the Add Agent Search dialog box.
- 2 Complete these options:
 - **Name** – enter the name of the agent you want to find. If you leave this field blank, the portal searches for all agents.

- Details – these optional selections match the codes in the Details column of the agent list in the Agent Manager detail pane. Choose from these options to narrow the search based on what is included in the agent.
 - Schedule – matches agents with an “S” details code.
 - Rules – matches agents with an “R” details code.
 - Content Delivery – matches agents with a “D” details code.
 - Send Notification – matches agents with an “N” details code.
 - Trigger Agent – matches agents with a “T” details code.
 - Error Management – matches agents with an “E” details code.

Note When you save an agent, the system assigns the agent a code based on the details contained in the agent.

- Status – select the status of the agents you are searching for—Ready, Stopped, or Deleted.
 - Owner – enter the user name of the person who owns the portlets for which you are searching.
- 3 Click Search. The agents that match the selected options display in the Results pane.
 - 4 In the Results pane, select the agent you want to trigger and click Add. You return to the Rule Editor.

Error Management tab

The Error Management tab allows processing to take place or continue when an error occurs while the agent is running.

Click the plus sign next to each error type—two options display for each error type:

- Set Agent Status To – resets the agent’s state to the selected status when an error occurs. For example, you can use this option to change an agent’s state to Stopped when an error occurs to prevent the error from happening again.

- Send Message to Error E-mail – sends an error message to the e-mail address specified in the Error E-mail field at the top of the Error Management tab.

The Active column displays a check mark if one or both of the options are set for the given error check. The error check types are:

- Any Error Occurred – encompasses all errors when the agent is running.
- System Error Occurred – an unexpected system failure like “Disk Full” or “Database Connection Refused.”
- Content Retrieval Error Occurred – an agent’s rule or delivery action is processing a portlet and the portal playback engine fails to retrieve the portlet’s content.
- Rules Error Occurred – a rule uses a regular expression and the rule fails because the content is not in the expected format.
- Actions: Trigger Error Occurred – the agent that should get triggered does not exist.
- Actions: Notify Error Occurred – there is a problem sending an e-mail message to the Notify recipient.
- Actions: Delivery Error Occurred – a problem occurs delivering a portlet set to the destination set defined in the delivery action.

Saving a new agent

When you finish creating a new agent’s schedule, actions, rules, and error management, you must save the agent.

- 1 When you save a new agent for the first time, click Save or Save As. You see the Agent Properties window.
- 2 Complete these fields:
 - Name – enter a name for the agent.
 - Description – enter a brief meaningful description of the agent.
 - Agent Server – values are “None” or the name of the assigned agent server.
 - None – the new agent runs on the first agent server that is recognized. Normally this is the default agent server.

- Assigned agent – clicked Find and select an agent server. The new agent runs only on the selected agent server.

Click Clear to remove the assigned agent server and return the value to None.

- Roles – select the available roles for who can edit the agent’s properties and view the agent’s log and click Add. Click Add All to assign all available roles.

3 Click OK to save your entries.

Handling “permission denied” error messages

The roles assigned to an agent are independent of the roles assigned to portlets that may run inside the agent. This means you can create an agent that includes portlets that require role “A,” but the agent is assigned role “B.” If another user of role “B” edits this agent and tries to preview the portlet of role “A,” they receive a permission denied error message.

Permission denied errors can occur in these areas:

- Portlet preview in the Actions Delivery panel displays “Permission Denied” when the user is not in the same role as the portlet being previewed.
- Portlet preview in the Rules panel displays “Permission Denied” if the user does not have the rights to view the portlet.
- Role-based views of the rules criteria shows “Permission Denied” when selecting the portlet to which the user does not have access rights.

Warning! When you preview a portlet and receive the error “Permission Denied,” the system disables the JavaScript Rules panel that is directly linked to the parsing of the previewed portlet.

Viewing or changing agent properties

To view an agent’s properties, select the agent in the Agent Manager detail pane, then click Properties in the Agent Builder. You see the Agent Properties window.

You can change the description, select an agent server, and change the agent's assigned roles.

When you finish viewing or changing the agent's properties, click OK. Click Cancel to exit the window without saving any changes you may have made.

Using adapters

Adapters allow you to write a portlet's content to a file, an e-mail message, or a database table. When you assign an adapter to an agent and the agent is triggered, the adapter writes the portlet's content to the specified destination. Three default adapter types are installed with Enterprise Portal—fs (file system), email, and db (database).

❖ Creating new adapters

- 1 Select Automate | Adapters from the Portal Studio menu.
- 2 Select New from the toolbar, then select the type of adapter you want to create:
 - File system (fs) – writes portlet content associated with an agent to a file system.
 - E-mail – writes portlet content associated with an agent to an e-mail message.
 - Database (db) – writes portlet content associated with an agent to a database table.

What you see next depends on the type of adapter you chose. For details, see:

- “Creating a file-system adapter” on page 227
- “Creating an e-mail adapter” on page 228
- “Creating a database adapter” on page 230

Creating a file-system adapter

- 1 Click New on the Adapter Manager toolbar and select “fs.” You see the Adapter Builder.

- 2 Complete these fields:
 - Name – the program fills in this field after you save the new adapter and enter the name.
 - ID – the adapter ID is a sequential number maintained by the system.
 - Description – the system automatically enters “FileSystem Adapter” in this field, which you can change if you want.
 - Type – displays the adapter type you chose after you clicked New.
 - Viewer – specifies the format in which to write the content. Currently only HTML is supported.
 - Roles – select the available roles that can edit the adapter’s properties and view the adapter’s log and click Add. Click Add All to assign all available roles.
 - Parameters – for a file-system adapter, the parameter is the directory where you want the portlet’s content written.
 - Name – the parameter name, which is “directory.” You cannot edit this field.
 - Display Name – change the display name or accept the default of “Storage Directory.”
 - Directory – enter the location where you want the content written. For example, *d:\temp\content*. If the folder does not exist, the adapter creates the folder when it writes the portlet content.
- 3 Click Save As.
- 4 In the Save As dialog box, enter an adapter name and click OK.
- 5 Click Close to exit the Adapter Builder.

Creating an e-mail adapter

- 1 Click New on the Adapter Manager toolbar and select “email.” You see the Adapter Builder.
- 2 Complete these options:
 - Name – the program fills in this field after you save the new adapter and enter the name.

- ID – the adapter ID is a sequential number maintained by the system.
- Description – the system automatically enters “Mail Adapter” in this field, which you can change if you want.
- Type – displays the adapter type you chose after you clicked New.
- Viewer – specifies the format in which to write the content. Currently, only HTML is supported.
- Roles – select the available roles that can edit the adapter’s parameters and view the adapter log, and click Add. Click Add All to assign all available roles.
- Parameters – for an e-mail adapter, the parameters are the values specific to writing the portlet’s content in an e-mail message; for example, the e-mail address, the subject line, and so on. You can change the display name or accept the default.
 - include_portlet – enter “true” or “false.” Set to true to write the portlet contents in the e-mail message. Set to false to write something in the msg_body field instead of including the portlet’s contents.
 - from_addr – enter the sender’s e-mail address.
 - subject – enter the text to appear as the message’s subject line.
 - msg_body – enter the text you want added to the body of the message. Normally, if you set include_portlet to true, you would not enter anything for this parameter.

If you set the first parameter to true and include a message, the text you enter displays before the portlet’s content and the entire e-mail message is sent to a MIME type of “text/plain.”

If you send only portlet content, the content type is set to a MIME type of “html.”
 - to_addr – enter the e-mail address of the recipient. To send the message to multiple recipients, enter the e-mail addresses as a comma-separated list.

- 3 Click Save As.
- 4 In the Save As dialog box, enter an adapter name and click OK.
- 5 Click Close to exit the Adapter Builder.

Creating a database adapter

- 1 Click New on the Adapter Manager toolbar and select “db.” You see the Adapter Builder.
- 2 Complete these options:
 - Name – the program fills in this field after you save the new adapter and enter the adapter’s name.
 - ID – the adapter ID is a sequential number maintained by the system.
 - Description – the system automatically enters “Database Adapter” in this field, which you can change if you want.
 - Type – displays the adapter type you chose after you clicked New.
 - Viewer – specifies the format in which to write the content. Currently, only HTML is supported.
 - Roles – select the assigned roles associated with who can edit the adapter’s parameters and view the adapter log and click Add. Click Add All to assign all available roles.
 - Parameters – the parameters for a database adapter are the database values that specific to writing the portlet’s content to a database. You can change the display name or accept the default.
 - Key_column_name – enter the table field into which the portlet ID is written.
 - Table_name – enter the name of the database table. This table must already exist.
 - Value_column_name – enter the table field into which the portlet’s content is written.
 - User – enter the database user for the database connection.
 - Url – enter the JDBC connection URL used for the database connection.
 - Password – enter the password for the user connection to the database.
 - Driver – enter the database driver used for the JDBC connection.
- 3 Click Save As.
- 4 In the Save As dialog box, enter an adapter name and click OK.
- 5 Click Close to exit the Adapter Builder.

Editing an adapter or viewing the log

To edit an adapter, select the adapter in the detail view and click Edit, or right-click the adapter and select Edit.

To view the adapter's activity, select the adapter in the detail view and click View Log, or right-click the adapter and select View Log. See "Viewing transaction logs" on page 232 for more details.

Using agent servers

Agents run on agent servers. When you install and start Portal Studio, the default agent server is created by the portal. All agents run on the default agent server unless you assign them to a different agent server.

Every onepage deployed WAR file (the portal) has an agent server thread running when it starts up. If you set up a cluster of onepage Web applications and want an agent to run on only one of the server machines (for example, maybe the database connection or file system only exist on one machine), then you can assign that server an ID in the *global.properties.xml* portal configuration file. When you restart the server and Portal Studio, the portal creates the agent server reference, which a user can then select for a specific agent.

You can also create the agent server reference using the following procedure. When you want the agent server to run, you must take the assigned ID and set it in *global.properties.xml* on the machine that would host the agent server; for example:

```
<Property name="AgentServerID" value="211"
description="Default Agent Server ID" menugroup="10"/>
```

❖ Creating new agent servers

- 1 Select Automate | Servers.
- 2 Select the New icon from the toolbar. You see the Agent Server Editor.
- 3 Complete these fields:
 - Name – the program fills in this field after you save the new agent server and enter the name.
 - ID – the agent server ID is a sequential number maintained by the system. The default agent server has an ID of 1.

- Description – enter a description of the agent server.
 - Status – select the agent server’s state —Ready, Stopped, Broken, or Deleted. Setting a server to a state other than Ready stops that agent server from retrieving agents to run. If an agent is assigned to an agent server with a state other than Ready, the agent does not run until the agent server state is reset to Ready.
 - Roles – select the available roles that can edit the agent server’s properties and view the agent server’s log and click Add. Click Add All to assign all available roles.
- 4 Click Save As to save the new agent server. A dialog box prompts you for the agent server’s name.
 - 5 Enter a name for the agent server and click OK.
 - 6 Click Close to exit the Agent Server Editor.

Editing an agent server

To edit an agent server, select the server in the detail pane and click Edit, or right-click the agent server in the detail pane and select Edit.

Viewing transaction logs

To view an agent server’s activity:

- 1 From the Portal Studio left pane, select Automate | Agent Servers.
- 2 From the Status menu, select the state in which the agent server resides.
- 3 In the detail pane, select the server and click View Log, or right-click the agent server name and select View Log.

The log window displays information about what agents have been started and the actions that occurred during the agent’s execution. The server log records all agent and adapter activity on all agent servers. Each agent and adapter has a log also and displays only the information pertinent to its type.

- Click Refresh to update the display.

- Click Clear Log to clear the content from the database. Clearing messages at the agent server level clears the messages from all agents and adapters that were running on that agent server. However, clearing messages from an agent or adapter log window clears only their references to the messages. The messages are still referenced in the agent server's log and must be cleared from the agent servers log for permanent deletion.

Limiting the log size

To prevent sending too much data to the browser and slowing down performance, define the number of message you want to display in `global.properties.xml`. For example:

```
<Property name="MaxLogRows" value="150"
description="The maximum number of log rows to return
on a view log request (set to zero for all)."
```

To view all messages in a sizable log, use SQL to display the log rather than View Log as the text returned over HTTP can be too enormous for the browser.

Logging is kept in the `app_logs` table. The objects that reference each table row are stored in the `app_log_object_refs` table. For example,

```
select * from app_logs
```

displays all log messages.

Note The agents transaction logging can fill a database quite fast. If the View Log or other agent functionality seems slow, check the database logs to see if the database is out of space. You may see an error like:

```
Tasks are sleeping waiting for space to become available
in the log segment for database tempdb
```

To solve this problem, increase the database size. See the *ASA Database Administration Guide* on the *Technical Library* CD that comes with the product. If you are using a database other than Adaptive Server Anywhere with EP, refer to that product's documentation for instructions on increasing the database size.

Importing, Exporting, and Deploying Portal Objects

This chapter describes how to import and export portal entities to and from Portal Studio installations. The chapter also explains how to distribute Portal Studio entities to a remote server.

Topic	Page
Introduction	235
Importing files into Portal Studio	236
Exporting objects from Portal Studio	238
Deploying portal objects to a remote server	239
Rules and restrictions	242

Introduction

Portal Studio allows you to move portal entities—catalogs, applications, page groups, pages, and agents—between different resources for the same EP installation and between installations by exporting their metadata to an XML document. The portlets, templates, and personalization adapter database table keys associated with these entities are exported along with their parent objects to the same XML document. This allows you to easily synchronize the portal content between an isolated development environment and a production environment.

Warning! To use the export, import, and deploy functionality, the Portal Studio user must be granted the PortalAdmin role. When a user is not granted this role, the Deploy option on the Portal Studio toolbar is not available.

Importing files into Portal Studio

If you are on the receiving side of an exported archive file from a different Portal Studio installation, use the Portal Studio–Deploy dialog box to import portal entities into your Portal Studio installation. You must place the archived XML file that you want to import on the same server as your Portal Studio installation.

❖ Importing a file into Portal Studio

- 1 Log in to the Portal Studio as a user that has the StudioAdmin and PortalAdmin role.
- 2 Select Build | Portlets from the Portal Studio left pane.
- 3 Click Deploy on the Portlet Manager toolbar.
- 4 When the Portal Studio–Deploy window displays, click the Import tab.
- 5 Complete these fields:
 - File Name – enter the name of the XML file to import, or click Browse to locate the file using the Choose File dialog box.
 - Click View to view the XML file contents.
 - Click Upload to upload the file to Portal Studio. When a prompt confirms that the upload was successful, click OK.
 - Import Catalog As Active – make the imported catalog the default display catalog. The imported XML document can contain more than one catalog, but only one catalog is marked as the default display catalog—approved and active. If you import this catalog as active, it deactivates the current default catalog and replaces it with the new catalog.
 - Override Existing Entities – have imported entities (pages, page groups, applications, and agents) replace an equivalent entity with the same name; for example, replace a page with an imported page with the same name. These entities do not use versioning.
 - Import Entities As Active – import the new entities as active but the catalog as inactive, which makes the imported entities (but not the catalog) available to Portal Interface users.

- Set Deferred Update For All Portal Users – indicates whether to notify all portal users of a change in the portal content so that they can refresh their view the next time they log in to the portal.

Note During the import process, if you do not select Set Deferred Update For All Portal Users, the Portal Interface user must perform an Update for each guest page or guest page group.

- 6 Click Import. When a prompt confirms that the import was successful, click OK.
- 7 Click View Log to review the import operation’s activities.
- 8 Click Close to exit the Portal Studio–Deploy window.
- 9 Select Build | Catalogs from the Portal Studio left pane.
- 10 Select Approved from the Catalog Manager Status menu. If you chose to import a new catalog as active, you should see the catalog in the detail view with a check mark in the Active column.
- 11 If you chose to import the catalog as active, click Update on the Catalog Manager toolbar to update Portal Interface on the same application server with the new default display catalog.
- 12 When you see confirmation that the operation was successful, click OK.
- 13 Log in to Portal Interface.
- 14 You can add portlets only to your personal page and page group.

If the page.group global property is turned on, create a page group, then add a new page to that page group.

If the page.group property is turned off, create a new page.

Note See Appendix A, “Configuring Global Properties,” for more information about setting the page.group property.

- 15 Click Add Portlet. When the Add Portlet window appears, it should display the contents of the new default display catalog.

Note When you import portal objects (templates, catalogs, and so on) they maintain their original owner; that is, the Owner column in the detail view for objects lists the person who originally created the object, not the person performing the import.

Exporting objects from Portal Studio

You can export catalogs, applications, page groups, pages, and agents to an archive file (with an XML format and extension) on the Portal Studio's server. That archive file and its contents can be imported into a different Portal Studio installation.

When you export portal entities to an archive file, child objects of those entities are also exported. For example, when you export page groups, the pages, portlets, and templates with which the portlets are associated are also exported.

❖ Exporting portal objects to an XML file

- 1 Log in to the Portal Studio as a user associated with a role—StudioAdmin, PortalAdmin—that has permission to perform all operations. See the *Enterprise Security Administration Guide* for more information about roles.
- 2 Select Build | Portlets from the Portal Studio left pane.
- 3 Click Deploy on the Portlet Manager toolbar.
- 4 When the Portal Studio–Deploy window displays, click the Export tab.
- 5 Complete these fields:
 - Export To Include – select or deselect the entities you want to export: Catalogs, Applications, Page Groups, Pages, and Agents.

You can export all or any combination of entities. When you select an entity, all other entities referred to by the selected entity are also exported. For example, if you select only page group, all pages, portlets, and templates referred to by the page group are included. If you select all entities, the export includes each unique entity once. That is, if a portlet is referenced by both the catalog and five pages, the portlet appears only once in the export XML document.

Note In EP versions earlier than 6.0, only the active catalog could be exported. With EP 6.0, more than one catalog can be exported at a time because agents can refer to catalogs that are inactive or not in an approved state.

- File Name – the name you want the XML file to have; for example:
`export.xml`

- 6 Click Export. The file is saved with the name you entered in the default Portal Studio location that is specified in the *global.properties.xml* configuration file.
- 7 When a prompt advises you that the export operation succeeded, click OK.
- 8 Click View Log to review the export operation activities.
- 9 Click the Download button to the right of the File Name field to download the file from the Portal Studio server to your local hard drive.
- 10 When the File Download dialog box displays, select Save.
- 11 When the Save As window appears, navigate to the location where you want to save the file, then enter a new file name—for example, `export.xml`—and click Save.
- 12 Click Close to exit the Portal Studio–Deploy window and return to the Portlet Manager.

Go to the location where the file was saved and send the file to another Portal Studio installation via an e-mail message or import the file into a different Portal Studio installation.

Deploying portal objects to a remote server

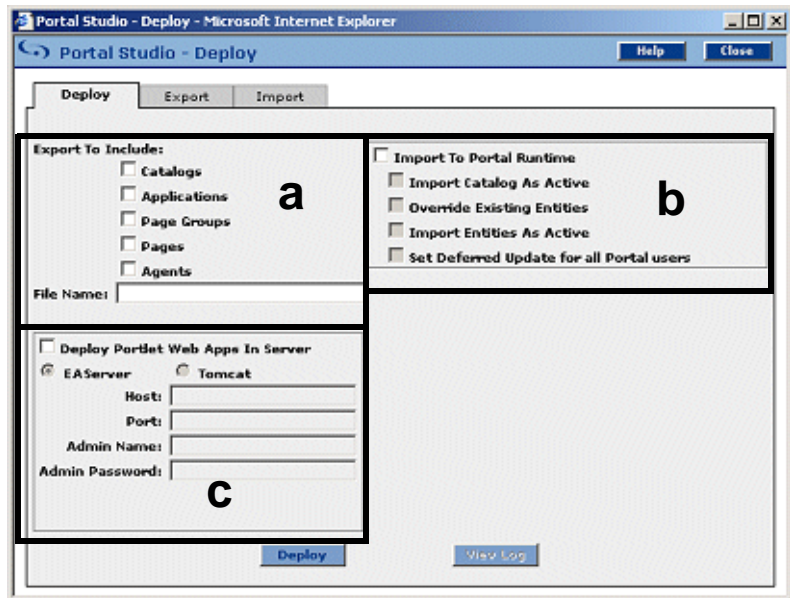
Some enterprises may have a distributed EP installation; that is, a Portal Studio and Portal Interface installation on one file system and another Portal Studio and Portal Interface installation on a remote file system.

In this scenario, you can deploy Portal Studio objects to another server.

❖ Pushing portal objects to a remote server

- 1 Log in to the Portal Studio as a user associated with a role—StudioAdmin, PortalAdmin—that has permission to perform all operations.
- 2 Select Build | Portlets from the Portal Studio left pane.

- Click Deploy on the Portlet Manager toolbar. You see the Portal Studio–Deploy window and the Deploy tab.



The window has three sections:

- Export to Include – exports portal object metadata to an XML file. Export also bundles the Web archive (WAR) files that contain Web applications required by the portlets you export. The export function on the Deploy tab then creates an enterprise archive (EAR) file that contains the XML file and WAR files (the export function on the Export tab only creates the XML file).

Note The WAR file must be manually packaged or exported from EAServer and placed in the */tmp/upload* directory to be included in the EAR file. By default, the upload directory is */tmp/upload* on UNIX and *x:\tmp\upload* on Windows.

- Import to Portal Runtime – loads the metadata from the XML file to the remote Portal Studio database. The XML is extracted from the EAR file.

- c Deploy Portlet Web Apps in Server – deploys Web archive (WAR) files found in the enterprise archive (EAR) file. The WAR files contain Web applications invoked by any of the portlets you are exporting, and places them in an enterprise archive (EAR) file. The system runs the necessary processes to deploy the WAR file on the remote application server.

You can run:

- (a), (b), and (c) together
- (a) alone
- (b) alone as long as you have the name of the EAR file from (a)
- (c) alone as long as you have the name of the EAR file from (a)
- (a) and (b) together
- (a) and (c) together
- (b) and (c) together

4 Complete these fields:

- Export To Include – select or deselect the objects to export: Catalogs, Applications, Page Groups, Pages, and Agents.
- File Name – enter the name and location of the EAR file to which you want to export objects.
- Import To Portal Runtime – import the XML portion of the EAR file into an instance of the Portal Interface on the remote application server.
- Import Catalog As Active – make the imported catalog the default display catalog. The imported XML document can contain more than one catalog, but only one catalog is marked as the default display catalog—approved and active. If you import this catalog as active, it deactivates the current default catalog and replaces it with the new catalog.
- Override Existing Entities – have imported pages override (receive a later version number) existing pages with the same name.
- Import Entities As Active – import the new entities as active, which makes the imported entities available to Portal Interface users.

- Set Deferred Update For All Portal Users – indicates whether to notify all portal users of a change in the portal content so that they can refresh their view the next time they log in to the portal.

Note During the import, if you do not select Set Deferred Update For All Portal Users, the Portal Interface user has to perform an Update for each guest page or guest page group.

- Deploy Portlet Web Apps In Server – have the system expand the EAR file on the remote server, extract the WAR file, and run the necessary processes to deploy the WAR file on the remote application server.

When you select Deploy Portlet Web Apps In Server, complete these options:

- EAServer/Tomcat – the type of application server running on the remote installation.
- Host – the host name of the remote application server.
- Port – the port number of the remote application server.
- Admin Name – the system administrator user name used to access the remote application server.
- Admin Password – the system administrator password used to access the remote application server.

5 When you finish, click Deploy.

6 When a prompt states that the operation was successful, click OK.

7 Click View Log to review the deploy operation's activities.

8 Click Close to exit the Portal Studio–Deploy window.

You should be able to see the deployed objects in the Portal Studio and Portal Interface remote installation.

Rules and restrictions

This section contains rules, restrictions, and additional information for importing and exporting specific portal objects.

Importing and exporting page groups

Page groups are a portal entity beginning with EP 6.0. Page groups created by developers in Portal Studio are called System Page Groups (SPG), as opposed to page groups created by Portal Interface users, which are personal page groups. SPGs represent default or guest page groups.

During the export process, all approved and active SPGs, including related entities, are written to the XML export document.

Like pages, SPGs are not version controlled. During the import process, existing page groups with the same name are replaced if the Replace Existing Entities option is selected in the Import tab of the Portal Studio–Deploy dialog box.

Pages within page groups are always exported regardless of their state (active or inactive). Even when a page is marked inactive within the Page Group Builder, it is still exported. Whether a page is imported as active or inactive depends on whether the Import Entities as Active option is selected on the Import tab. However, if a page is marked as inactive using the Page Group Builder, the page does not display in Portal Interface regardless of whether the Import Entities as Active option is selected.

There are two places in Portal Studio where you can deactivate or activate a page:

- **Page Group Builder** – select Build | Page Groups, then right-click an Approved page group in the detail pane and select Edit. The Page Group Builder displays all pages in the page group and each page name is followed by an Active option. When the Active option is not selected, the page is deactivated. The page will not display in only this page group. Other page groups that include this page are not affected; that is, the active/inactive page state in the Page Group Builder is pertinent only to the selected page group.
- **Page Properties** – select Build | Pages, then right-click an approved page in the detail pane and select Properties. The Page Properties window displays an Active option. Deselect Active to deactivate the page.

Note The same relationship exists between pages and portlets. Each page has an Active option for each portlet in the page. If Active is deselected for a specific portlet in the Page Manager, the portlet does not display on that page. If Active is deselected in the Portlet Manager, the selected portlet is deactivated and every page that contains the portlet does not display it.

Exporting applications

An application typically consists of pages and page groups. When you choose to export applications, all active and approved applications, including related entities, are exported. Applications do not have versioning control. Existing applications with the same name are replaced if the Replace Existing Entities option is selected in the Import tab of the Portal Studio–Deploy dialog box.

Exporting agents

The agent architecture contains multiple entities that may need to be exported when you export an agent—agents, agent servers, and adapters. An agent can also refer to pages, catalogs, and other agents. When you choose to export agents, all agent-related entities are exported.

Roles and security integration

The roles within the XML export document are the union of all the roles used by all the exported entities (page groups, pages, catalogs, portlets, templates, agents, and applications).

Note The portal administrator must create the required J2EE roles and verify that those roles are mapped to the underlying roles provided by EAServer and Enterprise Security or the Tomcat security realm. See the *Enterprise Security Administration Guide* for details.

At the time of import, export, and deploy, Enterprise Security authenticates the Portal Studio user to ensure that he or she has the appropriate role to perform the operation. The default role is PortalAdmin, which is specified by the value of the PortalAdministrationRole property in the *global.properties.xml* configuration file.

❖ Enabling deploy functionality

Enterprise Portal lets you enable the deploy functionality for Portal Studio users with the PortalAdmin role, as described in the *global.properties.xml* portal configuration file:

```
<Property name="PortalAdministrationRole"
value="PortalAdmin" description="The J2EE role
required to administer the Portal performing
```

```
export/import and update operations." menugroup="-1" />
```

Warning! This property is designed for a single role and not a list of roles. There is no parsing of the string to look for multiple roles.

- 1 To grant a Portal Studio user the Portal Admin role, you must be the PSO or have the StudioAdmin role.
- 2 Log in to Portal Studio and select Administer | Organizations from the left pane.
- 3 Select the organization in which the user resides to whom you want to give find and replace access.
- 4 Select Users in the Organization Manager menu.
- 5 Right-click the user in the detail view and select Edit Roles.
- 6 When the Edit User Roles window displays, grant the user the PortalAdmin role, then click OK.

In addition, the system also verifies that the user has the appropriate access permissions—the PortalAdmin role—against the asset EPStudio.Portal to perform the operations. These permissions are EPStudio.Deploy, EPStudio.Import, EPStudio.Export, EPStudio.UpdateCatalog, and EPStudio.UpdatePage. The portal administrator must have the PortalAdmin role to export, import, and deploy portal entities.

Importing Portal Studio objects protected by custom J2EE roles

When you import new Portal Studio objects, new objects may be protected by newly created custom J2EE roles. To see the new objects, verify that the user is granted the PortalAdmin role that allows him or her to view those objects. Once the user is granted the appropriate role, when he or she logs in to Portal Studio again, the new objects are visible and accessible.

Alternately, grant the user the PortalAdmin role. This allows him to add all Portal Studio objects regardless of his current J2EE roles. Users with this configuration do not need to log out and log back in to view the newly imported objects.

Creating Co-brands

This chapter explains how to enable your enterprise to host multiple portals from one Enterprise Portal (EP) installation.

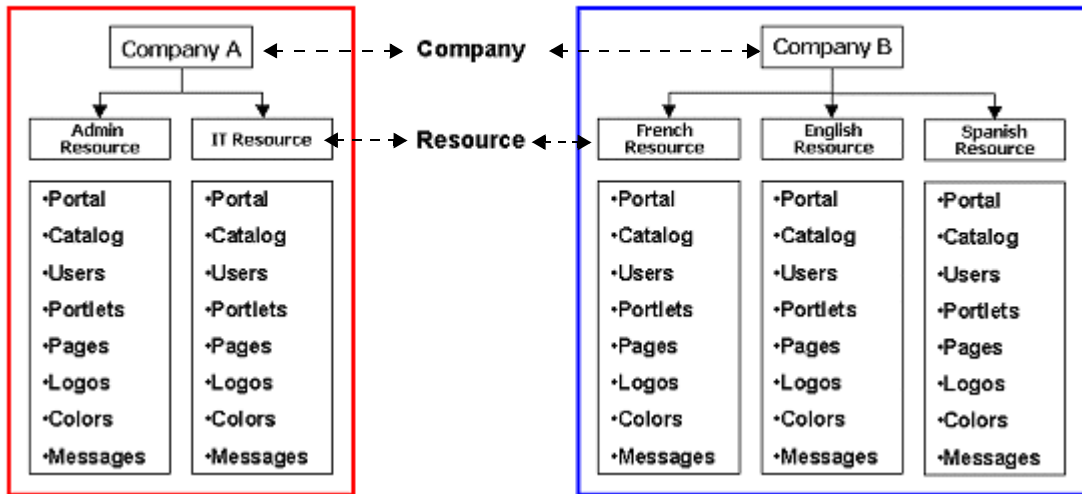
Topic	Page
Introduction	247
Creating co-brands	250
Implementing a co-brand	253

Introduction

Enterprise Portal allows you to create multiple portals (co-brands) hosted on one EP installation. Multiple portals allow you to create a different look and feel for different portal audiences. EP lets you easily create and manage an unlimited number of portals. You can configure each portal with different logos, icons, text, and colors. You can also create more complex portals by customizing portal component functionality, such as, global navigation buttons and self-registration forms.

Note Although this chapter describes how to create multiple co-brands that display differently, you can also use these instructions to modify the default onepage installation located in `%SYBASE%\EAServer\Repository\WebApplication\onepage\fw\cobrands\onepage-1` on Windows and in `$SYBASE/EAServer/Repository/WebApplication/onepage/fw/cobrands/onepage-1` on UNIX.

Figure 14-1: Single company, multiple resources, one installation



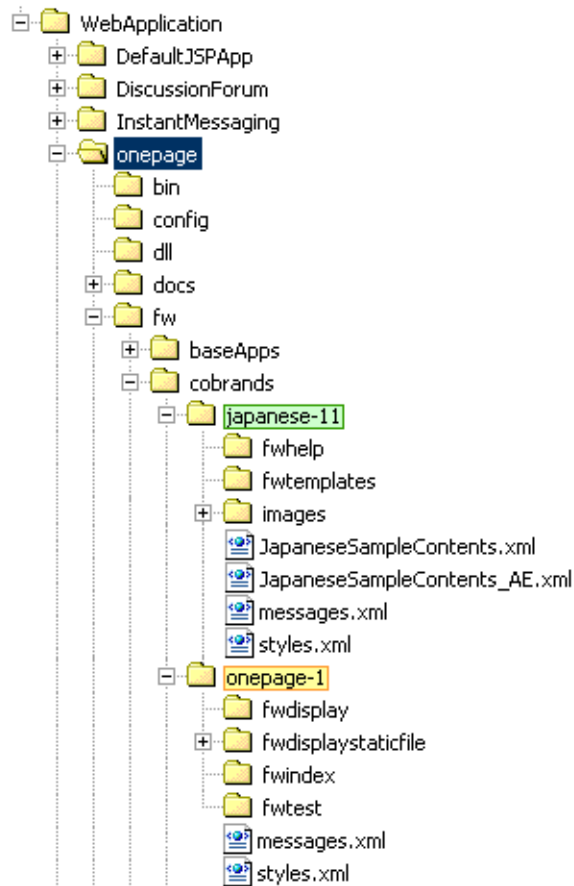
Because each co-brand has its own text messages, image files, and JSPs that produce dynamic content for the portal, you can also use co-brands to localize your portal, as illustrated in Figure 14-1 where Company B has a different resource for their English, French, and Spanish customers.

You can define co-brands to inherit their attributes from other co-brands, which allows you to make minor user interface changes without having to reimplement large amounts of content.

Each portal corresponds directly to a Portal Studio resource. When you add your domain, Portal Studio creates a default resource with a resource ID. The resource ID (RID) is the unique key that Portal Studio assigns to all portal objects associated with that resource.

To create additional co-brands for the same domain, you create an additional resource for that domain. Each resource and that resource's associated portal has a unique set of portal objects—portlets, templates, catalogs, page groups, and pages.

When you have several resources established, the directory structure resembles the structure shown in Figure 14-2.

Figure 14-2: Cobrand directory structure

In Figure 14-2, “onepage” is the default sample resource that comes with Enterprise Portal. onepage-1 and japanese-11 are examples of additional portals for the same installation. There must be a definition for each portal in the *cobrand.xml* file. In addition, each resource subdirectory has its own set of configuration files that override the properties in the master configuration file.

The onepage default installation gets its configuration from the contents of the onepage subdirectories, for example, the *JSP* login page is in *onepage\fw\baseApps\fwlogin*.

When you create a new resource, it has no portal objects. You must create or import the objects before you can access the portal from a Web browser. You can move portal objects between resources by importing and exporting those objects within Portal Studio (see Chapter 13, “Importing, Exporting, and Deploying Portal Objects,” for more information).

Creating co-brands

To create additional co-brands, repeat these steps for each co-brand you want to create. Each step references a procedure that gives detailed instructions.

- 1 Add an new resource. This step must be done by the Portal Security Officer (PSO). See “Adding a resource” on page 250.
- 2 Create a directory for the new resource and edit the *cobrand.xml* file. This step can be done by the system administrator. See “Editing the cobrand.xml file” on page 251.
- 3 Copy resource-specific files to the new co-brand directory. See “Copying resource-specific files” on page 252.
- 4 Edit the resource-specific file to fit the new co-brand. The system administrator can also perform this step. See Chapter 15, “Changing the Portal’s Look And Feel,” for specific instructions.
- 5 Implement the co-brand. See “Implementing a co-brand” on page 253.

Note You must have a unique name for each portal you create and users must have a unique user name across co-brands. For example, if you set up two portals, users must use two different user names to log in, one for each portal.

Adding a resource

- 1 Log in to Portal Studio as the Portal Security Officer or another Portal Studio user with all permissions on the SYBDOMAIN_DefaultDomain_OrgCtrlAsset asset.
- 2 Select Manage | Studio from the Portal Studio left pane.
- 3 Select Resources from the Studio Manager Objects menu.

- 4 Click New on the toolbar, or right-click in the detail view and select New Resource. The Resource Editor displays.
- 5 Complete these fields:
 - Resource Name – enter the resource’s name. This field becomes read-only after you save the resource.
 - Active – controls whether users assigned to the default resource for the selected company can log in to Portal Studio. The default is Active.
 - Portal Setup Complete – select this option when you have completed the other steps (updated *cobrand.xml*, copied configuration files to the resource directory, and so on) necessary to activate the portal associated with this resource. The remaining fields are filled in and updated automatically.
 - Description – enter a description if desired; for example, “Software company focusing on financial institutions” or “Training division”.
 - Owner, Modified By, Date Created, Date Modified – these fields are filled in and updated automatically.
- 6 Click Save.
- 7 When you see the “Resource saved” message, click OK. The new resource and resource ID display in the Studio Manager detail view.

Editing the *cobrand.xml* file

The *cobrand.xml* file defines the active co-brands for your portal installation. When you add a new resource, you must manually add the resource’s information to *cobrand.xml*, which is located in `%SYBASE%\EAServer\Repository\WebApplication\onpage\config` on Windows and in `$$SYBASE/EAServer/Repository/WebApplication/onpage/config` on UNIX.

- 1 In the `%SYBASE%\EAServer\Repository\WebApplication\onpage\fw\cobrand` on Windows and in `$$SYBASE/EAServer/Repository/WebApplication/onpage/fw/cobrand` on UNIX, create a new folder using this naming format:

```
resource name-resource_ID
```

where *resource name* is the name entered when you created the resource in Portal Studio, and *resource ID* is the portal-assigned identification number.

- 2 Open *cobrand.xml* in any text editor and copy an existing <CobrandDef> section and paste it below the last <CobrandDef> section in the file before the </CobrandDef> tag. For example, copy and paste the Sybase section.
- 3 In the section you copied, update the information to reflect the new resource. Enter:
 - Rid – the resource ID generated by Portal Studio.
 - Name – the new resource’s name.
 - Dir – the directory you created for the new resource where you copy the co-brand-specific files.
 - Charset – the character set you want the co-brand to use.
 - EmailCharset – the character set you want e-mail to use.
 - EmailerSetting – indicates whether the Portal Studio sends an e-mail message to the user when you create a new account. A setting of “1” indicates send e-mail; a setting of “0” (zero) indicates not to send a message.

Note When no SMTP server is available to send e-mail or you do not want an e-mail message to be sent, set the value of `emailerSetting` to “0.” An e-mail message is written to the `\work_root\email` directory. The value of the `work_root` property is specified in the `global.properties.xml` file, which you can change (see Appendix A, “Configuring Global Properties”). The name of the file is the e-mail address of the user.

- 4 Save and close the *cobrand.xml* file.

Copying resource-specific files

- 1 Go to
`%SYBASE%\EAServer\Repository\WebApplication\onpage\fw\baseApps`
on Windows and
`$SYBASE/EAServer/Repository/WebApplication/onpage/fw/baseApps`.
- 2 Copy *messages.xml* and *styles.xml* to the new co-brand directory.

- 3 Create the subdirectory *fwdisplaystatic\style* (or *fwdisplaystatic/style* on UNIX) in new co-brand directory
- 4 From the *baseApps\fwdisplaystaticfile\style* directory on Windows and the *baseApps\fwdisplaystaticfile/style/* directory on UNIX, copy *css.css* and *css.jsp* into the co-brand */style/* directory that you created in the previous step.
- 5 In a text editor, open *css.jsp* and change the HREF location reference to the *css.css* file to the new *css.css* file's location in the co-brand style subdirectory.
- 6 Save and close the file.

Editing resource-specific files

Edit the resource-specific files to fit look and feel you want for the new co-brand. The system administrator can also perform this step. See Chapter 15, “Changing the Portal’s Look And Feel,” for specific instructions.

Implementing a co-brand

- 1 Restart the application server.
- 2 Before you load the newly created portal, portal objects must exist in the new resource. The minimum requirement is that at least one guest page and page group is approved and active.

Log in to the Portal Studio with the user account you created for the new resource and use the instructions in Chapter 4, “Building Portlets,” and Chapter 10, “Building Pages,” to create a guest page with a portlet.
- 3 To mark the process as completed, select Manage | Studio from the Portal Studio left pane.
- 4 Right-click the resource listing in the detail view and select Edit Resource.
- 5 When the Resource Editor displays, select Portal Setup Complete, then click Save.

- 6 When the confirmation message appears, click OK, then click Close to exit the Resource Editor.
- 7 When you finish, click Logout to exit Portal Studio.
- 8 Load the portal by entering this URL in the address or location field of your browser:

`http://LOCALHOST.DOMAIN:PORT/onepage/index.jsp?rid=RID`

where *LOCALHOST* is the machine on which EP is installed, *DOMAIN* is the network domain in which that machine is registered, *PORT* is the port number used to connect to the portal, and *RID* is the resource ID assigned by the portal and viewable in Portal Studio using Manage | Studio.

- 9 Click Join Now to register as a new user and view the guest page you created.

Changing the Portal's Look And Feel

This chapter describes the portal's primary configuration files and explains how to change the look and feel of a portal. Use these instructions for single portals or when you have multiple co-brands.

Topic	Page
Introduction	255
Editing styles.xml	257
Editing css.css	258
Editing messages.xml	258
Editing global.properties.xml	259

Introduction

The most basic co-branding is accomplished by changing the logos, icons, colors, and visible text in Portal Interface. These attributes are specified in *styles.xml* and *messages.xml* that you copied to the new co-brand directory and in *css.css* that you copied to the new co-brand *style* subdirectory.

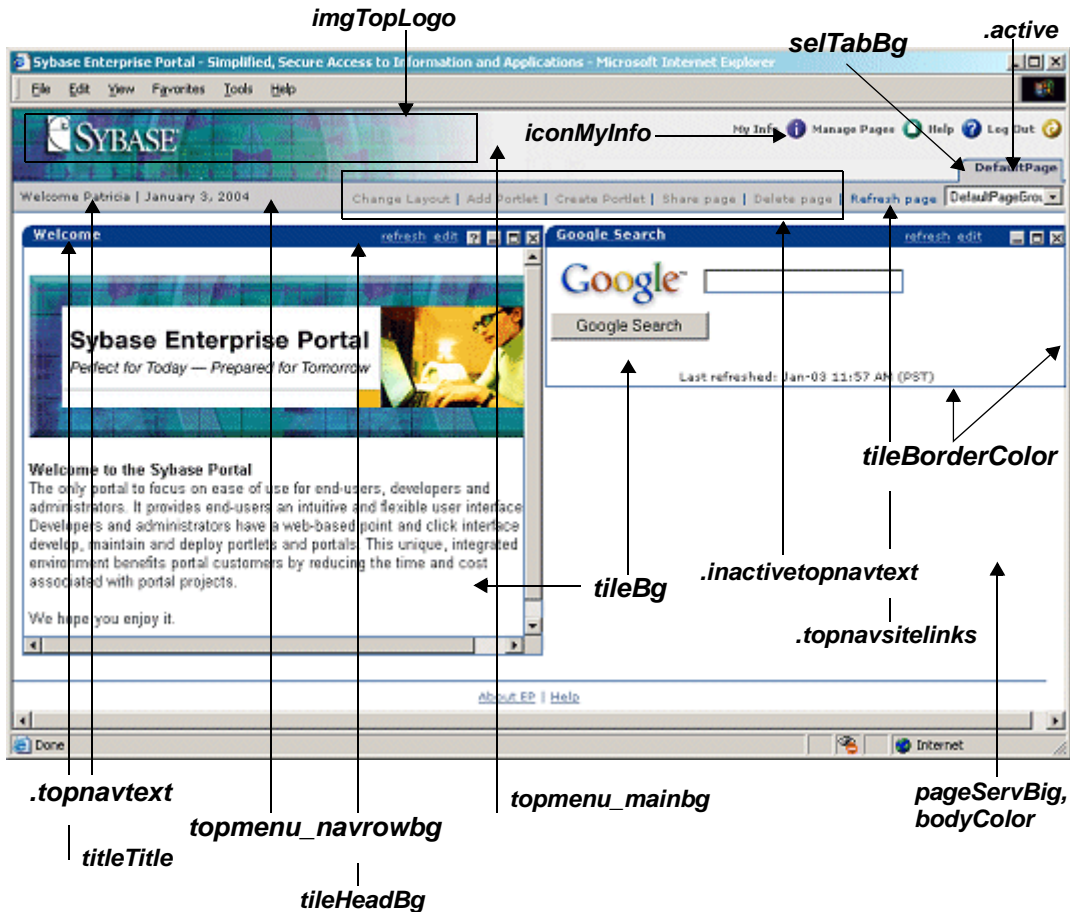
The settings you change in these resource-specific files override the settings in the onepage default files located in `\onepage\fw\baseApps` (or `/onepage/fw/baseApps` on UNIX). You can also make changes to the master configuration file `global.properties.xml`, located in

Note Enter color values as hexadecimal code rather than color names; for example, “FFFFFF” is white. Use a Web-safe color chart for reference, such as the one located at http://www.w3schools.com/html/html_colors.asp or at <http://www.visibone.com/color/>.

When you replace Enterprise Portal’s default images, icons, or tabs across all resources for an installation, put the new graphics in the appropriate subdirectory on
 %SYBASE%\EAServer\Repository\WebApplication\onpage\sy_images
 Windows and in
 \$SYBASE/EAServer/Repository/WebApplication/onpage/sy_images/
 UNIX.

To replace the images only for a particular resource, create an *images* directory within the co-brand’s subdirectory and put a copy of the new files there. When you edit *styles.xml*, you reference the location from which you want that co-brand’s images pulled.

Figure 15-1: Portal Interface style attributes



Editing *styles.xml*

The default *styles.xml* file contains the settings for Portal Interface logos, icons, and colors. Table 15-1 lists the most commonly changed properties. Use any text editor to explore and edit the file.

This file is divided into three major groups, colors, images, and misc.

- Colors – stores font colors, background colors, table colors, and so on.
- Images – stores all the images used on the site. It has been subdivided into smaller groups to keep it more manageable. Keep images in the appropriate subgroups.
- Misc – stores content setting that do not fit into color or image.

Table 15-1: *styles.xml* settings

Setting	Description
bodyColorTopMenu	The body HTML of the Portal Interface menu at the top of the Portal Interface window.
CorporateURL	URL link.
deleteIcon	Delete portlet icon.
expandIcon	Maximize portlet icon.
framesetAllIE	Frameset specification; for example, the default is 89,0,0,0.
iconAddPage	The Add Page icon at the top of the Portal Interface window.
iconHelp	The online Help icon at the top of the Portal Interface window.
iconLogout	The Logout icon at the top of the Portal Interface window.
iconManagePages	The Manage Pages icon at the top of the Portal Interface window.
iconManageTabs	Reserved for future use.
iconMyInfo	The My Info icon at the top of the Portal Interface window.
imgTopLogo, imgBottomLogo	The masthead image at the top and bottom of the Portal Interface window.
minimizeIcon	Minimize portlet icon.
mopTabCorner	Left unselected tab corner image.
nonSelTab	Unselected page group background color.
pageServBg, bodyColor	Page background.
portletHelpIcon, helpIcon	Portlet online help icon.
selTabBg	The selected page's background color.
tileBg	Portlet background color.
tileBorderColor	Portlet border color.
tileCorner	Portlet header corner image. The default is blue.
tileFont	Portlet font color.

Setting	Description
tileHeadBg	Portlet header background color.
tileLock	Secure lock portlet icon.
topmenu_mainbg	The Portal Interface top menu background color.
topmenu_navrowbg	The Portal Interface navigation bar background color.
topmenu_tabrowbg	Portal Interface page group tab background color.

Editing *css.css*

The *css.css* file is the cascading style sheet (CSS) used for the portal. Use any text editor to open *css.css* and change the settings you want to override. Save the file using the same file name “*css.css*.” If you use a different file name, the portal does not recognize the file.

Table 15-2: *css.css* settings

Setting	Description
.active	Selected page group font style (font, size, color, and so on).
.nonactive	Unselected page group font style.
.topnavtext	User greeting font style.
.topnavsitelinks	Secondary navigation active links font style.
.inactivetopnavtext	Secondary navigation inactive links font style.
.topabsetdropdown	Page group drop-down menu style.
.tileTitle	Portlet title font style.
.tileFunctions	Portlet header functions (refresh, edit).

Editing *messages.xml*

The *messages.xml* file contains settings for Portal Interface text that displays in menus, messages, and so on. Table 15-3 lists additional message and text groups that you can customize. Use any text editor to explore and edit the file.

Table 15-3: *messages.xml* settings

Setting	Description
DSF_ABOUT_EP	Link message.
TM_HELP	Help message.

The setting names in an application element tag must be unique across all applications. To ensure that the names are unique, use the application name (short name) as a prefix. See Table 15-4 for reference.

Table 15-4: message group short names

Message group	Setting section
Days and months	DaysAndMonths short name/prefix=DATE
Page titles	PageTitles short name/prefix=PT
Portal login	LoginPage short name/prefix=LP
Top menu names	FWTopMenu short name/prefix=TM
Portlet edit	FWTileEdit short name/prefix=TE
Portal registration	FWRegistration short name/prefix=RG
User's information	FWMyInfo short name/prefix=MI

Editing *global.properties.xml*

The *global.properties.xml* file describes database table contents that must be modified when administering some features such as Default Tabs and Pages, and Default Tiles on the Default Tabs and Pages.

See Appendix A, “Configuring Global Properties,” for more information on viewing and modifying this file’s contents.

Note When you make changes to *global.properties.xml*, you must restart the application server for your changes to take effect.

Configuring Global Properties

The *global.properties.xml* file is the master configuration file for Portal Studio and Portal Interface. This appendix describes the properties that *global.properties.xml* contains and how you can change these properties for different Enterprise Portal configurations.

Topic	Page
Introduction	261
Viewing <i>global.properties.xml</i> contents	262
Updating global properties	286
Limiting access to <i>global.properties.xml</i>	286

Introduction

This appendix lists the key configuration properties in *global.properties.xml*, provides tips on using this file, and gives examples of how to change property values for different installation scenarios.

As you set up Enterprise Portal and modify *global.properties.xml*, note that:

- In a default EP installation, *global.properties.xml* is located in the *Repository/WebApplication/onepage/config* subdirectory of your EA Server installation.
- You must have a separate *global.properties.xml* for each portal installation. See Chapter 14, “Creating Co-brands,” for details.
- When you change the contents of *global.properties.xml*, you must restart the application server to implement the changes.

Note Some of the remarks in the *global.properties.xml* file are no longer applicable. Use the information in this appendix instead.

Viewing *global.properties.xml* contents

Once you install Enterprise Portal, there are several properties in *global.properties.xml* that you can modify.

The contents of *global.properties.xml* are organized into these property groups:

- Global property group – properties for general portal settings.
- Audit property group – properties for Portal Interface and Portal Studio auditing.
- UWP property group – portlet playback engine settings.
- Product configuration property group – registers configuration file names.
- Docservers property group – Portal Studio application settings.
- Portal property group – properties for the portal framework.
- Agents property group – agent configuration properties.
- Clickthru property group – click-through settings. Not included in this document.
- Nav property group – properties specific to Web element portlet navigation.
- Operation events – not included in this document; legacy property settings that are not used.
- Capture logging properties – properties for capture logging events. Not included in this document.
- Database property group – properties for the data servers the portal is using.

The *global.properties.xml* file has a section for each property group in the preceding list. Property groups and properties that are not discussed in this appendix are either not used, or use a default value that you should not change.

Note The strings “*eashost*”, “*asehost*”, “*hostname*”, “*aseport*”, and “*domain*” in *global.properties.xml* are replaced by the appropriate values during the installation of Enterprise Portal. When configuration is successful, the administrator sees the actual values for these properties instead of the above string variables.

Global property group

This group contains general settings, including server names, addresses, mail properties, and portal properties.

Properties are listed in alphabetical order for ease of reference.

Global property name	Default value	Description
alwaysValidateSession	true	(true/false) When this value is set to “true,” the user’s portal session is validated on every request. There is a slight performance penalty for this. When this value is set to “false,” the session is checked only when security configuration requires an explicit authorization.
Anonymous.Roles	everybody PortalGuest	Lists the J2EE role values for anonymous users. Values are comma separated.
catalog_cache_refresh_list	127.0.0.1:8080/onepage	<p>Keep the default entry for a single-machine installation.</p> <p>For a multimachine installation, enter the list of machines with catalogs that should be refreshed when a cache refresh is triggered. Use commas to separate entries in the list.</p> <p>If you use the machine name, there must be a DNS entry inside the machine to resolve the entry.</p> <p>This comma-separated list should consist of machine names that are internally addressable from within the servers themselves and should include the port number followed by the name of the Web application.</p> <p>Example:</p> <pre>catalog_cache_refresh_list= demo1.\$DOMAIN:8080/onepage, demo2.\$DOMAIN:8080/onepage"</pre> <hr/> <p>Note If the <code>xml_catalog_serialize</code> property is set to “yes,” you must also set a value for <code>catalog_cache_refresh_list</code>.</p>

Global property name	Default value	Description
changeMyInfo	true	(true/false) When this value is set to “false”, the MyInfo links for “change password”, “personal information”, and “join now” do not display.
client_virtual_path	/onepage/	Used for JSP forward/include; for example: BEA=/onepage/ iplanet=/onepage/
default_http_port	8080	The application server port number. You provide the initial value at installation. This value must match the default application server HTTP port number and must be the same as the application server settings. Note If you change EAServer listener port numbers, you must also change this value to match.
default_https_port	8081	The application server HTTPS port number. This value must match the default application server HTTP port number and must line up with the application server settings. Note If you change listener port numbers, you must also change this value.
default_tile_href	“ “	Default HREF that appears on portlet title bars. The default is none. If you wanted the same HREF to display on most portlet title bars, enter that HREF here. This value can be overridden during portlet creation.
DefaultResourceID	1	The resource ID (RID) that is used if one is not included in a request.
DefaultUserID	1	The user ID that is used if one is not included in a request.
deferupdate	on	(on/off) When this value is set to “on”, the user’s view is updated only during login. When an operation to update user pages is performed, the command is recorded for later execution instead of executing immediately.

Global property name	Default value	Description
disableAutomaticReauthentication	false	(true/false) Set to “true” to disable the automatic re-authentication feature of the portal when using EPSecurity. The PortalSession Primary Key information will not be wrapped in the SessionID (SID) parameter.
DocumentRoot	<i>/PortletDocs</i>	Indicates the directory where portlet documents reside.
domain	<i>subdomain.domain</i>	The DNS name of your portal Web server in the format “.xxx.xxx”. Include any subdomain. This value is used by the portal’s servers. If you change your installation’s domain, you must change the value here to the new domain name.
ep_security_eas_guest_pwd	guest	Password of EP guest account used to connect to EAServer during self-registration.
ep_security_eas_url	<i>iiop://hostname.domain:9000</i>	URL for EAServer IIOP service when EP security is remotely installed. This may be a semicolon separated list if security is deployed to a cluster.
https.playback	on	(on/off) Enables secure bypass for Web applications. Turn “off” when using Oracle as the portal database.
javascript.domain	<i>domain</i>	DNS name of your portal JavaScript Web server in the same format as the “domain” without the leading period (.). Included any subdomain. This value is used by the browser from which you are running the portal.
mail.host	10.22.97.100	Fully qualified DNS name or IP address of the mail server. This should be a server that is internally available to you. Note You must change this value after installation to enter your own mail.host. The default entry is not a valid mail host.
mail.transport.protocol	smtp	(smtp/imap) Mail protocol. Currently, only SMTP is supported.
markupNoJSFile	<i>/config/noJavascriptOcMarkup.txt</i>	If you cannot navigate through a Web site because of JavaScript, enter that site here and Portal Studio ignores or removes the JavaScript.

Global property name	Default value	Description
maxPageName	25	Maximum number of characters allowed per tab/page name.
maxTabSetChar	5000	Maximum number of characters per tabset.
maxTabSets	99	Maximum number of page groups (tabsets) allowed. Valid range is 1- 99.
multipleGuestPages	true	(true/false) Controls the number of guest pages to display. Set to “false” to display only one guest page.
page.group	on	(off/on) Set to “on” to enable page groups in the portal.
passwordExpirationWarningWindow	30	When the portal.epSecurity property is set to “true,” users see warning messages a number of days prior to their password expiration equal to the value of this property. Set this property to 0 (zero) to disable password expiration warnings.
PortalAdministrationRole	PortalAdmin	<p>The J2EE role required to administer the portal, perform export/import and update operations.</p> <hr/> <p>Warning! This property is designed for a single role and not a list of roles. There is no parsing of the string to look for multiple roles.</p> <hr/>
portal.defaultRID	1	Default resource ID (RID) for the portal. If a request does not include a RID, the systems falls back to the RID specified here. When you add a new resource in Portal Studio, you may want to change the value here.
portal.epSecurity	true	(true/false) Set to “true” to register users with Enterprise Security.
portal.host	<i>hostname</i>	Host name of the page playback server. The initial value is supplied at installation.
portal.portlet_create_lock	E,T,U	<p>Lists the page types on which create, add, and move portlet cannot be done:</p> <p>E – default T – catalog U – guest page</p>

Global property name	Default value	Description
proxy	off	(on/off) Set this value to “on” only if you are planning to use a HTTP proxy server, and this proxy server is installed and configured.
proxy.bypass_list	127.0.0.1 localhost	Value – host1 host2 The addresses and host names of Enterprise Portal machines that must bypass the proxy server. When there are machines that cannot go through the proxy server (typically for internal sites), create the list of machines names; separate machine names using a bar (). Note See the HTTPConnection Javadocs for information on dontProxyFor() method.
proxy.host	127.0.0.1	IP address of the proxy server. Enter this value only if the proxy property is set to “on.”
proxy.password	DefaultProxyUserPassword	Configure this property only if the proxy property is set to “on” and the user exists on the proxy server
proxy.port	3128	Port number of the proxy server cache. Configure this property only if the proxy property is set to “on” and the user exists on the proxy server.
proxy.user	DefaultProxyUserName	Configure this property only if the proxy property is set to “on” and the user exists on the proxy server
redirfile	/config/redirlist.txt	The location and content to which a user is redirected if he or she enters a URL that does not exist or is unavailable.
registration.organization	o=Sybase Inc,c=us"	Defines the EP security organization where self-registered users are added. The default organization is the root organization
registration.userJ2EERoles	“ “	Lists the J2EE role values. Values are comma-separated.
RequiredRoles	everybody	Lists required J2EE role values in conjunction with roles specified in <i>web.xml</i> . Values are comma-separated.
resetPasswordEmail	pso@yourCompany.com	A semicolon-delimited list of e-mail addresses to which reset password notifications are to be sent.

Global property name	Default value	Description
RoleBaseDisplay	true	(true/false) Indicates whether listings should show objects with only the same role as the user.
RoleBaseDisplaySeeAllRoles	superuser	Supersedes the RoleBaseDisplay property. Any roles listed here see every item as if RoleBaseDisplay was set to "false". Values are comma-separated.
secure	off	(off/on) Set to "on" to enable HTTPS for the Web container. HTTPS will be used for log in and registration.
SecureHostname	<i>LoadBalanceSecure Serverhost.domain</i>	Fully qualified host name for the load-balanced secure publish/subscribe server used by Portal Studio and the portlet playback server (UWP or UPP).
secure_login	off	(off/on) Set to "off" to have Portal Interface login and self-registration occur over HTTP. When this value is set to "on", Portal Interface uses HTTPS during login and self-registration. Note Before setting this value to "on", you must configure EAServer for HTTPS. See the <i>Enterprise Security Administration Guide</i> for more information.
secure.tile.host	<i>hostname</i>	Hostname of the secure HTTPS portlet playback server. The initial value is supplied at installation. This value should be the same as tile.host.
send.mail	true	(true/false) Set this value to "true" if you want SMTP-based e-mail sent when you create a new accounts in Portal Studio, or when Portal Interface users register for the first time or request to change their password. If set to "false," e-mail content is written to the <i>workroot/e-mail</i> directory. Note You can leave this value set to "true," then set the value to "false" for individual portals in the <i>cobrand.xml</i> file. See Chapter 14, "Creating Co-brands."

Global property name	Default value	Description
server_virtual_path	/	Used for JSP forward/include; for example: BEA=/ JRun=/ iplanet=/onepage/
sharePage	on	(on/off) Enables page sharing in Portal Interface.
SOAPServer	127.0.0.1	The SOAP server machine name.
tabingChar	4	The number of characters that tabs use for the corners.
tabset	on	(off/on) Enables tabsets.
tile.host	<i>hostname</i>	Host name of the portlet playback server. The initial value is supplied at installation.
Note Tile is synonymous with portlet.		

Global property name	Default value	Description
tile_style	3	<p>Values – 1, 2, 3</p> <p>1 <IFRAME> tag only. Defines an inline frame for the inclusion of external objects including other HTML documents. Unlike frames created using <FRAMESET> and <FRAME>, <IFRAME> creates a frame that sits in the middle of a regular nonframed Web page. <IFRAME> works like , only instead of putting a picture on the page, it puts another Web page on the page.</p> <hr/> <p>Note Internet Explorer versions 5.x and 6.x support <IFRAME>. Netscape 6 supports <IFRAME> but not all of the attributes. Versions of Netscape earlier than version 6 do not support <IFRAME>.</p> <hr/> <p>2 <DIV> tag only. The <DIV> tag has a single attribute—align—with four possible values—left, center, right, and justify. The <DIV> tag is used primarily to create divisions that require the same alignment attribute within an HTML document.</p> <hr/> <p>Note Does not work on Netscape 7, which does not support Web data within <DIV> tags.</p> <hr/> <p>3 Mixed mode. The portlet is formatted using either an <IFRAME> or a <DIV> tag, depending on which tag is more appropriate to the portlet’s content.</p>
turnOffSecurePassword	<i>secure_passwd_off</i>	(on/off) Toggles use of secure password prompt by Portal Interface when using http portlets created from Portal Studio.
use_https	on	(on/off) Enables secure navigation.
versioning	on	(on/off) Enables versioning of portlets and catalogs.

Global property name	Default value	Description
WebServiceRoot	<i>/ws</i>	<p>This property indicates the root directory for web services.</p> <p>There are two ways to enter this information:</p> <ul style="list-style-type: none"> • Enter an absolute location. For example, if you enter <i>d:\work\service\ws</i> on Windows or <i>/work0-/service/ws</i> on UNIX, the system adds files directly to the specified location. You must ensure that the location can be accessed from a URL, like <i>http://localhost:8085/onepage/ws</i>. • Enter a relative location. For example, if you enter <i>service\ws</i> on Windows or <i>service/ws</i> on UNIX, the system adds files to the location relative to the current Web application; specifically, <i>root/work0/tomcat/webapps/onepage/service/ws</i> (on UNIX). You must ensure that the location can be accessed from a URL, like <i>http://localhost:8085/onepage/service/ws</i>.
wireless_portal	off	(on/off) Set to “on” to enable wireless_portal, which uses the Windows player to play captured portlets.
WorkRoot	<i>/tmp</i>	Base working directory to which the portal can write log files, upload files, and so on.
xml_catalog_serialize	yes	<p>(yes/no) Enter “yes” to create local serialized objects of the XML catalog in your local <i>/onepage/lib</i> directory. This allows a quicker-loading catalog.</p> <p>Enter “no” to disable catalog serialization.</p>

Audit property group

This group contains properties for configuring auditing in Portal Interface and Portal Studio. Properties are listed in alphabetical order for ease of reference.

Audit property name	Default value	Description
auditEnabled	false	(true/false) Set this value to “true” to enable auditing in Enterprise Portal and Portal Studio. Note This is not audit filtering. Audit filtering is set using the Jaguar Manager Configuration service.
auditPortletPlayback	false	(true/false) Set this value to “true” to enable auditing of portlet playbacks.
auditRoleFilter	false	(true/false) Set this value to “true” to enable auditing of role filtering on auditable resources.

UWP property group

UWP is the Universal Window (portlet) Player, also referred to as UPP— Universal Portlet Playback engine. This is the portlet content handling framework that provides a uniform mechanism for portlet execution. UWP provides a portlet container and a personalization engine.

Properties are listed in alphabetical order for ease of reference.

UWP property name	Default value	Description
CacheRefreshList	127.0.0.1	Comma-separated list of internally addressable machines names to refresh the cache. Example: CacheRefreshList= machine1.\$DOMAIN, machine2.\$DOMAIN Note You should update this list for a multimachine EP configuration.
ContentCacheMaxEntries	250	The maximum number of entries in the portlet ContentCache object. This is used for caching resolved portlet contents.
ContentCacheLRU ReductionSize	5	The number of entries removed at time from the Portlet ContentCache object when implementing the LRU policy. This is used with caching content of resolved portlets.

UWP property name	Default value	Description
DefaultTemplateName	Template1	<p>The default template used for single-element portlet. See Chapter 8, “Building Templates.” You can change this to any template you want. For example, if you wanted every portlet to use the same header logo or footer, you could create that template and enter that template’s name here.</p> <p>When you change the template name here, it changes the default template for all resource IDs.</p>
ErrorFile	<i>/logs/uwp.err</i>	Location and name of the file to log error messages, relative to the value for WorkRoot.
ObjectCacheLRUReduction Size	5	<p>The number of least recently used (LRU) entries removed at a time from the UWPObjectCache object when implementing the LRU policy.</p> <hr/> <p>Note The LRU policy is a page-replacement policy that removes from main memory the pages that show the least amount of recent activity. This policy is based on the assumption that these pages are the least likely to be used again in the immediate future.</p> <hr/> <p>This value is used for <code>clickthru_cache</code>, <code>windows_def_omt_cache</code>, and <code>parsed_template_cache</code>.</p>
ObjectCacheMaxEntries	250	Every object that goes into and out of the database gets cached. This value is the maximum number of entries allowed in the UWPObjectCache object. This value is used for <code>clickthru_cache</code> , <code>windows_def_omt_cache</code> , and <code>parsed_template_cache</code> .
StorageConnector	onepage-generic	<p>Values – <code>onepage-instantdb</code>, <code>onepage-generic</code>, <code>onepage-cloudscape</code>, <code>onepage-cloudscape-iplanet</code>.</p> <hr/> <p>Warning! Do not change this value.</p> <hr/>
ThisMachineName	127.0.0.1	The host name or IP address of this machine. If this machine is also a cache server, the value of ThisMachineName and UWPCacheServer must match.
TraceFile	<i>/logs/uwp.trc</i>	Location and name of the file to log trace messages, relative to the value for WorkRoot.
TraceOn	false	<p>(true/false) Used for basic tracing.</p> <hr/> <p>Note This value should be set to “off” in a production environment.</p> <hr/>

UWP property name	Default value	Description
UsePreCompiledJSP	false	(true/false) Set to “true” to have the application use precompiled JSP that are built ahead of time (.WAR file mode).
UWPCacheServer	127.0.0.1	The cache server is used for clickthru. If this machine is also a cache server, the value of ThisMachineName and UWPCacheServer must be the same. Warning! Do not change this value.
UWPIntegrationHost	<i>localhost.domain</i>	Used by BEA/Yahoo integrations to make requests for portlet playback, edit screens, and returning to the OEM portal screens. This name must be visible from outside the firewall.
UWPWebServiceHost	<i>localhost.domain</i>	This property must be the actual name of the machine that is a server for the Web services WSDL files. This name must be visible from outside the firewall.
VerboseTrace	false	(true/false) Used for detailed tracing. Note This value should be set to “false” in a production environment.

Product configuration property group

This group describes where to register configuration file names. Properties are listed in alphabetical order for ease of reference.

Audit property name	Default value	Description
epsecurity.xml	true	(true/false) Set this value to “true” to enable Enterprise Enterprise Security.
epstudio.xml	true	(true/false) Set this value to “true” to enable Portal Studio.

Docservers property group

Docservers retrieve portlet content from the database. Properties are listed in alphabetical order for ease of reference.

Docservers property name	Default value	Description
AuthenticationUsing	EPSecurity	Values – EP Security, Database, LDAP Authentication technique to use. Currently, Enterprise Portal supports only EP Security and database authentication for the doc server.
CacheRefreshList	127.0.0.1	Comma-separated list of internally addressable machines on which to refresh the cache. Example: CacheRefreshList= demo1.\$DOMAIN, demo2.\$DOMAIN
ClickacrossOn	true	(true/false) Indicates whether to expose the click-across user interface.
DBLookUpDocument Caching	true	(true/false) Whether to cache documents that are typically database lookups but change only with manual updates to the database; for example, category, subcategory. Change this value to “false” when someone makes changes to database settings using something other than Portal Studio.
DeployHost	<i>hostname.domain</i>	Real name of the machine used for import, export, and deploy functions. This name should be visible from outside the firewall
EPSecurityHost	<i>hostname.domain</i>	Real host name of the application server where Enterprise Security middleware is located, performing user authentication.
EPSecurityHostPortNumber	9000	Port number of application server used for the iiop connections.
ErrorFile	<i>/logs/ds.err</i>	Location and name of the file in which to log error messages, relative to the value of WorkRoot.
FrequentlyChangable DocumentCaching	true	(true/false) Whether to cache documents that change because of changes to other documents.
MailOn	false	(true/false) Whether to use a mail server for Agent servers.
MaxBytesForDocuments ToCache	3000000	The maximum number of bytes cached. 3000000= 3MB.

Docservers property name	Default value	Description
MaxNumberOfDocumentsToCache	100	The maximum number of documents cached.
NonfrequentlyChangedDocumentCaching	true	(true/false) Whether to cache documents that are updated infrequently; for example, a list of resources. The default maximum number of documents cached is 100.
UseAccessControl	true	(true/false) Whether to apply access control. Warning! Do not change this value.
UseStudioDatabase	true	(true/false) Set this value to “false” to use the Portal Studio framework without loading the Portal Studio database.
TraceFile	<i>/logs/ds.trc</i>	Location and name of the file in which to log trace messages, relative to the value of <i>WorkRoot</i> .
TraceOn	false	(true/false) Used for basic tracing. Note This value should be set to “off” in a production environment.
VerboseTrace	false	(true/false) Used for detailed tracing. Note This value should be set to “false” in a production environment.
WebServerOn	true	When this value is set to “true,” the Web service user interface displays and Web service elements can be created.

Portal property group

This group lists portal settings. Properties are listed in alphabetical order for ease of reference.

Portal property name	Default value	Description
ErrorFile	<i>/logs/portal.err</i>	Location and name of the file in which to log error messages, relative to <i>WorkRoot</i> .
TraceFile	<i>/logs/portal.trc</i>	Location and name of the file in which to log trace messages, relative to <i>WorkRoot</i> .

Portal property name	Default value	Description
TraceOn	false	(true/false) Used for basic tracing. Note This value should be set to “off” in a production environment.
VerboseTrace	false	(true/false) Used for detailed tracing. Note This value should be set to “false” in a production environment.

Agents property group

This group contains agent server settings. Properties are listed in alphabetical order for ease of reference.

Agents property name	Default value	Description
AgentServerID	0	Default Agent Server ID. See Chapter 12, “Using Enterprise Portal Agents.”
ErrorFile	<i>/logs/as.err</i>	Location and name of the file in which to log error messages, relative to WorkRoot.
HeartBeat	300000	Default heartbeat time in millions of seconds
MaxLogRows	150	Sets the maximum number of data (rows) returned from the database to the browser for log information on a view log request. Set to zero for all.
PushAgentLog	2	The number of heartbeat internalize after which the agent transaction log will be pushed to the PS server.
PSHost	<i>hostname.domain</i>	Fully qualified name of the Portal Studio/agent server machine.
PSPort	8080	Port on which the PSHost server is listening for requests.
StatusUpdateFreq	10	The default status report/update frequency.
TimeZone	PST	Publish Server (database) Time zone.
TraceFile	<i>/logs/as.trc</i>	Location and name of the file in which to log trace messages, relative to WorkRoot.
TraceOn	false	(true/false) Used for basic tracing. Note This value should be set to “off” in a production environment.

Agents property name	Default value	Description
TransactionFile	<i>/logs/transaction.log</i>	File to log agent execution events, relative to the directory in the WorkRoot property.
VerboseTrace	false	(true/false) Used for detailed tracing. Note This value should be set to “false” in a production environment.

Nav property group

This group contains one-click capture properties. The nav server creates the Content Capture Language (CCL) statements that define portlets in Portal Builder. Properties are listed in alphabetical order for ease of reference.

Nav property name	Default value	Description
acx	local	<p>(local/remote/none) ActiveX (ACX) control location. Setting this value to “none” disables ActiveX support completely and the option disappears from the portlet capture wizard interface.</p> <p>For Windows, this value is best set to “local” so that ActiveX support is carried out on the local machine. However, it can be set to “none” or “remote” if required.</p> <hr/> <p>Warning! Setting ACX to “local” on UNIX causes errors to appear in the application server’s log file, and causes capture and play back to fail.</p> <p>To use the ActiveX component during capture and playback on UNIX, set this property to “remote” and set “acx.host” and “acx.servlet.url” properties. The request to use ActiveX is delegated to a remote EP installation on Windows that handles the request and returns the result. “acx.remote” must point to the host and port where the ActiveX support servlet is located; for example:</p> <p style="text-align: center;"><code>http://demo.sybase.com:4040/</code></p> <p>The URL must be a Windows machine with EP installed.</p> <p>The “acx.servlet.url” property is set to <code>/onepage/servlet/ACX</code> and should remain that value unless you are directed to change the value by a qualified Sybase support engineer.</p> <hr/>
acx.host	<code>http://hostname:port</code>	Indicates the host/port where ACX servlet is located.
acx.servlet.url	<code>/onepage/servlet/ACX</code>	Indicates the URL of the ACX servlet.
		<hr/> <p>Warning! Sybase recommends that you change this property only with the guidance of a Sybase support engineer.</p> <hr/>
includelistfile	<code>/config/includelist.txt</code>	OEM data file.
log	<code>logs/nav.log</code>	The location and name of the navigation log.

Nav property name	Default value	Description
LogAllNav	off	(on/off) Set this value to “on” to turn on tracing for nav, HTTP, navigate, and clickthru. Note When this value is set to “on,” all other loggers in the Nav property group are also be turned on.
nav.general.host	<i>hostname</i>	Value – http://<host>.<subdomain> Host name of the fully qualified DNS name of the aliased general host for all nav servers.
nav.general.secure.host	https:// <i>hostname</i>	Value – https://<host>.<subdomain> Host portion of the fully qualified DNS name of the aliased general HTTPS content capture secure server.
nav.host	<i>hostname</i>	Value – http://<host>.<subdomain> The host portion of the DNS name of the server.
nav.secure.host	<i>hostname</i>	Value – https://<host>.<subdomain> The host portion of the DNS name of the server.
NavTraceOn	off	(off/on) Indicates whether to perform navigation tracing.
use_include	off	Set this value to “on” to limit navigation scope.

Capture logging properties

Capture logging property name	Default value	Description
LogFile	<i>/logs/capture.log</i>	The location and name of the log file for capture output.
CaptureTraceOn	off	Values – on, off Set this value to “on” for tracing. Note Fatal and error levels are always logged. Setting this property value to “on” sets the trace level to debug. Sybase recommends that you set TraceOn to “off” in a production environment.

Database property group

This group contains the database properties for each server.

Properties are listed in alphabetical order for k of reference.

Property name	Default value	Description
enable.tracing	false	Set this value to “true” to write traces to a log file.

Database property-specific parameters

Table A-1: DataPool property changes and additions

Property name	Property description	What's changed?
DatabaseImpl value="oracle-javaprocs"	Selected database implementation.	New database implementation for use with Oracle database.
<pre><DatabaseImpl name="oracle-javaprocs" vendor="oracle"> <DatabaseService name="home" pools="op_portal_oracle" balancer="com.onepage.db.UserDistrib" stored_proc_handler="com.onepage.db.JavaProcHandler" platform_handler="com.onepage.db.OraclePlatformHandler" description="service descript" /> <DatabaseService name="auth" pools="op_portal_oracle" balancer="" stored_proc_handler="com.onepage.db.JavaProcHandler" platform_handler="com.onepage.db.OraclePlatformHandler" description="service descript" /> <DatabaseService name="tools" pools="op_portal_oracle" balancer="" stored_proc_handler="com.onepage.db.JavaProcHandler" platform_handler="com.onepage.db.OraclePlatformHandler" description="service descript" /> <DatabaseService name="uwp" pools="op_portal_oracle" balancer="" stored_proc_handler="com.onepage.db.JavaProcHandler" platform_handler="com.onepage.db.OraclePlatformHandler" description="this is for UWP connectors" /> </DatabaseImpl></pre>		
DatabaseImpl = op_portal_asa	Selected data pool implementation.	Data pool implementation added for Adaptive Server Anywhere with parameters for all implementations—password_encrypted.
<pre><DataPool name="op_portal_asa" class="com.onepage.db.DbConnectionBroker" driver="com.sybase.jdbc2.jdbc.SybDriver" url="jdbc:sybase:Tds:@ASAHOST@:@ASAPORT@" user="dba" password="SQL" password_encrypted="false" minConnections="1" maxConnections="50" unusedtimer="360" checkouttimer="60" reuse="30" debug="true" /></pre>		

Property name	Property description	What's changed?
DataPool name = op_portal_oracle password_encrypted = false	<p>Values – true, false (default)</p> <ul style="list-style-type: none"> Encrypts the database passwords based on the password provided here. You must put the encrypted version of the password into the “password” property parameter in the “datapool” tag, and set the encrypted property to “true.” During allocation of the datapools, if the “password encrypted” parameter is set to “true” the password specified is decrypted and used. If “password encrypted” is not set or set to “false,” the password is used as specified in <i>global.properties.xml</i>. <p>By default, the “password encrypted” property is set to “false.”</p>	New attribute “password_encrypted” added.
DataPool name = op_portal_ase password_encrypted = false debug = false	<ul style="list-style-type: none"> password_encrypted values – true, false (default) See op_portal_oracle for details. debug values – true, false (default) When set to “false,” debugging is turned off. This value should remain off in a production system. 	New attributes “password_encrypted” and “debug” added.

Updating global properties

The *global.properties.xml* file is the master configuration file for Portal Studio and Portal Interface. After you install Enterprise Portal, there are properties you must change to match your enterprise. This section describes the changes you need to make.

Read through all the procedures in this section to determine which properties you must change. After changing each property, save the file, but do not close it, until you have made all of the necessary changes. When you are finished, save and exit the file, close the text editor, then stop and restart the application server.

❖ **Sending e-mail to a file**

The `send.mail` property, by default, is set to “true,” which means that SMTP-based e-mail is sent when a new user account is created. If you do not have SMTP-based e-mail, set this value to “false,” which writes the e-mail content to `<workroot>/e-mail`.

- 1 Use any text editor to open `global.properties.xml`, which is located in the `Repository/WebApplication/onepage/config` subdirectory of your EAServer installation.
- 2 Search for the `send.mail` property and change the value to “false.”
- 3 Save the file but do not close it.

❖ **Sending e-mail to users**

- 1 If you have an SMTP-based e-mail host and leave the `send.mail` value set to “true,” you must change the `mail.host` property value for this functionality to work.
- 2 Search for the property called `mail.host`. Change the value to the name of your company’s SMTP-based e-mail host server.
- 3 Save the file.

❖ **Configuring multi-machine installations**

If you installed EPon multiple machines, there are several properties in `global.properties.xml` that you must update to reflect this configuration.

- 1 Search for the `catalog_cache_refresh_list` property in the Global properties section of the configuration file.
- 2 Enter a comma-separated list of machines with catalogs that must be refreshed when a cache refresh is triggered. If you use the machine name, there must be a DNS entry inside the machine to resolve the entry.

This list must be machines that are internally addressable from within the servers themselves and must include the port number, followed by the Web application name, which is “onepage.” For example:

```
"catalog_cache_refresh_list=
demo1.$DOMAIN:8080/onepage,
demo2.$DOMAIN:8080/onepage"
```

- 3 Search for the `CacheRefreshList` property in the UWP properties section of the configuration file.

Enter a comma-separated list of internally addressable machines names to refresh the cache. If you use the machine name, there must be a DNS entry inside the machine to resolve the entry.

This list must be machines that are internally addressable from within the servers themselves and must include the port number, followed by the Web application name, which is “onepage.” For Example:

```
"CacheRefreshList=
demo1.$DOMAIN:8080/onepage,
demo2.$DOMAIN:8080/onepage"
```

- 4 Search for the `CacheRefreshList` property in the Docserver properties section of the configuration file.

Enter a comma-separated list of internally addressable machines names to refresh the cache. If you use the machine name, there must be a DNS entry inside the machine to resolve the entry.

This list must be machines that are internally addressable from within the servers themselves and must include the port number, followed by the Web application name, which is “onepage.” For example:

```
"CacheRefreshList=
demo1.$DOMAIN:8080/onepage,
demo2.$DOMAIN:8080/onepage"
```

❖ **Configuring different EAServers and ports for the same domain**

This example illustrates the settings used to configure a different machine and port for the same domain as other machines in your EP installation.

- 1 Open *global.properties.xml* in any text editor.
- 2 Search for *hostname* and replace it with the name of the EAServer machine host name that is on a different port, for example, “tahiti”. *hostname* is the machine name entered at installation.
- 3 Change the value of the `default_http_port` property to the new port for the different machine hosting EAServer:

```
<Property name="default_http_port" value="new http port"
description="the application server port" menugroup="10"/>
```

- 4 Verify that the value to which you changed the `default_http_port` property is the value for which the application server is configured.

In EAServer, the portal requires that there is a listener configured for both the IP address 127.0.0.1 and the new port that you configured for the `default_http_port` property:

```
hostname.domain    new http port
127.0.0.1          new http port
```

5 Stop and restart the application server.

❖ **Turning trace functionality off for production**

If you are running the portal in a production environment, you may want to turn off the trace settings, which are turned on by default.

- 1 In the UWP properties section, set `TraceOn` and `VerboseTrace` to “false.”
- 2 In the Docservers properties section, set `TraceOn` and `VerboseTrace` to “false.”
- 3 In the Portal properties section, set `TraceOn` and `VerboseTrace` to “false.”
- 4 In the Capture properties section, set `CaptureTraceOn` to “off.”

❖ **Enabling proxy servers**

If you are using a HTTP proxy server, there are several settings in *global.properties.xml* that you must modify.

- 1 In the Global properties section, set the proxy property to “true.”
- 2 Set the remaining proxy properties (`proxy.host`, `proxy.port`, `proxy.bypass_list`) to the correct values for your proxy configuration.
- 3 Save and exit the file, close the text editor, then stop and restart EAServer.

Limiting access to *global.properties.xml*

Currently, Enterprise Portal does not encrypt the user name and password used to log in to Portal Interface. To minimize the impact of this, set up *global.properties.xml* for minimal file access.

You should also change the database user name found in *global.properties.xml* from the default to something else. You must coordinate these changes between the portal database and the *global.properties.xml* file.

❖ **Limiting access to *global.properties.xml***

- 1 Use Sybase Central or isql to create a new user in the portal database and make this user the owner of *portaldatabase* and *portalsearchdb*. Here is the input using isql:

```
// Create a user portalowner/portalowner whose default database
// is the portaldatabase.

sp_addlogin portalowner, portalowner, portaldatabase
go

// Make this user the owner of portaldatabase.

use portaldatabase
go

sp_changedbowner portalowner, true
go
// Make this user the owner of portalsearchdb

use portalsearchdb
go

sp_changedbowner portalowner, true
go
```

- 2 Update the *global.properties.xml* DataPool entry:

```
<DataPool name="op_portal_ase"
  class="com.onepage.db.DbConnectionBroker"
  driver="com.sybase.jdbc2.jdbc.SybDriver"
  url="jdbc:sybase:Tds:hostname:port?ServiceName=portaldatabase"
  user="dba"
  password="SQL"
  password_encrypted="false"
  minConnections="1"
  maxConnections="50"
  unusedtimer="360"
  checkouttimer="60"
  reuse="30"
  debug="true" />
```

- 3 Restart the application server.

Using the Search Services

Enterprise Portal uses an embedded search engine—the Dynamic Reasoning Engine (DRE)—to perform concept-based searches. You can call search services methods to perform concept-based searches or you can perform concept-based searches using the portal’s search portlets.

This chapter describes the concept-based search services available in Enterprise Portal and the development information necessary to implement searches in your portal applications.

See Appendix C, “Setting Up Automatic Indexing and Search Functionality,” for instructions on using the portal’s search portlets.

Topic	Page
Introduction	289
Search services Enterprise JavaBean	291
Search example	311
SearchHome example	313
User agent and channel examples	313
Advanced search by proper name	314
Advanced search APIs	314

Introduction

The search engine uses high-performance matching algorithms to identify patterns within text that can then be matched with similar patterns from other data sources. The identified patterns (concept agents) improve their ability to find similar, relevant patterns by “learning” which search hits the user prefers.

You can use the concept-based searching capability with any language to search data that has been indexed into the search engine.

Concept-based searching includes:

- Natural language, Boolean, proximity, and concept queries.

- Fast query speed. The speed is linked to the number of results requested rather than the number of documents indexed.
- Retrieval of original document or its plain text content.
- Automatic elimination of duplicates by content or reference on indexing.

There are several options for indexing data in the search engine, including:

- WebFetch for searching Web data
- Lotus Notes Fetch
- ODBC Fetch

Caching

The search services Enterprise JavaBean (EJB) caches agents, channels, and subchannels to minimize the number of accesses to the database. Caching is based on the user name. If multiple clients have the same user name, they share the same cache. Clients of the search services EJB must be aware that:

- Each user must make a call to `setupUserMemory(user)` before any other calls for agents or channels.
- Each user must make a call to `cleanupUserMemory(user)` when that user is finished with agents and channels.
- While iterating through agents, channels, or subchannels, another client using the same user name may be creating or removing items. This can result in items becoming unavailable and an exception being thrown.
- The first call to `getCountOfChannels(roles, user)` loads the channels and subchannels from the database into the cache for that user name. Subsequent calls to `getCountOfChannels(roles, user)` with the same user name, even by another client, uses information in the cache. Subsequent calls do *not* reload the channels from the database, even if different roles are given.

Calling methods

The protocol for using agents and channels is described in the following steps and illustrated in “Search and SearchHome Examples” later in this chapter.

❖ Calling agents

- 1 Call `setupUserMemory`.
- 2 Call `getCountOfUserAgents` before making any other calls. The cache is loaded with the user agents.
- 3 Call the `getNextUserAgentName` method.
- 4 When you finish, call `cleanupUserMemory` to flush the cache.

❖ Calling channels and subchannels

- 1 Call `setupUserMemory`.
- 2 Call `getCountOfUserChannels` before making any other calls. The cache is then loaded with all the channels and subchannels to which the user has access.
- 3 Call `getNextChannelName` to go through the list of channels.
- 4 Call `getCountOfSubchannels`, then call `getNextSubChannelName` to go through the resulting list.
- 5 When you finish, call `cleanupUserMemory` to flush the cache.

Search services Enterprise JavaBean

Enterprise Portal includes an Enterprise JavaBean (EJB) that allows developers to access and use concept-based search features. The EJB component is installed automatically, resides in `EAServer`, and is readily available from a Java client.

Hosting the EJB on `EAServer` ensures that the EJB is:

- Available to servlets and other components on the application server
- Available to different content management solutions

The Java package `com.sybase.ep.search.PortalSearchServices` performs ad hoc natural language searches against documents that have been indexed into the `PortalSearch` search engine. It contains the `SearchHome` and `Search` classes, which are used to invoke the `PortalSearchServices` EJB.

- `Search` – contains methods to perform searches on the DRE. It also contains the methods used with agents, channels, subchannels, and roles.

See “com.sybase.ep.search.PortalSearchServices.Search” later in this chapter for method descriptions.

- SearchHome – contains methods to create an instance of the search services EJB and connect to the DRE.

See “com.sybase.ep.search.PortalSearchServices.SearchHome” later in this chapter for method descriptions.

See “Search and SearchHome Examples” later in this chapter for code samples for both classes.

com.sybase.ep.search.PortalSearchServices. SearchHome

Description	Contains the methods that create an instance of the search services EJB and connects to the DRE.
Constructors	create – connects to the DRE using these parameters:
Properties	<ul style="list-style-type: none">• Values from the [Server] section of the <i>PortalSearchqueryh.cfg</i> or <i>DRE.INI</i> file, or• Specified host and port names, or• Specified file and section. <p>See “Search and SearchHome Examples” later in this chapter for a code sample using the SearchHome class.</p>

create

Description	Connects to the DRE. When used without parameters, uses default values from the [Server] section of the <i>PortalSearchqueryh.cfg</i> or <i>DRE.INI</i> file located in <i>%SYBASE%\PortalSearch\Engine</i> on Windows 2000 and <i>\$SYBASE/PortalSearch/Engine</i> on UNIX.
Syntax	<pre>public void create(); public void create(java.lang.String host, int port); public void create(java.lang.String file, java.lang.String section);</pre>

throws `java.rmi.RemoteException`,
`javax.ejb.CreateException`,
`com.sybase.ep.search.PortalSearchServices.SearchException`

Parameters

- host* – host address of the DRE.
- port* – query port of the DRE.
- file* – file specification of *PortalSearchqueryh.cfg* or *DRE.INI*.
- section* – section of the *PortalSearchqueryh.cfg* or *DRE.INI* file

com.sybase.ep.search.PortalSearchServices.Search

Description

Contains methods that search indexed data in the DRE. Additional methods are used for agent and channel support.

Constructors

- `remove` – disconnects from the DRE.
- `setTracing` – sets tracing on or off.
- `doQuery` – sends a new natural language query to the search engine.
- `doNameSearchQuery` – sends a new name search query to the DRE.
- `suggestMore` – searches for documents like the named document.
- `getNextRecord` – retrieves the next record (document) in the result set.
- `getDocContent` – retrieves the contents of the current document or the contents of a document with the named ID. The option to store content must be selected when the document is indexed. Otherwise, there is no content.
- `getDocId` – retrieves the ID of the current document.
- `getDocSummary` – retrieves the summary of the current document.
- `getDocTitle` – retrieves the title of the current document.
- `getDocUrl` – retrieves the URL of the current document.
- `getDocWeight` – retrieves the weight of the current document.
- `getTotalDocs` – retrieves the total number of documents in the DRE.
- `getDatabaseNames` – retrieves a list of databases indexed in the DRE.
- `getDocField` – retrieves the specified field in the current document. Used to access user-defined fields of character data.

- `getLongDocField` – retrieves the specified field in the current document. The field must contain long data.

The following methods are used exclusively to instantiate agents and channels:

- `addRoleForChannel` – associates a role with a channel.
- `addRoleForUserAgent` – associates a role with a user agent.
- `cleanupUserMemory` – cleans up memory for a given user.
- `copyUserAgent` – copies a user agent to another agent.
- `createAgent` – creates a user agent or a subchannel in the database.
- `createChannel` – creates a channel.
- `doQueryWithAgent` – makes a query to the DRE using a user agent or subchannel.
- `editAgent` – modifies the definition of a user agent or subchannel.
- `getCountOfChannels` – gets the number of channels for a given set of roles and loads the user cache with the channels and subchannels to which the user has access.
- `getCountOfRolesForChannel` – counts the number of roles for a given channel.
- `getCountOfRolesForUserAgent` – counts the number of roles for a given user agent.
- `getCountOfSubChannels` – gets the number of subchannels for a given channel.
- `getCountOfUserAgents` – gets the number of user agents owned by a user and loads the user cache.
- `getCurrentSubChannelAge` – returns age information for the current subchannel.
- `getCurrentSubChannelMaxDocs` – returns the maximum number of documents for the current subchannel.
- `getCurrentSubChannelRelevance` – returns the relevance for the current subchannel.
- `getCurrentSubChannelTrainingText` – returns the training text for the current subchannel.
- `getCurrentUserAgentAge` – gets information on the age of the current user agent.

- `getCurrentUserAgentMaxDocs` – gets the maximum documents of the current user agent.
- `getCurrentUserAgentRelevance` – gets the relevance or minimum weight of the current user agent.
- `getCurrentUserAgentTrainingText` – gets the training text for the current user agent.
- `getNextChannelName` – returns next channel from the list of channels loaded by `getCountOfChannels` method.
- `getNextRoleForChannel` – returns next role for a current channel.
- `getNextRoleForCurrentUserAgent` – gets the name of the next role of the current user agent.
- `getNextSubChannelName` – returns the next subchannel name of the current channel.
- `getNextUserAgentName` – gets the next user agent name in the list.
- `removeChannel` – removes a channel.
- `removeRoleFromChannel` – removes a role from a channel.
- `removeRoleFromUserAgent` – removes a role from a user agent.
- `removeSubChannel` – removes a subchannel.
- `removeUserAgent` – removes a user agent.
- `renameAgent` – renames a user agent or subchannel.
- `retrainSubChannelByResults` – retrains a subchannel based on doc IDs.
- `retrainUserAgentByResults` – retrains user agent based on doc IDs.
- `setupUserMemory` – initializes user cache for a given user.
`setupUserMemory` must be called before any agent or channel methods.

See also

“Search and SearchHome Examples” later in this chapter for a code sample using the `Search` class.

remove

Description

Disconnects from the DRE.

```
public void remove()  
    throws java.rmi.RemoteException
```

setTracing

Description	Sets tracing on or off.
Syntax	<pre>public void setTracing (boolean <i>trace</i>) throws java.rmi.RemoteException</pre>
Parameters	<i>trace</i> – true or false

doQuery

Description	Sends a new natural language query to the search engine. The more descriptive the query, the more accurate the result.
Syntax	<pre>public int doQuery(java.lang.String <i>query</i>, java.lang.String <i>database</i>, java.lang.String <i>dates</i>, int <i>maxDocs</i>, int <i>relevance</i>) throws java.rmi.RemoteException, com.sybase.ep.search.PortalSearchServices.SearchException</pre>
Parameters	<i>query</i> – natural language query <i>database</i> – list of search engine databases, separated by commas <i>dates</i> – date range in the form dd/mm/yyyy,dd/mm/yyyy <i>maxDocs</i> – maximum number of documents to return <i>relevance</i> – relevance of the document (0 – 100)

Note By default, values are calculated by absolute value in the DRE; that is, a number that is not limited to 100. The norm is to have the results returned in a percentage result of 0 – 100. However, depending on how a query is sent to the DRE, you may get the absolute value instead; specifically, a value greater than 100.

Return value	Number of results found.
--------------	--------------------------

doNameSearchQuery

Description	Sends a new name search query to the DRE.
Syntax	<pre>public int doNameSearchQuery (java.lang.String <i>query</i>, java.lang.String <i>database</i>,</pre>

```

java.lang.String dates,
int maxDocs,
int relevance)
    throws java.rmi.RemoteException,
           com.sybase.ep.search.PortalSearchServices.SearchException

```

Parameters	<p><i>query</i> – natural language query</p> <p><i>database</i> – list of DRE databases, separated by commas</p> <p><i>dates</i> – date range in the form dd/mm/yyyy,dd/mm/yyyy</p> <p><i>maxDocs</i> – maximum number of documents to return</p> <p><i>relevance</i> – relevance of the retrieved document (0 – 100)</p>
Return value	Number of results found.

suggestMore

Description	Searches for documents similar to the given document.
Syntax	<pre> public int suggestMore (java.lang.String <i>docID</i>, java.lang.String <i>database</i>, java.lang.String <i>dates</i>, int <i>maxDocs</i>, int <i>relevance</i>, boolean <i>excludeInputDoc</i>) throws java.rmi.RemoteException, com.sybase.ep.search.PortalSearchServices.SearchException </pre>
Parameters	<p><i>docID</i> – document ID</p> <p><i>database</i> – comma-separated list of search engine databases</p> <p><i>dates</i> – date range in the form dd/mm/yyyy,dd/mm/yyyy</p> <p><i>maxDocs</i> – maximum number of documents to return</p> <p><i>relevance</i> – relevance of the retrieved document (0 – 100)</p> <p><i>excludeInputDoc</i> – exclude the input document from the results</p>
Return value	Number of results found.

getNextRecord

Description	Retrieves the next record (document) in the result set. A result set is initially positioned before the first record. The first call to <code>getNextRecord</code> makes the first record the current record; the second call makes the second record the current record, and so on.
Syntax	<pre>public boolean getNextRecord() throws java.rmi.RemoteException, com.sybase.ep.search.PortalSearchServices.SearchException</pre>
Return value	Returns true if a document is available; returns false if no more documents exist.

getDocContent

Description	Retrieves the contents of the current document (when used without parameters) or the contents of a document within the current result set identified by ID. The option to store content must be selected when the document is indexed; otherwise, there is no content to return.
Syntax	<pre>public java.lang.String getDocContent(); public java.lang.String getDocContent(long docID); throws java.rmi.RemoteException, com.sybase.ep.search.PortalSearchServices.SearchException</pre>
Parameters	<i>docID</i> – document ID within the current result set
Return value	Document text.

getDocId

Description	Retrieves the current document ID.
Syntax	<pre>public long getDocId() throws java.rmi.RemoteException, com.sybase.ep.search.PortalSearchServices.SearchException</pre>
Return value	Document ID.

getDocSummary

Description	Retrieves the current document summary.
-------------	---

Syntax `public java.lang.String getDocSummary()`
 throws `java.rmi.RemoteException`,
 `com.sybase.ep.search.PortalSearchServices.SearchException`

Return value Document summary.

getDocTitle

Description Retrieves the current document title.

Syntax `public java.lang.String getDocTitle()`
 throws `java.rmi.RemoteException`,
 `com.sybase.ep.search.PortalSearchServices.SearchException`

Return value Document title.

getDocUrl

Description Retrieves the current document URL.

Syntax `public java.lang.String getDocUrl()`
 throws `java.rmi.RemoteException`,
 `com.sybase.ep.search.PortalSearchServices.SearchException`

Return value Document URL.

getDocWeight

Description Retrieves the current document weight (relevance).

Syntax `public long getDocWeight()`
 throws `java.rmi.RemoteException`,
 `com.sybase.ep.search.PortalSearchServices.SearchException`

Return value Document weight (0 – 100).

getTotalDocs

Description Retrieves the total number of documents in the search engine.

Syntax `public long getTotalDocs()`
 throws `java.rmi.RemoteException`

Return value Total number of documents indexed in the DRE.

getDatabaseNames

Description	Retrieves the list of databases in the search engine.
Syntax	<pre>public java.lang.String get DatabaseNames() throws java.rmi.RemoteException</pre>
Return value	A list of databases, separated by commas.

getDocField

Description	Retrieves the specified field of the current document. The field must contain character data. This method accesses user-defined fields.
Syntax	<pre>public java.lang.String getDocField (java.lang.String <i>fieldName</i>) throws java.rmi.RemoteException, com.sybase.ep.search.PortalSearchServices.SearchException</pre>
Parameters	<i>fieldName</i> – name of the character field
Return value	Document field.

getLongDocField

Description	Retrieves the specified field of the current document. The field must contain long data. This method accesses user-defined fields.
Syntax	<pre>public java.lang.String getLongDocField (java.lang.String <i>fieldName</i>) throws java.rmi.RemoteException, com.sybase.ep.search.PortalSearchServices.SearchException</pre>
Parameters	<i>fieldName</i> – name of the character field
Return value	Document field.

addRoleForChannel

Description	Associates a role with a channel.
Syntax	<pre>public int addRoleForChannel(String <i>user</i>, String <i>channel</i>, String <i>role</i>) throws RemoteException, SearchException</pre>
Parameters	<i>user</i> – user name <i>channel</i> – channel name

role – associated role name

Return value Returns a status of: 0 if failure; otherwise, successful.

addRoleForUserAgent

Description Associates a role with a user agent.

Syntax `public int addRoleForUserAgent(String user,
String agent,
String role) throws RemoteException, SearchException`

Parameters *user* – user name

agent – agent name

role – associated role name

Return value Returns a status of: 0 if failure; otherwise, successful.

cleanupUserMemory

Description Cleans up memory for a given user; it frees the user's cache and should be called whenever a user is finished. If it is not called, the cache is not freed.

Syntax `public void cleanupUserMemory(String user) throws RemoteException,
SearchException`

Parameters *user* – user name

Return value Returns a status of: 0 if failure; otherwise, successful.

copyUserAgent

Description Copies a user agent to another user agent. You must call `setupUserMemory` before calling this method. You can copy agents only to your current user name.

Syntax `public int copyUserAgent(String sourceUser,
String sourceAgent,
String destUser,
String destAgent) throws RemoteException, SearchException`

Parameters *sourceUser* – name of a source user

sourceAgent – name of a user agent to be copied

destUser – name of a destination user; must match your current user name

destAgent – name of a destination user agent

Return value Returns a status of: 0 if failure; otherwise, successful.

createAgent

Description Creates a user agent or subchannel.

Syntax

```
public int createAgent(String user,
                      String agentOrChannel,
                      String subchannel,
                      String trainingText,
                      int ageOfInformation,
                      int relevance,
                      int maxDocs,
                      int agentType) throws RemoteException, SearchException
```

Parameters

user – user name

agentOrChannel – name of a user agent or channel

subchannel – name of a subchannel, or null if it is an agent

trainingText – text used in training of this agent

ageOfInformation – age of documents to be searched

relevance – minimum weight of documents to be searched

maxDocs – maximum number of documents

agentType – type of agent

Return value Returns a status of: 0 if failure; otherwise, successful.

createChannel

Description Creates a channel.

Syntax

```
public int createChannel(String user,
                        String channel) throws RemoteException, SearchException
```

Parameters

user – user name

channel – name of channel

Return value Returns a status of: 0 if failure; otherwise, successful.

doQueryWithAgent

Description	Makes a query to the DRE with a user agent or subchannel.
Syntax	<pre>public int doQueryWithAgent(String user, String agentOrChannel, String subChannelName, int agentType) throws RemoteException, SearchException</pre>
Parameters	<p><i>user</i> – user name</p> <p><i>agentOrChannel</i> – name of a user agent or channel</p> <p><i>subChannelName</i> – name of a subchannel, or null if it is an agent</p> <p><i>agentType</i> – type of agent</p>
Return value	Number of documents obtained by doing query.

editAgent

Description	Modifies a user agent or a subchannel.
Syntax	<pre>public int editAgent(String user, String agentOrChannel, String subchannel, String trainingText, int ageOfInformation, int relevance, int maxDocs, int agentType) throws RemoteException, SearchException</pre>
Parameters	<p><i>user</i> – user name</p> <p><i>agentOrChannel</i> – name of an agent or channel</p> <p><i>subchannel</i> – name of a subchannel or null if it is an agent</p> <p><i>trainingText</i> – text used in training of this agent</p> <p><i>ageOfInformation</i> – age of documents to be searched</p> <p><i>relevance</i> – minimum weight of documents to be searched</p> <p><i>maxDocs</i> – maximum number of documents</p> <p><i>agentType</i> – type of agent</p>
Return value	Returns a status of: 0 if failure; otherwise, successful.

getCountOfChannels

Description	Gets number of channels for a given set of roles. This method also resets a pointer to the beginning of the list of channels so getNextChannelName can be used to iterate through the list of channels. getCountOfChannels also loads the cache with channels and subchannels for that user.
Syntax	<pre>public int getCountOfChannels(String rolesOfUser, String user) throws RemoteException, SearchException</pre>
Parameters	<i>rolesOfUser</i> – set of roles user has <i>user</i> – name of a user
Return value	Returns the number of channels if successful; 0 if failure.

getCountOfRolesForChannel

Description	Counts the number of roles for a channel. Resets to the beginning of the list of roles for that channel so getNextRoleForChannel can be used to iterate through the list of roles.
Syntax	<pre>public int getCountOfRolesForChannel(String user, String channel) throws RemoteException, SearchException</pre>
Parameters	<i>user</i> – name of a user <i>channel</i> – name of channel
Return value	Returns the number of channels if successful; 0 if failure.

getCountOfRolesForUserAgent

Description	Counts number of roles of an agent. Resets to the beginning of the list of roles for agents so getNextRoleForCurrentUserAgent can be used to iterate through the list of roles for the current user agent.
Syntax	<pre>public int getCountOfRolesForUserAgent(String user, String agent) throws RemoteException, SearchException</pre>
Parameters	<i>user</i> – name of a user <i>agent</i> – name of agent
Return value	Returns the number of roles; 0 if failure, otherwise, successful.

getCountOfSubChannels

Description	Gets number of subchannels for a given channel. Resets to the beginning of the list of subchannels so getNextChannelName can be used to iterate through the list of subchannels.
Syntax	<code>public int getCountOfSubChannels(String user, String channel)</code> throws RemoteException, SearchException
Parameters	<i>user</i> – name of a user <i>channel</i> – name of channel
Return value	Returns the number of channels if successful; 0 if failure.

getCountOfUserAgents

Description	Gets number of agents owned by a user. Resets to the beginning of the list of user agents so getNextUserAgentName can be used to iterate through the list of agents.
Syntax	<code>public int getCountOfUserAgents(String user)</code> throws RemoteException, SearchException
Parameters	<i>user</i> – name of a user
Return value	Returns the number of agents if successful; 0 if failure.

getCurrentSubChannelAge

Description	Retrieves the age of information for a current subchannel. The current subchannel is the one returned by the last call to getNextSubChannelName.
Syntax	<code>public int getCurrentSubChannelAge()</code> throws RemoteException, SearchException
Return value	Returns the age of information if successful; 0 if failure.

getCurrentSubChannelMaxDocs

Description	Returns maximum number of documents for the current subchannel. The current subchannel is the one returned by the last call to getNextSubChannelName.
-------------	---

Syntax `public int getCurrentSubChannelMaxDocs() throws RemoteException, SearchException`

Return value The maximum number of documents if successful; 0 if failure.

getCurrentSubChannelRelevance

Description Returns relevance for the current subchannel. The current subchannel is the one returned by the last call to `getNextSubChannelName`.

Syntax `public int getCurrentSubChannelRelevance() throws RemoteException, SearchException`

Return value Returns the relevance if successful; 0 if failure.

getCurrentSubChannelTrainingText

Description Returns training text for the current subchannel. The current subchannel is the one returned by the last call to `getNextSubChannelName`.

Syntax `public String getCurrentSubChannelTrainingText() throws RemoteException, SearchException`

Return value Returns the training text if successful; 0 if failure.

getCurrentUserAgentAge

Description Gets age of information of the current user agent. The current agent is the one returned by the last call to `getCurrentUserAgentName`.

Syntax `public int getCurrentUserAgentAge() throws RemoteException, SearchException`

Return value Returns the age of information if successful; 0 if failure.

getCurrentUserAgentMaxDocs

Description Gets maximum documents of the current user agent. The current agent is the one returned by the last call to `getCurrentUserAgentName`.

Syntax `public int getCurrentUserAgentMaxDocs() throws RemoteException, SearchException`

Return value Returns the maximum documents if successful; 0 if failure.

getCurrentUserAgentRelevance

Description Gets relevance or minimum weight of the current user agent. The current agent is the one returned by the last call to `getCurrentUserAgentName`.

Syntax `public int getCurrentUserAgentRelevance() throws RemoteException, SearchException`

Return value Returns the relevance if successful; 0 if failure.

getCurrentUserAgentTrainingText

Description Gets training text of the current user agent. The current agent is the one returned by the last call to `getCurrentUserAgentName`.

Syntax `public String getCurrentUserAgentTrainingText() throws RemoteException, SearchException`

Return value Returns the training text if successful; 0 if failure.

getNextChannelName

Description Returns next channel from the list of channels loaded by `getCountOfChannels` method.

Syntax `public String getNextChannelName() throws RemoteException, SearchException`

Return value Returns the channel name if successful; throws an exception if failure.

getNextRoleForChannel

Description Returns next role for a current channel from the list of roles loaded by `getCountofRolesForChannels`.

Syntax `public String getNextRoleForChannel() throws RemoteException, SearchException`

Return value Returns the role name if successful; throws an exception if failure.

getNextRoleForCurrentUserAgent

Description	Returns next role of the current user agent from the list of roles loaded by <code>getCountOfRolesForUserAgent</code> .
Syntax	<code>public String getNextRoleForCurrentUserAgent()</code> throws <code>RemoteException</code> , <code>SearchException</code>
Return value	Returns the valid role name if successful; null if failure.

getNextSubChannelName

Description	Returns next subchannel name for a current channel. Resets to the beginning of the subchannels list so that <code>getNextChannelName</code> can be used to iterate through the list of subchannels from a list of subchannels loaded by the one returned by the last call to <code>getNextSubChannelName</code> .
Syntax	<code>public String getNextSubChannelName()</code> throws <code>RemoteException</code> , <code>SearchException</code>
Return value	Returns the subchannel name if successful; throws exception if failure.

getNextUserAgentName

Description	Gets name of the next user agent in the list loaded by <code>getCountOfUserAgents</code> .
Syntax	<code>public String getNextUserAgentName()</code> throws <code>RemoteException</code> , <code>SearchException</code>
Return value	Returns the agent name if successful; null if failure.

removeChannel

Description	Removes a channel and all of its subchannels and associated roles.
Syntax	<code>public int removeChannel(String user, String channel)</code> throws <code>RemoteException</code> , <code>SearchException</code>
Parameters	<i>user</i> – name of a user <i>channel</i> – name of a channel
Return value	Returns the status: 0 if failure, otherwise, successful.

removeRoleFromChannel

Description	Removes a role from a channel.
Syntax	<code>public int removeRoleFromChannel(String <i>user</i>, String <i>channel</i>, String <i>role</i>)</code> throws RemoteException, SearchException
Parameters	<i>user</i> – name of a user <i>channel</i> – name of a channel <i>role</i> – name of a role
Return value	Returns the status: 0 if failure, otherwise, successful.

removeRoleFromUserAgent

Description	Removes a role from an agent.
Syntax	<code>public int removeRoleFromUserAgent(String <i>user</i>, String <i>agent</i>, String <i>role</i>)</code> throws RemoteException, SearchException
Parameters	<i>user</i> – name of a user <i>agent</i> – name of an agent <i>role</i> – name of a role
Return value	Returns the status: 0 if failure, otherwise, successful.

removeSubChannel

Description	Removes a subchannel.
Syntax	<code>public int removeSubChannel(String <i>user</i>, String <i>channel</i>, String <i>subchannel</i>)</code> throws RemoteException, SearchException
Parameters	<i>user</i> – name of a user <i>channel</i> – name of an channel <i>subchannel</i> – name of a subchannel
Return value	Returns the training text if successful; 0 if failure.

removeUserAgent

Description	Removes a user agent and its associated roles.
Syntax	<pre>public int removeUserAgent(String user, String agent) throws RemoteException, SearchException</pre>
Parameters	<p><i>user</i> – name of a user</p> <p><i>agent</i> – name of an agent to be removed</p>
Return value	Returns the status: 0 if failure, otherwise, successful.

renameAgent

Description	Renames the supplied agent or subchannel.
Syntax	<pre>public int renameAgent(String user, String agent, String subchannel, String newAgent, int agentType) throws RemoteException, SearchException</pre>
Parameters	<p><i>user</i> – name of a user</p> <p><i>agent</i> – name of an agent to be removed</p> <p><i>subchannel</i> – name of a subchannel</p> <p><i>newAgent</i> – name of a new agent</p> <p><i>agentType</i> – name of an agent, in other words, agent or channel</p>
Return value	Returns the status: 0 if failure, otherwise, successful.

retrainSubChannelByResults

Description	Retrains a subchannel based on doc IDs.
Syntax	<pre>public int retrainSubChannelByResults(String user, String channel, String subchannel, String docIds) throws RemoteException, SearchException</pre>
Parameters	<p><i>user</i> – name of a user</p> <p><i>channel</i> – name of channel containing subchannel</p> <p><i>subchannel</i> – name of subchannel to be retrained</p>

docIds – list of doc IDs

Return value Returns the status: 0 if failure, otherwise, successful.

retrainUserAgentByResults

Description Retrains user agent based on doc IDs.

Syntax `public int retrainUserAgentByResults(String user,
String agent,
String docIds)` throws `RemoteException`, `SearchException`

Parameters *user* – name of a user
agent – name of an agent to be retrained
docIds – list of doc IDs

Return value Returns the status: 0 if failure, otherwise, successful.

setupUserMemory

Description Initializes user cache for a given user. `setupUserMemory` must be called before any agent or channel methods.

Syntax `public synchronized int setupUserMemory(String user)` throws `RemoteException`, `SearchException`

Parameters *user* – user name

Return value Returns the status: 0 if failure, otherwise, successful.

Search and SearchHome examples

This section provides code examples for the `Search` and `SearchHome` classes.

Search example

The following is sample code for the `Search` class:

```
PortalSession session = null;
Search search;
```

```
try
{
/*
* Issue a natural language query
*/
    int numResults = search.doQuery(session,
        "the text of the query goes here",
        "", // search all databases
        "", // no date range specified
        50, // return docs with a weight
            of 50 or more
        10); // limit results to 10 documents
    System.out.println("Number of results
        found = " +
numResults);
/*
* Loop through the result set and print out all
* the available information for each document.
*/
    int j = 0;
    while (search.getNextRecord())
    {
        j++;
        System.out.println("\nRecord " + j + ":");
        System.out.println("Doc id = " +
search.getDocId());
        System.out.println("Weight = " +
search.getDocWeight());
        System.out.println("Url = " +
search.getDocUrl());
        System.out.println("Title = " +
search.getDocTitle());
        System.out.println("Summary: " +
search.getDocSummary());
        System.out.println("Content: "
search.getDocContent());
    }
/*
* Disconnect for the search engine
*/
    search.remove();
}
catch (Exception e)
{
    System.out.println("Error: " + e.toString());
}
```

SearchHome example

The following is sample code for the SearchHome class:

```

/* Example of how to get the home interface of the
 * PortalSearchServices EJB */
Context          ctx;
SearchHome       home;
Search           search;

try
{
    /*
     * Get the home interface
     */
    home = (SearchHome)

ctx.lookup("com.sybase.ep.search.
PortalSearchServices/SearchHome");

    /*
     * Connect to the search engine specified in the
     * [Server] section of
     *
%SYBASE%/PortalSearch/Engine/<PortalSearchqueryh.cfg
or DRE.INI>
     */
    search = home.create();
}
catch (Exception e)
{
    System.out.println("exception " + e.toString());
}

```

User agent and channel examples

You can find examples of code that implement agents and channels in the *PortalSearch/Samples* directory of your Enterprise Portal installation.

Advanced search by proper name

To use the advanced search by proper name feature, documents containing proper names must be indexed in the DRE database. A proper name is defined as any two or more consecutive words that start with capital letters, such as “John E. Smith.”

❖ Enabling searching by proper name

- 1 Stop the Autonomy services/processes. See “Starting and stopping the automatic indexing utility” on page 335.
- 2 Go to the *PortalSearchqueryh.cfg* or *DRE.INI* file located in:
\$SYBASE/PortalSearch/Engine/PortalSearchqueryh.cfg or *DRE.INI* (UNIX)
%SYBASE%\PortalSearch\Engine\PortalSearchqueryh.cfg or *DRE.INI* (Windows)
- 3 With an ASCII text editor, add `PROPERNAMES=1` in the [Server] section of the *PortalSearchqueryh.cfg* or *DRE.INI* file.
- 4 Start the Autonomy services/processes. See “Restarting the indexing utility” on page 336.
- 5 Index documents that have names in the DRE. Any two consecutive words that begin with uppercase letters are considered names. See “Automatic indexing” on page 334.

Note Do not use this functionality to index documents that do not contain proper names.

Advanced search APIs

Enterprise Portal allows you to perform an advanced search using more exact queries. With a word filter, query results can be restricted to contain specific terms, ignore specific terms, or to have an exact match. You can also perform searches by date range, with different options in addition to the number of days. You can specify data sources for the query to restrict users from accessing certain data sources.

addDatabaseForUserAgent

Description	Associates a database with a given agent.
Syntax	<code>int addDatabaseForUserAgent (String user, String agent, String dbName throws RemoteException, SearchException</code>
Parameters	<i>user</i> – name of a user <i>agent</i> – name of an agent <i>dbName</i> – name of the database
Return value	Returns the value: ≥ 0 if successful; otherwise, failure.

removeDatabaseFromUserAgent

Description	Removes a database associated with a given agent.
Syntax	<code>int removeDatabaseFromUserAgent (String user, String agent, String dbName throws RemoteException, SearchException</code>
Parameters	<i>user</i> – name of a user <i>agent</i> – name of an agent <i>dbName</i> – name of the database
Return value	Returns the value: ≥ 0 if successful; otherwise, failure.

addDatabaseForChannel

Description	Associates a database with a given channel.
Syntax	<code>int addDatabaseForChannel (String user, String channel, String dbName throws RemoteException, SearchException</code>
Parameters	<i>user</i> – name of a user <i>channel</i> – name of a channel

dbName – name of the database

Return value Returns the value: ≥ 0 if successful; otherwise, failure.

removeDatabaseFromChannel

Description Removes a database associated with a given channel.

Syntax `int removeDatabaseFromChannel ('String user,
String channel,
String dbName')`
throws `RemoteException,`
`SearchException`

Parameters *user* – name of a user
channel – name of a channel
dbName – name of the database

Return value Returns the value: ≥ 0 if successful; otherwise, failure.

getCountOfDatabasesForUserAgent

Description Returns the number of databases associated with a given agent.

Syntax `int getCountOfDatabasesForUserAgent (String user,
String agent),`
throws `RemoteException,`
`SearchException`

Parameters *user* – name of a user
agent – name of an agent

Return value Returns the value: ≥ 0 if successful; otherwise, failure.

getNextDatabaseForUserAgent

Description Returns the name of the next database in the list.

Syntax `String getNextDatabaseForUserAgent ()`
throws `RemoteException,`
`SearchException`
Returns name of the next database in the list.

Return value Name of the database.

getCountOfDatabasesForChannel

Description	Returns the number of databases associated with a given channel.
Syntax	<code>int getCountOfDatabasesForChannel ('String user, String channel'),</code> throws <code>RemoteException,</code> <code>SearchException</code>
Parameters	<i>user</i> – name of a user <i>channel</i> – name of a channel
Return value	Returns the value: ≥ 0 if successful; otherwise, failure.

getNextDatabaseForChannel

Description	Returns the name of the next database in the list.
Syntax	<code>String getNextDatabasesForChannel ()</code> throws <code>RemoteException,</code> <code>SearchException</code>
Return value	Name of database.

doAdvancedQuery

Description	Sends an advanced query to the search engine.									
Syntax	<code>int doAdvancedQuery ('String query, String queryexpr, String databases, int MaxDocs int relevance int ageType String ageOfInformation String queryField')</code> throws <code>RemoteException,</code> <code>SearchException</code>									
Parameters	<i>query</i> – an equation representing query text, along with query type and query operator. If query is entered, <i>queryexpr</i> is ignored. Combination of query type and query operators include: <table> <tr> <td><code>MUST_WORD</code></td> <td><code>SHOULD_WORD</code></td> <td><code>MUSTNOT_WORD</code></td> </tr> <tr> <td><code>MUST_PERSON</code></td> <td><code>SHOULD_PERSON</code></td> <td><code>MUSTNOT_PERSON</code></td> </tr> <tr> <td><code>MUST_PHRASE</code></td> <td><code>SHOULD_PHRASE</code></td> <td><code>MUSTNOT_PHRASE</code></td> </tr> </table>	<code>MUST_WORD</code>	<code>SHOULD_WORD</code>	<code>MUSTNOT_WORD</code>	<code>MUST_PERSON</code>	<code>SHOULD_PERSON</code>	<code>MUSTNOT_PERSON</code>	<code>MUST_PHRASE</code>	<code>SHOULD_PHRASE</code>	<code>MUSTNOT_PHRASE</code>
<code>MUST_WORD</code>	<code>SHOULD_WORD</code>	<code>MUSTNOT_WORD</code>								
<code>MUST_PERSON</code>	<code>SHOULD_PERSON</code>	<code>MUSTNOT_PERSON</code>								
<code>MUST_PHRASE</code>	<code>SHOULD_PHRASE</code>	<code>MUSTNOT_PHRASE</code>								

For example, querywithopr looks like:

```
MUST_WORD=Sybase EP&&MUSTNOT_WORD=Enterprise Portal && .&&
```

where the query results must include the words “Sybase EP” and exclude the words “Enterprise Portal.”

queryexpr – logical query expression used to query the DRE only if query is NULL.

databases – a comma-delimited list of search engine databases. An empty string means all databases from all DREs.

maxDocs – maximum number of documents to return.

relevance – the relevance of the document from 0 – 100, with 100 being the highest.

ageType – 1=days, 2=weeks, 3=months, 4=years, or 5=date range.

ageOfInformation – timeframe for which the query should be done, expressed in number of days, weeks, months, years, or in the form dd/mm/yyyy, dd/mm/yyyy.

queryField – an expression presenting a combination of query fields.

For example:

```
Title=ASE 12.5 design document&&Summary=ASE 12.5 design document
```

returns the number of result documents that contain the unique title or summary (headings after the document title) “ASE 12.5 design document.”

Return value

Number of result documents.

Returns the value: ≥ 0 if successful; otherwise, failure.

getCountOfDatabases

Description

Returns the total number of databases DREs have.

Syntax

```
int getCountOfDatabases ( )  
throws RemoteException,  
SearchException
```

Return value

Total number of databases DREs have.

getNextDatabaseForDre

Description	Returns the name of the next database from the DRE.
Syntax	String getNextDatabaseForDre () throws RemoteException, SearchException
Return value	Returns name of the database.

existsDatabase

Description	Checks the existence of the database in any of the DREs.
Syntax	boolean existsDatabase (String dbName) throws RemoteException, SearchException
Parameters	<i>dbName</i> – name of the database.
Return value	True if database exists; otherwise, false.

Setting Up Automatic Indexing and Search Functionality

Enterprise Portal includes technology for several types of searches. This chapter describes the search services that are available in Enterprise Portal and how to configure the automatic indexing utility to enable concept-based searches.

Once you configure the search functionality and automatic indexing, perform searches using Portal Interface—see the *Portal Interface User's Guide*.

Topic	Page
Overview	321
Performing channel, subchannel, and agent searches	326
Performing concept-based searches	328
Search options	333
DRE administration tool	334
Managing the connection cache	334
Automatic indexing	334

Overview

Enterprise Portal uses an embedded search engine called the Dynamic Reasoning Engine (DRE) to perform concept-based searches. The DRE uses high-performance matching algorithms to identify patterns within text that can then be matched with similar patterns from other data sources. The identified patterns, also known as concept agents, can facilitate finding similar, relevant patterns by “learning” which search hits the user prefers.

Concept-based searches

Concept-based searching includes these features:

- Natural language, Boolean, proximity, and concept queries.

- Fast query speed. The speed is linked to the number of results requested rather than the number of documents indexed.
- Retrieval of original document or its plain text content.
- Automatic elimination of duplicates by content or reference on indexing.

You can use concept-based searching with any language to search data that has been indexed into the search engine. Several options are available to index data, which include:

- WebFetch for searching Web data
- Lotus Notes Fetch for searching Lotus Notes databases
- ODBC Fetch for searching ODBC databases

Dynamic Reasoning Engine

The DRE is a scalable, multithreaded database and process based on neural-network pattern-matching technology. It performs these essential tasks:

- Concept matching – accepts text as input and returns references to documents in another text source (or prebuilt index) with the most relevance.
- Standard text search – accepts a Boolean term or natural language query and returns a list of documents containing the terms ordered by relevance to the query.
- Stored queries – accepts a set of text (training phrase, document, or set of documents) and returns documents that match the concept. These queries can be “retrained” to make stored queries more efficient. Agents and channels are types of stored queries.

Automatic indexing

The automatic indexing utility is part of Enterprise Portal’s concept-based search service. Use the indexing utility to schedule the indexing of documents in specified data sources. You must index the data sources before you can search them.

Stored queries

A stored query is a search description that is stored in an Enterprise Portal database using agents and channels. An agent can be trained using natural language or by supplying documentation that describes the topic of interest. An agent can be retrained based on results, enabling future searches to be more specific and faster.

A user agent is a stored query that is created and assigned a search mission by an individual user. The user agent finds information and creates a personalized result set specifically for each user. For information about user agents, see the *Portal Interface User’s Guide*.

A channel is a stored query created by the system administrator. A channel is a logical category that contains subchannels. You can assign roles to a channel, which are inherited by its subchannels. For information about creating channels and subchannels, see “Performing channel, subchannel, and agent searches” on page 323.

Security

Agents and channels are available to users with the appropriate authority, which is granted using roles. Unauthorized users are denied access to agents or channels. System administrators must create roles for an agent or channel during the same session that they create the agent or channel. If the system administrator grants permissions to a user, those permissions are not available until the user logs out and logs in again.

Performing channel, subchannel, and agent searches

A subchannel is a search agent created by an administrator that can be used by anyone across the organization. Channels are a way to group similar types of subchannels into categories. For example, you might have a channel called “Company” with subchannels for “Finance,” “HR,” and “Administration.”

Administrators create channels and subchannels, and users can retrain a subchannel to narrow the search to their particular area of interest. For example, a human resources representative might want to customize the HR subchannel to search for employees who were hired within the last year.

First, you create a channel, then you create associated subchannels that have similar search criteria.

Note Only users with the PortalAdmin role can create channels and subchannels.

Creating channels and subchannels

Channels and subchannels can be created only by an administrator working in Admin mode. You create channels and subchannels using a Manage Channels portlet. You can either add a Manage Channels portlet to a page of your choice, or use the portlet on the predefined Channel Search page.

❖ **Adding a Manage Channels portlet to a page**

- 1 Log in to Portal Interface.
 - a Open a Web browser, and enter:

`http://HOSTNAME.DOMAIN:PORT/onepage/index.jsp`

Where *machine_name* is the name of the machine where the onepage Web application is installed. This displays the Portal Login page.

- b Enter your member name and password, then click Sign In.
- 2 Open the page group and page where you want to create the portlet, and click Add Portlet.

Note You can only add portlets to your personal page groups and pages. See Chapter 11, “Building Page Groups,” Chapter 10, “Building Pages,” and Chapter 4, “Building Portlets,” for instructions.

- 3 When the Add Portlet window displays, select Search, select Manage Channels, then click Done.

❖ **Creating channels**

- 1 On the page where you added the Manage Channels portlet, click the New button to the right of the Channels list box.
- 2 In the Edit Channel window, enter a name for the channel.
- 3 Select one or more roles (hold down the Ctrl key), then click the right-facing double arrow button to move those roles to the Assigned list.
- 4 Click OK.

❖ **Creating subchannels**

- 1 In the Manage Channels portlet, select a channel from the Available Channels list.

Note When you select a channel, wait for the portlet to refresh before continuing.

- 2 Click New to the right of the Subchannels list.
- 3 In the Edit Subchannel window, complete these fields:

- Name – enter a name that describes the type of information this search retrieves. Be specific so you can distinguish between subchannels if you use more than one (for example, “Programs, C++” for one, and “Programs, Java/JavaScript” for another).
- Training – enter a word or phrase that defines the results you want. Using specific criteria to more narrowly define your search is called “training.”

To train subchannels, use these guidelines:

- Type in a few descriptive words to return a list of relevant results.
 - To restrict a search further, include more terms.
 - To get fewer results, use a minus sign before a word to exclude pages containing that word. Include a space before the minus sign, but not after. For example, type “animals -cats” to find pages that contain the word “animals” but not the word “cats.”
 - To get more results, use wildcards. Type an asterisk at the end of a word or partial word. For example, typing “part*” returns pages that include “partner,” “particle,” “participate,” and so on.
 - Portal Interface searches are not case sensitive. All letters, regardless of how you type them, are understood as lowercase. For example, searches for “george washington”, “George Washington”, and “gEoRgE wAsHiNgToN” all return the same results.
4. Minimum Relevancy – an integer between 0 and 100 that represents the relevancy of the results as a percentage. For example, if you enter “Adaptive Server”, results that contain information about Adaptive Server are more relevant than results that contain information about other servers. If you enter 100, you only get results that contain “Adaptive Server”.
 - Number of Returned Results – choose the number of results you want to display each time you activate this subchannel’s predefined search.
 - Age of Information – select a time period in which to search for Web results, between 1 and 14 days. Leave this option blank to see results from any time period.
4. Click OK. You return to the Manage Channels portlet, and the new subchannel displays in the Subchannels list.

Editing channels and subchannels

❖ Renaming a channel

- 1 In the Manage Channels portlet, select a channel from the Channels list, and click Edit.
- 2 Enter the new channel name and click OK.
The Channels list is updated with the new name.

❖ Deleting a channel

- 1 In the Manage Channels portlet, select a channel from the Channels list, and click Delete.
- 2 In the confirmation dialog box, click OK. The channel is removed from the list.

❖ Renaming a subchannel

- 1 In the Manage Channels portlet, select a subchannel's corresponding channel.
- 2 After the screen refreshes, select from the Subchannels list, and click Edit.
- 3 Enter the new subchannel name, and click OK.
The Subchannels list is updated with the new name.

❖ Deleting a subchannel

- 1 In the Manage Channels portlet, select a subchannel from the Subchannels list, and click Delete.
- 2 In the confirmation dialog box, click OK. The channel is removed from the list.

Performing concept-based searches

❖ Performing concept-based searches

- 1 Verify that the DRE is running.
On UNIX, the `PortalSearchqueryh` process displays when you run:

```
ps -ef | grep Portal
```


On Windows, `PortalSearchqueryh` displays in the Task Manager.

The DRE starts automatically during the initial installation and configuration of Enterprise Portal. To restart the DRE, see “Starting the DRE” on page 327.

- 2 Index the files you want to search.

Use the automatic indexing utility to continually index files in the background—see “Automatic indexing” on page 334.

Dynamic Reasoning Engine

By default, logging is disabled for the DRE. To enable logging:

- 1 Using a standard text editor, open `$SYBASE/PortalSearch/Engine/PortalSearchqueryh.cfg` or `DRE.INI` on UNIX and `%SYBASE%\PortalSearch\Engine\PortalSearchqueryh.cfg` or `DRE.INI` on Windows.
- 2 In the [Server] section, set `LOGGING=1`.
- 3 Stop and restart the DRE for the changes to take effect.

To stop the DRE on UNIX, find the `PortalSearchqueryh` process ID, and kill the process:

```
ps -ef | grep PortalSearchqueryh
kill -9 processID
```

To stop the DRE on Windows, right-click the `PortalSearchqueryh` process in the Task Manager, and select End Process.

To start the DRE, see “Starting the DRE” on page 327.

All messages are logged in the `queryh.log` file.

Starting the DRE

The DRE starts automatically during the initial installation and configuration of Enterprise Portal. If you need to restart the DRE, follow these instructions.

❖ Starting the DRE on UNIX

- Change to the `$SYBASE/PortalSearch/bin` directory, and run:

```
StartQuery.sh $SYBASE/PortalSearch
```

❖ **Starting the DRE on Windows 2000**

- 1 Select Start | Settings | Control Panel.
- 2 Double-click the Services icon in Control Panel.
- 3 When the Services dialog appears, scroll down the list, right-click Sybase Search – Auto Indexer to select that service, then select Start.
- 4 When the Status changes to “Started,” click the close box to exit the window.

After the DRE is running, you can use the DRE administration tool (on Windows) or the automatic indexing utility to index your database files and prepare them for searches. For information on the automatic indexing utility, see “Automatic indexing” on page 334.

Search options

Passing security information to the DRE

When a user submits a search query, that user’s security information is embedded in the query string and passed to the DRE. The DRE uses a security plug-in to validate that the user has permission to see the requested result set. To use the DRE this way, pass the following Windows arguments in the query:

```
user=<User Name>  
pass=<Password>  
domain=<Domain Name>  
group=<Group Name>
```

For example:

```
/qmethod=q&querytext=accountWindowsNTs&user=  
John&pass=111&domain=Global&grou=Admin, Developers
```

Note This example requires that the appropriate user security information has been retrieved from the session before the query is run.

The mandatory variable/value pairs vary according to the type of security implemented. Table C-1 shows the mandatory values for each security type.

Table C-1: Values for security types

Security type	Unmapped	Mapped
Notes	user: Notes user name group: comma-separated list of groups to which that user belongs domain: Notes domain name	user: Notes user name group: comma-separated list of groups to which the user belongs
ODBC	Dependent on the stored function or procedure	N/A
Windows	user: Windows user name pass: Windows password domain: Windows domain name	user: Windows user name domain: Windows domain name group: comma-separated list of groups to which that user belongs
Exchange	user: Exchange user profile name domain: Exchange server domain name	N/A
UNIX	N/A	N/A

DRE and security DLLs

Autonomy currently supplies the following security DLL plug-ins:

	Unmapped	Mapped
Lotus Notes	Y	Y
ODBC/Oracle	Y	N
Windows	Y	Y
Exchange	Y	Y
UNIX	N	Y
NDS	N	Y

The DLLs act as plug-ins for the DRE. The DRE uses them when instructed to do so in the configuration file. The DRE goes through the following process:

- 1 On start-up, the DRE loads the DLLs.
- 2 When the DRE receives a query, it looks for the best matching results.

- 3 For each result, it calls the security DLL to find out if the requesting user has permission to see that document. If yes, then the document is shown; if no, then the document is not shown. The DRE makes sure that if *n* results are requested, the user can access the results for that many documents.

Note By default, results are calculated by absolute value in the DRE; that is, a number not limited to 100. Generally, results are returned as a percentage, ranging from 0 to 100. However, depending on how a query is sent to the DRE, you may get the absolute value instead; specifically, a value over 100.

The input parameters for the security DLL are:

- User name, password, domain, group (passed from the front-end application)
- Document reference (configured to be any structured field in the DRE)

DRE generic security configuration

Configure the *DRE.CFG* file the same way for all types of security, with specific values for certain settings. Add a security entry for the DRE database in *DRE.CFG*. For example:

```
[Databases]
NUMDBS=1
0=ARCHIVE
[ARCHIVE]
Security=MySecuritySection
```

Add a `Security=security_section` entry under the database name, which is “ARCHIVE” in this example. The value of this entry is the name of the security section, in this example `[MySecuritySection]`.

```
[MySecuritySection]
Library=
Logging=
Type=
DebugDecrypt=
ReferenceField=
CacheMaxSize=
CacheExpiryMinutes=
HideField=
```

Table C-2 shows the possible values for use in `[MySecuritySection]`.

Table C-2: Values for [MySecuritySection]

Entry	Description	Default
Library	The name of the DLL file to be used. This should be the name of the DLL supplied by Autonomy.	Not specified. No security enabled.

Entry	Description	Default
Logging	For troubleshooting, set this value to 1 to write logging information into a file called <i>security.log</i> . Entries in this file show exactly what the DLL is doing. Once the system is configured properly, turn this setting off to avoid slowing it down.	FALSE = 0 = NO
Type	The type of security you are using. Possible values are: <ul style="list-style-type: none"> • AUTONOMY_SECURITY_NOTES • MAPPEDAUTONOMY_SECURITY_ODBC • AUTONOMY_SECURITY_ORACLE • AUTONOMY_SECURITY_NT • AUTONOMY_SECURITY_NT_MAPPED • AUTONOMY_SECURITY_EXCHANGE • AUTONOMY_SECURITY_EXCHANGE_MAPPED • AUTONOMY_SECURITY_UNIX_MAPPED 	Not specified. No security enabled.
DebugDecrypt	Provides a finer grain of logging. Set to 1 to print the access control list (ACL) against which the DLL matches. This is recommended only for debugging because it is a security risk to have ACL information printed in a plain text file, and the DRE's performance is hindered. This setting is in effect only when logging is set to 1. This setting is available only for the mapped version of the security plug-ins.	FALSE = 0 = NO
ReferenceField	The field used to store the ACL against which the security entitlement is compared. You can specify alternative fields in this entry. For example, in the mapped version of the security modules, this entry is usually: ReferenceField=AUTONOMYMETADATA	AUTONOMYMETADATA
CacheMaxSize CacheExpiryMinutes	The unmapped version of the DLLs can cache security information as it becomes available. The next time the same request is made, the results are returned faster. <ul style="list-style-type: none"> • CacheMaxSize – maximum cache size; the default is 1000 bytes. • CacheExpiryMinutes – time in minutes after which each cached entry expires; the default is 0 (never expires). 	1000 0

Entry	Description	Default
HideField	<p>If the value of ReferenceField is set to anything other than the DREREFERENCE, set the value of HideField to 1 so that the field is not shown in the results that the DRE sends back. This is primarily to be used when implementing mapped security, so that the encrypted ACLs are not shown as part of result sets.</p> <hr/> <p>Note The DREREFERENCE is a unique identifier within each <i>idx</i> file.</p>	FALSE = 0 = NO

Sorting queries in the DRE

To sort a search result list, you must include an `xoptions` parameter in a query. Table C-3 describes the values you can use to sort the results.

Table C-3: *xoptions* parameters

Parameter	Description
<code>sortdate</code>	Sorts results by the date and time at which the document was added to the DRE.
<code>sortboth</code>	Sorts results by date and relevance (preferred method for sorting by date).
<code>absweight</code>	Sorts results by the sum of weights of the terms present in the document, rather than by a percentage score.

For example, you can sort by date using this HTTP request string:

```
http://HOST:QUERYPORT/qmethod=q&querytext=
sybase&qnum=10&xoption=sortdate
```

Before submitting the query to the search bean's `doQuery` method, you must modify the query to include the `xoptions` parameter. This example shows one method of controlling the sort order of search results:

```
public static final int SORT_DATE = 0;
public static final int SORT_BOTH = 1;
public static final int SORT_WEIGHT = 2;
public int numOfResults = 15;
public int minRelevance = 75;
```

`&xoptions=<sort_option>` is appended to the query. The ampersand (&) is required, and the syntax must be exactly as illustrated in this example:

```
/**
```

```
* Application's search method -- the decision of the
* order in which to sort the search results* is made here.*/

public void doSearch (Search searchBean,
                     String query,
                     int sortOrder)
{
    switch(sortOrder)
    {
        case SORT_DATE:
            query = query + "&xoptions=sortdate";
            break;
        case SORT_BOTH:
            query = query + "&xoptions=sortboth";
            break;
        case SORT_WEIGHT:
            query = query + "&xoptions=absweight";
            break;
        default:
            // unrecognized sort order
    }

    searchBean.doQuery(query, "", "", numOfResults, minRelevance);
}
```

DRE administration tool

The DRE administration tool performs the indexing and administration tasks required by the DRE. The DRE must be running before you use the administration tool.

This tool is supplied with both Windows and UNIX, but you must run the executable from a Windows machine. If your Sybase installation is on UNIX, copy the *\$SYBASE/PortalSearch/Engine/<Windows directory>* subdirectory from UNIX to Windows.

Online help is available from the DRE administration tool.

❖ Starting the DRE administration tool

- 1 Select Start | Run.

- 2 Use the Windows Explorer to locate the executable in `%SYBASE%\PortalSearch\Engine`, then double-click `guidadmin.exe` to execute the program
- 3 To set the host and query port, click Change DRE Settings.
- 4 In the Change DRE Settings dialog box, enter:
 - Host Name or IP Address – machine where the DRE is running.
 - Query Port – the default is 8500.
- 5 Click OK. The DRE administration tool connects to the DRE, and the main window displays.

Managing the connection cache

The `epSearchCache` connection cache is the interface to the portal search database. The default connection cache size is configured during installation. To change the size of the cache, use Jaguar Manager—see Chapter 4, “Database Access,” in the *EAServer System Administration Guide*.

Automatic indexing

The automatic indexing utility is part of Enterprise Portal’s concept-based search service. Use the indexing utility to schedule the indexing of documents in specified data sources. You must index the data sources before you can search them. See “Performing concept-based searches” on page 326 for a description of concept-based searching.

The automatic indexing utility runs continuously and performs operations from one or more queue files or directories. Specify the names of the queue files or directories in the `autoindexer.cfg` file. See “Configuration file” on page 337.

Note The automatic indexing utility `PortalSearchAutoindexer` runs as a process on UNIX and as both a process and a service on Windows.

A queue file contains a list of file names with a full or relative path. The automatic indexing utility automatically processes new files that appear in the directory specified in the queue file, together with any files that have been replaced or updated. A file's indexed contents are saved to another file with an *idx* extension, which can be read by the DRE. The automatic indexing utility also processes data formats such as HTML, Microsoft Word documents, and PDF files.

On Windows platforms, once you start the DRE, you can start the indexing utility as a Windows service that adds new data content dynamically. Communication between the indexing utility and the DRE is fully automated and, once you invoke it, the process can run continuously.

The *PortalSearchautoindexer.log* file records all the actions performed by the indexing utility, and is located in the *PortalSearch/autoindexer* subdirectory of your EP installation.

The indexing utility automatically attempts to import and index any *new* files that appear in the data source directories. Be aware of any applications that create temporary files in a data source directory. For example, if you run Microsoft Word from a data source directory, the temporary files it creates are processed by the indexing utility.

You can define the data source directories, file types, the polling period for data files, and so on, in the autoindexer's configuration file. The indexing utility uses one configuration file, *PortalSearchautoindexer.cfg*, which is located in the *PortalSearch/autoindexer* subdirectory of your EP installation.

Note In some older installations, the indexing utility configuration file is *DRE.INI*.

Starting and stopping the automatic indexing utility

❖ Starting the automatic indexing utility on UNIX

- Change to *\$SYBASE/PortalSearch/autoindexer*, and run:

```
PortalSearchautoindexer
```

❖ Starting the automatic indexing utility on Windows 2000

- 1 Select Start | Settings | Control Panel | Administrative Tools.
- 2 Select Services.

- 3 When the Services dialog appears, scroll down the list, click Sybase Search – Auto Indexer to select that service, then click Start.
- 4 When the Status changes to “Started,” click the close box to exit the window.

❖ **Stopping the indexing utility on UNIX**

- 1 Find the `PortalSearchautoindexer` process ID:

```
ps -ef | grep PortalSearchautoindexer
```

- 2 Kill the `PortalSearchautoindexer` process:

```
kill -9 processID
```

❖ **Stopping the indexing utility in Windows 2000**

- 1 Select Start | Settings | Control Panel | Administrative Tools.
- 2 Select Services.
- 3 In the Services dialog box, scroll down the list and click `PortalSearchAutoindexer`, then click Stop.
- 4 When the Status becomes blank, click Close.

Restarting the indexing utility

When you restart the indexing utility, you can either reprocess the last file that was processed, or start from scratch with a new file.

The indexing utility generates these files:

- `<AppName>.dirstat` – keeps a list of all the files that have been processed.
- `<AppName>.dirstat.bak` – a copy of `<AppName>.dirstat` before the last file was processed.
- `<QueueFile>.pos` – records the current position in the queue of files to be processed.
- `<QueueFile>.pos.bak` – a copy of `<QueueFile>.pos` before the last file was processed.

❖ **Reprocessing the most recent file**

- 1 Replace the contents of `<AppName>.dirstat` with the contents of `<AppName>.dirstat.bak`.

- 2 Replace the contents of *<QueueFile>.pos* with the contents of *<QueueFile>.pos.bak*.

❖ **Starting from scratch**

- Delete these files from the *PortalSearch/autoindexer* subdirectory of your EP installation:
 - *<AppName>.dirstat*
 - *<AppName>.dirstat.bak*
 - *<QueueFile>.pos*
 - *<QueueFile>.pos.bak*

❖ **Restarting the automatic indexing utility on UNIX**

- Run the script
\$SYBASE/PortalSearch/autoindexer/PortalSearchAutoindexer.

❖ **Restarting the automatic indexing utility on Windows 2000**

- 1 Select Start | Settings | Control Panel | Administrative Tools.
- 2 Select Services.
- 3 When the Services dialog box displays, scroll down the list, right-click Sybase Portal Search – Auto Indexer to select that service, then click Start.
- 4 When the Status reads “Started,” click the close box to exit the window.

Configuration file

The indexing utility uses a configuration file, *PortalSearchautoindexer.cfg*, which is located in the *PortalSearch/autoindexer* subdirectory of your EP installation.

Note In some older installations, the indexing utility configuration file is *DRE.INI*.

The configuration file has a format similar to a standard Windows *.ini* file, with a number of sections defined by square brackets. Each section contains a number of *key=value* pairs. The three sections in the configuration file govern different aspects of the operation of the DRE and are arranged in this format:

```
[Configuration]
```

```
[Default]
[job_name]
```

[*job_name*] represents the section containing details on a particular job.

Use the indexing utility to index, import to, and delete *idx* files. To perform more than one action, create multiple jobs within your configuration file. You can also perform these actions within more than one DRE. See the DRE host settings in the [Default] section of *PortalSearchautoindexer.cfg*.

Basic configuration parameters

Table C-4 describes the automatic indexing configuration parameters. The values in this table are used in the [Configuration] section of *PortalSearchautoindexer.cfg*. Each description column includes a sample *key=value* pair.

Table C-4: Configuration parameters

Key	Description
PollingPeriod	Specifies the time interval in milliseconds between each autoindexing poll. PollingPeriod=5000
PollingMethod	Specifies whether the files to be processed by the indexing utility are read from a file or from a directory. The value of this parameter can be either: <ul style="list-style-type: none"> • 1 – file. • 2 – directory. For example: PollingMethod=2 When processing a large number of files on Windows, you must use file polling instead of directory polling. File polling simply looks at the names in a file, then processes them. This is more scalable because it does not require directory tree polling. On UNIX, use “1,” and send the output to a file. On Windows, make a directory listing, and set the following in the <i>PortalSearch\autoindexer\PortalSearchautoindexer.cfg</i> configuration file: [configuration] FilenameOutputMode=1 This prints a list of all the file names that would be processed in directory mode. All settings for directory polling, recursion, musthaves, canthaves, dates, and so on, are considered when listing the file names. Insert the <code>PollingMethod</code> key in the configuration section instead of in each job section. If you have more than one job, and you want to create more than one listing of file names, move the appropriate job to the top and run it. If you do not specify <code>FilenameOutputMode</code> , the file names print to stdout.

Key	Description
RemoveLogFileOnStart	Indicates whether to remove the log file before running the autoindexing utility. The value for this parameter can be: <ul style="list-style-type: none"> • 1 or on – yes • 0 or off – no For example: RemoveLogFileOnStart=0 The default value of this key value pair is off (0).
Number	The number of jobs that are to be performed by a particular indexing utility.
n	The name of job number “n”; n=0 for the first job.
MaxLogKBytes	The maximum size (in kilobytes) that the log file can reach before it is renamed <i>as.log.previous</i> and a new log file is created.

Default parameters

Table C-5 describes the default configuration settings. The values in this table are used in the [Configuration] section for any job that does not have its own definition entry.

Table C-5: Default parameters

Key	Description
DREHost	IP address where the DRE resides. To index into more than one DRE, separate the IP addresses by commas. DreHost=localhost,120.7.0.0
QueryPort	The port number by which queries are sent to the DRE. To use more than one DRE, specify the query ports separated by commas. QUERYPORT=2000,6000
IndexPath	The port number by which indexing commands are sent to the DRE. If this line is not present or is set to 0, then indexing is not available. To use more than one DRE, specify the index ports separated by commas. INDEXPORT=2001,6001
PollingAction	The action to perform when polling. <ul style="list-style-type: none"> • 1 – index <i>.idx</i> files to a DRE (<i>.idx</i> files are files that are in hash form). • 2 – import files in various formats and index them into a DRE. • 7 – import files in various formats and index them into a DRE. Index <i>.idx</i> files to a DRE, then delete the files. • 8 – delete from a DRE. PollingAction=2

Key	Description (Continued)
PollingPostAction	<p>The action to perform after the file has been processed. The value of this parameter can be:</p> <ul style="list-style-type: none"> • 0 – do nothing. • 1 – delete the file after processing. • 2 – move the file to another directory. It retains the subdirectory structure. <p>The default value is 0.</p> <pre>PollingPostAction=0</pre> <p>When moving the originally scanned files, the subdirectory structure is retained.</p> <pre>Bool FileCopyKeepDirStructure(char*szRoot1, char*szFilename, char*szRoot2, char*szMode)</pre>
MoveToDirectory	<p>Destination directory when PollingPostAction is set to 2.</p> <pre>MoveToDirectory=D:\projects\autoindexer\processed\</pre>
PollingMaxNumber	<p>Maximum number of files to be processed at each poll.</p> <pre>PollingMaxNumber=100</pre> <p>The default value is 100.</p>
FilePollFilename	<p>File name from which the list of files to be processed is read.</p> <pre>FilePollFilename=queue</pre>
fileBaseDirectory	<p>Directory path to attach to each file in FilePollFilename. Specify either the full path or only the file names inside the queue file.</p> <pre>BaseDirectory=c:\Files\</pre>
directoryPathCSVs	<p>The directory where the files to be processed reside.</p> <pre>directoryPathCSVs=D:\projects\autoindexer\files</pre> <p>To process more than one directory, separate paths with commas.</p>
DirectoryFileMatch	<p>Specifies files using wildcards.</p> <pre>directoryFileMatch=*</pre>
directoryRecurse	<p>Indicates whether to recurse into directories.</p> <ul style="list-style-type: none"> • 1 or on – recurse. • 0 or off – do not recurse. <pre>directoryRecurse=off</pre>
directoryMustHaveCSVs	<p>Specifies the string that must appear in the directory path of a document indexed via Web Fetch.</p> <pre>directoryMustHaveCSVs=*/temp/*</pre>
directoryCantHaveCSVs	<p>Specifies the string that cannot appear in the directory path of a document indexed via the Web Fetch.</p> <pre>directoryCantHaveCSVs=*.sys,*.bat,*.exe</pre>
directorybeforeDate	<p>The number of days after today that the document must be modified for AutoIndexer to process them.</p> <pre>directoryBeforeDate=5</pre> <p>(5 days from today)</p>

Key	Description (Continued)
directoryAfterDate	The number of days before today that the document must have been modified for AutoIndexer to process them. <code>directoryAfterDate=-1</code> (yesterday)
importIDXFilesAction	Specifies what to do with <i>idx</i> files that have been processed. <ul style="list-style-type: none"> • 0 – delete the <i>idx</i> file. • 1 – move the <i>idx</i> file to another directory. • 2 – do nothing.
importIDXFilesMoveTo	If importIDXFilesAction is set to 1, this specifies where to move the <i>idx</i> files.
deleteRefReplaceCSVs	If PollingAction is set to either 7 or 8, this specifies the string to be replaced in the reference when looking for the document to delete.
deleteRefReplaceWithCSVs	If PollingAction is set to either 7 or 8, this specifies the string that is to replace the string specified by deleteRefReplaceCSVs in the document reference.
ImportPathReplaceString	Specifies the string to replace the substrings of files in the queue.
deletePathReplaceUpToSlash	If PollingAction is set to either 7 or 8, this specifies a string to be replaced. You can use this to replace a portion of many strings that contain different substrings, as opposed to replacing a whole word. If the files in the queue are <i>C:\a\b\hello,c:\a\c\hello and c:\b\a\hello</i> , this key replaces the portion up to the third backslash in each string: <code>deletePathReplaceUpToSlash=3</code>
deleteEscapeReferences	Boolean value that when set to 1, escapes references when doing deletion. This is useful when trying to remove documents with spaces in the reference.
FilenameOutputMode	Boolean value that when set to 1, creates a file that lists all the files to be processed. This is useful if you are doing file-based processing.

Import parameters

This section describes the import parameters. There is one table for each import parameter type.

Table C-6: Input parameters

Key	Default	Value	Description
TextImportExtns	None	String	Any extra extensions for text import.
HTMLImportExtns	None	String	Any extra extensions for HTML import.
ImportDefaultExtension	None	String	If the import module finds a file with an unknown extension, it treats it as the type you enter here, for example, <i>.txt</i> or <i>.html</i> .

Key	Default	Value	Description (Continued)
ImportTempDir	.\	String	A temporary directory for the importer to use when importing nonstandard formats; in other words, documents that are not <i>.txt</i> or <i>.html</i> .
ImportRecursive DirectoryImporting	FALSE	TRUE/ FALSE	When set to TRUE, recursively imports subdirectories.

Table C-7: Content parameters

Key	Default	Value	Description
ImportMinLength	0	Integer	Any document with size smaller than this number of bytes is not imported. The size of the document is measured according to the plain text content.
ImportMaxLength	100000000	Integer	Any document with size greater than this number of bytes is not imported. The size of the document is measured according to the plain text content.
ImportMinLengthWords	0	Integer	Any document with less than this number of words is not imported.
ImportMaxLengthWords	1000000	Integer	Any document with more than this number of words is not imported.
ImportStoreContent	TRUE	TRUE/ FALSE	Specifies whether to store the content of the document in the destination DRE.
ImportSummary	FALSE	TRUE/ FALSE	Imports a field called “summary” that contains the quick summary information generated by the importing module.
ImportSummarySize	3	Integer	Number of sentences to be used in the quick summary generated by the importing module.
ImportBreaking	FALSE	TRUE/ FALSE	Specifies whether to break the document into sections when importing. To enable, you must also set the value of combine to 1 in the <i>PortalSearchqueryh.cfg</i> or <i>DRE.INI</i> file of the engine into which the data is being entered.
ImportBreakingMin ParagraphWords	160	Integer	The smallest section that can be created when attempting to break the document at paragraph boundaries.
ImportBreakingMax ParagraphWords	360	Integer	The largest section that can be created when attempting to break the document at paragraph boundaries.
ImportBreakingMinDoc Words	600	Integer	The size above which document breaking is implemented. Smaller documents remain whole.

Key	Default	Value	Description (Continued)
ImportIntelligentTitle Summary	FALSE	TRUE/ FALSE	When set to TRUE, the import module attempts to find unique titles and summaries by comparing the title or summary of each document with the next. For example, if the first document has the title “News Today,” then the import module sets this as its title. If a second document has the same title, the import module attempts to find a unique title either from the headings (<H1>, <H2>, and so on) or failing that, from the content.

Table C-8: Field parameters

Key	Default	Value	Description
FixedFieldName <i>n</i>	None	String	The name of the field in <i>PortalSearchqueryh.cfg</i> or <i>DRE.INI</i> that is used to store the fixed field value.
FixedFieldValue <i>n</i>	None	String	The value that is stored in the DRE field denoted by FixedFieldName <i>n</i> .
FieldName <i>n</i>	None	String	The name of the field in <i>PortalSearchqueryh.cfg</i> or <i>DRE.INI</i> that stores the dynamic field value.
FieldStart <i>n</i>	None	String	Specifies the start of the value to be stored in the DRE field named in FieldName <i>n</i> .
FieldStop <i>n</i>	None	String	Specifies the end of the value to be stored in the DRE field named in FieldName <i>n</i> .
ImportRemapField <i>n</i>	None	String	The name of the field whose value should be used as the value of an alternative field specified by ImportRemapFieldTon.
ImportRemapFieldTon	None	String	The name of the field that takes its value from an alternative field specified by ImportRemapField <i>n</i> .
ImportFieldHTML ConvertChars	FALSE	TRUE/ FALSE	Set this parameter to TRUE if you want HTML entities in fields to be replaced with an equivalent character. For example, is replaced with a space character.
ImportMetaToFields	TRUE	TRUE/ FALSE	Specifies whether to use HTML meta tags for DRE fields.
ImportChecksum	FALSE	TRUE/ FALSE	If set to TRUE, a value is added to the Checksum field in the [Field] section in <i>PortalSearchqueryh.cfg</i> or <i>DRE.INI</i> . This field value is used to determine whether to show a document result in the front end.
HTMLFieldName <i>n</i>	None	String	The name of the field in <i>PortalSearchqueryh.cfg</i> or <i>DRE.INI</i> that is used to store the dynamic field value in an HTML document. The value stored is not HTML-stripped.

Key	Default	Value	Description (Continued)
HTMLFieldStart <i>n</i>	None	String	Specifies the start of the value to be stored in the DRE field named in HTMLFieldName <i>n</i> .
HTMLFieldStop <i>n</i>	None	String	Specifies the end of the value to be stored in the DRE field named in HTMLFieldName <i>n</i> .

Table C-9: DRE parameter

Key	Default	Value	Description
Database		String	The database into which documents are indexed.

Table C-10: Data parameters

Key	Default	Value	Description
ImportExtractDateFrom	0	Integer *	The date to be extracted from the document. 0 – nothing. 1 – current time. 2 – last accessed date. 3 – last modified date. 16 – from DRE field. 32 – from content. 64 – from file name.
ImportExtractDateFrom Field	None	String	The name of a date field in the DRE whose value is to be extracted.
ImportExtractDateTo Field	DREDATE	String	The name of a date field in the DRE whose value is taken from ImportExtractDateFromField.

Key	Default	Value	Description
ImportExtractDate FormatCSVs	None	String	<p>The comma-separated format in which dates are to be extracted. The format specifiers are used whenever dates can be specified in the DDMMYY format style strings:</p> <ul style="list-style-type: none"> • YY – 2-digit year, for example, 99 or 00 or 01. • YYYY – 4-digit year, for example, 1999, 2000, 2001. • LONGMONTH – January, March, August. • SHORTMONTH – Jan, Mar, Aug. • MM – 2-digit month, for example, 01, 10, 12. • M+ – 1- or 2-digit month, for example, 1,2,3,10. • DD – 2-digit day, for example, 01, 02, 03, 12, 23. • D+ – 1- or 2-digit day, for example, 1, 2, 12, 13, 31. • HH – 2-digit hour, for example, 01, 12, 13. • H= – 1- or 2-digit hour, for example, 1, 2, 12, 13, 31. • NN – 2-digit minute, for example, 01, 12, 13. • N+ – 1- or 2-digit minute, for example, 1, 2, 12, 13, 31. • SS – 2-digit second, for example, 01, 12, 13. • S+ – 1- or 2-digit second, for example, 1, 2, 12, 13, 31. • ZZZ – time zone, for example, GMT, EST, PST. <p>The correct syntax is illustrated in the following examples:</p> <ul style="list-style-type: none"> • Dates with 1- or 2-digit days: <ul style="list-style-type: none"> DateFormats=D+/SHORTMONTH YYYY, DDMMYY • Quoted string to allow spaces, commas, and so on, within the format: <ul style="list-style-type: none"> DateFormats= "D+SHORTMONTH YYYY", "Date: D+ LONGMONTH, YYYY" • Directory style dates : <ul style="list-style-type: none"> DateFormats+D+/M+/YY, MM/DD/YYYY
ImportExtractDateTo Format	None	String	<p>Specifies the format of the dates extracted once in the DRE. If the value of ImportExtractDateToField is set to DREDATE, then ImportExtractDateToFormat is set to the current date in this format: <i>yyyy/mm/dd</i>.</p>
ImportExtractLength	FALSE	TRUE/ FALSE	<p>Specifies whether to allow the extraction of the file length.</p>

Key	Default	Value	Description
ImportExtractLengthToField	FILELENG TH	String	If ImportExtractLength is set to 1, the file length is extracted into the DRE field specified by this parameter; for example: ImportExtractLengthToField=FileLength

Table C-11: Page layout parameters

Key	Default	Value	Description
ImportStartDefCSVs	None	String	Comma-separated list of strings that marks the beginning of the text in a document to be indexed into the DRE.
ImportEndDefCSVs	None	String	Comma-separated list of strings that marks the end of the text in a document to be indexed into the DRE.
HTMLImportStartDefCSVs	None	String	Comma-separated list of strings that marks the beginning of text in an HTML document to be indexed into the DRE.
HTMLImportEndDefCSVs	None	String	Comma-separated list of strings that marks the end of text in an HTML document to be indexed into the DRE.
ImportPageBreakDefs	None	String	Specifies a string used to mark a document break. Use this parameter to split documents into segments with each segment contained in one <i>idx</i> format. After the <i>idx</i> files have been indexed into the DRE, the individual segments are joined back together to produce the original document.
ImportStartSkipWords	0	Integer	Specifies the number of words to skip from the beginning of a document when importing a document.
ImportStartSkipSentences	0	Integer	Specifies the number of sentences to skip from the beginning of a document when importing a document.

Table C-12: Path parameters

Key	Default	Value	Description
ImportPathReplaceUpToSlash	-1	Integer	Specifies the slash through which the string is replaced with the string specified in ImportPathReplaceString. You can use this parameter to replace a single portion of many strings that contain different substrings, as opposed replacing a single word. For example, if the files in the queue are <i>c:\a\b\hello</i> , <i>c:\a\c\hello</i> , and <i>c:\b\a\hello</i> , setting ImportPathReplaceUpToSlash to 3 will replace the portion of the strings up to the third backslash in each string. By default, nothings is replaced.
ImportPathReplaceString	None	String	Specifies a string to replace the string through the slash indicated by the ImportPathReplaceUpToSlash parameter.

Key	Default	Value	Description (Continued)
ImportRefTruncateAfter	-1	Integer	Truncates the reference after the nth occurrence of the string specified in ImportRefTruncateString. By default, no truncation occurs.
ImportRefTruncateString	None	String	Specifies the string used to truncate the reference after the nth occurrence.
ImportRefReplaceCSVs	None	String	Specifies the string to be replaced.
ImportRefReplaceWithCSVs	None	String	Specifies the string to replace the original reference of the file.
ImportRegisterExtnCSVsN=	None	String	Specifies the files with certain extensions that will be imported; for example: <code>ImportRegisterExtnCSVs0=HTML, XML</code>
ImportRegisterExecutableN=	None	String	This specifies the import slave that imports the corresponding files (see ImportRegisterExtnCSVs); for example: <code>ImportRegisterExecutable0=testslave.exe</code> In this example, <i>testslave.exe</i> imports HTML and XML files.

Using the ImportRefReplace parameters

When a directory is imported, references to the files are similar to `/work/Inetpub/wwwroot/filename`. To store the actual reference for a file so you can access its URL, replace `/work/Inetpub/wwwroot` with `http://hostname/` in the configuration file:

```
ImportRefReplaceCSVs=/work/Inetpub/wwwroot
ImportRefReplaceWithCSVs=http://hostname/
```

Strings are replaced from left to right. For example:

```
ImportRefReplaceCSVs=/work/Inetpub/wwwroot
ImportRefReplaceWithCSVs=http://hostname/
```

replaces:

```
/work/Inetpub/wwwroot/directoryname/myfile
```

with:

```
http://hostname/directoryname/myfile
```

You can apply the Import Reference Replace function more than once by setting the value of the `ImportRefReplaceCSVs` and `ImportRefReplaceWithCSVs` parameters to a list of comma-separated paths; for example:

```
ImportRefReplaceCSVs=C:\Inetpub\wwwroot\,D:\Inetpub\wwwroot\
ImportRefReplaceWithCSVs=http://193.128.174.23/,http://193.128.174.38/
```

This replaces the string “C:\Inetpub\wwwroot\” with “http://193.128.174.23/” and replaces “D:\Inetpub\wwwroot\” with “http://193.128.174.38/” whenever they appear in the queue of files to be processed. You must ensure that the number of paths specified in `ImportRefReplaceCSVs` is equal to the number of paths in `ImportRefReplaceWithCSVs`.

Reference replacement is recursive, which means that if you want to replace:

```
C:\Inetpub\wwwroot\data\data1
```

with:

```
http://127.0.0.1/data/data1
```

enter:

```
ImportRefReplaceCSVs=C:\Inetpub\wwwroot,\
ImportRefReplaceWithCSVs=http://127.0.0.1,/
```

`ImportPathReplaceUpToSlash` and `ImportPathReplaceString` are both performed before `ImportRefReplaceCSVs` and `ImportRefReplaceWithCSVs`.

The import rendering parameters are described in Table C-13.

Table C-13: Import rendering parameters

Key	Default	Value	Description
ImportRenderedHTMLExtensions	None	String	Specifies the extension of files on which to perform HTML rendering.
ImportRenderedHTMLFieldName	None	String	For any non-HTML file that gets imported, the HTML file that comes out of <i>Keyview/PDF</i> gets moved into this field.
ImportRenderedHTMLMoveToDir	None	String	For any non-HTML file that gets imported, the HTML file that comes out of <i>Keyview/PDF</i> gets moved into this directory.
ImportRenderedHTMLPathReplaceUpToSlash	-1	Integer	Specifies a string that is to be replaced up to a certain ‘\’. Used to set the right reference for the <code>ImportRenderedHTMLFieldName</code> field.
ImportRenderedHTMLPathReplaceString	None	String	Specifies the replacement string of the string immediately preceding the nth ‘\’.
ImportRenderedHTMLRefReplaceCSVs	None	String	Specifies the string that is to be replaced. Used to set the right reference for the <code>ImportRenderedHTMLFieldName</code> field.

Key	Default	Value	Description (Continued)
ImportRenderedHTMLRefReplaceWithCSVs	None	String	Specifies the string that is to replace the original reference of the ImportRenderedHTMLFieldName field. Used to set the right reference for the ImportRenderedHTMLFieldName field.

Using the ImportRendered parameters

These parameters are used for import rendering:

- `ImportRenderedHTMLMoveToDir`
- `ImportRenderedHTMLFilenameExtensions`
- `ImportRenderedHTMLFieldName`

You can also perform any reference replacement necessary to get the right path. These settings are equivalent to those used for the DRE document reference.

- `ImportRenderedHTMLPathReplaceUpToSlash`
- `ImportRenderedHTMLPathReplaceString`
- `ImportRenderedHTMLPathReplaceCSVs`
- `ImportRenderedHTMLRefReplaceWithCSVs`

The keyview and PDF slaves temporarily convert all the various formats (for example, Microsoft Word, PDF, Excel, PPT) into HTML. The rendered HTML is then processed to create data in *idx* ASCII format. You can store the temporary HTML data in a repository on the Web server.

The tables that follow describe each type of import rendering parameter.

Table C-14: Parsing parameters

Key	Default	Value	Description
ImportStripTagsCSVs	None	String	Strips the content between begin and end tags; for example: <code>ImportStripTagCSVs=</code> strips all content between <code></code> and <code><\B></code> .
ImportStripStringCSVs	None	String	Comma-separated list of strings that when matched in a document are stripped from the <i>idx</i> format.
ImportStripLinks	TRUE	TRUE/ FALSE	Specifies whether to remove hypertext links.

Table C-15: Delimited parameters

Key	Default	Value	Description
ImportDelimitedExtns	None	String	Specifies the extension of the files that are to be treated as delimited files; for example, .csv files.
ImportDelimitedDoc Start	None	String	Specifies the beginning of a record when you are importing a delimited file.
ImportDelimitedDoc End	None	String	Specifies the end of a record when you are importing a delimited file.
ImportDelimitedStart <i>n</i>	-1	Integer	Specifies the start string of data to be inserted into the <i>idx</i> field. This <i>idx</i> field is denoted by <code>ImportDelimitedField<i>n</i></code> .
ImportDelimitedEnd <i>n</i>	None	String	Specifies the end string of data to be inserted into the <i>idx</i> field. The name of this field is specified by <code>ImportDelimitedField<i>n</i></code> .
ImportDelimitedField <i>n</i>	None	String	Specifies the name of the field where data from a delimited file is stored. <code>ImportDelimitedStart<i>n</i></code> and <code>ImportDelimitedEnd<i>n</i></code> specify the data to include in that field.
ImportDelimitedSkip Chars <i>n</i>	0	Integer	When an <i>idx</i> file is produced, a number of characters are skipped at the beginning of a document. This specifies how many characters to skip.

Table C-16: XML parameters

Key	Default	Value	Description
ImportXMLField <i>n</i>	None	String	<p>Specifies the field in the DRE from which to take the content from the XML file, as defined by the parameter <code>ImportXMLSearchCSVTags<i>n</i></code>; for example, if:</p> <pre>ImportXMLSearchCSVTags0=head,title ImportXMLField0=DRREFERENCE</pre> <p>and the XML file contains the following strings:</p> <pre><head> XML Example </head> <title> Example 1 </title></pre> <p>the DRE field <code>DRREFERENCE</code> gets the value:</p> <pre>XML Example 1</pre>

Key	Default	Value	Description (Continued)
ImportXMLSearch CSVTags <i>n</i>	None	String	Searches for the specified XML tags. This can be a hierarchical level of search tags; for example: <code>ImportXMLSearchTags0=body,head,title</code> extracts the content between these tags: <ul style="list-style-type: none"> • <code><body></body></code> • <code><head></head></code> • <code><title></title></code>
ImportXMLStripTags <i>n</i>	FALSE	TRUE/ FALSE	Specifies whether to strip the strings between the specified XML tags when importing a file.
ImportXMLExtns	None	String	A comma-separated list of all XML file extensions; for example: <code>ImportXMLExtns=*.xml</code>
ImportXMLEntryOnly <i>n</i>	FALSE	TRUE/ FALSE	Specifies whether to import only the content between the searched tags at that level; for example, if you set: <code>ImportXMLSearchCSVTags0=TAG1,TAG2</code> <code>ImportXMLEntryOnly0=TRUE</code> and the XML file contains the following strings: <pre><TAG1> <TAG3> Some Other Tag3 entry </TAG3> <TAG2> Some data <TAG3> The entry for Tag3 </TAG3> </TAG2> </TAG1></pre> The imported content is: <pre>Some data</pre> Whereas, if you set: <code>ImportXMLEntryOnly0=FALSE</code> The imported content is: <pre>Some data The entry for Tag3</pre>

Table C-17: Logging parameters

Key	Default	Value	Description
ImportLogFile	None	String	Specifies the name of the file used to record the import process.

Key	Default	Value	Description (Continued)
ImportLogLevel	Errors	Errors Warnings None Full	Specifies the type of logging to be used.
ImportLogFileAppend	FALSE	TRUE/ FALSE	Specifies whether to append details to an existing import log file.

Table C-18: Binary content parameters

Key	Default	Value	Description
ImportFilterBinary Content	FALSE	TRUE/ FALSE	Specifies whether to process binary content.
ImportMaxBinaryChars PerHundred	3	Integer	Maximum number of binary characters allowed per 100 characters. Anything exceeding this number is ignored.
ImportMaxAverage WordLength	25	Integer	Binary words with more than this number of characters are not imported.
ImportMinAllowed ASCII CODE	0	Integer	The minimum ASCII value allowed in characters in the binary content. Anything lower is treated as binary and removed.
ImportMaxAllowed ASCII CODE	175	Integer	The maximum ASCII value allowed in characters in the binary content. Anything higher is treated as binary and removed.

Table C-19: E-mail parsing parameters

Key	Default	Value	Description
ImportAttachments	FALSE	TRUE/ FALSE	Specifies whether to import e-mail attachments.
ImportSaveAttachments	FALSE	TRUE/ FALSE	Specifies whether to save e-mail attachments in the directory specified by ImportSaveAttachmentDir.
ImportSaveAttachment Dir	.\	String	Specifies the directory path where e-mail attachments are to be saved.

Import anchors

The import module automatically breaks HTML pages on anchors. It retains the reference to the actual anchor, so that when you click on a result, you go to that anchor section, rather than the top of the page.

To use this feature, set the value of `ImportBreaking` to `true`, and verify that you have set the breaking parameters according to the size of the data in each anchor.

Set the value of `ImportIntelligentTitleSummary` to `1` so titles and summaries are of the individual anchor sections rather than the page in general. Otherwise, all summaries and most titles will be the same.

[job_name] section of configuration file

All *key=value* pairs that appear in the `[Default]` section can also appear in the `[job_name]` section. Entries in `[job_name]` overwrite the settings in `[Default]`. If an entry is not in `[job_name]`, the indexing utility uses the value from `[Default]`.

An example of the automatic indexing configuration file is shown below:

```
[Configuration]
PollingMethod=2
PollingPeriod=5000
RemoveLogFileOnStart=on
Number=1
0=Job0

[Default]
PollingAction=2
PollingPostAction=0
MoveToDirectory=
PollingMaxNumber=1000
filePollFilename=queue1
fileBaseDirectory=
directoryPathCSVs=
directoryFileMatch=*
directoryRecurse=on
DreHost=
QueryPort=
IndexPort=
importIDXFilesAction=1
importIDXFilesMoveTo=
Database=Test
ImportStoreContent=on
ImportTempDir=
ImportStripTagCSVs=B,H1,ZZ
ImportPathReplaceUpToSlash=4
ImportPathReplaceString=http://www.southpark.com
```

```
ImportRefReplaceCSVs=I:\inetpub\wwwroot\,\
ImportRefReplaceWithCSVs=http://193.128.174.142/,/
ImportMinLength=0
ImportMinLengthWords=0
ImportMaxLength=10000
ImportMaxLengthWords=500000
ImportSummary=on
ImportBreaking=off
ImportBreakingMinParagraphWords=10
ImportBreakingMaxParagraphWords=200
ImportBreakingMinDocWords=1500
ImportStripLinks=off
ImportMetaToFields=off
TextImportExtns=.ini
HtmlImportExtns=.tmpl
FixedFieldName0=SiteImage
FixedFieldValue0=/sourceimages/times.gif
FieldName0=URL_NAME
FieldStart0=<!--URL_NAME=
FieldStop0=?
deleteRefReplaceCSVs=
deleteRefReplaceWithCSVs=
```

```
[Job0]
PollingAction=2
PollingPostAction=0
MoveToDirectory=
PollingMaxNumber=1000
filePollFilename=queue2
fileBaseDirectory=
directoryPathCSVs=
directoryFileMatch=*
directoryRecurse=on
DreHost=
QueryPort=
IndexPort=
importIDXFilesAction=1
importIDXFilesMoveTo=
Database=Test
ImportStoreContent=on
ImportTempDir=
ImportStripTagCSVs=B,H1,ZZ
ImportPathReplaceUpToSlash=4
ImportPathReplaceString=http://www.southpark.com
ImportRefReplaceCSVs=I:\inetpub\wwwroot\,\
ImportRefReplaceWithCSVs=http://193.128.174.142/,/
```

```
ImportMinLength=0
ImportMinLengthWords=0
ImportMaxLength=10000
ImportMaxLengthWords=500000
ImportSummary=on
ImportBreaking=off
ImportBreakingMinParagraphWords=10
ImportBreakingMaxParagraphWords=200
ImportBreakingMinDocWords=1500
ImportStripLinks=off
ImportMetaToFields=off
TextImportExtns=.ini
HtmlImportExtns=.tmpl
FixedFieldName0=SiteImage
FixedFieldValue0=/sourceimages/times.gif
FieldName0=URL_NAME
FieldStart0=<!--URL_NAME=
FieldStop0=-->
deleteRefReplaceCSVs=
deleteRefReplaceWithCSVs=
```


Using the Enterprise Portal Samples

Enterprise Portal includes several samples. This chapter discusses how to use these samples.

Topic	Page
Using the single sign-on sample	357
Creating single-sign-on portlets with Web applications	368
Creating single-sign-on portlets	375
Using the sample search portlets	385
Using the Content Explorer sample portlet	390
Using the Address Book sample	395

Using the single sign-on sample

Note This sample uses the <IFRAME> HTML tag and therefore works only with Internet Explorer. Netscape 6.0 supports the <IFRAME> tag, but not all of the attributes. Versions of Netscape earlier than version 6.0 do not support the <IFRAME> tag.

This section describes how to create a portlet that allows users to access back-end systems, like e-mail servers, using the single sign-on capability provided by Enterprise Portal.

For example, users may have different accounts with different passwords on several systems. A user with a portal account may also have an e-mail account on a mail server. These two accounts may have different user IDs and passwords. Developers can create portlets to access the e-mail server, and have the Security Officer (PSO) set up proxy authentication using Enterprise Security. This enables users to access the e-mail server from the portal without logging in to the e-mail server separately.

Note All of the source files for this sample are included in the Enterprise Portal installation.

Creating a portlet that provides single sign-on access to back-end systems requires:

- Creating a secure Web application
- Creating a deployable WAR file
- Deploying the WAR file
- Testing the single sign-on example as a standalone Web application
- Creating a page with a portlet that executes the JSP
- Creating proxy authentication information

Sybase has performed the first two steps for you, but we include the procedures for your reference.

Creating a secure Web application

You can use your preferred development environment to create a Web application that contains the JSPs to access your back-end application.

After deploying *sso.war* in EAServer (“Deploying the WAR file” on page 360), you can find *web.xml* under */Repository/WebApplication/sso/WEB-INF* of your EAServer installation.

Here is the sample *web.xml* file that defines a Web application called “SSO” with a single JSP in it:

```
<?xml version='1.0' encoding='UTF-8'?>
<!DOCTYPE web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web Application
2.3//EN" 'http://java.sun.com/dtd/web-app_2_3.dtd'>

<web-app>
```



```
<display-name>sso</display-name>

<servlet>
  <servlet-name>SSO</servlet-name>
  <display-name>SingleSignon</display-name>
  <jsp-file>/SSO.jsp</jsp-file>
</servlet>

<security-constraint>
  <web-resource-collection>
    <web-resource-name>ssoZone</web-resource-name>
    <url-pattern>/SSO.jsp</url-pattern>
    <http-method>POST</http-method>
    <http-method>GET</http-method>
  </web-resource-collection>
  <auth-constraint>
    <description>Role required to access the SSO jsp</description>
    <role-name>PortalUser</role-name>
  </auth-constraint>
  <user-data-constraint>
    <transport-guarantee>NONE</transport-guarantee>
  </user-data-constraint>
</security-constraint>

<login-config>
  <auth-method>BASIC</auth-method>
  <realm-name></realm-name>
</login-config>

<security-role>
<description>Whatever</description>
<role-name>PortalUser</role-name>
</security-role>

</web-app>
```

The JSP is a protected Web resource, and anyone who wants to execute it must be assigned the PortalUser role. If the Web container receives an unauthenticated request for this JSP, the Web container uses basic authentication to validate the user before verifying that the user has the PortalUser role.

Creating a deployable WAR file

This section describes how to create a deployable WAR file for the sample. The process is shown only for reference. The deployable WAR file—*sso.war*—is located in */PortalStudio/samples* in the EP installation directory.

The development environment directory structure looks like this:

```
temp/SSO.jsp
temp/WEB-INF/web.xml
temp/WEB-INF/lib/<JAR files in Web application>
```

For this sample, Sybase created a deployable Web application archive (WAR) file using the `jar` command (entered in at a command line) from the Java JDK:

```
cd temp
jar -cf /tmp/sso.war *
```

These commands create the *sso.war* deployment file in the */tmp* directory.

The *security_stubs.jar* file must be in the *WEB-INF/lib* directory so that *sso.jsp* can use Enterprise Security to access proxy authentication information. We have packaged *security_stubs.jar*, and after you deploy the WAR file in the next step, it is located under *WEB-INF/lib*.

Deploying the WAR file

If you are using EAServer, use either Jaguar Manager or jagtool to deploy the WAR file into EAServer and make it an active application. See the *EAServer System Administration Guide*, Chapter 10, “Importing and Exporting Application Components,” or Chapter 13, “Using jagtool and jagant.” To view the HTML documentation, use your Web browser to load *html/docs/sysadmin/main.htm* from the EAServer installation directory.

Testing the single sign-on example as a standalone Web application

Open a Web browser. In the URL locator, enter:

```
http://HOSTNAME:8080/sso/sso.jsp
```

where *HOSTNAME* is the name of the machine where EAServer is installed and where you deployed *sso.jsp*.

If you have not logged in to this EAServer previously, a dialog box prompts for a user name and password. Enter a valid user name and password for a user that has the PortalUser role and who is recognized by Enterprise Security.

Because you have not yet defined proxy authentication information, there are several items marked “null.” See “Creating proxy authentication information” on page 363.

Creating a page with a portlet that executes the JSP

To add single sign-on access to your portal, create a portlet that executes the secure Web application, create a portal page and add the portlet to it, and create the proxy authentication information for the back-end application.

❖ Creating a JSP portlet

- 1 Log in to the Portal Studio.
- 2 Select Build | Portlets from the Portal Studio left pane.
- 3 Click New on the Portlet Manager toolbar.
- 4 Right-click in the left pane of the Portlet Builder and select New JSP Element from the Element Menu.
- 5 In the JSP Element Definition window, select Use Web Application , then enter these values:

- WAR File to Upload – if you deployed the WAR file using the instructions in “Deploying the WAR file” on page 360, leave this field blank.

If you did not deploy the WAR file, specify the path to *sso.war*, and click Upload. When you are notified that uploading is complete, use the instructions in “Deploying the WAR file” on page 360 to deploy the WAR file into EAServer, then restart EAServer.

- WAR file – “*sso.war*”.
- Web App Display Name – “sso” in lowercase letters.
- Initial Resource – “*/SSO.jsp*”.
- Select “Web App Qualified URLs”.
- Select the “Single Sign-On Required” and “JSP Include” options.

Leave the remaining fields blank and the remaining options unselected.

- 6 Click Next.
Enter “ProxyAuth” for the Element Name and click Finish.
- 7 In the Portlet Builder window, click Save As. When the Finish window displays, complete these options:
 - Portlet name – enter “ProxyAuth”.
 - Active – select this option.
 - iFrame – select this option.
 - Category – select Business.
 - Subcategory – select Investing.
 - Refresh Interval – choose Daily.
 - Default Size – choose Large.
 - Show Last Refresh – select this option.
 - Roles – click to select PortalUser in the Available Roles list, then click Add to move that role to the Assigned Roles list.Leave the remaining fields blank and the remaining options unselected.
- 8 Click Finish, then click OK in the confirmation dialog box
- 9 Click Close to exit the window.
- 10 At the Portlet Manager window, select New from the Status menu. A listing for the new portlet displays in the detail view.
- 11 Right-click the portlet listing in the detail view and select Approval Statusn | Approved.

❖ **Creating a page and adding the portlet to it**

- 1 Select Build | Pages from the Portal Studio left pane.
- 2 Click the New icon on the Page Manager tool bar. The Page Builder window displays.
- 3 Click Add. The Universal Find Search window displays.
- 4 Enter the portlet name—in this sample, ProxyAuth—and click Search. The portlet displays in the Results pane.
- 5 Verify that the Active option is selected, click the 100% layout (far right icon), then click Save As. The Save Page window displays.
- 6 Complete the fields and options:

- Name – enter “ProxyAuthPage”.
- Type – select Catalog.
- Roles – select PortalUser and StudioAdmin from the Available Roles list and click Add to move the role to the Assigned Roles list and define what roles can access this page. See the *Enterprise Security Administration Guide* for more information on security roles.

Note User accounts assigned the roles in the Assigned Roles list are the only user accounts that can access this page.

- 7 Click OK, then click Close to exit the Page Builder.
- 8 At the Page Manager window, select New from the Status menu. A listing for the new page displays in the detail view.
- 9 Right-click the page listing in the detail view and select Status | Approved.

Creating proxy authentication information

When the new page displays in Portal Interface, the contents of the new portlet indicate that the current user does not have any proxy authentication information for the target back-end application. This section describes how to create the proxy authentication information, define the back-end-application asset type, and represent the new portlet as an asset in the Enterprise Security framework.

Note If you create proxy authentication information at the asset level, any user who has READ permission on the asset—through any role—can use this information to connect to the back-end application. You should either limit the users with access rights to the back-end application, or carefully select the users to whom you grant READ permission to the back-end application.

❖ Defining the asset

- 1 Log in to Portal Studio as a user with the Portal Security Officer (PSO) role. If you accepted the defaults during EP installation, the user name is “pso” and the password is “123qwe”.
- 2 In the Portal Studio window, select Administer | Organizations, then select Assets under the Organization Manager.
- 3 Click New on the toolbar.

- 4 When the Create Asset window displays, complete the fields:
 - Asset Name – enter “Application_1”.
 - Asset Type – select “Unspecified” from the drop-down list.
 - Description – enter “ProxyAuth Application”.
- 5 Click OK.

Now that you have created the asset, you must specify how to protect access to that asset. For this example, you will provide proxy authentication information for accessing the asset at the role level.

Warning! Grant READ permissions on assets to all users in the PortalUser role. If you do not have READ permission on an asset, proxy authentication information is not returned, regardless of whether you set up proxy authentication information at the asset, role, or subject level.

❖ **Creating proxy authentication information**

To define a user's proxy authentication information for an asset:

- 1 Log into Portal Studio with a user that has the Portal Security Officer (PSO) role. If you accepted the defaults during EP installation, the user name is “pso” and the password is “123qwe.”
- 2 In the Portal Studio window, select Administer | Organizations, then select Assets under the Organization Manager.
- 3 Right-click Application_1 asset and select Edit. The Edit Asset window displays.
- 4 Make a note of the AssetDN. You may want to write it down or copy it using Ctrl+C.
- 5 Connect to Jaguar Manager and select Servers | Jaguar | Installed Web Applications | SSO.
- 6 Right-click the SSO Web application in the right pane and select Edit JSP. The Edit JSP SSO window displays.
- 7 Locate the following line of code:

```
String asset = "a1=Application_1,o=sybase,c=us";
```
- 8 Edit the value of the AssetDN (inside the quotes) to match the value of the Application_1 asset you created.
- 9 Select File | Save and click OK.

- 10 Select File | Exit to close the Edit JSP SSO window.
- 11 Right-click the SSO Web application in the left pane and select Refresh.
- 12 Select File | Exit to close Jaguar Manager.

❖ **Granting permissions to the asset**

- 1 Log into Portal Studio with a user that has the Portal Security Officer (PSO) role. If you accepted the defaults during EP installation, the user name is “pso” and the password is “123qwe.”
- 2 In the Portal Studio window, select Administer | Organizations, then select Assets under the Organization Manager.
- 3 Right-click Application_1 in the detail view and select Manage Access Permission. The Manage Access Permission on Asset window displays.
- 4 Select PortalUser from the list of roles. Select READ from the Available Permissions list and click Add to move it to the Assigned Permissions list. Confirm that the newly granted permission displays in the Access Permissions Granted On The Asset list at the bottom of the window.
- 5 Click OK to save the granted permissions.

❖ **Create a proxy authentication information**

- 1 Log into Portal Studio with a user that has the Portal Security Officer (PSO) role. If you accepted the defaults during EP installation, the user name is “pso” and the password is “123qwe.”
- 2 Select Administer | Organizations, select Users under the Organization Manager, and right-click the user for whom you want to create access to proxy authentication and select Proxy Authentication.

Note If you have not created users yet, create a new user and assign them the PortalUser role. then click New on the toolbar. Refer to the Portal Studio online help—click Help in the Create New User window—or the *Enterprise Security Administration Guide* for details

- 3 When the Manage User Proxy Authentication Information window displays, click New, then click Find in the Create User Proxy Authentication Information window.
- 4 Select Application_1 and click OK.
- 5 Enter the following information to complete the user proxy authentication information:

- Service URL – leave this field blank.
 - User Name – enter the “ProxyUser”.
 - Password – enter “ProxyPassword”.
 - Verify Password – enter “ProxyPassword” again.
- 6 Click OK. The Manage User Proxy Authentication Information window displays the new proxy authentication information entry.
 - 7 Click OK to exit the window.

You have finished creating the role-level proxy authentication information. Anyone who has the PortalUser role and uses *sso.jsp* to access the back-end application authenticates to that application using the user name and password provided in this last procedure.

Creating the single sign-on JSP

For demonstration purposes, *sso.jsp* looks up proxy authentication information for the current user and prints the user name, password, and URL from that proxy authentication information record to the JSP.

Here is the sample *sso.jsp*:

```
<%@ page import="javax.servlet.*"%>
<%@ page import="java.security.*"%>
<%@ page import="javax.servlet.http.*"%>
<%@ page import="javax.naming.*"%>
<%@ page import="javax.rmi.PortableRemoteObject"%>
<%@ page import="com.sybase.ep.security.management.*"%>
<%@ page import="com.sybase.ep.security.sessionsvcs.*"%>
<%
String asset = "a1=Application_1,o=sybase,c=us";
// You can change the assetDN to something more readable using
// Enterprise Security. For example, if there is only one asset
// in your system with the name "Application_1", then you can
// change the AssetDN to "Application_1", and correspondingly
// String asset = "Application_1";
//in this code.

Principal p = null;
String user = null;
String pwd = null;
String backendURL = null;

    try
```



```

{
    //Get the user principal
    p = request.getUserPrincipal ();
    //System.out.println("SSO user principal is: " + p);
    Context ctx = new InitialContext();

    //Get the EP PortalSession object to get the
    //Proxy authentication information
    PortalSessionHome psHome = (PortalSessionHome)
        PortableRemoteObject.narrow(
            ctx.lookup("com.sybase.ep.security.sessionsvcs/PortalSession"),
            PortalSessionHome.class);
    PortalSession ps = psHome.findByCurrentSession();
    ProxyAuthenticationInfo pai = ps.findProxyAuthenticationInfo(asset);
    //Get the username
    user = pai.authenticationName;
    //Get the password
    byte[] password = pai.authenticationCredential;
    pwd = new String(password);
    backendURL = pai.serviceURL;

    //System.out.println(Asset " + asset + " provided proxy auth user: "
    //+ user + " with password: " + pwd
    //+ " and extraURL info: " + backend URL);
}
catch (Exception e)
{
    e.printStackTrace();
}
%>
<h1 align="center"> Single Sign On Example </h1>
SSO user Principal is <%=p.getName()%>

<P> Proxy Authentication info for Asset: <%=asset%>
<P> User name: <%=user%>
<P> Password: <%=pwd%>
<P> URL: <%=backendURL%>

```

In this example, the asset variable “Application_1” matches the asset created using the Enterprise Security Manager, and the complete DN (distinguished name) for the asset is provided.

❖ Running the JSP

- 1 Log in to Portal Interface.
 - a Open a Web browser, and enter:

`http://machine_name.domain:8080/onepage/index.jsp`

where *machine_name* is the name of the machine where the onepage Web application is installed and *domain* is the network domain in which the machine is registered. This displays the Portal Login page.

- b Enter your member name and password, then click Sign In.
- 2 Click the Add Page icon.
- 3 In the Add Page window, select the page containing the SSO portlet that you created (ProxyAuthPage), and click Done.
- 4 The ProxyAuthPage (which contains the ProxyAuth portlet) displays. When this JSP is executed for an authenticated user who has the PortalUser role and READ access to the Application_1 asset, it prints:

```
SSO user Principal is
uid=userID, cn=UserName
o=Sybase, c=us
```

```
Proxy Authentication info for Asset:
a1=Application_1,o=Sybase,c=us
```

```
User name: proxyUser
Password: proxyPasswordURL:
```

This output differs based on the proxy authentication information Enterprise Security determines is appropriate for the current user.

Creating single-sign-on portlets with Web applications

This section describes how to create single sign-on (SSO) portlets with Web applications running within the same Web container as Portal Interface.

All secured Web applications running inside Portal Interface as portlets are not forced to re-authenticate if the user has already been authenticated to the portal. Not forcing re-authentication requires that the secured Web application must run in the same container as Portal Interface and the user must have the required roles to access the secured Web application.

To create a single-sign on portlet:

- 1 Create a secured Web application with security constraints specified in the deployment descriptor file—*web.xml*.

- 2 Deploy the secured Web application to the container running Portal Interface.
- 3 Create a single-sign on JSP portlet in Portal Studio and deploy it in Portal Interface.

Creating a secured Web application

For this example, create a Web application with two JSP's—*sso.jsp* and *authorized_error.jsp*. This sample secured Web application consists of these two files and the *WEB-INF/web.xml* that defines the Web application to the Web container.

Deployment descriptor

To have a secured Web application, you need to declare security constraints in the *web.xml* deployment descriptor. You can use any supported authentication method; for this example, we used BASIC.

This is what the *web.xml* file looks like:

```
<?xml version="1.0" encoding="UTF-8"?><!DOCTYPE
web-app PUBLIC "-//Sun Microsystems, Inc.//DTD Web
Application 2.3//EN" 'http://java.sun.com/dtd/web
app_2_3.dtd' ><web-app>

<display-name>ssoapp</display-name>
<servlet>
<servlet-name>SSO</servlet-name>
<display-name>SSO</display-name>
<jsp-file>/sso.jsp</jsp-file>
</servlet>

<session-config>
<session-timeout>30</session-timeout>
</session-config>

<error-page>
<error-code>403</error-code>
<location>/authorize_error.jsp</location>
</error-page>

<error-page>
<error-code>401</error-code>
<location>/authorize_error.jsp</location>
```

```
</error-page>

<security-constraint>
<web-resource-collection>
<web-resource-name>ssoZone</web-resource-name>
<url-pattern>/*</url-pattern>
<http-method>GET</http-method>
<http-method>POST</http-method>
</web-resource-collection>

<auth-constraint>
<description>Role required to access the sso
jsp</description>
<role-name>PortalUser</role-name>
</auth-constraint>
<user-data-constraint>
<transport-guarantee>NONE</transport-guarantee>
</user-data-constraint>
</security-constraint>

<login-config>
<auth-method>BASIC</auth-method>
<realm-name>SSO App</realm-name>
</login-config>

<security-role>
<description>Whatever</description>
<role-name>PortalUser</role-name>
</security-role>
</web-app>
```

This code secures a Web application—ssoapp—with *sso.jsp*. This JSP is a protected Web resource.

Programming notes

- To access *sso.jsp*, users must have the PortalUser role. If an authenticated user does not have the PortalUser role, the container responds by presenting *authorize_error.jsp* to the user.
- If the portal receives an unauthenticated request for *sso.jsp*, the container uses BASIC authentication to authenticate the user before checking to see if the user has the PortalUser role.
- Since “/*” is used in the URL, all resources—if there is more than one JSP—under the ssoapp Web application are protected.

- If you use roles that are not in the Portal Studio Web container, be sure to follow the Web container's documentation to add the required roles and add authorized portal users to those roles. You must also add the required roles in Portal Studio.

The protected JSP

`sso.jsp` displays the user principal name. You can place any secured content there.

```
<%@ page import="javax.servlet.*"%>
<%@ page import="java.security.*"%>
<%@ page import="javax.servlet.http.*"%>
<%
Principal p = null;
    p = request.getUserPrincipal();
%>
<h1 align="center"> Single Sign On Example</h1>
SSO user Principal Name is <%=p.getName()%>
```

Deploying the secured Web application

Building the deployable WAR file

The directory structure in your development environment should look like this:

UNIX:

```
../temp/sso.jsp
../temp/authorized_error.jsp
../temp/WEB-INF/web.xml
../temp/WEB-INF/lib/<JAR files your Web application might use>
```

Windows:

```
c:\temp\sso.jsp
c:\temp\authorized_error.jsp
c:\temp\WEB-INF\web.xml
c:\temp\WEB-INF\lib\<JAR files your Web application might use>
```

Create the deployable Web application archive (WAR) file using the `jar` command from the location of your Java JDK:

```
cd <temp>
jar -cf /tmp/ssoapp.war *
```

```
cd <temp>
jar -cf \temp\ssoapp.war *
```

This creates the *ssoapp.war* deployment file in */tmp* on UNIX and in *\temp* on Windows.

Deploying *ssoapp.war*

To deploy the *ssoapp.war* Web application, follow the Web container's documentation.

If you are using EAServer, use either Jaguar Manager or jagtool to deploy *ssoapp.war* into EAServer and make it an active application. For example, if you use jagtool, enter:

```
jagtool deploy -type war <path to ssoapp.war>
```

Testing the *sso.jsp* as a standalone Web application

To test the *sso.jsp*, enter the following in your browser's URL:

```
http://<yourhost>:<port>/ssoapp/sso.jsp
```

You should see a user name and password login screen if you have not previously logged in through your browser.

To see the *sso.jsp* page, enter a valid user name and password for a user with the PortalUser role. You should see the *sso.jsp* page.

Creating a single sign-on JSP portlet

Creating the SSO sample JSP portlet

Note This procedure assumes that you have already deployed *ssoapp.war* into the Web container.

- 1 Launch Portal Studio. In a Web browser, enter
`http://HOSTNAME.DOMAIN:8080/onepage/index.html`.
- 2 Log in to Portal Studio and select Build | Portlets.
- 3 Click the down arrow beside the New icon and select Aggregated Portlet.

- 4 In the Choose Portlet Type window, select JSP and click Next.
- 5 In the JSP Element Definition window, select Use Web Application, enter the following for *ssoapp.war*, then click Next.
 - a WAR File To Upload – leave this field blank.
 - b WAR File– *ssoapp.war*
 - c Web App Display Name – *ssoapp*
 - d Initial Resource – */sso.jsp*
 - e Web App Qualified URL's – select this option.
 - f Single-Sign On Required– select this option.
- 6 When the Finish window appears, enter the following:
 - Portlet Name – enter *sso sample*
 - Active/Iframe/Editable – select these options. Leave the No Popup option unselected.
 - Inline Portlet Clickthru – (only available when you select iFrame) select this option to enable links in this portlet to display inline; that is, links will not display in a separate pop-up window.
 - Title URL – leave blank.
 - Help URL – leave blank.
 - Category/Subcategory – accept the defaults.
 - Refresh Interval – accept the default.
 - Default Size – accept the default.
 - Shared By – leave blank.
 - Show Last Refresh – accept the default.
 - J2EE Roles – select roles from the Available Roles list and add them to the Assigned Roles list to define what roles can access this portlet.

Note User accounts with the roles in the Assigned Roles list are the only user accounts that can access this portlet.

 - Description/Notes – add an optional description or note.
- 7 Click Finish.

- 8 When you return to the Portlet Manager window, select New from the Status menu. You should see the SSO sample portlet listed in the detail view.
- 9 Right-click “SSO sample” and select Approval Status | Approved.
- 10 Select Approved from the Status menu and verify that the SSO sample portlet displays in the Approved detail view.

Creating a new page for the secured JSP portlet

- 1 Select Build | Pages from the Portal Studio left pane.
- 2 Click the New icon.
- 3 When the Page Builder window displays, click Add button and click on search. SSO sample appears on the right-hand side.
- 4 Highlight SSO sample and click Add.
- 5 When the Search window appears, click Search. The SSO sample portlet displays in the Results pane, along with other existing portlets.
- 6 Click “SSO sample” to select that portlet, then click Add. The portlet shows up in the Page Builder on the representation of the new page you are constructing.
- 7 Click Save As.
- 8 In the Save Page window, enter the following, then click OK.
 - Name – enter `SSO Sample`.
 - Type – select Catalog.
 - Active – select this option.
 - Reg. Page Catalog – select this option.

The number you enter in the numeric (#) text field determines the position of the page in the list of available pages.

 - Page Catalog – do not select this option.
 - J2EE Roles – select roles from the Available Roles list and add them to the Assigned Roles list to define what roles can access this page.

Note User accounts with the roles in the Assigned Roles list are the only user accounts that can access this page.

- 9 Click Close to exit the Page Builder window.
- 10 When you return to the Page Manager window, select New from the Status menu. You should see the SSO Sample page listed in the detail view.
- 11 Right-click “SSO Sample” and select Status | Approved.
- 12 Select Approved from the Status menu and verify that the SSO Sample page displays in the Approved detail view.

Adding the page to a user’s page group

- 1 Launch Portal Interface. In a Web browser, enter
`http://HOSTNAME.DOMAIN:8080/onepage/index.jsp`
- 2 Log in to Portal Interface.
- 3 Click Add Page
- 4 If you have the required role, you see the SSO Sample page in the premade page list. Select that page and click Done.

You should see the secured Web application as a portlet without having the portal reauthenticate your role.

Creating single-sign-on portlets

Note Creating single-sign-on portlets with EP and Enterprise Security is supported only in the Enterprise Edition.

This section describes how Enterprise Portal is integrated with J2EE container security and how to use Enterprise Security to provide single-sign-on access to back-end systems via proxy authentication.

Creating a secured Web application

Use a development environment to create a Web application that contains the JSPs that you want to access your back end application.

For this example, create a Web application with one JSP—*ProxyAuth.jsp*. This sample secured Web application consists of *ProxyAuth.jsp* and the *WEB-INF/web.xml* that defines the Web application to the Web container.

The *web.xml* file looks like this:

```
<?xml version="1.0" encoding="UTF-8"?>

<!DOCTYPE web-app PUBLIC "-//Sun Microsystems,
Inc.//DTD Web Application 2.3//EN"
'http://java.sun.com/dtd/web-app_2_3.dtd'>

<web-app>
<display-name>ProxyAuth</display-name>
<servlet>
<servlet-name>ProxyAuth</servlet-name>
<display-name>ProxyAuth</display-name>
<jsp-file>/ProxyAuth.jsp</jsp-file>
</servlet>

<session-config>
<session-timeout>30</session-timeout>
</session-config>

<security-constraint>
<web-resource-collection>
<web-resource-name>ProxyAuthZone
</web-resource-name>
<url-pattern>/*</url-pattern>
<http-method>GET</http-method>
<http-method>POST</http-method>
</web-resource-collection>

<auth-constraint>
<description>Role required to access the ProxyAuth
jsp</description>
<role-name>PortalUser</role-name>
</auth-constraint>

<user-data-constraint>
<transport-guarantee>NONE</transport-guarantee>
</user-data-constraint>
</security-constraint>

<login-config>
<auth-method>BASIC</auth-method>
<realm-name></realm-name>
</login-config>

<security-role>
```

```

<description>Whatever</description>
<role-name>PortalUser</role-name>
</security-role>

</web-app>

```

The *web.xml* file is saying “here is a Web application named ‘ProxyAuth’ with a single JSP in it”. The JSP is a protected Web resource; that is, users that want to execute the JSP must have the PortalUser role. When the container receives an unauthenticated request for this JSP, the container should use BASIC authentication to validate the user before checking to see if the user has the PortalUser role.

Building the deployment file

The directory structure in your development environment should look similar to this:

UNIX:

```

.../temp/ProxyAuth.jsp
.../temp/WEB-INF/web.xml
.../temp/WEB-INF/lib/<JAR files your Web application might use>

```

Windows:

```

c:\temp\ProxyAuth.jsp
c:\temp\WEB-INF\web.xml
c:\temp\WEB-INF\lib\<JAR files your Web application might use>

```

- 1 Create the deployable Web application archive (WAR) file using the `jar` command from the location of your Java JDK:

UNIX:

```

cd <basedir>
jar -cf /tmp/ProxyAuth.war *

```

Windows:

```

cd <basedir>
jar -cf \temp\ProxyAuth.war *

```

This creates the *ProxyAuth.war* deployment file in your */tmp* on UNIX and in your *\temp* directory on Windows.

- 2 Copy the *security_stubs.jar* file to the *WEB-INF/lib* directory on UNIX or the *WEB-INF\lib* directory on Windows so the ProxyAuth JSP can use Enterprise Security to access proxy authentication information.

Deploying *ProxyAuth.war*

To deploy the *ProxyAuth.war* Web application, follow the Web container's documentation.

If you are using EAServer, use either Jaguar Manager or jagtool to deploy *ProxyAuth.war* into EAServer and make it an active application. For example, if you use jagtool, enter:

```
jagtool deploy -type war <path to ProxyAuth.war>
```

Testing *ProxyAuth.jsp* as a standalone Web application

To test the *ProxyAuth.jsp*, enter the following in your browser's URL:

```
http://<HOSTNAME>:8080/ProxyAuth/ProxyAuth.jsp
```

You should see a user name and password login screen if you have not previously logged in through your browser. Enter a valid user name and password for an Enterprise Security user that has the Portal User role

To see the *ProxyAuth.jsp* page, enter a valid user name and password for a user with the PortalUser role. You should be able to see the *ProxyAuth.jsp* page.

Because you have not defined any proxy authentication information yet, you see several items that are "null."

Creating a catalog page with a portlet that executes *ProxyAuth.jsp*

The procedures in this section assume that you have already deployed *ProxyAuth.war* into the Web container.

❖ Creating the *ProxyAuth* sample JSP portlet

- 1 Launch Portal Studio. In a Web browser, enter

```
http://HOSTNAME.DOMAIN:8080/onepage/index.html
```

- 2 Log in to Portal Studio and select Build | Portlets.
- 3 Click the down arrow beside the New icon and select Aggregated Portlet.
- 4 In the Choose Portlet Type window, select JSP and click Next.
- 5 In the JSP Element Definition window, select Use Web Application, enter the following for *ProxyAuth.war*, then click Next.

- 1 WAR File To Upload – if you already deployed *ProxyAuth.war* in the Web container, you do not need to upload that file now. If you have not deployed *ProxyAuth.war*, specify the path to *ProxyAuth.war* and click Upload. There will be a success dialog when the upload has completed.
 - 2 WAR File– `ProxyAuth.war`
 - 3 Web App Display Name – `ProxyAuth`
 - 4 Initial Resource – `/ProxyAuth.jsp`
 - 5 Web App Qualified URL's – select this option.
 - 6 Single Sign-On Required– select this option.
 - 6 When the Finish window appears, enter:
 - Portlet Name – `ProxAuth`.
 - Active/Iframe/Editable – select these options. Leave the remaining option on this line unselected.
 - Inline Portlet Clickthru – (only available when you select iFrame) select this option to enable links in this portlet to display inline; that is, links will not display in a separate pop-up window.
 - Title URL – leave blank.
 - Help URL – leave blank.
 - Category/Subcategory – accept the defaults.
 - Refresh Interval – accept the default.
 - Default Size – accept the default.
 - Shared By – leave blank.
 - Show Last Refresh – accept the default.
 - J2EE Roles – select roles from the Available Roles list and add them to the Assigned Roles list to define what roles can access this portlet.
-
- Note** User accounts with the roles in the Assigned Roles list are the only user accounts that can access this portlet.
-
- Description/Notes – add an optional description or note.
- 7 Click Finish.

- 8 When you return to the Portlet Manager window, select New from the Status menu. You should see the ProxAuth sample portlet listed in the detail view.

Right-click “ProxAuth” and select Approval Status | Approved.

- 9 Select Approved from the Status menu and verify that the ProxAuth sample portlet displays in the Approved detail view.

❖ **Creating a new page for the secured JSP portlet**

- 1 Select Build | Pages from the Portal Studio left pane.
- 2 Click the New icon.
- 3 When the Page Builder window displays, click Add button and click on search. ProxAuth sample appears on the right-hand side.
- 4 Highlight ProxAuth and click Add.
- 5 When the Search window appears, click Search. The ProxAuth sample portlet displays in the Results pane, along with other existing portlets.
- 6 Click “ProxAuth” to select that portlet, then click Add. The portlet shows up in the Page Builder on the representation of the new page you are constructing.
- 7 Click Save As.
- 8 In the Save Page window, enter the following, then click OK.

- Name – enter ProxAuth
- Type – select Catalog.
- Active – select this option.
- Reg. Page Catalog – select this option.

The number you enter in the numeric (#) text field determines the position of the page in the list of available pages.

- Page Catalog – do not select this option.
- J2EE Roles – select roles from the Available Roles list and add them to the Assigned Roles list to define what roles can access this page.

Note User accounts with the roles in the Assigned Roles list are the only user accounts that can access this page.

- 9 Click Close to exit the Page Builder window.

- 10 When you return to the Page Manager window, select New from the Status menu. You should see the ProxAuth page listed in the detail view.
- 11 Right-click ProxAuth and select Status | Approved.
- 12 Select Approved from the Status menu and verify that the ProxAuth page displays in the Approved detail view.

❖ **Adding the page to a user's page group**

- 1 Launch Portal Interface. In a Web browser, enter:

```
http://HOSTNAME.DOMAIN:8080/onepage/index.jsp
```

- 2 Log in to Portal Interface.
- 3 Click Add Page.
- 4 If you have the required role, you see the ProxAuth page in the premade page list. Select that page and click Done.

The page should display, but the contents of the ProxyAuth portlet should indicate that the current user does not have any proxy authentication information for the target back end application.

Entering portal authentication data

In this example you represent the back-end application as an asset that needs to be protected by Enterprise Security.

❖ **Creating a new asset**

Define an asset that corresponds to back-end applications.

- 1 Log in to Portal Studio using “pso” for the user name and “123qwe” for the password and click OK.
- 2 Select Administer | Organizations in the left pane and select the organization for which you want to create the asset.
- 3 Select Assets from the organization's menu and click New. Complete the fields in the Create New Asset dialog box.
 - Asset Name – enter `Application_1`.
 - Asset Type – select `STUDIO.Applications`.
 - Description – enter a description.
- 4 Click OK.

Protecting the asset

Now that the asset has been created, you must provide proxy authentication information to access the asset at the role level.

Before any proxy authentication information can be returned for a user, you must grant READ permission on the asset. If the asset does not have READ permission, it does not matter what kind of proxy authentication information is set up at the subject, role, or asset level—no information can be returned for a user.

❖ **Creating user-based proxy authentication information**

Before you can create proxy authentication information, users and assets must exist in the enterprise environment. To define a user's proxy authentication information for an asset:

- 1 In the Manage User Proxy Authentication Information dialog box, click New. The Create User Proxy Authentication Information dialog box displays.

Enter:

- Asset DN – the distinguished name (DN) for the asset. To look up the DN, click Find.

In the Choose Asset dialog box, select the organization to which the asset belongs, highlight the asset name, then click OK. The asset DN displays in the Create User Proxy Authentication Information dialog box.

- Service URL – the connection request uses this URL to establish a connection with the selected asset.
 - User Name – the name used to log in to the asset specified by the URL.
 - Password – a valid password for the user name.
 - Verify Password – the same password.
- 2 Click OK to save your changes. The asset now has proxy authentication information associated with it.

Repeat this procedure for each asset for which you want to create proxy authentication information.

The role-level proxy authentication information has been created. The next section describes the *ProxyAuth.jsp* that uses the proxy authentication information.

single-sign-on JSP programming

ProxyAuth.jsp looks up proxy authentication information for the current user and prints the user name, password, and URL from that information to the JSP.

The *ProxyAuth.jsp* source file looks like this:

```
<%@ page import="javax.servlet.*"%>
<%@ page import="java.security.*"%>
<%@ page import="javax.servlet.http.*"%>
<%@ page import="javax.naming.*"%>
<%@ page import="javax.rmi.PortableRemoteObject"%>
<%@ page
import="com.sybase.ep.security.management.*"%>
<%@ page
import="com.sybase.ep.security.sessionsvcs.*"%>

<%
String asset = "a1=Application_1,o=Sybase,c=us";

//If you place that asset in the "root" organization,
//you can use this shortened naming.
// String asset = "Application_1";
Principal p = null;
String user = null;
String pwd = null;
String backendURL = null;

try
{
//Get the user principal
p = request.getUesrPrincipal();
//System.out.println("ProxyAuth user
principal is: " + p);
Context ctx = new InitialContext();
//Get the EP PortalSession object to
get the
//Proxy authentication information
PortalSessionHome psHome =
(PortalSessionHome)
PortableRemoteObject.narrow(

ctx.lookup("com.sybase.ep.security.sessionsvcs/
PortalSession"),
PortalSessionHome.class);
PortalSession ps=
psHome.findByCurrentSession();
```

```

ProxyAuthenticationInfo pai =
    ps.findProxyAuthenticationInfo(asset);
//Get the username
user = pai.authenticationName;
//Get the password
byte[] password =
    pai.authenticationCredential;
    pwd = new String(password);
    backendURL = pai.serviceURL;
//System.out.println("Asset " + asset +
"provided proxy auth user: "
//+ user + " with password: " + pwd
//+ " and extraURL info: " + backendURL);
}
catch (Exception e)
{
    e.printStackTrace();
}
}
%>
<h1 align="center"> Single Sign On Example </h1>
ProxyAuth user Principal is <%=p.getName()%>
<P> Proxy Authentication info for Asset: <%=asset%>
<P> User name: <%=user%>
<P> Password: <%=pwd%>
<P> URL: <%=backendURL%>

```

Programming notes

- The asset variable matches the asset you created using the Security Manager:

```

// String asset = "Application_1";

```
- In the above example provides the complete distinguished name (DN) for the asset.
- When this JSP is executed for an authenticated user with the PortalUser role and READ access to the Application_1 asset, the ProxyAuth portlet content displays.

Viewing the ProxyAuth portlet

- 1 Launch Portal Interface. In a Web browser, enter

```

http://HOSTNAME.DOMAIN:8080/onepage/index.jsp

```

- 2 Log in to Portal Interface with your user name and password.
- 3 Click Add Page.
- 4 If you have the required role, you see the ProxAuth page in the premade page list. Select that page and click Done.

The output that displays in the ProxyAuth portlet varies for different users, based on the information Enterprise Security decides is appropriate for the current user to view.

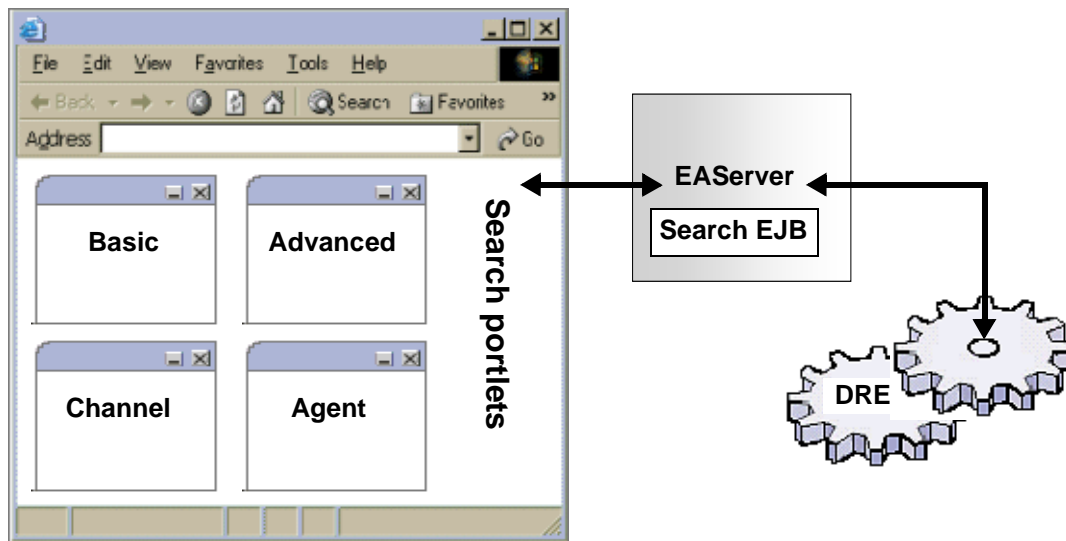
Using the sample search portlets

Note This section applies only to the EP Enterprise Edition; that is, the search portlets are not available in the Demo or Information Edition of EP.

This document describes the search portlets that are available in Enterprise Portal. You can use these portlets to execute actual searches. The search portlets can also be used as examples to create your own search portlets or to customize the search portlets provided with EP.

Enterprise Portal uses a search engine—the Dynamic Reasoning Engine (DRE)—to perform concept-based searches. EP provides six search portlets implemented as JavaServer Pages (JSPs) that access the DRE via the Portal Search Enterprise JavaBean (EJB). These search portlets allow you to execute a single search across multiple databases and display the combined results to the user.

Figure D-1: Portal search architecture



Dynamic Reasoning Engine

The DRE uses high-performance matching algorithms to identify patterns within text that can then be matched with similar patterns from other data sources. The identified patterns, also known as concept agents, can facilitate finding similar, relevant patterns by learning which search hits the user prefers.

For more information about setting up and configuring the server and automatic indexing utility, see Appendix C, “Setting Up Automatic Indexing and Search Functionality.”

Search portlets call search EJB functions and get queries executed in the DRE. These search portlets allow you to execute a search across multiple sites at one time and to show the search results from each site to users.

After you install and configure Enterprise Portal, verify that the DRE is properly configured and running by accessing the following URL from your browser:

```
http://<DREHost>:<DREPort>/qmethod=v
```

where *DREHost* is the name of the machine on which the DRE was installed, and *DREPort* is the port on which the DRE is configured to listen. By default, the DRE is configured to listen on port 8500 for Solaris and port 2000 for Windows. You should see a window similar to the one shown in Figure D-2.

Figure D-2: DRE test results

Autonomy AXE Version 4.3.10 Build 37203

Query port: 2000, Index port: 2001

835 documents
 1150 document sections
 1150 committed document slots
 10492 Terms
 Process ID: 932
 Directory: D:\Sybase\PortalSearch\Engine

1 Database(s):

Name	Expiry Hours (Default)	Expiry Action	Documents	Document Sections
0: Sample	-	-	835	1150

1 Language Type(s):

Language Type	Documents	Document Sections
english	835	1150

Threads:

0 : Indexer idle
 1 : Index Server Port 2001 (Waiting for connection)
 3 : File Command Thread waiting
 4 : GET /qmethod=v HTTP/1.1
 5 : Query Server idle.
 6 : Query Server idle.
 7 : Query Server idle.

Search EJB

During the EP installation, the search Enterprise JavaBean (EJB) is installed into EAServer. To verify that the bean has been installed and configured correctly, use Jaguar Manager to connect to EAServer and verify the existence of PortalSearchServices of the configured EAServer.

❖ **Verifying the search EJB installation**

- 1 Start Jaguar Manager and connect to the Jaguar server that has the search EJB installed. For example, on Solaris, enter:

```
% cd $SYBASE/EAServer/bin
% jagmgr
```

On Windows, select Start | Programs | Sybase | EAServer 4.2.2 | Jaguar Manager.

- 2 When Sybase Central displays, select Tools | Connect | Jaguar Manager.
- 3 When the Login dialog box displays:
 - Username – enter “jagadmin”.
 - Password – if you have not changed the “jagadmin” password, leave this field blank. If you have changed the password for “jagadmin,” enter that password here.
 - Host Name – enter the name of the machine or host name for your installation.
 - Port Number – accept the default.
- 4 Click Connect.
- 5 Select Jaguar Manager | Servers | Jaguar | Installed Packages to verify that the Portal SearchServices package is installed.

Search portlets

Enterprise Portal includes a set of portlets that take advantage of the various PortalSearch EJB application interfaces (APIs). These portlets are Basic Search, Advanced Search, Agent Search, Manage Agents, Channel Search, and Manage Channels.

- Basic Search – performs simple searches against the DRE or the Internet. The query you enter is sent to the DRE search engine.
- Advanced Search – provides a detailed search. Queries can incorporate details such as word or phrase filters, proper name, date, and document title and summary. You can direct queries to specific sets of databases, and you can limit the results by the number of documents returned and minimum relevancy. Result sets can be sorted by date or relevancy.

- Agent Search – provides an interface to the DRE agent functionality, known also as automated search processes. You train an agent by defining a query. Users create agents and those agents cannot be shared with other users. The agent search functionality is divided into two portlets:
 - Agent Search – displays the list of available agents and the results of an executed agent.
 - Manage Agents – used to create, delete, rename, edit, and copy agents; also used to run and execute agents.
- Channel Search – channels are groupings of queries. These groupings are defined by the Portal Administrator, and contain subchannels—stored queries similar to agents—that are relevant to the channel. For example, you might have a channel called Company Information with subchannels for Finance, HR, and Administration.

The Portal Administrator creates and edits channels and subchannels. Access to a channel is limited by the roles assigned to the channel.

Channel Search functionality is divided into two portlets:

- Channel Search – lists the channels and subchannels available to the user according to the user’s roles. The display is divided into two sections—a list of channels and a list of subchannels. When a user selects a channel, the subchannel list is populated with the subchannels currently available to that user. The user executes the subchannel by selecting it.
- Manage Channels – access to this portlet is limited to users with an Admin role—any role that contains “Admin” in its name. The Manage Channels portlet functionality includes:
 - Channel functions – used to create, rename, delete, and assign roles to a channel.
 - Subchannel functions – used to create, rename, delete, edit, and perform subchannel searches.

Testing search as a standalone Web application

To test each portlet’s JSP file as a standalone Web application, enter the following URLs in your browser’s Address or Location field. You can change the default port 8080 to any real port number of the Web application server’s HTTP listener.

Basic Search:

```
http://<LOCALHOST>:8080/search/basicsearch/basicsearch.jsp
```

Advanced Search:

```
http://<LOCALHOST>:8080/search/advancedsearch  
/advancedsearch.jsp
```

Manage Agents:

```
http://<LOCALHOST>:8080/search/agent  
/agentcontrol.jsp?uid=<uid>
```

Manage Agents requires *uid*, which can be any string.

Agent Search:

```
http://<LOCALHOST>:8080/search/agent  
/listagent.jsp?uid=<uid>
```

Agent Search requires a *uid*, which can be any string.

Manage Channels:

```
http://<LOCALHOST>:8080/search/channel  
/managechannels.jsp?uid=<uid>&role=<role>
```

Manager channels requires *uid* and *role*. Use any string for *uid* and enter any role that uses "Admin" in its name for *role*; for example, "PortalAdmin."

Channel Search:

```
http://<LOCALHOST>:8080/search/channel  
/channellist.jsp?uid=<uid>&role=<role>
```

Channel Search requires *uid* and *role*. Use any string for *uid* and enter any role for *role*.

Using the Content Explorer sample portlet

Note The Content Explorer sample portlet can be used only with Internet Explorer versions 5.0, 5.5, and 6.0 on Windows and Netscape 7.0 on UNIX.

Also, when using the Content Explorer portlet with Internet Explorer, you must select Tools | Internet Options in Internet Explorer. In the Temporary Internet Files section of the dialog box, click Settings and select the Every Visit to the Page option (the default is Automatically), click OK, then click OK again to exit the dialog box.

The Content Explorer is a basic content management tool for the creation and management of documents, shortcuts, and URLs. It uses folders and collections (sets of folders) to allow users to browse for the content to which they have access.

The Content Explorer portlet uses role-based security to ensure that end users see only the content to which they have access.

Depending on your role, you have access to either the administrator view (PortalAdmin) where you have full management capabilities, or to the user view (PortalGuest or PortalUser) where you can only browse for content to which you have access.

Sample data is included with the Content Explorer to help you understand how it works.

An administrator organizes the assets in a way that makes them more accessible to users of a collection. System administrators, sales, marketing, and customers can have different roles, and consequently different views of the Content Explorer.

Both collections and assets are created with roles that indicate access rights. If a user's role does not have access rights to an asset or collection, that asset or collection never displays in the Content Explorer. When a user or administrator logs in, the Content Explorer displays only the collections and assets their role has permission to access.

Users

Browsing is the ability to start at a known collection and navigate for relevant information.

Browsing collections

The user automatically sees the Browse Collections page if they do not have the PortalAdmin role. When a user browses, Content Explorer:

- Queries the database for the user's visible collections
- Displays visible collections

Note The Browse Collections window does not appear for administrators because the administrator browses the Content Explorer using the Manage Collections window.

There are two controls in the Browse Collections portlet. On the left is the tree view, which displays the available collections to browse. On the right is the list view, which displays the contents of the collection selected in the tree view.

Tree view

The root of the tree view displays a list of collections sorted by name. Each collection contains a list of assets sorted by name.

There are two icons next to each object in the tree view:

- The icon to expand or collapse the object
- The icon that specifies the object type; for example, folder, filing cabinet, and so on

Selecting the object type icon or the name of an object displays that object's contents in the list view.

List view

Several object types display in the list view:

- Folder – double-click a folder in the list view is the same as double-clicking a folder in the tree view.
- Document – depending on the browser, double-click a document to:
 - Launch the application associated with the document; or
 - Prompt for a location to store the file if an application is not associated with the document; or
 - Asks if you want to store the file or launch an application.
- URL and a shortcut – click a URL or shortcut to launch the associated application. If the URL is an HTML document, a new browser window appears with the page loaded. If the URL is a “mailto:” URL, a mail utility launches.

Administrator functions

Only administrators can catalog assets for users. Collection management allows an administrator to create, modify, delete, and control collections and the objects or assets the collections contain.

Managing collections

❖ Adding objects

- 1 In the tree view, select the collection or folder where you want the object created and click New.

To create an object at the root of the tree view, select the Content Explorer node and click New.

- 2 Depending on the type of object, a portlet appears where you enter that object's values.
 - Collection – the Add a Collection portlet appears where you enter the collection's unique name and access rights.
 - Asset – the New Asset portlet appears where you enter an asset name.

Note Depending on the asset type (document, URL, and so on), you are asked to enter additional values; such as, author, location or path, search query, and so on.

- 3 Do one of the following:
 - Click Cancel to abort the operation.
 - Click Done to complete the operation.

You return to the Manage Collections page.

❖ Editing objects

Editing a collection or asset allows you to change the same information that you entered when you created the object. You can edit any object except the Content Explorer node.

- To edit a collection or asset, select the object and click Edit.

❖ Deleting objects

Deleting a collection or asset removes the selected object and all of that object's descendents. For example, if you delete a folder, all subfolders within that folder are also deleted.

- 1 To delete a collection or asset, select the object you want to delete and click Delete. You can delete any object except the Content Explorer node.
- 2 A confirmation dialog box displays asking you to confirm that you really want to delete the selected object.

- Click Yes to delete the object and any objects that exist within the object.
- Click Cancel or No to abort the operation.

❖ **Moving objects**

When you move an object, you change that object's hierarchical association. If the new association is in a different collection, the selected object and all object's contained within that object reflect that the objects are in the new collection.

The object you are moving is the "source." The object into which the source is moved is the "destination." All objects are moved using cut and paste functionality.

- 1 Select the object you want to move and click Cut Selected.

Information about the selected object is moved to an internal paste buffer.

- 2 Select the new object into which you are moving the source object and click Paste.

Once the paste operation is completed, the selected object is removed from the internal paste buffer.

Restrictions

When you move objects, note that:

- If you move an object to a new collection, that object's access rights may be incompatible with the destination collection's access rights.
- If the source and destination are within the same object, the move fails.
- If the destination object is located within the source, the move fails.
- If the destination object has the same name as the source, the move fails.
- If objects within the source are modified before the paste as been completed, only the source and the objects that currently reside within it are moved.
- If the source is deleted after selecting Cut, the paste fails.
- If the source is modified after selecting Cut, the new object is pasted.

Using the Address Book sample

The Address Book sample application creates a personalized list of contacts for portal users. You can:

- Separate an address book into categories (for example, business contacts, personal friends, home improvement contacts, and so on), then assign address book entries to specific categories.
- Update or delete address book entries.
- Search for addresses using the last name, first name, or category, or view all addresses and filter them based on categories.
- Perform searches using partial information; for example, you can search for all the last names that begin with a specific character within a category.

The Address Book application is developed as a standard J2EE Web application. It is packaged in the *addressbook.war* file, which you can deploy as a JSP portlet into Portal Interface. The application includes security constraints that restrict access to its resources.

Building and deploying the Address Book portlet

❖ Deploying *addressbook.war* and creating a JSP portlet

- 1 Log in to Portal Studio as a user with access to all operations. For example:
 - Username – enter “pso”.
 - Password – enter “123qwe”
- 2 Select Build | Portlets from the Portal Studio left pane.
- 3 Click New on the Portlet Manager toolbar.
- 4 When the Portlet Builder displays, click the down arrow beside the Add button and select JSP Element. The JSP Element Definition window appears.
- 5 Click the Browse button and navigate to `%SYBASE%\PortalStudio\samples\` on Windows and `$$SYBASE/PortalStudio/samples` on UNIX. Select *addressbook.war*, and click Open.
- 6 Click Upload to upload the file to the machine where the portal Web application is installed on EAServer or Tomcat.

- 7 When the file has been uploaded successfully, the WAR File and Web App Display Name are automatically entered for you and you see a prompt confirming the upload was successful.
Click OK.
- 8 In the Initial Resource field, enter:
`index.jsp`
- 9 Keep the Web App Qualified URL option selected.
- 10 Click Deploy to deploy the WAR file. You see the Portal Studio – Deploy window.
- 11 Complete these fields:
 - EAServer/Tomcat – select the application server on which to deploy the WAR file.
 - Host – enter the host name of the machine on which the application server is installed.
 - Port – enter the local host port number. Enter “9000” for EAServer, “8080” for Tomcat.
 - Admin Name – enter the administrator login used to log in to the application server. For Tomcat, the user must have a “manager” role.
 - Admin Password – enter the password for the login name entered.
- 12 Click Deploy. When you see a prompt that the deploy operation succeeded, click OK.
- 13 Enter the Element Name and click Finish.
- 14 In the Portlet Builder, click Save As.
- 15 Complete the options on the Finish window.
- 16 Complete the new element’s properties:
 - Portlet Name – enter “Address Book”.
 - Active – this option is selected by default. Deselect the option if you do not want this portlet to be active. Only portlets with an approved status and marked as active are available to portal users.

- Iframe – select this option to display the portlet in an inline frame that can be placed inside of another HTML document or placed directly on a page with other elements flowing around it.

Note Unlike frames created using <FRAMESET> and <FRAME>, <IFRAME> creates a frame that sits in the middle of a regular nonframed Web page. <IFRAME> works like , only instead of putting a picture on the page, it puts another Web page on the page.

- No Popup – available only when you select iFrame. Select this option to enable links in this portlet to display inline; that is, links will not display in a separate pop-up window.
- Editable – whether the portlet can be edited by the users with the access roles to view the portlet.
- Secure – whether this is a secure portlet.
- Version – select this option to create a new version of the portlet each time you save it. Subsequent versions have new version numbers.
- Parameter – used to create an invisible parameter to which you can assign click-across events.

Note This property does not display in the Portlet Properties window that appears when you click Properties in the Portlet Builder.

- In Context – indicates that the click-across session needs to keep the HTTP client in loop.

Note In Context helps address one-click capture Yahoo e-mail portlet problems where there is a session value in the click-thru links and the existing solution does not work. There is no way to know when you need to use this option other than to try the normal click-thru then select this option if the normal process is not giving the desired results.

- Title URL – shows the URL of the content you captured.
- Help URL – enter the location of the online help file to associate with this portlet, or click Select to browse to and select the file.
- Content Type – select the type of content this portlet will display.

- Charset – select the character set to use in displaying this portlet’s content.
- Category/Subcategory – select a category and subcategory that best fit this portlet.
- Portlet Refresh Interval – select the interval at which you want this portlet’s content to refresh. The default is five minutes. The browser automatically makes a request for the portlet content when this interval runs down to zero.
 - The default is Daily, which translates into 1440 minutes.
 - Specify custom values by selecting Add New in the drop-down list. Specify this value minutes or seconds.
- Default Size – select “Show All.”
- Content Cache Interval – the interval at which you want the content of a portlet to be refreshed for the defined source. If this value is not Real Time, the content is held by the UWP cache for the duration of the specified interval. All requests made during this interval receive the cached content.
 - The default is Real Time, which means retrieve the portlet content from its source on every playback request.
 - Specify custom values by selecting Add New in the drop-down list. Select the minute/second drop-down list and enter a value in the text box.
- Show Last Refresh – display the latest refresh image when the portlet displays. By default, this option is turned “on.”
- Roles – select roles from the Available Roles list and add them to the Assigned Roles list to define what roles can access this portlet. See the *Enterprise Security Administration Guide* for more information on roles

Note User accounts with the roles in the Assigned Roles list are the only user accounts that can access this portlet.

- Description/Notes – add an optional description or note.
- 17 Click Finish. After a message displays that the portlet was saved successfully, click OK to return the Portlet Builder.

- 18 Select Status from the View By drop-down list, then click New on the Portlet Manager Status menu to see the new portlet in the detail view. All portlets are saved with a New status.
- 19 Right-click the Address Book listing in the detail view and select Status | Approved from the pop-up. When the confirmation prompt displays, click OK.
- 20 Right-click the Address Book listing in the detail view and select Active Status | Activate from the pop-up.

You have deployed the Address Book Web application. You have also created, saved, approved, and activated a JSP portlet in which to display the Address Book Web application. Next, add the portlet to a page.

❖ **Adding the JSP portlet to a page**

- 1 Select Build | Pages from the Portal Studio left pane.
- 2 Click New on the Page Manager toolbar. The Page Builder displays.
- 3 Click the layout icon that indicates the portlet will display on 100% of the page.
- 4 Click Add. The Search window displays.
- 5 In the Name field, type “Address Book” and click Search. A listing for the Address Book portlet displays in the Results pane.
- 6 Click the Address Book listing and click Add above the Results pane. In the Page Builder, you see Address Book in the portlet list in the left pane and a cell representing the portlet displays in the right pane.
- 7 Click Save As. The Save Page window displays.
- 8 Complete these options:
 - Name – enter “Address Book”.
 - Type – select Catalog from the drop-down list.
 - Active – select this option.
 - Roles – hold down the Ctrl key and select PortalAdmin, PortalUser, and StudioAdmin, then click Add.
- 9 Click OK.
- 10 When a prompt states that the page was saved, click OK.
- 11 Click Close to exit the Page Builder.

- 12 When you return to the Page Manager, select New from the Status menu. You see the new page listed in the detail view, marked as active.
- 13 Right-click the Address Book listing in the detail view and select Approval Status | Approved from the pop-up. When the confirmation prompt displays, click OK.
- 14 Select Approved from the Status menu. The page now displays in the approved status detail view.
- 15 When a prompt confirms that the user pages have been updated, click OK.
- 16 Log out of Portal Studio.

❖ **Creating a servlet alias**

- 1 Start Jaguar Manager and connect to the Jaguar server that has the search EJB installed.

For example, on Solaris, enter:

```
% cd $SYBASE/EAServer/bin
% jagmgr
```

On Windows, select Start | Programs | Sybase | EAServer 4.2.2 | Jaguar Manager.

- 2 When Sybase Central displays, select Tools | Connect | Jaguar Manager.
- 3 When the Login dialog box displays, enter:
 - Username – “jagadmin”.
 - Password – if you have not changed the “jagadmin” password, leave this field blank. If you have changed the jagadmin password, enter that password here.
 - Host Name – the name of the machine or host name for your installation.
 - Port Number – accept the default.
- 4 Click Connect.
- 5 Select Jaguar Manager | Servers | Jaguar | Installed Web Applications.
- 6 Right-click “addressbook” in the right pane and select Web Application Properties.
- 7 Select the Servlet Mapping tab.
- 8 Click Add and change the URL Pattern for “Address” servlet to:

```
/servlet/com.sybase.sample.addressbook.AddressBookServlet
```

- 9 Click OK to save your changes.
- 10 Right-click the Address Book Web application and select Refresh.
- 11 Select File | Exit to log out of Jaguar Manager.

❖ **Customizing *personalization.properties***

- 1 In any text editor, open *personalization.properties*, located in `%EAServer%\Repository\WebApplication\onpage\WEB-INF\classes\` on Windows and in `$EAServer/Repository/WebApplication/onpage/WEB-INF/classes/` on UNIX.

The current contents are:

```
connection_url = jdbc:sybase:Tds:changask:5000/portaldatabase
db_username = sa
db_password =
driver_classname = com.sybase.jdbc2.jdbc.SybDriver
```

- 2 Change the following:
 - `connection_url` – modify “changask:5000” to the host name and port number of the Portal Interface database.
 - `db_username` – enter the user name used to access the database.
 - `db_password` – enter the password for the user name you entered.
- 3 Save the file in the directory that started the EAServer process. If EAServer is started in `$JAGUAR/bin` (UNIX) or `%JAGUAR%\bin` (Windows), save the file there, then close the text editor.
- 4 Stop and restart EAServer.

Using the Address Book

When the Address Book application is running, you can perform a variety of functions from the portlet.

❖ **Adding the Address Book in Portal Interface**

- 1 Log in to Portal Interface.
- 2 Select Add Page.
- 3 When the Add Page window displays, select Add Pre-made Pages, select Address Book, and click Done.

- 4 You see the Address Book page and portlet.

Note The Address Book must display in an inline frame (iframe option). If this is not the case, click the edit hyperlink on the title bar of the portlet, select the Iframe display type, then click OK.

❖ **Creating a category**

Before you can add any address book entries, you must create at least one category.

- 1 Select Add a Category.
- 2 In the Add Category window, enter a category name and description. Do not use spaces in the name.

❖ **Creating an address book entry**

- 1 Select Add a New Contact.
- 2 In the Add Contact window, select the category to which you want to add the address, then enter the name, address, and telephone number information.
- 3 Click Save Contact to save the entry.

❖ **Updating an address book entry**

- 1 Select View all Contacts.
- 2 Find the address book entry by choosing the category and the first letter of the last name. Select the entry.
- 3 In the Address Details window, click Edit.
- 4 In the Edit Address entry window, update the information you want to change, and click Save.

❖ **Deleting an address book entry**

- 1 Select View all Contacts.
- 2 Find the address book entry by choosing the category and the first letter of the last name. Select the entry.
- 3 In the Address Details window, click Delete.

❖ **Search for addresses**

- 1 Select Search for Addresses.

- 2 In the Search window, enter the search criteria, and select Search.

You can search for addresses based on all or part of the first name, last name, or category. For example, if you enter `JOE` for the first name and `FR` for the last name, the application returns both Joe Francis and Joey Fritsch.

❖ **Filter addresses**

- 1 Select View all Contacts.
- 2 To view all your entries, select All Categories.

To view all the entries in a specific category, select the category.

To view all the last names that begin with a specific letter, select All Categories, then select the first letter of the last name.

To view everyone with last names that begin with a specific letter and that are in a specific category, select the category and the first letter of the last name.

For example, to see everyone in the Engineering category whose last name begins with 'J', select the category Engineering, and click 'J'.

Application files

The Address Book sample application includes the Java class files listed in Table D-1 and the JavaServer Page (JSP) files listed in Table D-2.

Table D-1: Java files

File name	Description
<i>AddressBookConstants.class</i>	Contains a list of constants that can be accessed from the Java code and the JSPs.
<i>AddressBookServlet.class</i>	The controller servlet for the application. All calls from the JSPs are directed to this servlet. The servlet determines the type of operation requested, invokes the appropriate data access methods, saves the results in <i>AddressBookJSPDataMediator.class</i> , and redirects the request to the appropriate JSP to render the data.

File name	Description
<i>AddressBook.JSPDataMediator.class</i>	Used to negotiate data between the JSPs and the Java code. An object of this class is stored in the session. The JSP gets the objects from the session to access the data and renders HTML windows.
<i>AddressBookDBAccess.class</i>	Responsible for all data access using the personalization APIs. See the comments in the Java code for more information on the individual methods of this class.
<i>AddressBookEntry.class</i>	The encapsulation of an individual address book entry. Additionally, this class contains support methods that ensure unique names when saving information using the personalization APIs.

Table D-2: JSP files

File name	Description
<i>addcategory.jsp</i>	Displays the window that allows you to add a category.
<i>addcontact.jsp</i>	Displays the window that enables you to create an address book entry.
<i>addressdetails.jsp</i>	Displays a read-only list of addresses. You can invoke this window from the Search Results window or the List Address window.
<i>confirmscreen.jsp</i>	When an address is entered, this file displays an address confirmation window.
<i>editcontact.jsp</i>	Displays a window that allows you to edit an address.
<i>error.jsp</i>	Displays error messages. The type of operation that caused the error determines the error message.
<i>index.jsp</i>	Displays the Welcome window, which allows you to: <ul style="list-style-type: none"> • Add a new contact • View all contacts • Add a category • Search for addresses
<i>listaddress.jsp</i>	Lists all the addresses in the address book based on the specified criteria.
<i>search.jsp</i>	Displays the Search window that allows users to search on first name, last name, or category.
<i>searchresults.jsp</i>	Displays the search results.
<i>success.jsp</i>	Notifies users when an operation has completed successfully.

File name	Description
<i>underconstruction.jsp</i>	A message page with one line of text, “Under Construction.”

Implementation details

Initially, the user ID (UID), page ID (PID), and address book title ID (TID) are passed to the initial resource page (*index.jsp*) in the request object. You can access these values using this code:

```
String uid = request.getParameter("uid");
String pid = request.getParameter("tileid");
String rid = request.getParameter("rid");
```

The `AddressBookServlet` class saves the UID, PID, and TID values in the session for subsequent calls.

See the JSP portlets documentation—

`%EASERVER%\Repository\WebApplication\onepage\docs\javadoc\index-all.html` on Windows and

`$EASERVER/Repository/WebApplication/onepage/docs/javadoc/index-all.html` on UNIX—for a list of variables that are accessible from pages through the request object.

Personalization APIs

The *addressbook.war* file includes the personalization APIs packaged in the *personalization.jar* file. The personalization APIs are a set of Java classes that implement the `com.onepage.db.personalization.PersonalDbInterface` interface.

Using the personalization APIs, the Address Book Web application can get and maintain information about portal users. The personalization APIs provide a convenient way for applications developed in Enterprise Portal to access databases and maintain personalized information about the users who are logged in. For example, you can develop a Web application with JSP pages and Java beans using the personalization APIs. You can then deploy the Web application into the application server running the portal. You can create portlets from the Web application, and add the portlets to pages for portal users.

Web applications developed using the personalization APIs have access to the personalization repository. You can develop personalized content for the users who access the application.

These API methods are included in the *personalization.jar* file:

- AddBinding(String UID, String PID, String TID, String name, String value)
- AddMultipleBinding(String UID, String PID, String TID, HashMap map, Boolean flag)
- RemoveBinding(String UID, String PID, String TID, String name)
- RemoveBinding(String UID, String PID, String TID, String [] array)
- GetBinding(String UID, String PID, String TID, String name)
- GetMultipleBindings(String UID, String PID, String TID)

Single sign-on

You can configure the Web application with security constraints that restrict direct access to the resources of the application. When a Web application user tries to access the Web application directly, the container prompts the user for a user ID and password, then authenticates the user. If a Web application is included as a portlet in a portal page, users of the page are not asked to log in to the Web application. This is a convenient way to implement single sign-on for Web applications that are integrated into the portal.

Accessing the database

The personalization APIs map name/value pairs to a unique set of UID, PID, and TID. The names and values are saved as strings. Using the personalization APIs, users can add, retrieve, and delete entries from the database. The personalization APIs use a JDBC connection—the data source, user name and password values are saved in the *personalization.properties* file.

The personalization APIs expect non-null UID, PID, and TID values; built-in triggers check for null values. Names and values are not validated. The developer must verify that names and values are correct and not duplicated. If you attempt to enter a name that already exists, the existing entry is updated.

Adding entries to the database

These APIs are used to add entries to the database:

- AddBinding(String UID, String PID, String TID, String name, String value)
- AddMultipleBinding(String UID, String PID, String TID, HashMap map, Boolean flag)

The Address Book application uses a padded-name convention for creating a unique column name. The padded name for a column is a fully-qualified name; for example, the fully-qualified column name for “FirstName” is:

`<category>+ “.” + <firstName> + “.” + <lastName>+ “.” + “FirstName”`

Where *<category>*, *<firstName>*, and *<lastNames>* are the values for the address book category, and the first name and last name of a user. For example, if a user adds Sean Dang to his address book in the “Golf-Buddies” category, the fully-qualified column name is `Golf-Buddies.Sean.Dang.FirstName`. The value is “Sean.” While this convention does not ensure uniqueness, it suffices for the purposes of this application. For more information, see the `getPaddedNameForThisColumn` method in *AddressBookEntry.java*.

Retrieving database entries

The following personalization APIs are used to retrieve entries from the database:

- `GetBinding(String UID, String PID, String TID, String name)`
- `GetMultipleBindings(String UID, String PID, String TID)`

`GetBinding` gets the value corresponding to a UID, PID, TID, and name set, and is used to form the fully-qualified name for a column. For more information, see the `AddressBookEntry.getFullEntryForThisUser` method in *AddressBookDBAccess.java*.

`GetMultipleBindings` uses the UID, PID, and TID to get a map containing all the key sets for the name value. `GetMultipleBindings` is used in the `listCategories` method, which gets a `HashMap`, then iterates over the list to check for the string “CATEGORY,” since all categories are stored as “CATEGORY.<category_name>,” where *<category_name>* is the string value of the category. Another method that uses the `GetMultipleBindings` API is `getAllLSCommaFSInThisCategory`, which iterates a `HashMap` and parses the names for the strings “FirstName” and “LastName.”

Updating and deleting database entries

You can update a database entry by inserting a row with the same UID, PID, TID, and name. Do this by calling the `AddBinding` API.

To delete a name value pair that is associated with a UID, PID and TID, use the `RemoveBinding` API. For more information, see the `deleteAllRowsForThisEntry` method in *AddressBookDBAccess.java*.

The Personalization APIs do not support transactions. To perform a bulk deletion, check the return values from the `RemoveBinding` method or catch the `DataAccessException` to determine the next course of action if a method call fails.

Accessing Portal Interface Content with Pocket PC

This chapter describes how to set up and configure Enterprise Portal 6.0 and your mobile terminal so that you can access Portal Interface content anytime, anywhere.

These instructions are for connecting to Portal Interface with the Pocket PC 2003 with Pocket Internet Explorer 4.01, or Pocket PC Emulator running on a machine in the same network as Enterprise Portal. Different environments may require different connection procedures. Refer to the documentation that came with your Pocket PC for instructions on making connections.

Note You can only view existing Portal Interface content with the Pocket PC. You can not add, delete, or change page groups, pages and portlets.

To access Portal Interface from a Pocket PC, install the following software on a machine that exists on the same network where Enterprise Portal is running:

- Microsoft ActiveSync 3.7
- Microsoft Embedded Visual C++ 4.0 and Service Pack 2.0
- Microsoft Pocket PC 2003 SDK

In addition, you must:

- Have a valid user name and password for logging in to Portal Interface.
- Verify that the `page.group` property in `global.properties.xml` is set to “on.”
- Verify that the page group containing the portlets is using the PocketPC navigation style.

Accessing content using Pocket PC 2003

- 1 Start Pocket PC from Start | Programs | Microsoft Pocket PC 2003 SDK | Pocket PC 2003.
- 2 Click in the screen area to bring up the Pocket PC 2003 desktop.
- 3 Start Internet Explorer from Start | Internet Explorer.
- 4 To access Portal Interface, enter the following URL, substituting your portal's host and domain name.

`http://HOSTNAME.PORTALDOMAIN:8080/onepage/index.jsp`

For example, if your machine name is "mymachine," and your portal domain is "sybase.com," enter:

`http://mymachine.sybase.com:8080/onepage/index.jsp`

- 5 Click Settings in the Cannot Connect message box.
 - a Select the Advanced tab at the bottom of the Connections window.
 - b Click Select Networks.
 - c Configure the drop-down box for programs that automatically connect to the Internet should connect using: My Work Network
 - d Click OK to save Network Management settings.
 - e Click OK to save Connections settings.

Accessing Portal Interface content using the Pocket PC Emulator

- 1 Start Pocket PC Emulator from Start | Programs | Microsoft Pocket PC 2003 SDK | Pocket PC 2003 Emulator.

The Emulator may take 15 - 30 seconds to start.

- 2 Click in the screen area to bring up the Pocket PC 2003 desktop.
- 3 Start Internet Explorer from Start | Internet Explorer.
- 4 To access Portal Interface, enter the following URL, substituting your portal's host and domain name.

`http://HOSTNAME.PORTALDOMAIN:8080/onepage/index.jsp`

For example, if your machine name is “mymachine,” and your portal domain is “sybase.com,” enter:

`http://mymachine.sybase.com:8080/onepage/index.jsp`

- 5 Click Settings in the Cannot Connect message box.
 - a Select the Advanced tab at the bottom of the Connections window.
 - b Click Select Networks.
 - c Configure the drop-down box for programs that automatically connect to the Internet should connect using: My Work Network
 - d Click OK to save Network Management settings.
 - e Click OK to save Connections settings.

ActiveX and portlet creation on UNIX

Description How can I use the ACX option when building portlets using a UNIX EP installation?

Solution To use the ACX option on UNIX when you create one-click capture portlets, the ActiveX processing must be delegated to a remote Windows machine that has EP installed. Specifically:

- 1 Install EP on a Solaris box. See the *Enterprise Portal 6.0 Installation Guide*.
- 2 Install EP on a Windows machine. See the *Enterprise Portal 6.0 Installation Guide*.
- 3 Edit *global.properties.xml* (see Appendix A, “Configuring Global Properties”) to:
 - Set the “acx” property to “remote”. On UNIX this property defaults to “none” and on Windows the default is “local.”
 - Set “acx.host” to the URL of the host (for example, `http://tanguy.sybase.com:8080`).
The `acx.servlet.url` property should normally be left as is (default is `/onpage/servlet/ACX`).
 - Restart the UNIX EP installation.

Now you can use the ACX option when building portlets using the UNIX EP installation. See “Building aggregated portlets” on page 48.

Agent log fills up the database

Description The agents transaction logging fills up the database too fast and the View Log or other agent functionality seems slow. Check the database logs to see if the database is out of space. You may see an error like this:

```
Tasks are sleeping waiting for space to become
available in the log segment for database tempdb
```

Solution Increase the database size. If you are using Adaptive Server Anywhere with your EP installation, see the *ASA Database Administration Guide* for instructions. If you are using Adaptive Server Enterprise, see the *System Administration Guide*. Both documents are on the *Technical Library* CD that ships with the product.

Auditing

Description Users cannot log in to Portal Interface when auditing is enabled.

Solution You must set up the message service and create the message topic before you enable auditing notifications; otherwise, users cannot log in to Portal Interface. See the *EAServer System Administration Guide* for instructions.

Auto indexer does not work

Description The autoindexer does not work.

Solution Restart the autoindexer:

- 1 Go to the `%SYBASE%\PortalSearch\Engine\main` directory on Windows and `$$SYBASE/PortalSearch/Engine/main` on UNIX.
- 2 Delete the `indexq.dat` file:
- 3 Restart the PortalSearchQueryh DRE service.

Chart text displays improperly

Description Text does not display correctly for your chart.

Solution If your chart text displays incorrectly, use this procedure to resolve the problem.

- 1 Start Jaguar Manager, select Tools | Connect | Jaguar Manager, then enter:
 - User Name – jagadmin
 - Password – leave blank.
 - Host Name – the name of your local machine.
 - Port Number – 9000
 Click Connect.
- 2 Expand the Servers folder, right-click the Jaguar tree node, and select Server Properties.
- 3 Select the Static Page Caching tab and do either of the following:
 - Deselect the Enable Static Page Caching option so no static pages or items are cached; or
 - Prevent caching of items stored in */onpage/jspfilter* by entering the following into the Exclude WebApp Files text box:


```
(onpage/jspfilter, *.*)
```
- 4 Select File | Refresh Static Cache.
- 5 Exit Jaguar Manager, then restart EAServer.

From there on, charts are no longer be cached.

Co-brands error

Description The following error displays in the *uwp.err* log:

```
ERROR Id='211' is NOT defined for OEM resources,
defaulting to OEM rid 1
```

There are no other signs that the co-brand is set up incorrectly.

Solution The co-brand name is misspelled.

Portal Security Officer password

Description You can't remember the password for the Portal Security Officer (PSO).

Solution Issue this SQL call in the Adaptive Server Enterprise entldb database to reset the PSO password to "123qwe":

```
UPDATE Subject
SET EncodedPassword= '{SHA:DktKCZJELMc=}OC0GOj+HRsPb6ZweCt3/NaKmagM='
WHERE SubjectDN LIKE "%uid=ps0%"
```

If the account is locked due to repeated invalid login attempts, remove the login lock record for the PSO user in the LoginLock table.

Note If you are using Adaptive Server Anywhere, see the *ASA Database Administration Guide* for instructions.

Portal Studio login

Description After you enter your user name and password to log in to Portal Studio, you see a mostly blank screen prompting you to enter the user name and password again and you see in this text in the status bar:

```
User: HARDCODED Resource: HARDCODED Company:
HARDCODED
```

Solution1 Configure the browser to accept cookies.

- 1 In Internet Explorer, select Tools | Internet Options.
- 2 In the Internet Options window, select the Privacy tab, then click Advanced.
- 3 Verify that the options are set to accept cookies. Click OK to exit the window.
- 4 Click OK to close the Internet Options window.

Solution 2 Change the Web container's proxy settings. During container login, the Web container uses HTTP 302 redirects to display the login form, authenticate the user, then take the browser to protected content. A cookie must be passed along with these 302 redirects, so it important that the Web container be configured so that the 302 redirects appears to be identical to the URL address that set the cookie.

- 1 Start and connect to Jaguar Manager.
- 2 Once you connect, select Jaguar Manager | Servers.
- 3 Right-click Jaguar and select Server Properties.
- 4 When the Server Properties window displays, select the HTTP Config tab.
- 5 In the Domain Name field, enter the host name + domain.
- 6 Verify that the Proxy HTTP Port and the Proxy HTTPS Port match the listeners configured for EAServer (Jaguar) and in *global.properties.xml*.
- 7 Click OK.
- 8 Right-click Jaguar and select Refresh to implement any changes you made.
- 9 Select File | Exit to close Jaguar Manager.

Solution 3 Remove any special characters from the host name or domain name. Some browsers cannot properly store and retrieve cookies for a site with special characters in the site's name, so the browser never sends back cookies to that site.

Change the name of the machine, or change the browser's character encoding to a character set that does contain special characters.

Glossary

ACDB	Access Control Database. A central database that stores all of the user's authorization and authentication information, such as user name and password credentials, digital certificates, and access permissions to the portal components. The ACDB structures data in the form of a Lightweight Directory Access Protocol (LDAP)-compliant directory and is stored on the Adaptive Server, unless you have specifically configured a third-party database to store user information.
API	An acronym for application program interface—a set of routines, protocols, and tools for building software applications that enables programs to communicate with each other.
ASP	Active Server Pages. An open, compile-free application environment in which Web developers can combine HTML, scripts, and reusable Active Server components. ASP technology enables server-side scripting for IIS with native support for both Visual Basic Scripting Edition and JScript.
access control	The process of controlling who has access to a data source.
adapter	A component that provides an interface between an internal application and external applications or messaging systems. An adapter detects events and validates event contents.
application integration	Usually, a solution designed for a specific industry that allows multiple programs to work together seamlessly. This is an approach that provides application logic and data to the application server via proxy components. During design, components are defined using metadata (sp_catalogs, COBOL copy books, IDL repositories) that associates the logic and data with a component name. Code generation and deployment into the application server are features of application integrators. When the component is available in the application server, any developer can use it without having to understand the specifics of how it works.
application service provider	Third-party companies that manage and distribute software-based services and solutions across a wide area network from a central data center.

applet	A small program in an HTML-based program built with Java that a browser temporarily downloads to, and runs from, a user's hard drive. Java applets can be downloaded and run by any Java-interpreting Web browser, such as Microsoft Internet Explorer and Netscape Navigator. Java applets can be used to add multimedia effects, such as background music, real-time video displays, animations, and interactivity, such as calculators and games, to Web pages without having to send a user request back to the server.
asset	Any object within the enterprise's computer systems, including but not limited to a document, database information, another computer system, an application, and so on.
audit, auditing	A method to provide individual accountability for users performing operational and administrative tasks. User actions are recorded in an audit log so the system administrator can see who is doing what while the user is logged in to the networked system.
authentication	The process of verifying the identity of the person trying to enter a network system.
authorization	The term used to describe the process of assigning permissions to users or groups of users to access portal assets.
B2B	An acronym for business-to-business. The ability of companies to deliver products, services, support, and information over the Internet to other companies with whom they do business.
B2Bi	An acronym for business-to-business integration. B2Bi enables a business to integrate its computer systems with those of its business partners (suppliers, vendors, customers), eliminating redundant data entry, speeding up order turnaround times, and generally reducing the "friction" involved in business-to-business transactions.
bean	A reusable software component. Beans can be combined to create an application.
binding	The association of a client and a server.
broker	A type of middleware that connects clients and servers. An example is an Object Request Broker.
buffered queue	A message queue that resides in memory.
business object	An application-level component you can use in unpredictable combinations. A business object is independent of any single application.

Business objects provide a natural way to describe application-independent concepts such as customer, order, competition, money, payment, car, and patient. They encourage a view of software that transcends tools, applications, databases, and other system concepts.

certificate authorities

Entities that validate identities and issue digital certificates. They can be either independent third parties or organizations running their own certificate-issuing server software. The certificate issued by the CA binds a particular public key to the name of the entity the certificate identifies.

CGI

Common Gateway Interface. A program written in any language that allows the program to be executed on a server machine. The program receives its input from an information server and sends its output to a client.

CipherSuites

As part of the SSL handshake, the client and server agree upon a common CipherSuite. The CipherSuite includes SSL/TLS support options, algorithms used for key exchange, and digital signatures.

class

In object-oriented programming, a category of objects. For example, there might be a class called shape that contains objects that are circles, rectangles, and triangles. The class defines all the common properties of the different objects that belong to it.

client/server

A network architecture in which one or more computers (servers) accept requests for services from one or more workstations (clients).

This may also refer to a back-end application (server) that accepts requests for information from a front-end application (client).

collaborate

The process of working together to build content. This usually involves cross-functional groups and multiple versions.

communications middleware

Software that provides inter-application connectivity based on communication styles such as message queuing, ORBs, and publish/subscribe.

communications protocol

A formally defined system for controlling the exchange of information over a network or communications channel.

component

In programming and engineering disciplines, a component is an identifiable part of a larger program or construction. Usually, a component provides a particular function or group of related functions.

In object-oriented programming and distributed object technology, a component is a reusable program building block that can be combined with other components in the same or other computers in a distributed network to form an application. Examples of a component include a single button in a graphical user interface, a small interest calculator, and an interface to a database manager.

Components can be deployed on different servers in a network and communicate with each other for needed services. A component runs in a context called a container. Examples of containers include pages on a Web site, Web browsers, and word processors.

connection pooling Connection pooling is a performance optimization based on using collections of preallocated resources, such as objects or database connections. Pooling results in more efficient resource allocation.

connectionless communications Communications that do not require a dedicated connection or session between applications.

content management Establishes control and business process on the portal content for versioning, staging, and approval.

continuous availability The ability of a computer to stay up and running 24 hours a day, 7 days a week. Continuous availability requires that solutions are both highly reliable and quickly recoverable in the event of failure. See also high availability.

control To control the development and distribution of the content via versioning and auditing.

CORBA Common Object Request Broker Architecture. CORBA is a distributed-objects standard developed and defined by the Object Management Group (OMG). CORBA provides the mechanisms by which objects transparently make requests and receive responses, as defined by OMG's Object Request Broker (ORB). CORBA is an application framework in which objects can communicate with each other, even if they are written in different programming languages, are running on different platforms, reside at different locations, or were developed by different vendors.

credentials User name and passwords pairs used for user authentication when logging in to a networked system.

DCE Distributed Computing Environment—from the Open Software Foundation, DCE provides key distributed technologies such as RPC, distributed naming service, time synchronization service, distributed file system, and network security.

DCOM	Distributed COM. A protocol that enables software components to communicate directly over a network in a reliable, secure, and efficient manner. Based on the Open Software Foundation's DCE-RPC specification, DCOM deploys across heterogeneous platforms and works with both Java applets and ActiveX components.
digital certificate	An electronic document used to identify an individual, a server, a company, or some other entity and associate that identification with a public key. See PKI.
distinguished name	A name that uniquely identifies an entity. The distinguished name is embedded in a digital certificate. Enterprise Portal security identifies an entity by its DN for authentication to the portal.
DOM	Document Object Model. The specification for how objects in a Web page (text, images, headers, links, and so on) are represented. The DOM defines what attributes are associated with each object, and how the objects and attributes can be manipulated. Dynamic HTML (DHTML) relies on the DOM to dynamically change the appearance of Web pages after they have been downloaded to a user's browser.
DSO	<p>Data source objects. A DSO provides data, embedded via data binding into an HTML page, that users can then sort and filter as they would a database, without needing to return to the server. DSOs supply data asynchronously to the page, similar to the way GIF images are displayed incrementally as they are transmitted.</p> <p>DSOs can be written as Java applets using Java and a JavaBeans interface, or as ActiveX controls using Visual C++.</p>
DTD	A document type definition is a specific definition that follows the rules of the Standard Generalized Markup Language (SGML). A DTD accompanies a document and identifies what the codes (or markup) are that separate paragraphs, identify topic headings, and so on, and how each is to be processed. When a DTD is mailed with a document, any location that has a DTD "reader" (or "SGML compiler") can process the document and display or print it as intended.

data binding	<p>The process by which a data source is linked to a Web page. You can present, manipulate, and update data on the client by linking data to HTML tags. Data binding is based on a component architecture consisting of four major pieces: DSO data consumers, the binding agent, and the table repetition agent. The DSO provides the data to the page, data-consuming HTML elements display the data, and the binding and table repetition agents ensure that both the provider and the consumer are synchronized. Data binding, combined with HTML 4.0 and the Document Object Model, is one of the contributing technologies to Dynamic HTML (DHTML).</p> <p>Because the data binding is done on the client side, the data displayed on the Web page is kept separate from the HTML that displays the data. Data binding does this by treating HTML in a Web page as a template for data supplied by a data source object. Then, using the Dynamic HTML support, the data supplied by data objects is merged with the HTML template on the client, producing a complete HTML page.</p>
data element	<p>Element that contains no element references or code lists.</p>
data mart	<p>One or more databases designed to help managers make strategic decisions about their businesses. A data mart usually focuses on a particular subject or department rather than on an enterprise-wide application.</p>
data store	<p>A physical repository that resides on a database server.</p>
data warehouse	<p>A collection of data designed to help managers make strategic decisions about their business. A data warehouse contains a wide variety of data that presents a coherent picture of business conditions at a single point in time. Unlike a data mart, a data warehouse usually refers to a set of databases that are integrated across an entire enterprise.</p>
database event	<p>Database actions that change database states, that can be captured and re-created, and that cannot occur (or be recorded) simultaneously. These can include begins, rollbacks, or commits; inserts, updates, or deletes; blobs (Java object, image, or text); or stored procedure invocations that result in a change in the source database.</p>
database middleware	<p>Allows clients to invoke SQL-based services across multivendor databases. Database middleware is defined by de facto standards such as ODBC, DRDA, RDA, and so on.</p>
decryption	<p>The process of unencoding information. Encryption and decryption allow communicating systems to disguise information they send.</p>

digital signatures	Digital signatures are created with a mathematical algorithm that generates a unique, fixed-length string of numbers from a text message; the result is called a hash or message digest. Digital signatures are used for tamper detection and nonrepudiation.
directory services	A way for clients to locate services. Usually contained in a single system image of available servers.
distributed database system	A computing system that contains a number of autonomous database management systems that are interconnected by a network and that cooperate with each other when performing data access and data capture tasks.
EAR	Enterprise archive file. Used to distribute a J2EE application. A standard JAR file with a .ear extension that may contain JAR and WAR files.
EJB	Enterprise JavaBeans is an architecture for setting up program components, written in Java, that run in the server parts of a client/server. EJBs are specific Java components that meet the Java specifications for thread management, container support, and so on.
encryption	A process wherein a cryptographic algorithm is used to encode information to safeguard it from anyone except the intended recipient. Encryption and decryption allow communicating systems to disguise information they send.
enterprise	A reference to all aspects of a large business organization—from manufacturing to finance, marketing to human resources. This term can also refer to an organization plus its partners, vendors, suppliers, and customers.
EP	An acronym for enterprise portal. An enterprise portal integrates all aspects of an organization's IT infrastructure and offers customers, partners, vendors, and employees a broad array of resources and services, including personalized information, online purchasing, e-mail, forums, search engines, and product support.
event	An event is a notification that occurs in response to some action. It can be a change in state or as a result of the user clicking or moving the mouse, pressing a keyboard key, or other actions that are focus-related, element-specific, or object-specific. Programmers write code that respond to these actions. An event can also be an object that is imported, passed between processors, and exported to an external database.
extensible	Capable of accepting new, user-defined commands.
extranet	An intranet that allows partial access to authorized outsiders via valid user names and passwords.

firewall	A system designed to prevent unauthorized access to or from a private network. Firewalls can be implemented in both hardware and software, or a combination of both. Firewalls are frequently used to prevent unauthorized Internet users from accessing private networks connected to the Internet, especially intranets. All messages entering or leaving the intranet pass through the firewall, which examines each message and blocks those that do not meet the specified security criteria.
gateway	A hardware and/or software setup that performs translations between disparate protocols.
hash	Also called a message digest, the hash is a unique, fixed-length string of numbers generated by a mathematical algorithm from a text message. The result is called a digital signature.
HTTP	HyperText Transport (or Transfer) Protocol is the set of rules that governs the exchange of text, graphic, sound, and video files on the World Wide Web.
HTTPS	The secure version of HTTP.
high availability	The ability of a computer to stay up and running most of the time. Also, the ability to perform most administration tasks with users still connected and working with the data in the database. See also continuous availability.
IDL	An interface definition language allows a program or object written in one language to communicate with another program written in an unknown language. For example, an Object Request Broker (ORB) uses an interface definition language to broker communication between two object programs.
IIOIP	Internet-Interoperable-ORB-Protocol is an object-oriented protocol that allows distributed programs written in different programming languages to communicate over the Internet. IIOIP is the transport protocol for CORBA.
ISP	An acronym for Internet service provider, which is a company that provides access to the Internet to companies or individual users for a monthly fee.
Internet	A global network connecting millions of computers.
intranet	A private network within an organization.
JDBC	JDBC is a data access interface based on ODBC and used with the Java programming language.
J2EE	Sun software: Java 2 platform, Enterprise Edition.

Java	<p>Developed by Sun Microsystems, Java is an object-oriented programming language, similar to C++. Java-based applications, or applets, can be quickly downloaded from a Web site and run using a Java-compatible Web browser such as Microsoft Internet Explorer or Netscape Navigator. Java applets are the most widespread use of Java on the Web.</p> <p>Java programs or source code files (.java) are compiled into a format known as bytecode files (.class). These files, once compiled, can be executed by a Java interpreter. Most operating systems, including Windows, Macintosh OS, and UNIX, have Java interpreters and runtime environments known as Java virtual machines.</p>
LDAP	<p>Lightweight Directory Access Protocol. LDAP is a software protocol that allows anyone to locate organizations, individuals, and other resources (files, devices, and so on) on the Internet or on a corporate intranet.</p>
load balancing	<p>The even distribution of processing and communications activity across a computer network so no single device is overwhelmed.</p>
mapper	<p>Sybase product that enables you to create transformations that transform XML documents based on one vocabulary into XML documents based on a different XML vocabulary.</p>
marshaling, unmarshaling	<p>Data marshaling converts native datatypes into an intermediate data stream that can pass safely between process boundaries. Unmarshaling converts it from the intermediate data stream to a datatype required at the other end of a communication.</p>
message	<p>A string of bytes that has meaning to the applications that use it. Messages are used for transferring information from one application to another between components in a single application. The applications can be running on the same platform or on different platforms.</p>
message broker	<p>A key component of EAI, a message broker is an intelligent intermediary that directs the flow of messages between applications, which become sources and consumers of information. Message brokers provide a flexible communications backbone and provide such services as data transformation, message routing and message warehousing.</p>
message digest	<p>Also called a hash; a unique, fixed-length string of numbers generated by a mathematical algorithm from a text message. The result is a digital signature.</p>

message queuing	A form of communication between programs. Application data is combined with a header (information about the data) to form a message. Messages are stored in queues, which can be buffered or persistent (see buffered queue and persistent queue). Message queuing is an asynchronous communications style and provides a loosely coupled exchange across multiple operating systems.
message routing	A process that routes messages to applications based on business rules. A particular message may be directed based on its subject or actual content.
message warehousing	A central repository for temporarily storing messages for analysis or transmission.
metadata	Data that describes other data. Any file or database that holds information about another database's structure, attributes, processing, or changes.
method	In object-oriented programming, a procedure that is executed when an object receives a message. A method is really the same as a procedure, function, or routine in procedural programming languages. The only difference is that in object-oriented programming, a method is always associated with a class.
middleware	Software that facilitates communication between two applications. Middleware provides an API through which applications invoke services and it controls the transmission of the data exchange over the network. There are three basic types: communications middleware, database middleware, and systems middleware.
migration	When referring to data, migration describes the process of translating data from one format to another. When referring to a computing environment, migration describes the process of moving from one type of hardware or software to another.
nonrepudiation	Digital signatures provide nonrepudiation, that is, senders cannot deny, or repudiate, that they sent a message, because their private key encrypted the message.
object middleware	Allows clients to invoke methods or objects that reside on a remote server. This middleware revolves around OMG's CORBA and Microsoft's DCOM.
ODBC	Open Database Connectivity. ODBC is a Windows standard API that is used for SQL communication to connect applications to a variety of data sources. Access is generally provided through the Control Panel, where data source names (DSNs) can be assigned to use specific ODBC drivers.
ORB	Object Request Broker. Software that allows objects to dynamically discover each other and interact across machines, operating systems, and networks.

PKI	A public-key infrastructure allows users of an insecure public network, such as the Internet, to securely exchange data and money using a public and a private cryptographic key pair obtained and shared through a trusted authority.
persistent queue	A message queue that resides on a permanent device, such as a disk, and can be recovered in case of system failure.
portal	A Web site that offers users access to a broad array of resources and services, such as e-mail, forums, search engines, and online shopping malls.
personalization	A concept of using continually adjusted user profiles to match content or services to individuals.
Portal Security Officer (PSO)	<p>The Portal Security Officer role is predefined in the Access Control Database. The Portal Security Officer manages Enterprise Portal and EAServer security using the Enterprise Portal Security Manager, a graphics-based administration tool.</p> <p>The default PSO role has all permissions and is assigned to a default login. You can use this to initially log in and create user name and password combinations for security officers, administrator, and grant the appropriate roles. You can then invalidate or delete the default login to secure the product against intruders who possess the default login information.</p>
private key	Part of the larger public-key infrastructure, a private key is kept secret and the public key is published. Typically, you use the private key to encrypt data before sending it over the Internet, and the recipient decrypts data with your public key.
public key	Part of the larger public-key infrastructure, a public key is published, and the corresponding private key is kept secret. Typically, the public key is used to decrypt information that is encrypted with a private key before being sent over the Internet. See PKI.
public-key cryptography	Public-key cryptography consists of encryption and decryption, digital signatures, keys, and digital certificates. It is part of the larger public key infrastructure. See PKI.
publish	Make an event available to an external application by placing it on the external application's queue.
publish/subscribe	A style of interapplication communications. Publishers are able to broadcast data to a community of information users or subscribers, which have issued the type of information they want to receive (normally defining topics or subjects of interest). An application or user can be both a publisher and subscriber.

queue	<p>A list constructed and maintained so that the next data element to be retrieved is the one stored first.</p> <p>For example, one application can put a message on a queue, and another application can retrieve the message from the same queue.</p>
RMI	<p>Remote Method Invocation is a set of protocols being developed by Sun's Java Software division that enables Java objects to communicate remotely with other Java objects. RMI is a relatively simple protocol, but unlike more complex protocols such as CORBA and DCOM, it works only with Java objects. CORBA and DCOM are designed to support objects created in any language.</p>
role-based access	<p>Users gain access to the information based on their role in the organization.</p>
RPC	<p>Remote procedure call. A form of application-to-application communication that hides the intricacies of the network by using an ordinary procedure call mechanism. It is a tightly coupled synchronous process.</p>
real time, real-time	<p>The immediate processing of input, such as the ability of a computer to respond or process information immediately with no interruption.</p>
replication	<p>The process of copying data to remote locations. The copied (replicated) data is then kept synchronized with the primary data. Data replication is distinct from data distribution. Replicated data is stored copies of data at particular sites throughout a system and is not necessarily distributed data.</p>
request/response	<p>See publish/subscribe.</p>
SNMP	<p>Simple Network Management Protocol is the protocol that governs network management and how network devices and their functions are monitored. It is not necessarily limited to TCP/IP networks.</p>
SOAP	<p>Simple Object Access Protocol. SOAP provides a way for applications to communicate with each other over the Internet, independent of platform. Remote objects can give a program almost unlimited power over the Internet, but most firewalls block non-HTTP requests. SOAP, an XML-based protocol, gets around this limitation to provide intraprocess communication across machines.</p> <p>In Enterprise Portal, the implementation of SOAP is intended to provide businesses with a way to expose corporate software functionality to their customers with minimal firewall constraints, platform dependencies or complex development implementations involving DCOM or CORBA.</p>

SOAP was developed by Microsoft, DevelopMentor, and Userland Software and has been proposed to the Internet Engineering Task Force (IETF) as a standard.

SPI	Service Provider Interface, the programming interface for developing Windows drivers to provide common access to services, the application (query, word processor, e-mail program, and so on) is written to a particular interface, such as ODBC or MAPI, and the developer of the service software (database manager, document manager, print spooler, and so on.) writes to the SPI for that class of service
SQL	Structured Query Language.
SSL	Secure Sockets Layer. A set of rules that govern server authentication, client authentication, and encrypted communication between servers and clients. SSL is widely used on the Internet, especially for interactions that involve exchanging confidential information.
SSL handshake	A series of I/O round trips between a server and a client to negotiate and agree upon a secure encrypted session.
SSO	An acronym for single sign-on. Single sign-on features allow a client to request access to protected assets within the portal without having to resubmit credentials or certificates for authentication.
scalability	The ability of an information system to provide high performance as greater demands are placed upon it, through the addition of extra computing power.
secure business object	The SBO is an EAServer service component that communicates to back-end data stores in a language and syntax that is understood by the data store, and performs access control checks on the user's component requests.
server	A computer or software package that provides specific capabilities to client software running on other computers.
servlet	<p>A servlet is a small, persistent, low-level program that runs on a server. The term was coined in the context of the Java applet, a small program that is sent as a separate file along with a Web (HTML) page.</p> <p>Some programs that access databases based on user input need to be on the server. These programs were most often implemented using a Common Gateway Interface (CGI) application. However, if a Java virtual machine is running in the server, servlets can be implemented in Java. A Java servlet can execute more quickly than a CGI application. Instead of creating a separate program process, each user request is invoked as a thread in a single daemon process, so that the system overhead for each request is slight.</p>

sockets	A portable standard for network application providers on TCP/IP networks.
stored procedure	A program that creates a named collection of SQL or other procedural statements and logic that is compiled, verified, and stored in a server database.
systems middleware	Software that provides value-add services as well as interprogram communications. An example is transaction processing monitors, which are required to control local resources and also cooperate with other resource managers to access nonlocal resources.
taxonomy	A classification system for items based on their relationship to one another.
TCP/IP	Transmission Control Protocol/Internet Protocol—the network protocol for the Internet that runs on virtually every operating system. IP is the network layer and TCP is the transport layer. TCP/IP is the primary transport protocol used in client/server computing, and is the protocol that governs the transmission of data over the Internet.
trade relationship	Business relationship between two trading partners in which EDI and XML documents are exchanged.
trading partner	Organization with which you trade (for example, a supplier or customer). Trading partners send and receive EDI and XML documents.
transaction log	The log of transactions kept by a database server. A log reader reads these logs to capture changes to primary databases.
transform	Process in which you convert a source document based on one XML vocabulary into a target document based on another XML vocabulary.
Transport Layer Security	A security protocol from the Internet Engineering Task Force (IETF) that is a merger of SSL and other protocols. TLS is backward compatible with SSL and uses Triple DES encryption.
trigger	A stored procedure that is automatically invoked on the basis of data-related events.
thin client	Thin client refers to the Net PC or the network computer, personal computers for businesses that are centrally-managed, configured with only essential equipment, and do not have CD players, diskette drives, or expansion. Since the idea is to limit such computers to essential applications, they tend to remain “thin” in terms of the client applications they include.
URI	Uniform Resource Identifier. A URI is compact string of characters for identifying an abstract or physical resource and provides a simple and extensible means for identifying resources. An example of a URI is a URL.

- URL** Uniform Resource Locator. A subset of a URI, a URL is like a networked extension of the standard file name concept: you can point to a file in a directory, but that file and directory can exist on any machine on the network. They can also be served by any of several different methods. URLs can also point to queries, documents stored deep within databases, and so on.
- WAR** Web application archive file. Used to distribute Web applications; it includes a deployment descriptor and Web components, and may contain server-side utility classes, HTML, image and sound files, applets, and client-side utility classes.
- workflow** Software used to automatically route events or work-items from one user or program to another. Workflow is synonymous with process flow, although traditionally has been used in the context of person-to-person information flows.
- XML** Extensible Markup Language—a simplified subset of Standard Generalized Markup Language (SGML)—is a way to that provides a file format for representing data, a method for describing data structure, and a mechanism for extending and annotating HTML with semantic information.
- As a universal data format, XML provides a standard for the server-to-server transfer of different types of structured data so that the information can be decoded, manipulated, and displayed consistently and correctly. In addition, it enables the development of three-tier Web applications, acting as the data transfer format between the middle-tier Web server and the client.
- XSL** Extensible Style Language. XSL is a transformational language. It can take an XML document (or a rigorously valid HTML document) and convert it to another XML document, an HTML document, a printable HTML document, a standard ASCII text file, a proprietary text format, or conceivably even a binary representation.

Index

Symbols

@OP parameters 89

A

ActiveX (ACX) portlet capture option 51
ActiveX and portlet creation on UNIX 413
ACX
 see ActiveX 51
acx (ActiveX) global property 279
acx.host global property 279
acx.servlet.url global property 279
adapters
 building portlets with 152
 creating security assets for 153
 database table 150
 database table, using 158
 EJB 150
 EJB, using 159
 granting permissions on security assets 153
 personalization 88, 149
 security proxy 150
 security proxy, using 152
 types 150
 user name/password 150
 user name/password, using 158
addDatabaseForChannel 315
addDatabaseForUserAgent 315
adding
 companies 250
 new catalog branches 183
 pages 194
 portletlets to catalogs 183
 secured JSP page to a page group 375
 split rules 133
 XML_sample portlet to a page 174
address book sample 395
addRoleForChannel 300

addRoleForUserAgent 301
Administer menu 7
advanced search APIs 314
agent log fills up the database 414
agents
 changing status 213
 creating 214
 deleting 326
 renaming 326
 running once 213
 search 322
AgentServerID global property 277
aggregate portlets 48
alwaysValidateSession global property 263
Anonymous.Roles global property 263
application portlets 106
applications
 managers 9
applications, creating single sign-on portlets for 368
applications, selecting 6
archive file, importing to 238
assets
 creating new 381
 creating security 153
 defining 363
 granting permissions 153
 protecting 382
assigning portlet labels 91
auditEnabled global property 272
auditing
 sample Portal Interface records 34
 troubleshooting 414
auditPortletPlayback 272
auditPortletPlayback global property 272
auditRoleFilter global property 272
AuthenticationUsing global property 275
auto indexer
 troubleshooting 414
Automate menu 6
automatic indexing 334

- reprocessing the last file 336
- starting from scratch 337
- utility, stopping 336
- automatic indexing basic configuration parameters
 - MaxLogKBytes 339
 - Number 339
 - PollingMethod 338
 - PollingPeriod 338
 - RemoveLogFileOnStart 339
- automatic indexing default parameters
 - deleteEscapeReferences 341
 - deletePathReplaceUpToSlash 341
 - deleteRefReplaceCSVs 341
 - deleteRefReplaceWithCSVs 341
 - directoryAfterDate 341
 - directorybeforeDate 340
 - directoryCantHaveCSVs 340
 - DirectoryFileMatch 340
 - directoryMustHaveCSVs 340
 - directoryPathCSVs 340
 - directoryRecurse 340
 - DREHost 339
 - fileBaseDirectory 340
 - FilenameOutputMode 341
 - FilePollFilename 340
 - importIDXFilesAction 341
 - importIDXFilesMoveTo 341
 - ImportPathReplaceString 341
 - IndexPort 339
 - MoveToDirectory 340
 - PollingAction 339
 - PollingMaxNumber 340
 - PollingPostAction 340
 - QueryPort 339
- automatic indexing import parameters
 - Database 344
 - FieldName 343
 - FieldStart 343
 - FieldStop 343
 - FixedFieldName 343
 - FixedFieldValue 343
 - HTMLFieldName 343
 - HTMLFieldStart 344
 - HTMLFieldStop 344
 - HTMLImportEndDefCSVs 346
 - HTMLImportExtns 341
 - HTMLImportStartDefCSVs 346
 - ImportAttachments 352
 - ImportBreaking 342
 - ImportBreakingMaxParagraphWords 342
 - ImportBreakingMinDocWords 342
 - ImportBreakingMinParagraphWords 342
 - ImportChecksum 343
 - ImportDefaultExtension 341
 - ImportDelimitedDocEnd 350
 - ImportDelimitedDocStart 350
 - ImportDelimitedEnd 350
 - ImportDelimitedExtns 350
 - ImportDelimitedField 350
 - ImportDelimitedSkipChars 350
 - ImportDelimitedStart 350
 - ImportEndDefCSVs 346
 - ImportExtractDateFormatCSVs 345
 - ImportExtractDateFrom 344
 - ImportExtractDateFromField 344
 - ImportExtractDateToField 344
 - ImportExtractDateToFormat 345
 - ImportExtractLength 345
 - ImportExtractLengthToField 346
 - ImportFieldHTMLConvertChars 343
 - ImportFilterBinaryContent 352
 - ImportIntelligentTitleSummary 343
 - ImportLogFile 351
 - ImportLogFileAppend 352
 - ImportLogLevel 352
 - ImportMaxAllowedASCIIICODE 352
 - ImportMaxAverageWordLength 352
 - ImportMaxBinaryCharsPerHundred 352
 - ImportMaxLength 342
 - ImportMaxLengthWords 342
 - ImportMetaToFields 343
 - ImportMinAllowedASCIIICODE 352
 - ImportMinLength 342
 - ImportMinLengthWords 342
 - ImportPageBreakDefs 346
 - ImportPathReplaceString 346
 - ImportPathReplaceUpToSlash 346
 - ImportRecursiveDirectoryImporting 342
 - ImportRefReplaceCSVs 347
 - ImportRefReplaceWithCSVs 347
 - ImportRefTruncateAfter 347
 - ImportRefTruncateString 347

- ImportRegisterExecutable 347
- ImportRegisterExtnCSVs 347
- ImportRemapField 343
- ImportRemapFieldTo 343
- ImportRenderedHTMLExtensions 348
- ImportRenderedHTMLFieldName 348
- ImportRenderedHTMLMoveToDir 348
- ImportRenderedHTMLPathReplaceString 348
- ImportRenderedHTMLPathReplaceUpToSlash 348
- ImportRenderedHTMLRefReplaceCSVs 348
- ImportRenderedHTMLRefReplaceWithCSVs 349
- ImportSaveAttachmentDir 352
- ImportSaveAttachments 352
- ImportStartDefCSVs 346
- ImportStartSkipSentences 346
- ImportStartSkipWords 346
- ImportStoreContent 342
- ImportStripLinks 349
- ImportStripStringCSVs 349
- ImportStripTagsCSVs 349
- ImportSummary 342
- ImportSummarySize 342
- ImportTempDir 342
- ImportXMLEntryOnly 351
- ImportXMLExtns 351
- ImportXMLField 350
- ImportXMLSearchCSVTags 351
- ImportXMLStripTags 351
- TextImportExtns 341
- automatic indexing utility 321
 - configuration file 335, 337
 - configuration parameters, basic 338
 - configuration parameters, default 339
 - import parameters, basic 341
 - Import Reference Replace parameter 347
 - import rendering parameters 349
 - [job_name] section of configuration file 353
 - PortalSearchAutoindexer**, starting 335
 - restarting 336, 337
- Autonomy, Inc.
 - DLL files 329, 330

B

- bar graph charts 143
- branch, catalog
 - defined 182
- branches 185
 - deleting catalog 186
 - renaming catalog 186
- branches, catalog
 - adding new 183
- broadcast events 101
- Build menu 6
- building
 - application portlets 106
 - page groups 206
 - pages 194
 - templates 164, 165

C

- CacheRefreshList global property 272, 275
- CacheRefreshList property 284
- caching 290
- calling methods 290
- CapAll 51
 - performance hit 51
- capture formats 49
- capture options
 - CapAll 51
- CaptureTraceOn global property 280
- catalog branch and subbranches, illustrated 179
- Catalog Manager 179–186
 - adding new branches 183
 - catalog branch defined 182
 - changing catalog properties 185
 - changing catalog status 185
 - creating a new catalog with 182
 - deleting existing branches 186
 - deleting portlets from catalogs 186
 - described 180
 - interface 180
 - moving existing branches 185
 - renaming existing branches 186
 - viewing catalog properties 185
- catalog_cache_refresh_list global property 263
- catalog_cache_refresh_list property 284

Index

- catalogs
 - adding new branches 183
 - adding portlets to 183
 - changing
 - catalog properties** 185
 - changing status 185
 - creating a 182
 - deleting branches 186
 - deleting portlets from 186
 - display 179
 - moving existing branches 185
 - renaming existing branches 186
 - updating changes to Portal Interface 186
 - viewing properties 185
- certifications on the Web xvii
- CGI parameters 87
- changeMyInfo global property 264
- changing
 - an agent's status 213
 - catalog properties 185
 - catalog status 185
 - page layout 191
 - template workflow status 171
- changing default express portlet templates 172
- channels
 - adding 323
- channels and subchannels 323
- charting portlets 141
- charts
 - bar graph 143
 - creating pie 145
 - creating portlet 142
 - display problems 147, 415
 - fonts 144
 - line 146
 - troubleshooting 415
 - turning off caching 147, 415
- classes
 - com.sybase.ep.search.PortalSearchServices.Search 293
 - com.sybase.ep.search.PortalSearchServices.SearchHome class 292
- cleanupUserMemory 301
- click-across portlets 96
- ClickacrossOn global property 275
- client_virtual_path global property 264
- cobrand
 - directory structure 249
- co-brands
 - troubleshooting 415
- co-brands, creating 247
- co-brands, multiple portals 247
- cobrand.xml file, editing 251
- columns
 - changing width for pages 193
- com.sybase.ep.search.PortalSearchServices.Search class 293
- com.sybase.ep.search.PortalSearchServices.SearchHome class 292
- commands
 - guiadmin.exe** 334
 - isql** 287
 - PortalSearchautoindexer** 335
 - StartQuery.sh** 327
- companies
 - adding to create cobranded portal 250
- concept-based searching 289, 326
 - description of 289, 321
 - DRE 322
 - EJB 291–300
 - executables 326
 - GUI administration tool 333
 - performing 326
 - SearchHome class 292–300
- configuration
 - automatic indexing basic parameters 338
 - automatic indexing default parameters 339
 - automatic indexing import parameters 341
 - automatic indexing Import Reference Replace parameter 347
 - automatic indexing import rendering parameters 349
 - automatic indexing utility file 335, 337
- Configure menu 7
- configuring
 - global properties 261
 - multimachine installations 284
- connection cache, epSearchCache 334
- Content Explorer
 - managing collections 393
 - sample portlet 390
- ContentCacheMaxEntries global property 272

- conventions xviii
- copyUserAgent 301
- createAgent 302
- createChannel 302
- creating
 - agents 214
 - catalogs 182
 - database elements 54
 - deployable WAR file 371
 - HTML elements 74
 - JSP elements 56
 - messaging portlets 96
 - new assets 381
 - new page for a secured JSP portlet 374
 - new templates 170
 - new UDDI registry entries 73
 - page groups 206
 - pages 362
 - pages with portlet that executes a JSP 361
 - personalization adapter keys 151
 - personalization keys 154
 - portlet charts 142
 - portletlets with personalization adapters 152
 - proxy authentication information 364
 - secure Web applications 358
 - secured Web application 369, 375
 - security assets 153
 - select lists for portlet input fields 136
 - single sign-on JSP portlet and deploying 372
 - single sign-on portlets 357
 - single sign-on portlets with EP and Enterprise Security 375
 - single sign-on portlets with Web applications 368
 - user-specific UI XSLT template 175
 - user-specific XML 173
 - WAR files 360
 - Web elements 49
 - Web portlets 51
 - Web service element using a service key 70
 - Web service element using WSDL 69
 - XML portlet 52, 54
 - XML sample portlet 173
- creating catalogs 179–186
 - catalog branch and subbranches 179
 - Catalog Manager 180

- css.css 258
- customizing
 - portlet look and feel 196
- customizing user XML element templates 173

D

- data
 - managing collections in Content Explorer 393
- database elements 54
- database table
 - adapter 150
 - database table adapter, using 158
- Database, autoindexing parameter 344
- DatabaseImpl global property 282
- data-capable elements 132
 - filter rules 134
 - record layout 134
 - split rules 133
- DataPool name global property 283
- DBLookupDocumentCaching global property 275
- default express portlet template, changing 172
- default, template 48
- default_http_port global property 264
- default_https_port global property 264
- default_tile_href global property 264
- DefaultResourceID global property 264
- DefaultTemplateName global property 273
- DefaultUserID global property 264
- deferupdate global property 264
- defining
 - broadcast events 101
- defining, assets 363
- definition, catalog branch 182
- deleteEscapeReferences, autoindexing parameter 341
- deletePathReplaceUpToSlash, autoindexing parameter 341
- deleteRefReplaceCSVs, autoindexing parameter 341
- deleteRefReplaceWithCSVs, autoindexing parameter 341
- deleting
 - catalog branches 186
 - page groups 209
 - pages 197
 - portletlets from catalogs 186

- templates 171
- deploy
 - enabling functionality 244
- DeployHost global property 275
- deploying
 - a WAR file 371
 - portal objects to a remote server 239
 - secured JSP portlet to Portal Interface 372
 - the secured Web application 371
 - WAR files 360
- deployment descriptor (single sign-on) 369
- directoryAfterDate, autoindexing parameter 341
- directoryBeforeDate, autoindexing parameter 340
- directoryCantHaveCSVs, autoindexing parameter 340
- DirectoryFileMatch, autoindexing parameter 340
- directoryMustHaveCSVs, autoindexing parameter 340
- directoryPathCSVs, autoindexing parameter 340
- directoryRecurse, autoindexing parameter 340
- disableAutomaticReauthentication global property 265
- display catalog 179
- display problems
 - chart caching 147, 415
- doAdvancedQuery 317
- docservers, *global.properties.xml* property group 275
- documentation
 - EAServer xv
 - Enterprise Portal xv
 - jConnect for JDBC xvi
 - other sources of information xvi
- DocumentRoot global property 265
- domain global property 265
- doQueryWithAgent 303
- DRE
 - GUI administration tool 333
 - host and port settings 334
 - passing security information to 328
 - security DLLs 329
 - sorting queries in 332
 - starting 327
- DRE administration tool, starting 333
- DREHost, autoindexing parameter 339
- Dynamic Reasoning Engine (DRE) 386
- Dynamic Reasoning Engine. *See* DRE

E

- EAServer
 - configuring multiple servers for the same domain 285
 - documentation xv
 - setting up for secure portlet creation 79
- editAgent 303
- editing
 - cobrand.xml file 251
 - global.properties.xml file 259
 - pages 197
 - portlet properties 86
 - portlets 85
 - styles.xml file 255
 - templates 171
- EJB
 - adapter 150
 - adapter, using 159
- element list 42
- elements
 - database 54
 - data-capable 132
 - HTML 74
 - JSP 56
 - saving 82
 - Web 49
 - Web from secure sites 79
 - XML 52, 54
- elements, enabling different templates for the same 175
- e-mail
 - sending to a file 284
 - sending to users 284
- enable.tracing global property 281
- Enabling
 - deploy functionality 244
- enabling
 - different templates for the same element 175
 - portlets and pages to support multiple languages simultaneously 137
- entering
 - user-specific personalization values in Portal Studio 160
- entering portal authentication data for single sign-on 381
- Enterprise JavaBeans (EJBs) 291

Enterprise Security
 creating single sign-on portlets in EP 375
 environment variables
 JAGUAR xviii
 SECURITY xviii
 SYBASE xviii
 ep_security_eas_guest_pwd global property 265
 ep_security_eas_url global property 265
 epSearchCache, managing 334
 epsecurity.xml global property
 global property
 epsecurity.xml 274
 EPSecurityHost global property
 global property
 EPSecurityHost 275
 EPSecurityHostPortNumber global property 275
 epstudio.xml global property 274
 ErrorFile global property 273, 275, 276, 277
 events
 defining broadcast 101
 spaces in event names 113, 115, 118, 127, 128
 examples
 search 311
 examples, Search and SearchHome classes 311
 examples,HTML template 165
 existsDatabase 319
 exporting
 objects from Portal Studio to an XML file 238
 personalization keys 161

F

FieldName, autoindexing parameter 343
 FieldStart, autoindexing parameter 343
 FieldStop, autoindexing parameter 343
 fileBaseDirectory, autoindexing parameter 340
 FilenameOutputMode, autoindexing parameter 341
 FilePollFilename, autoindexing parameter 340
 filter rules 134
 filter rules, adding 134
 FixedFieldName, autoindexing parameter 343
 FixedFieldValue, autoindexing parameter 343
 fonts
 charting 144
 formats, portlets 49

FrequentlyChangableDocumentCaching global property
 275

G

getCountOfChannels 304
 getCountOfDatabases 318
 getCountOfDatabasesForChannel 317
 getCountOfDatabasesForUserAgent 316
 getCountOfRolesForChannel 304
 getCountOfRolesForUserAgent 304
 getCountOfSubChannels 305
 getCountOfUserAgents 305
 getCurrentSubChannelAge 305
 getCurrentSubChannelMaxDocs 305
 getCurrentSubChannelRelevance 306
 getCurrentSubChannelTrainingText 306
 getCurrentUserAgentAge 306
 getCurrentUserAgentMaxDocs 306
 getCurrentUserAgentRelevance 307
 getCurrentUserAgentTrainingText 307
 getNextChannelName 307
 getNextDatabaseForChannel 317
 getNextDatabaseForDre 319
 getNextDatabaseForUserAgent 316
 getNextRoleForChannel 307
 getNextRoleForCurrentUserAgent 308
 getNextSubChannelName 308
 getNextUserAgentName 308
 Getting Started CD xvi
 global properties, updating 283
 global property 272
 acx (ActiveX) 279
 acx.host 279
 acx.servlet.url 279
 AgentServerID 277
 alwaysValidateSession 263
 Anonymous.Roles 263
 auditEnabled 272
 auditRoleFilter 272
 AuthenticationUsing 275
 CacheRefreshList 272, 275
 CaptureTraceOn 280
 catalog_cache_refresh_list 263
 changeMyInfo 264

Index

- ClickacrossOn 275
- client_virtual_path 264
- ContentCacheMaxEntries 272
- DatabaseImpl 282
- DataPool name 283
- DBLookUpDocumentCaching 275
- default_http_port 264
- default_https_port 264
- default_tile_href 264
- DefaultResourceID 264
- DefaultTemplateName 273
- DefaultUserID 264
- deferupdate 264
- DeployHost 275
- disableAutomaticReauthentication 265
- DocumentRoot 265
- domain 265
- enable.tracing 281
- ep_security_eas_guest_pwd 265
- ep_security_eas_url 265
- EPSecurityHostPortNumber 275
- epstudio.xml 274
- ErrorFile 273, 275, 276, 277
- FrequentlyChangableDocumentCaching 275
- HeartBeat 277
- https.playback 265
- includelistfile 279
- javascript.domain 265
- log 279
- LogAllNav 280
- LogFile 280
- mail.host 265
- mail.transport.protocol 265
- MailOn 275
- markupNoJSFile 265
- MaxBytesForDocumentsToCache 275
- MaxLogRows 277
- MaxNumberOfDocumentsToCache 276
- maxPageName 266
- maxTabSets 266
- nav.general.host 280
- nav.general.secure.host 280
- nav.host 280
- nav.secure.host 280
- NavTraceOn 280
- NonfrequentlyChangedDocumentCaching 276
- ObjectCacheLRUReductionSize 272, 273
- ObjectCacheMaxEntries 273
- page.group 266
- portal.defaultRID 266
- portal.epSecurity 266
- portal.host 266
- portal.portlet_create_lock 266
- PortalAdministrationRole 266
- proxy 267
- proxy.bypass_list 267
- proxy.host 267
- proxy.password 267
- proxy.port 267
- proxy.user 267
- PSHost 277
- PSPort 277
- PushAgentLog 277
- redirfile 267
- registration.userJ2EERoles 267
- RequiredRoles 267
- resetPasswordEmail 267
- RoleBaseDisplay 268
- RoleBaseDisplaySeeAllRoles 268
- secure 268
- secure_login 268
- SecureHostname 268
- security.title.host 268
- send.mail 268
- server_virtual_path 269
- sharePage 269
- SOAPServer 269
- StatusUpdateFreq 277
- StorageConnector 273
- tabingChar 269
- tabset 269
- ThisMachineName 273
- tile.host 269
- tile_style 270
- TimeZone 277
- TraceFile 273, 276, 277
- TraceOn 273, 276, 277
- TransactionFile 278
- turnOffSecurePassword 270
- use_https 270
- use_include 280
- UseAccessControl 276

- UsePreCompiledJSP 274
- UseStudioDatabase 276
- UWPCacheServer 274
- UWPIntegrationHost 274
- UWPWebServiceHost 274
- VerboseTrace 274, 276, 277, 278
- versioning 270
- WebServerOn 276
- WebServiceRoot 271
- wireless_portal 271
- WorkRoot 271
- xml_catalog_serialize 271
- global, global.properties.xml property group 263
- global.properties.xml
 - global property group 263
- global.properties.xml*
 - capture logging properties 280
 - database property group 280
 - docservers property group 275
 - limiting access to 286
 - nav property group 278
 - portal property group 276
 - UWP property group 272
- global.properties.xml configuration file 261
- global.properties.xml file, editing 259
- granting
 - permissions on security assets 153
- guiadmin.exe** command 334

H

- HeartBeat global property 277
- help templates, building 167
- host name
 - DRE 334
- HTML
 - elements 74
 - templates 165
- HTML templates 165
 - OPContent tag 165
- HTMLFieldName, autoindexing parameter 343
- HTMLFieldStart, autoindexing parameter 344
- HTMLFieldStop, autoindexing parameter 344
- HTMLImportEndDefCSVs, autoindexing parameter 346

- HTMLImportExtns, autoindexing parameter 341
- HTMLImportStartDefCSVs, autoindexing parameter 346
- https.playback global property 265

I

- iFrames 82, 397
- import parameters
 - auto indexing utility Import Reference Replace parameter 347
 - auto indexing utility import rendering parameters 349
 - autoindexing utility 341
- ImportAttachments, autoindexing parameter 352
- ImportBreaking, autoindexing parameter 342
- ImportBreakingMaxParagraphWords, autoindexing parameter 342
- ImportBreakingMinDocWords, autoindexing parameter 342
- ImportBreakingMinParagraphWords, autoindexing parameter 342
- ImportChecksum, autoindexing parameter 343
- ImportDefaultExtension, autoindexing parameter 341
- ImportDelimitedDocEnd, autoindexing parameter 350
- ImportDelimitedDocStart, autoindexing parameter 350
- ImportDelimitedEnd, autoindexing parameter 350
- ImportDelimitedExtns, autoindexing parameter 350
- ImportDelimitedField, autoindexing parameter 350
- ImportDelimitedSkipChars, autoindexing parameter 350
- ImportDelimitedStart, autoindexing parameter 350
- ImportEndDefCSVs, autoindexing parameter 346
- ImportExtractDateFormatCSVs, autoindexing parameter 345
- ImportExtractDateFrom, autoindexing parameter 344
- ImportExtractDateFromField, autoindexing parameter 344
- ImportExtractDateToField, autoindexing parameter 344
- ImportExtractDateToFormat, autoindexing parameter 345
- ImportExtractLength, autoindexing parameter 345

Index

- ImportExtractLengthToField, autoindexing parameter 346
 - ImportFieldHTMLConvertChars, autoindexing parameter 343
 - ImportFilterBinaryContent, autoindexing parameter 352
 - importIDXFilesAction, autoindexing parameter 341
 - importIDXFilesMoveTo, autoindexing parameter 341
 - ImportImportXMLSearchCSVTags, autoindexing parameter 351
 - importing
 - personalization keys 161
 - XML files into Portal Studio 236
 - ImportIntelligentTitleSummary, autoindexing parameter 343
 - ImportLogFile, autoindexing parameter 351
 - ImportLogFileAppend, autoindexing parameter 352
 - ImportLogLevel, autoindexing parameter 352
 - ImportMaxAllowedASCIIICODE, autoindexing parameter 352
 - ImportMaxAverageWordLength, autoindexing parameter 352
 - ImportMaxBinaryCharsPerHundred, autoindexing parameter 352
 - ImportMaxLength, autoindexing parameter 342
 - ImportMaxLengthWords, autoindexing parameter 342
 - ImportMetaToFields, autoindexing parameter 343
 - ImportMinAllowedASCIIICODE, autoindexing parameter 352
 - ImportMinLength, autoindexing parameter 342
 - ImportMinLengthWords, autoindexing parameter 342
 - ImportPageBreakDefs, autoindexing parameter 346
 - ImportPathReplaceString, autoindexing parameter 341, 346
 - ImportPathReplaceUpToSlash, autoindexing parameter 346
 - ImportRecursiveDirectoryImporting, autoindexing parameter 342
 - ImportRefReplaceCSVs, autoindexing parameter 347
 - ImportRefReplaceWithCSVs, autoindexing parameter 347
 - ImportRefTruncateAfter, autoindexing parameter 347
 - ImportRefTruncateString, autoindexing parameter 347
 - ImportRegisterExecutable, autoindexing parameter 347
 - ImportRegisterExtnCSVs, autoindexing parameter 347
 - ImportRemapField, autoindexing parameter 343
 - ImportRemapFieldTo, autoindexing parameter 343
 - ImportRenderedHTMLExtensions, autoindexing parameter 348
 - ImportRenderedHTMLFieldName, autoindexing parameter 348
 - ImportRenderedHTMLMoveToDir, autoindexing parameter 348
 - ImportRenderedHTMLPathReplaceString, autoindexing parameter 348
 - ImportRenderedHTMLPathReplaceUpToSlash, autoindexing parameter 348
 - ImportRenderedHTMLRefReplaceCSVs, autoindexing parameter 348
 - ImportRenderedHTMLRefReplaceWithCSVs, autoindexing parameter 349
 - ImportSaveAttachmentDir, autoindexing parameter 352
 - ImportSaveAttachments, autoindexing parameter 352
 - ImportStartDefCSVs, autoindexing parameter 346
 - ImportStartSkipSentences, autoindexing parameter 346
 - ImportStartSkipWords, autoindexing parameter 346
 - ImportStoreContent, autoindexing parameter 342
 - ImportStripLinks, autoindexing parameter 349
 - ImportStripStringCSVs, autoindexing parameter 349
 - ImportStripTagsCSVs, autoindexing parameter 349
 - ImportSummary, autoindexing parameter 342
 - ImportSummarySize, autoindexing parameter 342
 - ImportTempDir, autoindexing parameter 342
 - ImportXMLEntryOnly, autoindexing parameter 351
 - ImportXMLExtns, autoindexing parameter 351
 - ImportXMLField, autoindexing parameter 350
 - ImportXMLStripTags, autoindexing parameter 351
 - includelistfile global property 279
 - IndexPort, autoindexing parameter 339
 - inline frames, using 82, 397
 - interface of Catalog Manager 180
 - interface, Portal Studio 5
 - isql** command line utility 287
- ## J
- JAGUAR environment variable xviii
 - javascript.domain global property 265
 - JSP
 - elements 56

- portlet, creating 361
- templates 168
- testing as a Web application 360
- JSP portlets using the Secure Web Proxy 59
- JSPs
 - creating a new page for a secured JSP portlet 374
 - creating single sign-on portlet and deploying 372
 - portlet SSO check box 62
 - protected 371
 - single sign-on programming 383

K

- keys
 - creating personalization 154
 - importing and exporting personalization 161

L

- labels
 - mapping portlet 91
 - portlet 90
- languages, portlet support for multiple languages 137
- layout, changing page 191
- limiting log size 233
- line charts 146
- lists, creating select 136
- log global property 279
- LogAllNav global property 280
- LogFile global property 280
- logging in to Portal Studio 4
- logs
 - limiting the log size 233
- look and feel
 - customizing portlet 196

M

- mail.host global property 265
- mail.transport.protocol global property 265
- MailOn global property 275
- Manage menu 7
- management, portlet 85

- managers
 - application 9
 - Portlet 40
- markupNoJSFile global property 265
- MaxBytesForDocumentsToCache global property 275
- MaxLogKBytes, autoindexing parameter 339
- MaxLogRows global property 277
- MaxNumberOfDocumentsToCache global property 276
- maxPageName global property 266
- maxTabSetChar global property
 - global property
 - maxTabSetChar 266
- maxTabSets global property 266
- menus
 - Administer 7
 - Automate 6
 - Build 6
 - Configure 7
 - Manage 7
- messages.xml file 258
- messaging portlets 96
 - creating 96
- MoveToDirectory, autoindexing parameter 340
- moving
 - existing catalog branches 185
 - moving catalog 185
- multimachine installations, configuring 284
- multiple portals 247

N

- nav.general.host global property 280
- nav.general.secure.host global property 280
- nav.host global property 280
- nav.secure.host global property 280
- NavTraceOn global property 280
- NonfrequentlyChangedDocumentCaching global property 276
- Number, autoindexing parameter 339

- O**
- ObjectCacheLRUReductionSize global property 272, 273
 - ObjectCacheMaxEntries global property 273
 - one-click capture 49
 - OPContent tags 165
- P**
- Page Builder, interface 189
 - Page Group Builder, interface 201
 - page groups 187
 - building 206
 - deleting 209
 - Page Manager
 - displaying 187
 - using 187
 - page.group global property 266
 - pages
 - adding 194
 - adding to page groups 206
 - adding XML portlets to portal 174
 - building 194
 - changing column width 193
 - changing layout 191, 192
 - creating 362
 - deleting 197
 - editing 187, 197
 - editing and deleting 197
 - introduction 187
 - page groups 187
 - status 171
 - types 190
 - viewing or editing 197
 - pages, viewing properties 197
 - parameters 86
 - @OP 89
 - CGI 87
 - passwordExpirationWarningWindow global property 266
 - performing concept-based searches 326
 - permissions
 - role needed to find and replace 23
 - personalization
 - adapters 149
 - keys, creating 151, 154
 - keys, importing and exporting 161
 - values, entering user-specific in Portal Studio 160
 - personalization adapters 88
 - pie chart 145
 - PollingAction, autoindexing parameter 339
 - PollingMaxNumber, autoindexing parameter 340
 - PollingMethod, autoindexing parameter 338
 - PollingPeriod, autoindexing parameter 338
 - PollingPostAction, autoindexing parameter 340
 - Portlet Builder 41
 - portal
 - authentication data, entering 381
 - content, managing with Content Explorer 393
 - customizing 196
 - Portal Interface
 - channels 323
 - logging in after modifying custom_*.jsp 24
 - subchannels 323
 - portal objects, deploying 235
 - portal page, creating and populating 194
 - portal pages
 - adding XML portlets to 174
 - portal search
 - automatic indexing 334
 - connection cache 334
 - Portal Security Officer password 416
 - Portal Studio
 - deploying objects from 239
 - interface 5
 - logging in 4
 - login 416
 - pages, creating 362
 - portal.defaultRID global property 266
 - portal.epSecurity global property 266
 - portal.host global property 266
 - portal.portlet_create_lock global property 266
 - PortalAdmin role 18, 244, 266
 - PortalAdministrationRole global property 266
 - portals, multiple 247
 - PortalSearchAutoindexer** 334
 - PortalSearchautoindexer** 335
 - PortalSearchautoindexer.cfg* 335, 337
 - Portlet Builder, element list 42
 - portlet labels
 - mapping 91
 - Portlet Manager 40
 - portlet refresh interval 83, 398

- portlets
 - @OP parameters 89
 - ACX capture option 51
 - adding to a catalog 183
 - aggregate 48
 - application 106
 - CGI parameters 87
 - charting 141
 - creating a new page for a secured JSP portlet 374
 - creating a secured JSP portlet and deploying 372
 - creating charts 142
 - creating messaging 96
 - creating select lists for input fields 136
 - creating single sign-on using Enterprise Security 375
 - creating single sign-on with Web applications 368
 - creating XML 173
 - customizing the look and feel 196
 - database 54
 - data-capable elements 132
 - dragging and dropping on a page 192
 - editing 85
 - element list 42
 - enabling support for multiple languages 137
 - entering portal authentication data 381
 - filter rules 134
 - finding 91
 - HTML elements 74
 - JSP 56
 - JSP elements 56
 - JSP SSO check box 62
 - JSP using Secure Web Proxy 59
 - labels 90
 - management functions 85
 - messaging 96
 - messaging and click-across 96
 - multiple element 48
 - properties 86
 - publishing to a UDDI registry as a Web service 71
 - record layout 134
 - removing 197
 - removing from page 197
 - role needed to find and replace 23
 - saving 82
 - secure 79
 - single sign-on, creating 357
 - split rules 133
 - Template Manager 163
 - types 41
 - using iFrames 82, 397
 - viewer 43
 - Web 49
 - Web elements from secure sites 79
 - with personalization adapters 152
 - XML 52, 54
 - production environment
 - settings 274, 276, 277, 278
 - programming
 - single sign-on JSPs 383
 - proper names
 - searching by 314
 - properties
 - catalog 185
 - changing catalog 185
 - configuring global 261
 - portlet 86
 - properties, editing portlet 86
 - properties, global 283
 - protected JSPs 371
 - protecting assets 382
 - proxy authentication information 363
 - creating 364
 - proxy global property 267
 - proxy properties 286
 - proxy servers, enabling 286
 - proxy.bypass_list global property 267
 - proxy.host global property 267
 - proxy.password global property 267
 - proxy.port global property 267
 - proxy.user global property 267
 - PSTHost global property 277
 - PSPort global property 277
 - publishing portlets to a UDDI registry as a Web service 71
 - PushAgentLog global property 277
- Q**
- queries, stored 322

QueryPort, autoindexing parameter 339

R

record layout 134
redirfile global property 267
registration.organization global property
 global property
 registration.organization 267
registration.userJ2EERoles global property 267
remote server, deploying portal objects to 239
removeChannel 308
removeDatabaseFromChannel 316
removeDatabaseFromUserAgent 315
RemoveLogFileOnStart, autoindexing parameter 339
removeRoleFromChannel 309
removeRoleFromUserAgent 309
removeSubChannel 309
removeUserAgent 310
renameAgent 310
renaming
 catalog branches 186
Replacing portlets
 necessary permission 23
reprocessing files and automatic indexing 336
RequiredRoles global property 267
resetPasswordEmail global property 267
restarting the automatic indexing utility 337
retrainSubChannelByResults 310
retrainUserAgentByResults 311
RoleBaseDisplay global property 268
RoleBaseDisplaySeeAllRoles global property 268
roles
 PortalAdmin 18, 266
running
 agents once 213

S

sample codes
 search 311
sample portlet, Content Explorer 390
samples 357
 address book 395

Portal Interface audit records 34
 search EJB 387
 search portlets 388
 single sign-on 357
saving
 elements 82
 portlets 82
Search
 example 311
search
 automatic indexing utility 321
 by proper name 314
 concept-based 326
 EJB 387
 finding portlets 91
 portlet samples 388
search APIs
 addDatabaseForChannel 315
 addDatabaseForUserAgent 311, 315
 addRoleForChannel 300
 addRoleForUserAgent 301
 advanced 314
 cleanupUserMemory 301
 copyUserAgent 301
 createAgent 302
 createChannel 302
 doAdvancedQuery 317
 doQueryWithAgent 303
 editAgent 303
 existsDatabase 319
 getCountOfChannels 304
 getCountOfDatabases 318
 getCountOfDatabasesForChannel 317
 getCountOfDatabasesForUserAgent 316
 getCountOfRolesForChannel 304
 getCountOfRolesForUserAgent 304
 getCountOfSubChannels 305
 getCountOfUserAgents 305
 getCurrentSubChannelAge 305
 getCurrentSubChannelMaxDocs 305
 getCurrentSubChannelRelevance 306
 getCurrentSubChannelTrainingText 306
 getCurrentUserAgentAge 306
 getCurrentUserAgentMaxDocs 306
 getCurrentUserAgentRelevance 307
 getCurrentUserAgentTrainingText 307

- getNextChannelName 307
 - getNextDatabaseForChannel 317
 - getNextDatabaseForDre 319
 - getNextDatabaseForUserAgent 316
 - getNextRoleForChannel 307
 - getNextRoleForCurrentUserAgent 308
 - getNextSubChannelName 308
 - getNextUserAgentName 308
 - removeChannel 308
 - removeDatabaseFromChannel 316
 - removeDatabaseFromUserAgent 315
 - removeRoleFromChannel 309
 - removeRoleFromUserAgent 309
 - removeSubChannel 309
 - removeUserAgent 310
 - renameAgent 310
 - retrainSubChannelByResults 310
 - retrainUserAgentByResults 311
 - SearchHome class 292–311
 - searching 289–313
 - secure global property 268
 - secure portlets 79
 - setting up EAServer 79
 - Secure Web Proxy 59
 - secure.title.host global property 268
 - secure_login global property 268
 - SecureHostname global property 268
 - security 323
 - creating assets 153
 - granting asset permissions 153
 - security configuration, DRE generic 330
 - SECURITY environment variable xviii
 - Security Manager 364
 - security proxy adapter 150, 152
 - selecting, applications 6
 - send.mail global property 268
 - send.mail property 284
 - sending
 - e-mail to users 284
 - sending e-mail
 - to a file 284
 - server_virtual_path global property 269
 - service keys, using to create Web service element 70
 - setupUserMemory 311
 - sharePage global property 269
 - single sign-on
 - creating JSP portlet and deploying 372
 - creating portlets with Web applications 368
 - deploying the secured Web application 371
 - deployment descriptor 369
 - entering portal authentication data 381
 - JSP programming 383
 - single sign-on sample 357
 - SOAPServer global property 269
 - software updates on the Web xvii
 - sorting queries in the DRE 332
 - SSL 79
 - starting
 - DRE administration tool 333
 - starting Portal Studio 4
 - StartQuery.sh** command 327
 - status
 - changing agent 213
 - changing catalog 185
 - page 171
 - StatusUpdateFreq global property 277
 - stopping
 - automatic indexing utility 336
 - StorageConnector global property 273
 - stored queries 322
 - styles.xml 257
 - styles.xml file, editing 255
 - subchannels
 - adding 324
 - training 325
 - SYBASE environment variable xviii
- ## T
- tabingChar global property 269
 - tables
 - filter rules 134
 - record layout 134
 - split rules 133
 - tabset global property 269
 - Technical Library CD xvi
 - template
 - changing active status 171
 - deleting 171
 - Template Manager 163, 170, 171, ??–172
 - building templates 165

- changing default express portlet templates 172
- deleting templates 171
- described 163
- editing templates 171
- help templates 167
- interface 164
- portletlets and 163
- Template Upload window 164
- XSL templates 169
- templates
 - building new 165
 - changing
 - workflow status** 171
 - changing default express portlet template 172
 - creating custom user-specific XML 173
 - creating new 170
 - default 48
 - deleting 171
 - enabling different for the same element 175
 - help 167
 - HTML 165
 - JSP 168
 - Template Manager interface 164
- TextImportExtns, autoindexing parameter 341
- ThisMachineName global property 273
- tile.host,global property 269
- tile_style global property 270
- TimeZone global property 277
- trace property, disabling for production 286
- TraceFile global property 273, 276, 277
- TraceOn global property 273, 276, 277
- training subchannels 325
- TransactionFile global property 278
- troubleshooting 413
 - ActiveX and portlet creation on UNIX 413
 - agent log fills up the database 414
 - auditing 414
 - auto indexer 414
 - charts 415
 - co-brands 415
 - ortal Security Officer password 416
 - Portal Studio login 416
- turnOffSecurePassword global property 270
- typographical conventions xviii

U

- UDDI registries
 - creating new 73
 - publishing portlets to as a Web service 71
- updating
 - catalog changes to Portal Interface 186
 - software via the Web xvii
- updating global properties 283
- use_https global property 270
- use_include global property 280
- UseAccessControl global property 276
- UsePreCompiledJSP global property 274
- user agents 322
- user name/password
 - adapter, using 158
 - adapters 150
- UseStudioDatabase global property 276
- UTF-8 137
 - and Web browsers 139
- utilities
 - automatic indexing utility 321
 - isql** 287
 - PortalSearchAutoindexer** 334
- UWP, *global.properties.xml* property group 272
- UWPCacheServer global property 274
- UWPIntegrationHost global property 274
- UWPWebServiceHost global property 274

V

- VerboseTrace global property 274, 276, 277, 278
- versioning
 - global property 270
- viewing
 - catalog properties 185
 - page properties 197
 - portletlets 43
 - portlet properties 86

W

- WAR files, creating deployable 371
- Web 368
 - applications, creating secured 369, 375

- applications, deploying secured 371
- browsers and UTF-8 139
- service elements 62
- Web applications
 - creating secure 358
 - creating WAR files 360
 - deploying WAR files 360
 - JSPs, testing as 360
 - web.xml* definition file 358
- Web elements 49, 79
- Web service elements
 - creating 69
 - creating new UDDI registry entries 73
 - creating using a service key 70
 - publishing to a UDDI registry 71
- web.xml* Web application file 358
- WebServerOn global property 276
- WebServiceRoot global property 271
- wireless_portal global property 271
- WorkRoot global property 271
- WSDL, using to create Web service elements 69

X

- XML
 - adding portlets to a page 174
 - creating portlets 173
 - creating user-specific XSLT templates 175
 - templates, creating custom user-specific 173
- XML file
 - exporting to 238
- XML files
 - importing into Portal Studio 236
- XML output DTD 45
- XML portlets 52, 54
- xml_catalog_serialize global property 271
- XSL templates
 - creating 169
 - sample 169
- XSLT
 - templates, creating user-specific 175
- XSLT, creating 169

