
Enterprise PeopleTools 8.50 PeopleBook: System and Server Administration

September 2009

Copyright © 1988, 2009, Oracle and/or its affiliates. All rights reserved.

Trademark Notice

Oracle is a registered trademark of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.

License Restrictions Warranty/Consequential Damages Disclaimer

This software and related documentation are provided under a license agreement containing restrictions on use and disclosure and are protected by intellectual property laws. Except as expressly permitted in your license agreement or allowed by law, you may not use, copy, reproduce, translate, broadcast, modify, license, transmit, distribute, exhibit, perform, publish or display any part, in any form, or by any means. Reverse engineering, disassembly, or decompilation of this software, unless required by law for interoperability, is prohibited.

Warranty Disclaimer

The information contained herein is subject to change without notice and is not warranted to be error-free. If you find any errors, please report them to us in writing.

Restricted Rights Notice

If this software or related documentation is delivered to the U.S. Government or anyone licensing it on behalf of the U.S. Government, the following notice is applicable:

U.S. GOVERNMENT RIGHTS

Programs, software, databases, and related documentation and technical data delivered to U.S. Government customers are "commercial computer software" or "commercial technical data" pursuant to the applicable Federal Acquisition Regulation and agency-specific supplemental regulations. As such, the use, duplication, disclosure, modification, and adaptation shall be subject to the restrictions and license terms set forth in the applicable Government contract, and, to the extent applicable by the terms of the Government contract, the additional rights set forth in FAR 52.227-19, Commercial Computer Software License (December 2007). Oracle USA, Inc., 500 Oracle Parkway, Redwood City, CA 94065.

Hazardous Applications Notice

This software is developed for general use in a variety of information management applications. It is not developed or intended for use in any inherently dangerous applications, including applications which may create a risk of personal injury. If you use this software in dangerous applications, then you shall be responsible to take all appropriate fail-safe, backup, redundancy and other measures to ensure the safe use of this software. Oracle Corporation and its affiliates disclaim any liability for any damages caused by use of this software in dangerous applications.

Third Party Content, Products, and Services Disclaimer

This software and documentation may provide access to or information on content, products and services from third parties. Oracle Corporation and its affiliates are not responsible for and expressly disclaim all warranties of any kind with respect to third party content, products and services. Oracle Corporation and its affiliates will not be responsible for any loss, costs, or damages incurred due to your access to or use of third party content, products or services.

Contents

Preface

System and Server Administration Preface	xix
System and Server Administration	xix

Chapter 1

Getting Started with System and Server Administration	1
System and Server Administration Overview	1
PSADMIN	1
Analytic Servers	2
Web Servers	3
Search Indexes	3
PeopleSoft Configuration Manager	4
PeopleTools Utilities	5
Tracing and Debugging	5
Jolt Configuration Options	6
Timeout Settings	6
System and Server Administration Implementation	6

Chapter 2

Understanding PeopleSoft Internet Architecture	9
PeopleSoft Architecture Fundamentals	9
Database Server	10
Application Servers	11
Application Servers	11
Oracle Tuxedo and Oracle Jolt	11
Domains	12
PeopleSoft Server Processes	12
Services	14
Listeners, Handlers, and Queues	14
PeopleSoft Process Scheduler Server	17
Web Server	18
Web server software	18
PeopleSoft Servlets	19

Oracle Jolt	20
Web Browser	21
Server Configuration Options	21
Logically Separate Server Configuration	21
Physically Separate Server Configuration	23
Implementation Options	24

Chapter 3

Working with Server Domain Configurations	27
Understanding PS_HOME and PS_CFG_HOME	27
Implementing Flexible Server Installations	28
Applying Security Restrictions	29
Working with the Default PS_CFG_HOME	29
Locating the Default PS_CFG_HOME	29
Using PSADMIN with the Default PS_CFG_HOME	30
Working with Alternate PS_CFG_HOME Locations	31
Specifying Alternate PS_CFG_HOME Locations	31
Using the %V Meta Variable	31
Configuring Domains in Alternate Locations of PS_CFG_HOME	32
Managing Domains	33

Chapter 4

Using the PSADMIN Utility	35
Understanding PSADMIN	35
Starting PSADMIN	36
Using PSADMIN	37
Using Configuration Templates	37
Using the Quick-Configure Menu	38
Using the PSADMIN Command-Line Interface	39
Understanding the PSADMIN Command-Line Interface	39
Using the Miscellaneous Commands	40
Using the Application Server Commands	40
Using the Process Scheduler Commands	45
Using the Search Server Commands	49
Using PSADMIN Executables and Configuration Files	50
Understanding PSADMIN Executables and Configuration Files	50
Configuring a Domain	51
Loading a Configuration	52
Archiving Application Server Configuration Files	52
Booting a Domain	53

Stopping a Domain	53
Monitoring a Domain	53
Configuring the Application Server to Handle Cache Files and Replay Files	53

Chapter 5

Using PSADMIN Menus	57
Using the Application Server Administration Menu	57
Accessing the Application Server Options	58
Administering a Domain	58
Importing Domain Configurations	59
Booting a Domain	60
Shutting Down a Domain	60
Checking the Domain Status	61
Purging the Domain Cache	62
Configuring a Domain	64
Editing Configuration and Log Files	65
Creating a Domain	66
Deleting a Domain	66
Configuring an Application Server Domain to Preload Non-Shared Cache	67
Cleaning Domain IPC Resources	69
Using the Process Scheduler Menu	69
Understanding the Process Scheduler Menu	70
Starting a Process Scheduler Server	70
Stopping a Process Scheduler Server	70
Configuring a Process Scheduler Server	71
Creating a Process Scheduler Server Configuration	71
Deleting a Process Scheduler Server	71
Editing the Process Scheduler Configuration File	72
Using the Process Scheduler Options	72
Using Process Scheduler Command-Line Options	72
Cleaning Domain IPC Resources	73
Using the Search Server Menu	73
Setting Up the PeopleSoft Windows Service	73
Understanding Microsoft Windows Services	73
Configuring the PeopleSoft Service	74
Testing the Service	75
Editing the Service Configuration File	76

Chapter 6

Setting Application Server Domain Parameters	77
---	-----------

Startup Options	78
DBName	78
DBType	78
UserID	78
UserPswd	79
Connect ID	79
Connect Password	79
ServerName	79
Database Options	79
SybasePacketSize	79
UseLocalOracleDB	80
EnableDBMonitoring	80
PSDB Maximum Cursors	80
Security Options	81
Validate Signon With Database	81
Workstation Listener Options	81
Address	81
Port	82
Encryption	82
Min Handlers	82
Max Handlers	82
Max Clients per Handler	82
Client Cleanup Timeout	83
Init Timeout	83
Tuxedo Compression	83
Jolt Listener Options	83
Address	83
Port	83
Encryption	84
Min Handlers	84
Max Handlers	84
Max Clients per Handler	84
Client Cleanup Timeout	84
Init Timeout	84
Client Connection Mode	84
Jolt Compression Threshold	85
Additional Prompt	85
Jolt Relay Adapter Options	86
Listener Address	86
Listener Port	86
Domain Settings	86
Domain ID	86
Add to PATH	86
Spawn Threshold	87
Log Directory	87

Restartable	88
Allow Dynamic Changes	88
LogFence	89
AppLogFence	89
Trace-Log File Character Set	89
PeopleCode Debugger Options	89
Trace Options	90
TraceSQL	90
TraceSQLMask	90
TracePC	90
TracePCMask	91
TracePPR and TracePPRMask	91
TracePIA and TracePIAMask	92
TraceAE	92
TraceAnalytic and Trace AnalyticMask	92
TracePPM	93
DumpMemoryImageAtCrash	93
DumpMemoryObjectsAtCrash	93
Log Error Report, Mail Error Report	93
Write Crash Dump to Separate File	93
Cache Settings	94
Cache Settings	94
EnableServerCaching	94
ServerCacheMode	94
CacheBaseDir	95
MaxCacheMemory	96
PreLoadFileCache and PreLoadMemoryCache	96
Remote Call Options	96
RCCBL Redirect	96
RCCBL PRDBIN	97
PSAPPSRV Options	97
Min Instances	97
Max Instances	98
Service Timeout	98
Recycle Count	98
Percentage of Memory Growth	98
Allowed Consec Service Failures	99
Max Fetch Size	100
Auto Select Prompt	100
Tuxedo Queue Size	100
PSANALYTICSRV Options	100
Min Instances	100
Max Instances	100
Analytic Instance Idle Timeout	100
PSSAMSRV Options	101

Min Instances	101
Max Instances	101
Service Timeout	101
Recycle Count	101
Allowed Consec Service Failures	101
Max Fetch Size	102
PSQCKSRV Options	102
Min Instances	102
Max Instances	102
Service Timeout	102
Recycle Count	102
Allowed Consec Service Failures	103
Max Fetch Size	103
PSQRYSRV Options	103
Min Instances	103
Max Instances	103
Service Timeout	103
Recycle Count	104
Allowed Consec Service Failures	104
Max Fetch Size	104
Use Dirty-Read	104
Integration Broker Server Processes	105
SMTP Settings	105
SMTPServer	105
SMTPPort	106
SMTPServer1	106
SMTPPort1	106
SMTPSender	106
SMTP BlackberryReplyTo	106
SMTPSourceMachine	106
SMTPCharacterSet	106
SMTPEncodingDLL	106
SMTPGuaranteed	107
SMTPTrace	107
SMTPSendTime	107
SMTPUserName	107
SMTPUserPassword	107
SMTPUserName1	107
SMTPUserPassword1	107
SMTPTimeToWaitForResult	108
SMTPSSLPort	108
SMTPUseSSL	108
SMTPClientCertAlias	108
SMTPSSLPort1	108
SMTPUseSSL1	108

SMTPClientCertAlias1	108
SMTPDNSTimeoutRetries	108
SMTP Further Considerations	109
Interface Driver Options	109
SCP_LOCALE	109
PSTOOLS Options	109
EnablePPM Agent	109
Add to CLASSPATH	110
JavaVM Options	110
Proxy Host	110
Proxy Port	111
Non Proxy Hosts	111
Character Set (UNIX or USS Only)	111
Suppress App Error Box (Microsoft Windows Only)	112
DbFlags	112
Suppress SQL Error	113
Integration Broker Options	114
Min Message Size for Compression	114
Thread Pool Size	114
Search	114
Search Indexes	114
PSRENSRV Options	115
log-severity_level	115
io_buffer_size	115
default_http_port	115
default_https_port	115
default_auth_token	115
PSPPMSSRV Options	115
Min Instances	116
Max Instances	116
Select Server Process Options	116
Do you want the Publish/Subscribe servers configured?	116
Move quick PSAPPSRV services into a second server (PSQCKSRV)?	116
Move long-running queries into a second server (PSQRYSRV)?	116
Do you want JOLT configured?	117
Do you want JRAD configured?	117
Do you want WSL Configured?	117
Do you want to enable PeopleCode Debugging?	117
Do you want Event Notification configured?	117
Do you want MCF Servers configured?	117
Do you want Performance Collators configured?	117
Do you want Analytic Servers configured?	118
Do you want Domains Gateway configured?	118

Chapter 7

Working with Oracle WebLogic	119
Understanding WebLogic	119
Working With the WebLogic Server Administration Console	120
Starting WebLogic	121
Starting WebLogic on Microsoft Windows	121
Starting WebLogic on UNIX	123
Stopping WebLogic	123
Stopping WebLogic Using the Administration Console	123
Stopping WebLogic Using the Command Line	124
Using WebLogic Server Administration Console to Monitor PeopleSoft Sessions	124
Setting Up Reverse Proxy Servers	125
Understanding Reverse Proxy Servers For PeopleSoft Implementations	126
Configuring Microsoft IIS as an RPS	126
Configuring WebLogic as an RPS	128
Configuring Sun Java System Web Server as an RPS	131
Configuring Apache HTTP as an RPS	134
Setting The HTTP Session Timeout	135
Setting Authentication Failure Timeout	135
Enabling or Disabling HTTP Keep Alive	135
Changing WebLogic User Passwords	136
Implementing WebLogic SSL Keys and Certificates	138
Understanding SSL Encryption with WebLogic	138
Obtaining Encryption Keys	138
Preparing Keys and Certificates for the Keystore	141
Importing Keys and Certificates Into the Keystore	143
Configuring WebLogic SSL Encryption Keys	145
Working With WebLogic Session Cookies	147
Securing Servlets on WebLogic	147
Adjusting the JVM Heap Size	149
Determining the Service Pack Level	150
Enabling HTTP Access Log	151

Chapter 8

Working with IBM WebSphere	153
Understanding WebSphere Application Server Within Your PeopleSoft Implementation	153
Deploying PeopleSoft Applications With WebSphere	153
Using The Integrated Solutions Console	154
WebSphere Application Server Profiles	155

IBM HTTP Server	156
Starting and Stopping WebSphere Application Servers	157
Starting the WebSphere Server	157
Stopping the WebSphere Server	157
Configuring Reverse Proxy Servers For WebSphere	157
Understanding Reverse Proxy Servers With IBM WebSphere	158
Configuring IBM HTTP Server as a Reverse Proxy Server	158
Configuring Microsoft IIS as a Reverse Proxy Server	159
Configuring Sun Java System Web Server as a Reverse Proxy Server	161
Setting Up SSL For WebSphere	163
Understanding WebSphere Key Stores	163
Generating a Certificate Using pskeymanager	163
Configuring the WebSphere Container to Support SSL	165
Securing The Administrative Console and Applications For WebSphere	166
Understanding WebSphere Security	166
Securing the Administrative Console	168
Configuring Application Security	170
Setting HTTP Session Timeout	173
Setting Authentication Failure Timeout	173
Working With JVM Heap Size	173
Working with Logging and Tracing Options	174
Enabling HTTP Access and HTTP Error Logging	174
Enabling General Logging and Tracing	175

Chapter 9

Configuring Search and Building Search Indexes	177
Understanding PeopleSoft Search Indexes	177
Overview of Search Indexes	177
Types of Indexes	178
Components of the Search Architecture	178
Index Building	180
Search Index Limitations	181
User Search Strategies	182
Configuring PeopleSoft Search	183
Understanding PeopleSoft Search Configurations	183
Configuring Search to run within the Application Server (Type-1)	185
Configuring Search to Run as a Separate Process (Type-2)	185
Configuring a Separate Search Server (Type-3)	186
Search Server Administration	190
Working with Indexes	192
Understanding Common Controls	192
Understanding Supported MIME Types	193

Opening Existing Collections	194
Creating New Collections	194
Building Record-Based Indexes	195
Modifying Record-Based Index Properties	195
Adding Subrecords to Search Indexes	198
Building File System (Spider) Indexes	199
Setting File System Options	199
Defining What to Index	200
Building HTTP Spider Indexes	201
Defining HTTP Gateway Settings	201
Defining What to Index	203
Administering Search Indexes	203
Specifying the Index Location	203
Administering the Search Index	204
Editing Properties	205
Scheduling Administration	206
Sharing Indexes Between Application Servers and PeopleSoft Process Scheduler	206
Modifying the VdkVgwKey Key	207

Chapter 10

Using PeopleSoft Configuration Manager	209
Understanding PeopleSoft Configuration Manager	209
Common Elements in PeopleSoft Configuration Manager	210
Starting PeopleSoft Configuration Manager	210
Specifying Startup Settings	210
Specifying Display Settings	212
Specifying Crystal Report, Business Interlink, and JDeveloper Settings	214
Specifying Trace Settings	215
Specifying Workflow Settings	216
Specifying Remote Call/AE Settings	216
Configuring Developer Workstations	217
Importing and Exporting Environment Settings	218
Configuring User Profiles	219
Defining a Profile	219
Specifying Databases and Application Servers	220
Configuring Process Scheduler	222
Configuring nVision	223
Specifying Common Settings	226
Specifying Command Line Options	227
Setting Up the PeopleTools Development Environment	228
Understanding the PeopleTools Development Environment	228
Understanding the Client Setup Process	229

Verifying PS_HOME Access	229
Verifying Connectivity	229
Verify Supporting Applications	229
Using the Configuration Manager	230
Running the Client Setup Process	230

Chapter 11

Using PeopleTools Utilities	233
Understanding the PeopleTools Utilities	233
Using the System Information Page	233
Understanding the System Information Page	234
Viewing the System Information Page	234
Using Administration Utilities	236
PeopleTools Options	237
Message Catalog	247
Spell Check System Dictionary	249
Translate Values	250
Load Application Server Cache	251
Tablespace Utilities	255
Tablespace Management	256
DDL Model Defaults	256
BOE Integration Administration	258
Strings Table	259
Lookup Exclusion	260
XML Link Function Registry	260
Merchant Integration Utilities	260
TableSet IDs	260
Record Group	261
TableSet Control	261
Convert Panels to Pages	262
Update Utilities	265
Remote Database Connection	265
URL Maintenance	266
Copy File Attachments	269
Query Administration	270
Sync ID Utilities	270
nVision Report Request Admin	270
Analytic Server Administration	270
Upgrade Conversion	271
Analytic Model Viewer	271
Analytic Instance Load/Unload	271
Analytic Instance Create/Del/Copy/	271

Pre-Load Cache Utilities	271
Gather Utility	271
QAS Administration	273
Oracle Resource Management	273
Using Audit Utilities	273
Using the Record Cross Reference Component	274
Performing a System Audit	275
Performing Database Level Auditing	276
Using Debug Utilities	276
Using the PeopleTools Test Utilities Page	276
Replay Appserver Crash	277
Using the Trace PeopleCode Utility	278
Using the Trace SQL Utility	278
Using International Utilities	278
Setting International Preferences	278
Setting Process Field Size	279
Administering Time Zones	279
Managing Languages	279
Using Optimization Utilities	281
Using PeopleSoft Ping	281
PeopleSoft Ping Chart	282
PeopleSoft Ping Delete	283
PeopleSoft Ping Options	283

Chapter 12

Tracing, Logging, and Debugging	285
Setting Up the PeopleCode Debugger	285
Debugging for a Two-Tier Connection	285
Debugging for a Three-Tier Connection	286
Using the PeopleCode Debugger	288
Configuring PeopleCode Trace	288
Configuring SQL Trace	290
Enabling IDDA Logging	290
Understanding IDDA Logging	291
Enabling IDDA Logging	292
Working with IDDA Functional Categories	292
Configuring Logging Options	293
Viewing IDDA Logging Output	294

Chapter 13

Working with Jolt Configuration Options	297
Configuring Jolt Failover and Load Balancing	297
Configuring Weighted Load Balancing	297
Configuring Jolt Failover	297
Configuring Jolt Session Pooling	298
Understanding Jolt Internet Relay	298
Jolt Internet Relay Architecture	299
Implementation Considerations	300
Configuring JRLY	300
Configuring JRAD	302
Running Jolt Relay	303
Using the JRLY Administration Program	303
Running Jolt Relay on Windows	303
Running Jolt Relay on UNIX	304

Appendix A

Securing PS_HOME and PS_CFG_HOME	305
Understanding PS_HOME and PS_CFG_HOME Security	305
Understanding PS_HOME Security	305
Understanding Minimum Access Required by The User Starting Domains	306
Understanding PS_CFG_HOME Security	307
Securing PS_HOME on UNIX	307
Managing a Secure PS_HOME on UNIX	308
Working with User Accounts	308
Configuring Partial PS_HOME Access	308
Securing PS_HOME on Windows	309
Multiple Administrator User Accounts	309
Local User Accounts	311
Managing a Secure PS_HOME on Windows	313
Working With Mapped Drives, UNC Paths, and TM_TUXIPC_MAPDRIVER	313
Working With Oracle ProcMGR Service	314
Managing TM_TUXIPC_MAPDRIVER	314
Resolving Initialization Timeout Issues	314
Implementing PS_CFG_HOME Security	315
Securing PS_CFG_HOME on UNIX	315
Securing PS_CFG_HOME on Windows	316

Appendix B

WebLogic Managed Server Architecture	319
PeopleSoft Internet Architecture Servlets and Applications	319
WebLogic Domain Types	320
Understanding WebLogic Domain Types	320
Single-Server Domains	320
Multi-Server Domains	322
Distributed Managed Servers	325
Common Default Settings	327
WebLogic Domain Directory Structure and Files	331
WebLogic Domain Directory Structure	331
WebLogic Domain File Listing by Type	332
J2EE Application Files	335
PIA Install and Reinstall Options	336
Administering a WebLogic Server Life Cycle	337
Understanding the WebLogic Server Life Cycle	337
Starting and Stopping Single-Server Processes	337
Starting and Stopping Multi-Server Processes	338
Starting and Stopping a Distributed Managed Server	342
Tuning Performance and Monitoring Resources	342
Managing JVM Heap Size	343
Monitoring HTTP Session Count for PeopleSoft Portal	344
Changing Configuration Settings	345
Understanding the WebLogic Server Configuration Files	345
Changing the WebLogicAdmin Server's Listen Ports	345
Changing Application and Server Deployment Targets	346

Appendix C

PeopleSoft Timeout Settings	349
Web Server Timeouts	349
Session-Timeout	351
Web Server Default System Timeout	351
Application Server Timeouts	352
Process Scheduler Timeouts	354
Search Server Timeouts	354
PIA Timeouts	356

Appendix D

Troubleshooting Server Issues 357

Uploading Files Using Non-Latin Characters 357

 Solution For UNIX 357

 Solution For Windows 357

WebSphere: Port Set In the Host Header of a Request Returned Incorrectly 358

 Scenario 358

 Solution 358

Index 359

System and Server Administration

Preface

This preface provides an overview of the contents discussed in the System and Server Administration PeopleBook and discusses PeopleBooks and the Online PeopleSoft Library.

System and Server Administration

This book includes several chapters relating to administration tools for the PeopleSoft application server, and web servers. It also contains information about building and maintaining search indexes, database level auditing, and PeopleTools utilities.

Note. PeopleSoft supports a number of versions of UNIX and Linux in addition to Microsoft Windows. Throughout this book, there are references to operating system configuration requirements. Where necessary, this book refers to specific operating systems by name (for example, Solaris, HP/UX, Linux, etc.). However, for simplicity the word UNIX is used to refer to all UNIX-like operating systems, including Linux.

PeopleBooks and the Online PeopleSoft Library

A companion PeopleBook called PeopleBooks and the Online PeopleSoft Library contains general information, including:

- Understanding the PeopleSoft online library and related documentation.
- How to send PeopleSoft documentation comments and suggestions to Oracle.
- How to access hosted PeopleBooks, downloadable HTML PeopleBooks, and downloadable PDF PeopleBooks as well as documentation updates.
- Understanding PeopleBook structure.
- Typographical conventions and visual cues used in PeopleBooks.
- ISO country codes and currency codes.
- PeopleBooks that are common across multiple applications.
- Common elements used in PeopleBooks.
- Navigating the PeopleBooks interface and searching the PeopleSoft online library.
- Displaying and printing screen shots and graphics in PeopleBooks.
- How to manage the PeopleSoft online library including full-text searching and configuring a reverse proxy server.
- Understanding documentation integration and how to integrate customized documentation into the library.

- Glossary of useful PeopleSoft terms that are used in PeopleBooks.

You can find this companion PeopleBook in your PeopleSoft online library.

Chapter 1

Getting Started with System and Server Administration

This chapter provides an overview of system and server administration and discusses system and server administration implementation.

System and Server Administration Overview

This section discusses:

- PSADMIN.
- Analytic servers.
- Web servers.
- Search indexes.
- PeopleSoft Configuration Manager.
- PeopleTools utilities.
- Tracing and debugging.
- Jolt Internet Relay.
- Timeout settings.

PSADMIN

You use PSADMIN for managing application server domains, PeopleSoft Process Scheduler domains, integration server processes, search domains, and so on. PSADMIN also enables you to configure and manage the behavior of servers with respect to a wide range of PeopleTools infrastructure elements, including:

- Tuxedo and Jolt.
- PeopleCode debugging.
- Caching.
- Analytic server framework.

- Transactional SQL requests.
- Performance enhancement.
- PeopleSoft Query.
- Integration Broker.
- Email.
- Real time event notification.
- Performance Monitor.
- MultiChannel Framework.

You launch and run PSADMIN using a command line interface.

See Also

[Chapter 4, "Using the PSADMIN Utility," page 35](#)

[Chapter 5, "Using PSADMIN Menus," page 57](#)

[Chapter 6, "Setting Application Server Domain Parameters," page 77](#)

Analytic Servers

The *analytic server framework* provided by PeopleSoft is a general server infrastructure designed to meet the needs of PeopleSoft products that process large amounts of data in memory. It provides a stateful model of client/server connectivity that these products require to be part of the PeopleTools system, by keeping track of configuration settings, transaction information, and other data for a session. For example, client software could request that an analytic model or optimization model be recalculated in one transaction, then retrieve the results of the calculation on that model at a later time. A server process handles these requests, and maintains the model state and calculated data in memory between the requests. Additional transactions can then modify the model and perform recalculations on it without shuffling all of the data between the client and the server or dumping all the data to a database, thus preserving in-memory performance.

When a program doesn't "maintain state" or when the infrastructure of a system prevents a program from maintaining state, it's known as a *stateless* program or system. It can't take information about the last session into the next session, such as settings the user makes or conditions that arise during processing. All session state is maintained by the client and is transferred to the server with each request. As long as an application server is up and running, a user's session remains active and functional, and any application server can perform requested transactions.

However, with some products, such as Analytic Calculation Engine or PeopleSoft Optimization Framework, running a calculation on a multi-dimensional model is likely to produce far more data than is reasonable to shuttle between a client and server to maintain a stateless connection. For performance reasons, the calculations are performed completely in memory. If these calculations were to be synchronized and stored in the database so that a stateless connection could be maintained, performance would suffer significantly.

Web Servers

PeopleSoft supports Oracle WebLogic and IBM WebSphere web servers, which provide the same basic functionality to support PeopleSoft applications, including a console interface, secure sockets layer (SSL), and reverse proxy servers (RPS).

Each web server has its own way of accomplishing its functionality, and each adds its own extra features that you might find useful to your PeopleSoft system. This PeopleBook provides supplemental information about configuring and administering the web servers where it has particular relevance to PeopleSoft.

Note. The information in this PeopleBook is not intended to replace any Oracle WebLogic or IBM WebSphere documentation. Always refer to the manufacturer's documentation for detailed information about your web server.

See Also

[Chapter 7, "Working with Oracle WebLogic," page 119](#)

[Chapter 8, "Working with IBM WebSphere," page 153](#)

[Appendix B, "WebLogic Managed Server Architecture," page 319](#)

Search Indexes

A search index is a collection of files that is used during a search to quickly find documents of interest. You build a search index to enable searching on a given set of documents. The set of files that make up the index is a *collection*. This collection contains a list of words in the indexed documents, an internal documents table containing document field information, and logical pointers to the actual document files.

Fields contain metadata about a document. For example, Author and Title might be fields in an index. *VdkVgwKey* is a special field that identifies each document and is unique to all of the documents in the collection.

Every search index can be modified by changing the configuration files that are associated with the index. These configuration files are known as *style* files and reside in the style directory under the database directory. A typical configuration of style files define fields for a particular index.

PeopleSoft software supports these types of search indexes:

- Record-based indexes.

Record-based indexes are used to create indexes of data in PeopleSoft tables. For example, if the PeopleSoft application has a catalog record that has two fields (Description and PartID), you can create a record-based index to index the contents of the Description and PartID fields.

- HTTP spider indexes.

HTTP spider indexes index a web repository by accessing the documents from a web server. You typically specify the starting uniform resource locator (URL). The indexer walks through all documents by following the document links and indexes the documents in that repository. You can control to what depth the indexer should traverse.

- File system indexes.

File system indexes are similar to HTTP spider indexes, except that the repository that is indexed is a file system. You typically specify the path to a file directory, then the indexer indexes all documents within that folder. HTTP spider indexes and file system indexes are sometimes collectively referred to as *spider* indexes. The indexer recognizes a wide variety of document formats, such as Word or Excel documents. Any document in an unknown format is skipped by the indexer.

See Also

Chapter 9, "Configuring Search and Building Search Indexes," page 177

PeopleSoft Configuration Manager

PeopleSoft Configuration Manager is a Microsoft Windows application that simplifies development workstation administration by enabling you to adjust PeopleSoft registry settings from a central location. You can set up one development workstation to reflect the environment at your site, then export the configuration file, which can be shared among other workstations. You can also define separate profiles for connecting to different PeopleSoft databases.

Note. The PeopleSoft Configuration Manager applies only to development environment workstations, such as workstations used to launch Application Designer and Data Mover on Windows.

PeopleSoft configuration parameters are grouped on the Configuration Manager pages according to the function, feature, or tool that they control, including:

- Startup settings.
- Display settings.
- Crystal report and Business Interlink settings.
- Trace settings.
- Workflow settings.
- Remote call settings.
- Developer workstations.
- Importing and exporting environment settings.
- Defining configuration profiles.

See Also

Chapter 10, "Using PeopleSoft Configuration Manager," page 209

PeopleTools Utilities

The PeopleTools utilities are a set of various configuration and administration interfaces that serve as a browser-based method of setting numerous system settings. These utilities, most of which are available through the PeopleTools Utilities menu, provide the ability to configure, maintain, or launch a wide range of features, including:

- The System Information page.
- The message catalog.
- The spell check dictionary.
- Translate values.
- Application server caching.
- SQR customization.
- Table management and sharing.
- Backward compatibility.
- Remote database connection.
- File attachments.
- Stored URLs.
- Mobile data synchronization (deprecated).
- Update tracking.
- Platform-specific database features.
- Database auditing.
- International settings.
- Optimization utilities.
- PeopleSoft Ping.

See Also

[Chapter 11, "Using PeopleTools Utilities," page 233](#)

Tracing and Debugging

You can use the PeopleCode Debugger to interactively debug a PeopleCode program's configurations of a two-tier connection to the database or a three-tier connection to the database. You can temporarily override the PeopleSoft Configuration Manager trace settings for PeopleCode and SQL programs.

See Also

Chapter 12, "Tracing, Logging, and Debugging," page 285

Jolt Configuration Options

With Jolt, PeopleSoft provides the options of configuring load balancing, session pooling, and (for some special configurations) Jolt Internet Relay. Load balancing enables you to route requests to servers according to the ability of a server to handle a given request load. Powerful, dedicated servers can take a higher load while less powerful servers can take a lighter load. Session pooling enables user sessions to share web server connections, which is a more efficient use of system resources. Jolt Internet Relay enables you to route connections from one web server to another, perhaps through a fire wall, for specific configuration or security needs.

See Also

Chapter 13, "Working with Jolt Configuration Options," page 297

Timeout Settings

This appendix lists the delivered default timeout settings for the web server, application server, PeopleSoft Process Scheduler, search servers, and PeopleSoft Internet Architecture (PIA).

See Also

Appendix C, "PeopleSoft Timeout Settings," page 349

System and Server Administration Implementation

The functionality of system and server administration for your PeopleSoft applications is delivered as part of the standard installation of PeopleTools, which is provided with all PeopleSoft products.

Several activities must be completed before you administer the system and servers for your implementation:

- Install Enterprise PeopleTools according to the installation guide for your database platform and operating system.
- Install your PeopleSoft application according to the installation guide for your database platform and application.
- Establish a user profile that gives you access to the tools, pages, and processes that you'll use.

Other Sources of Information

In addition to implementation considerations presented in this section, take advantage of all PeopleSoft sources of information, including the installation guides, release notes, PeopleBooks, and Red Papers.

See Also

"System and Server Administration Preface," page xix

Enterprise PeopleTools 8.50 PeopleBook: Getting Started with Enterprise PeopleTools

Chapter 2

Understanding PeopleSoft Internet Architecture

This chapter discusses:

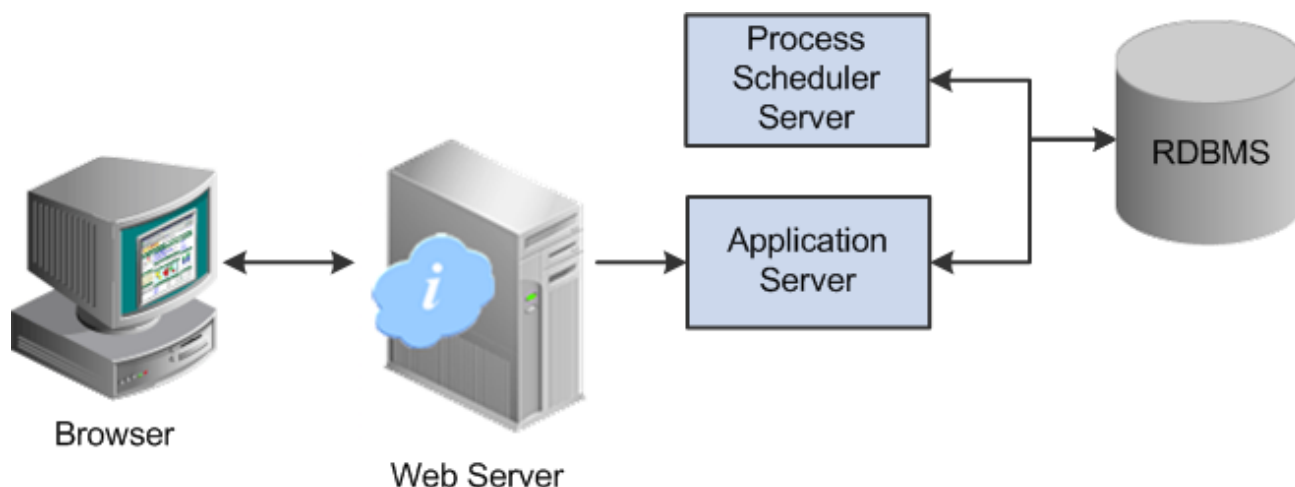
- PeopleSoft Architecture fundamentals.
- Database server.
- Application servers.
- Process Scheduler server environment.
- Web server.
- Web browser.
- Configuration options.
- Implementation options.

PeopleSoft Architecture Fundamentals

Your PeopleSoft application runs within the PeopleSoft Internet Architecture (PIA), which requires a variety of software and hardware elements, including:

- relational database management system (RDBMS).
- application server(s).
- Process Scheduler server(s).
- web server(s).
- web browsers.

It's important to understand the role of each element before you can decide which configuration options will work best for your implementation. The following diagram illustrates, at a high level, the physical relationship between the basic elements of the PeopleSoft architecture:



Distinct tiers of the architecture starting with the database, moving to the application server and Process Scheduler server, to the web server, then out to the browser

Configuring the PeopleSoft infrastructure is not just about enabling internet application deployment through a browser. PeopleSoft enables you to take advantage of numerous PeopleSoft intranet, internet, and back-end solutions, including:

- Integration Broker, our service oriented architecture (SOA).
- PeopleSoft Portal Solutions.
- Analytic Calculation Engine.
- MultiChannel Framework.
- Performance Monitor.

All of these additional technologies stem from the basic architecture depicted in the previous diagram.

Database Server

The database server houses a database engine and your PeopleSoft application database, which includes all the PeopleTools metadata, application definitions, system tables, application tables, and application data. The database server simultaneously handles the application server connections, development environment connections, and batch programs running against it.

The PeopleSoft database is the repository for all information managed by your PeopleSoft application. Both application data and PeopleSoft metadata are stored and maintained in the database. Application Designer, the main tool of the development environment, enables you to define, modify, and maintain this metadata, which the system uses to drive the runtime architecture. This collection of metadata defines a PeopleSoft application.

With Application Designer you can create dozens of different types of application definitions, such as fields, records, and pages. When an application developer saves an application definition, the system saves this definition to the metadata repository in the PeopleSoft database. At runtime, the application server retrieves the most recent application object definitions from the metadata repository, compiles and caches the application definition into memory, and runs the business rules based on the most current definitions.

Application Servers

This section discusses:

- Application servers.
- Tuxedo and Jolt.
- Domains.
- PeopleSoft server processes.
- Services.
- Listeners, handlers, and queues
- Database connectivity.

Application Servers

The application server is the core of PeopleSoft Internet Architecture. It runs business logic and submits SQL to the database server. An application server consists of numerous PeopleSoft server processes, grouped in domains. Each server process within a domain provides unique processing abilities, enabling the application server to respond effectively to a multitude of transaction requests generated throughout the PeopleSoft architecture.

Application servers require database connectivity software installed locally to maintain the SQL connection with the RDBMS. You must install the required connectivity software and associated utilities for your RDBMS on any server on which you intend to run the PeopleSoft Application Server.

After the application server establishes a connection to the database, any device that initiates a transaction request through the application server takes advantage of the application server's direct connection to the database.

Oracle Tuxedo and Oracle Jolt

PeopleSoft uses Oracle Tuxedo, a middleware framework and transaction monitor, to manage transactions between the application server and the database. PeopleSoft also uses Oracle Jolt, a Java API and class library, to facilitate communication between Tuxedo running on the application server and the PeopleSoft software running on the web server. Tuxedo and Jolt are required elements of the PeopleSoft application server.

Note. Tuxedo doesn't actually perform processing on the application server; it schedules PeopleSoft server processes to perform the transactions.

Domains

A domain is the collection of server processes, supporting processes, and resource managers that enable the database connections required to fulfill application requests. You manage each domain with a separate configuration file, and you configure each application server domain to connect to a single database. A single application server machine can support multiple application server domains running on it. You configure an application server domain using the PSADMIN utility.

There can be a one-to-one or a many-to-one relationship between application server domains and a database. In the simplest case, you configure a single application server domain to connect to a single PeopleSoft database. In a more sophisticated environment, you may configure multiple application server domains, with each domain connecting to the same PeopleSoft database. The opposite is not valid; a single application server domain cannot be used to connect to multiple PeopleSoft databases.

For example, suppose you have installed three application databases. In this case, you must configure at least three application server domains, one for each database. As demand increases, you may need to configure multiple application server domains per database, for redundancy, fail-over, and performance reasons.

You can configure multiple application server domains under a single PeopleSoft configuration home directory, or *PS_CFG_HOME*. In this context, *PS_CFG_HOME* refers to the PeopleSoft high-level directory on the application server. *PS_CFG_HOME* is the directory to which you installed the PeopleSoft application server configuration files creating a domain.

PSADMIN creates a directory beneath *PS_CFG_HOME*\appserv for each application server domain that you configure. For example, suppose you create three domains: HCMDMO1, HCMDMO2, and HCMDMO3. In this case, PSADMIN creates subdirectories \HCMDMO1, \HCMDMO2, and \HCMDMO3 beneath the *PS_CFG_HOME*\appserv directory.

PeopleSoft Server Processes

When you boot an application server domain, it starts the set of server processes associated with that domain. Numerous server processes run in a domain. Each server process establishes a persistent connection to a PeopleSoft database, and this connection acts as a generic SQL pipeline that the server process uses to send and receive SQL. Each server process uses its own, exclusive, SQL connection to facilitate requests from multiple sources. From the RDBMS perspective, each server process within a domain represents a connected user.

A server process is executable code that receives incoming transaction requests. The server process carries out a request by making calls to a service, such as MgrGetObject. The server process waits for the service to complete, then returns information to the device that initiated the request, such as a browser. While a server process waits for a service to complete, other transaction requests wait in a queue until the current service completes. A service may take a fraction of a second to complete or several seconds, depending on the type and complexity of the service. When the service completes, the server process is then available to process the next request in the corresponding queue.

The number of server processes of each type is configurable and typically varies within a domain, depending on the requirements of your application or the main purpose of the domain. For example, if a domain's primary function is to handle application requests, you might see more of the server processes devoted to that task within that domain, such as PSAPPSRV, PSQCKSRV, and PSQRYSRV. Likewise, if a domain's primary function is to handle Integration Broker SOA requests, you might see more of the server processes devoted to that task in the domain, such as the messaging server processes.

You need to configure only those server processes that your implementation requires per domain. The minimum server processes that a domain requires are PSAPPSRV, PSSAMSRV, and PSWATCHSRV.

Note. PSWATCHSRV is a process that always starts in a domain. It is not an optional server process.

You can configure multiple instances of the same server processes to start when you boot the application server domain. This helps you to handle predicted workloads.

The following tables describes the PeopleSoft server processes. Not all of the server processes will necessarily be a part of every domain as that depends on the configuration options you select.

The required server processes are:

Server Process	Description
PSAPPSRV	This process performs functional requests, such as building and loading components. It also provides the memory and disk-caching feature for PeopleTools objects on the application server. PSAPPSRV is required to be running in <i>any</i> domain.
PSSAMSRV	This SQL application manager process handles the conversational SQL that is mainly associated with Application Designer. This process is required to be running on any domain.
PSWATCHSRV	Monitors the domain and detects any orphaned application server processes.

Other server processes that you can elect to configure based on your system requirements, include:

Server Process	Description
PSQCKSRV	Handles quick, read-only SQL requests. This is an optional process designed to improve performance by reducing the workload of PSAPPSRV.
PSQRYSRV	Handles any query run by PeopleSoft Query. This is an optional process designed to improve performance by reducing the workload of PSAPPSRV.
PSMSGDSP, PSMSGHND, PSPUBDSP, PSPUBHND, PSSUBDSP, PSSUBHND	Used for "Publish-Subscribe" processing for Integration Broker and the PeopleSoft service oriented architecture (SOA).
PSANALYTICSRV	Performs processing and requests required for Analytic Calculation Engine.
PSRENSRV	Enables real-time event notification (REN) used in various PeopleTools technology, such as MultiChannel Framework and report distribution.
PSUQSRV, PSMCFLOG	Used within the MultiChannel Framework to manage queues and transaction logs.

Server Process	Description
PSPPMSSRV	Handles the processing of all the data recorded by the Performance Monitor.
PSDBGSRV	When debugging PeopleCode, this server process maintains an independent connection to the database to avoid conflict.

Services

When a PeopleSoft application sends a request to the application server, it sends a service name and a set of parameters, such as MgrGetObject and its parameters. Tuxedo then queues the transaction request to a specific server process that is designed to handle certain services.

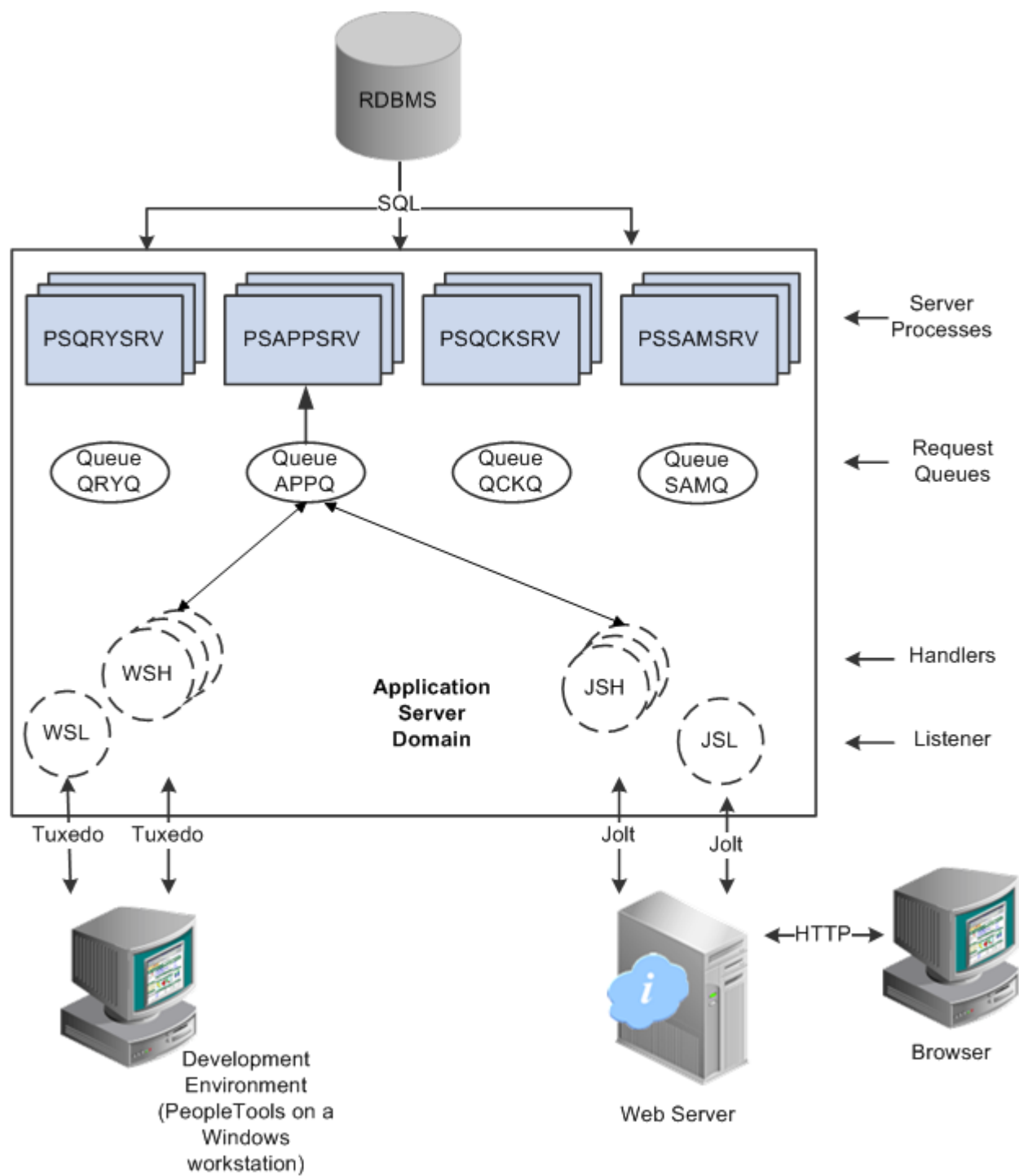
When a server process boots, it advertises to the system the predefined services it handles.

Note. When discussing PeopleSoft architecture mechanics and system features, the term *service* is used in various contexts. The following statement may help to clarify this term within the context of an application server domain: An application server domain calls server processes, such as PSAPPSRV, which in turn invoke services, such as MgrGetObject, which run against the database. The services called by domain server processes are not to be confused with the services and service operations associated with the service oriented architecture (SOA) provided by Integration Broker.

Listeners, Handlers, and Queues

Listeners, handlers, and queues provide the basis of a domain's functionality. They receive requests, route requests, store requests, monitor requests, and return request responses.

The following diagram illustrates how listeners, handlers, and queues interact with incoming requests and server processes:



Listeners route application requests to handlers where they are placed in queues monitored by server processes that submit requests to the database

Note. For simplicity, the diagram does not depict every server process that runs on the application server.

The following table describes each component depicted in the previous diagram:

<i>Item</i>	<i>Description</i>
Browser	Enables internet and intranet access to all PeopleSoft applications, including many PeopleTools system administration interfaces.
Development Environment	<p>A Microsoft Windows workstation used for specific application development and system administration tasks. For example, an application developer accesses Application Designer in the development environment to modify a page or record definition. Likewise, a system administrator would access PeopleSoft Data Mover in the development environment to import a data file. Using the development environment you can connect directly to the database (a two-tier connection) or through the application server (a three-tier connection).</p> <p>Note. Using the PeopleTools development environment in Microsoft Windows, you can connect directly to the database, or through an application server.</p> <p>Note. No end user applications are deployed to the C++/Windows platform. All end user applications are deployed through the browser.</p>
Workstation listener (WSL)	The workstation listener monitors Tuxedo ports for initial connection requests sent from the PeopleTools development environment (as in Application Designer). After the workstation listener accepts a connection from a workstation, it directs the request to a workstation handler. From that point, the workstation interacts only with the workstation handler to which it is assigned.
Workstation handler (WSH)	The workstation handler processes the requests it receives from the workstation listener. A unique port number identifies a workstation handler. The port numbers for the workstation handler are selected (internally by Tuxedo) from a specified range of numbers. You can configure multiple workstation handlers to start automatically if demand increases and other handlers become overloaded.
Jolt server listener (JSL)	The Jolt server listener applies only to browser requests. The Jolt server listener monitors the Jolt port for connection requests sent from the browser through the web server. After the Jolt server listener accepts a connection, it directs the request to a Jolt server handler. From that point, the browser interacts with the Jolt server handler. This is analogous to the relationship between the workstation server listener and workstation server handler.

<i>Item</i>	<i>Description</i>
Jolt server handler (JSH)	The Jolt server handler applies only to browser requests. The Jolt server handler processes the requests it receives from the Java server listener. The port numbers for the Jolt server handler are selected internally by Tuxedo in sequential order.
Request queues	Each type of server process has a service request queue that it shares with other server processes of the same type. For example, PSAPPSRV processes use the queue, APPQ, and PSQCKSRV processes use the queue, QCKQ. The workstation handler and Jolt server handler insert requests into the appropriate queue, and then the individual server processes complete each request in the order that it arrives.
Server processes	The server processes act as the core of the application server domain. They maintain the SQL connection and make sure that each transaction request gets processed on the database and that the results are returned to the appropriate origin.

PeopleSoft Process Scheduler Server

The PeopleSoft Process Scheduler environment can also be thought of as the "batch" environment. It is the location where many of your batch programs, such as Application Engine programs, run, and in most situations, this is also where you have COBOL and SQR executables installed.

In a multiserver environment, you can decide where your Process Scheduler environment resides based on available servers and performance requirements. In the PeopleSoft topology, you can install Process Scheduler on a separate server, or it can run on either the application server or the database server.

While you can install PeopleSoft Process Scheduler on any supported application server, database server, or batch server, it's important that you choose a location that's supported in your PeopleSoft environment, which varies per RDBMS platform. In most database environments, you have at least two options.

You use PSADMIN to configure and administer both the application server and PeopleSoft Process Scheduler server. The PeopleSoft Process Scheduler setup procedure in PSADMIN provides a menu-driven interface to configure PeopleSoft Process Scheduler parameters and administer the Process Scheduler server agent, and it is very similar to the PSADMIN menus for configuring an application server domain.

Even though the application server and PeopleSoft Process Scheduler have PSADMIN as a common interface, take advantage of Tuxedo middleware, and share directories under *PS_HOME* and *PS_CFG_HOME*, they are separate entities. For instance, you boot, configure, and shut down the application server and the PeopleSoft Process Scheduler server separately, and the two environments can exist on separate server machines.

See Also

Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Process Scheduler, "Understanding PeopleSoft Process Scheduler"

Web Server

A Java-enabled web server is required to extend the PeopleSoft architecture to the internet and intranet. When you install the PeopleSoft Internet Architecture on the web server, you install a collection of PeopleSoft Java servlets designed to handle a wide range of PeopleSoft transactions originating from the internet or an intranet.

This section discusses:

- Web server software
- PeopleSoft servlets
- Jolt

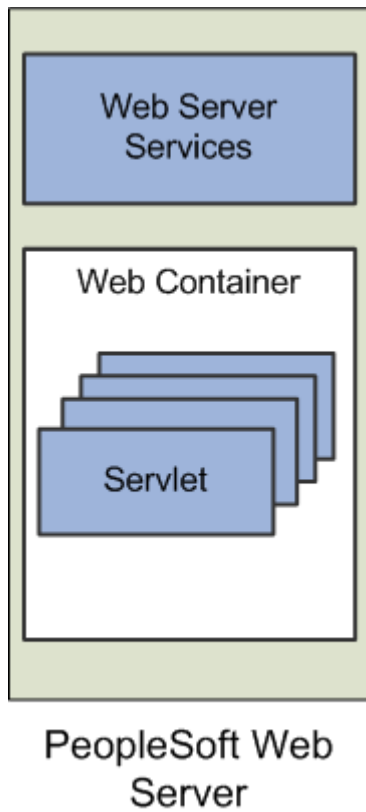
Web server software

PeopleTools provides and supports the following industry-standard web servers for use within your PeopleSoft implementation:

- Oracle WebLogic
- IBM WebSphere

The following software runs on the PeopleSoft web server:

Software Component	Description
Web server services	Web server services software manages the web server and provides the HTTP/S 'listener' for browser and remote system requests.
J2EE Web container	The web container (or servlet engine) is the J2EE environment in which the PeopleSoft servlets run. This component is embedded within the web server software.
Java servlets	Java is the platform-independent programming language used widely for web-based programs, web applications, and servlets. Servlets are Java programs that run on the web server, unlike 'applets' which are downloaded to a client browser to run. During the PeopleSoft installation, a variety of PeopleSoft Java servlets are installed on the web server.



Web server running web server services and the web container where PeopleSoft servlets are installed

PeopleSoft Servlets

The following PeopleSoft servlets reside on the web server:

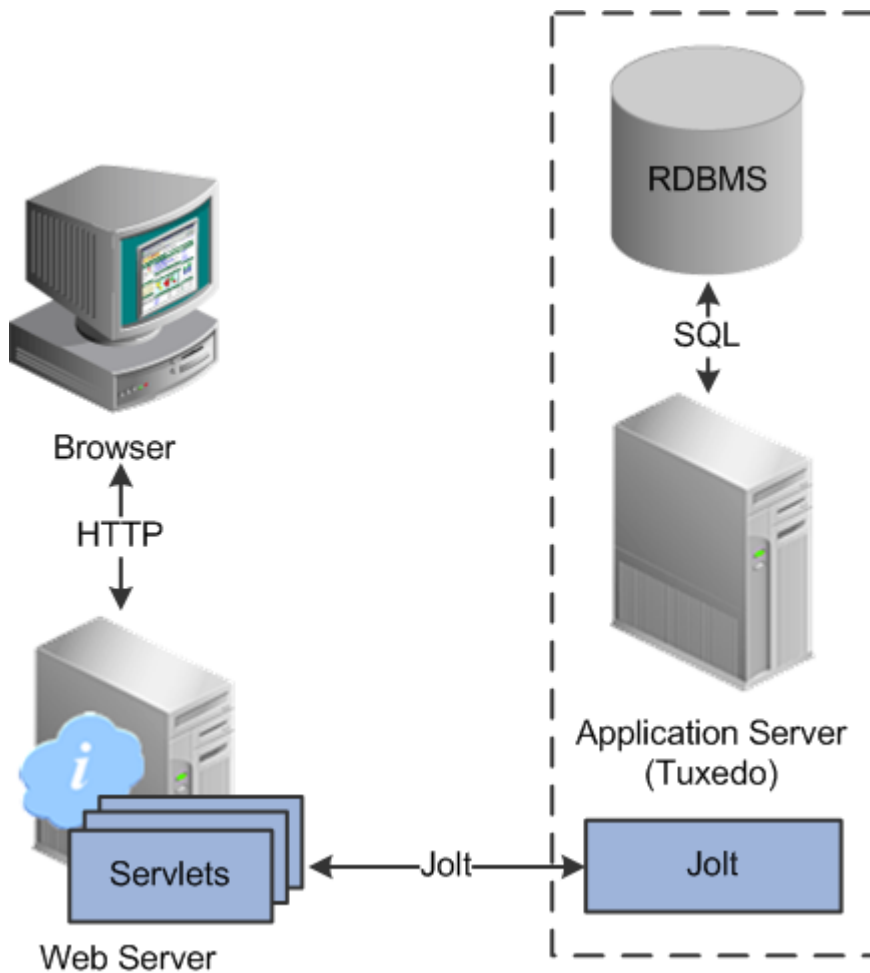
<i>Servlet</i>	<i>Description</i>
PORTAL	The portal servlet handles all of the requests and formatting for the users accessing PeopleSoft through PeopleSoft portal technologies. It manages various functionality, such as browser requests, search, content management, and homepage personalization.
PSIGW	The gateway servlet transmits service requests and responses between defined nodes as part of the PeopleSoft Service Oriented Architecture. The gateway handles PeopleSoft-to-PeopleSoft, PeopleSoft-to-third party, and third party-to-PeopleSoft services.
PSEMHUB	Supports the Environment Management Framework, used by various lifecycle management tools, such as Change Assistant, to collect and monitor system information.
PSOL	Supports the 'PeopleSoft Online Library,' which refers to the HTML, context-sensitive access to Enterprise PeopleTools PeopleBooks.

Servlet	Description
Report Repository	This report repository servlet enables users to easily access and distribute the output of batch reports, such as Crystal and SQR, run through PeopleSoft Process Scheduler. This servlet retrieves the report output in the report repository and serves it to the browser.
PSINTERLINKS	Used with the deprecated product PeopleSoft Business Interlinks.

Oracle Jolt

When you install Tuxedo, Jolt gets installed by default. The PeopleSoft servlets on the web server transmit requests and data through a connection to Jolt, which runs on the application server. Jolt extends Tuxedo capabilities by acting as the communication layer between the Java-based environment of the servlets and the C++ environment of the application server. You configure the servlets to direct requests from the web server to a predefined Jolt port on the application server.

The following diagram shows the relationship between Jolt and servlets on the web server:



PeopleSoft servlets on the web server sending messages to the application server through Jolt

Web browsers and integrated systems don't send requests directly to the application server. Instead, they send HTTP/S requests to the PeopleSoft servlets running on the web server. The web server translates the HTTP/S request into a Jolt request that is sent to a specified Jolt port. Then the application server, running on Tuxedo, submits the appropriate SQL to the database.

Web Browser

A supported web browser is the primary means by which end users access PeopleSoft applications and system administrators access administrative tools. You do not need to install other software on the workstation running the browser, such as connectivity software or downloaded applets.

The PeopleSoft system sends only the following to the browser:

- HTML
- JavaScript
- Cookies

Because the browser processes only this basic internet content, the client workstation is not burdened with unnecessary processing responsibility. All processing occurs at the server level.

After the system authenticates a user during signon, PeopleTools security deploys web browser cookies to store a unique access token for each user session. As the user navigates around the system, perhaps to a separate PeopleSoft application, the token in the browser cookie is used to reauthenticate the user and bypass the sign-in process. The browser authentication cookie is:

- an in-memory cookie, never written to disk.
- encrypted to prevent snooping.
- check-summed to prevent tampering.

Note. To access PeopleSoft applications, the browser option to allow session cookies *must* be enabled.

Server Configuration Options

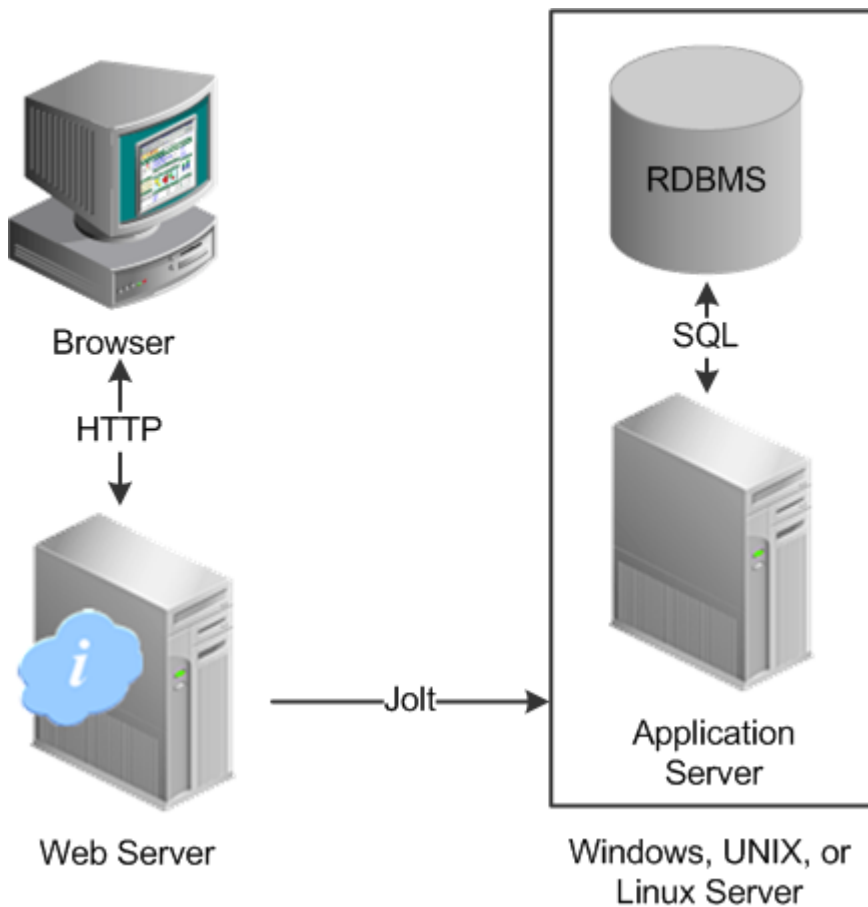
Depending on a variety of variables, such as your server operating system, your hardware resources, and your site's performance requirements, you can configure your environment to support physically separate or logically separate servers. In some cases, the PeopleSoft standard installation procedure recommends one or the other depending on, for example, the combination of your database type and operating system. Any platform-dependent configuration requirements are discussed in your PeopleTools installation documentation.

See *Enterprise PeopleTools 8.50 Installation for your database*

Logically Separate Server Configuration

A logically separate server environment means that multiple servers share the same physical machine. The servers are *logically*, but not *physically*, separate.

The following diagram depicts a logical configuration with two server machines. One server contains the web server, and the other contains the application server and database server:



Logical separation between the RDBMS and the application server with the two elements running on the same physical server machine

The solid line surrounding the application server and the database server represents one physical machine.

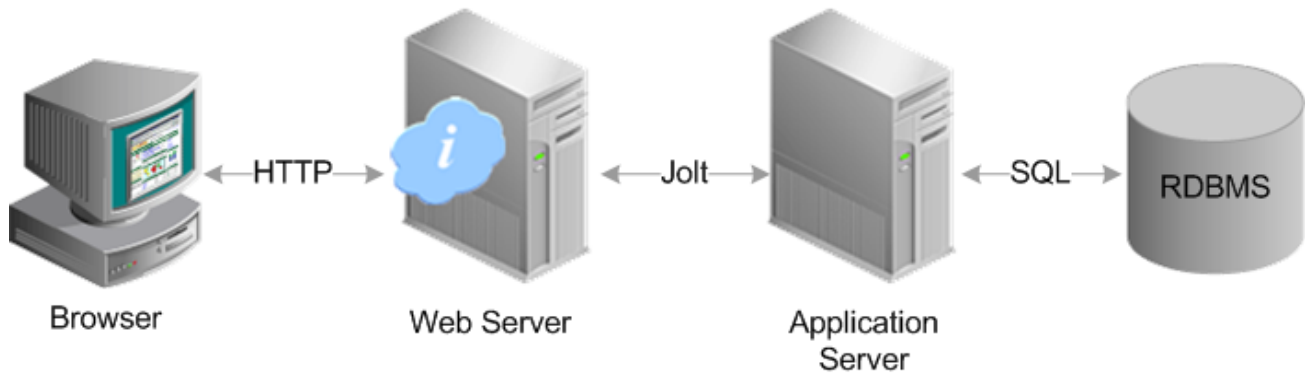
Although this diagram depicts a separate web server, the web server software could also reside on the same machine with both the application server and the database server. When installing multiple PeopleSoft architecture elements onto the same machine, the only requirement is that each element be supported by the underlying operating system.

If all servers are located on the same machine, however, you should consider security and performance issues. If you're deploying PeopleSoft applications to the internet, you will most likely want your web server outside of your network firewall and not on the same machine as the database server. Generally, having your application server on the same physical machine as the database server provides the best performance as this configuration has no network layer between the application server and the database.

Note. For development, testing, or training purposes, you might want to have all PeopleSoft architecture elements (or as many as possible) on the same server machine.

Physically Separate Server Configuration

A physically separate server configuration means that the web server, application server, and database server each reside on separate machines. The following diagram depicts a physically separate server configuration:

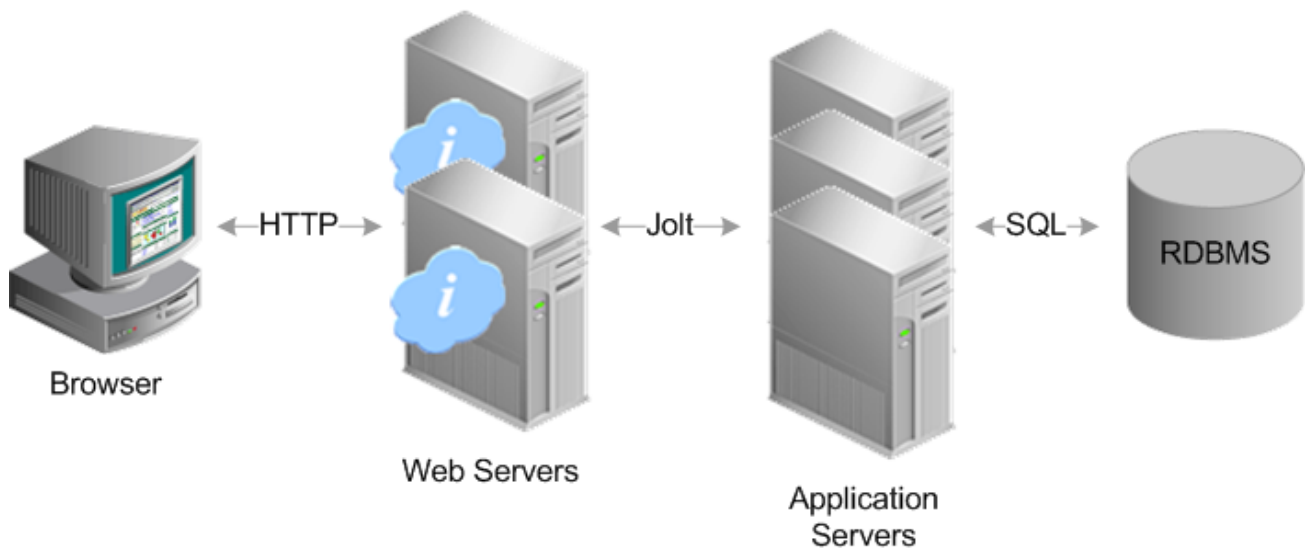


Physical separation between web server, application server, and RDBMS server with each server residing on different physical server machines

If the application server and database server don't reside on the same machine, then the application server and the database server should be connected to the same high-performance, backbone network. This ensures optimum performance.

Regarding performance, one advantage to keeping the architecture elements separate is that as demand on your system increases, you can add more server machines at each 'tier' to house more application servers or web servers in a modular fashion. If the server elements are installed onto a single machine, your options of increasing system performance are limited by the resources of that machine.

The following diagram illustrates how you can add multiple web servers and application servers when increased demand has pushed previous hardware to its limits.



Physical separation between web server, application server, and RDBMS promotes scalability, enabling the addition of multiple web servers and application servers to meet increased system demand

Within a PeopleSoft system, you can configure multiple PeopleSoft Internet Architecture installations on a web server as well as configure multiple domains on a single application server machine. If you reach the resource limits of a single server machine, you can incorporate more server machines to house additional PeopleSoft Internet Architecture installations or application server domains. Incorporating multiple, physically separate server machines provides increased:

- scalability
- fail-over
- availability

Implementation Options

Once you have the basic PeopleSoft Internet Architecture elements configured, you can elect to incorporate additional PeopleTools technology to run within that architectural foundation, including:

Technology	Description
Portal	<p>Oracle Enterprise provides a variety of portal options ranging from the base portal technology provided by PeopleTools, to Enterprise Portal and Application Portal Pack solutions. These portal solutions enable you to provide simple end user navigation, aggregate web-based content, share content with other portals, and more.</p> <p><i>See Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies</i></p> <p><i>See PeopleSoft Enterprise Portal 9.1 PeopleBooks</i></p> <p>See Your PeopleSoft application portal pack documentation</p>
Integration Broker	<p>Integration Broker provides a service oriented architecture (SOA) for exposing PeopleSoft business logic as services and consuming external web services for PeopleSoft applications to invoke. Integration Broker supports synchronous and asynchronous messaging with other PeopleSoft applications and with third-party systems.</p> <p><i>See Enterprise PeopleTools 8.50 PeopleBook: Integration Broker</i></p>
Performance Monitor	<p>Performance Monitor enables you to capture and store detailed performance information for any transaction occurring within your PeopleSoft implementation. You can monitor performance in real-time as well as analyze stored, historical data to help identify trends and problem areas.</p> <p><i>See Enterprise PeopleTools 8.50 PeopleBook: Performance Monitor</i></p>
MultiChannel Framework	<p>MultiChannel Framework delivers an integrated infrastructure to support multiple interaction channels for call center agents or other PeopleSoft users who must respond to incoming requests and notifications on these channels. MultiChannel Framework supports voice, email, web-based chat, instant messaging, and generic event channels.</p> <p><i>See Enterprise PeopleTools 8.50 PeopleBook: MultiChannel Framework</i></p>

<i>Technology</i>	<i>Description</i>
Analytic Calculation Engine	<p>Analytic Calculation Engine enables application developers to define both the calculation rules and the display of calculated data within PeopleSoft applications for the purposes of multidimensional reporting, data editing, and analysis.</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: Analytic Calculation Engine</i></p>
Search server	<p>Configuring a search server enables you to off-load Verity search processing onto a remote server, rather than configuring search capabilities per domain. In this configuration, Tuxedo routes search requests from application server domains to the search domain running on the remote search server. Multiple application server domains may use the same search server to execute search requests.</p> <p>See <u>Chapter 9, "Configuring Search and Building Search Indexes," Configuring PeopleSoft Search, page 183.</u></p>

Chapter 3

Working with Server Domain Configurations

This chapter provides an overview and discusses how to:

- Work with the default PS_CFG_HOME.
- Work with alternate PS_CFG_HOME locations.

Understanding PS_HOME and PS_CFG_HOME

On any server that you install the PeopleSoft software, the installation program installs the required files for that server into one high-level directory structure, referred to as PS_HOME. After creating a domain, the configuration files associated with that domain reside in a directory structure referred to as "PS_CFG_HOME".

By default, the system separates the binary files (executables and libraries) stored in PS_HOME from the ASCII files (configuration and log files) associated with a domain stored in PS_CFG_HOME. This separation of the binary and ASCII files applies only to these servers:

- PeopleSoft Application Server.
- PeopleSoft Process Scheduler Server.
- PeopleSoft Search Server.

Note. Decoupling binary files from ASCII files does not apply to any other PeopleSoft servers, such as file servers, database servers, or web servers.

The following table describes the two file types and provides examples of these types within the PeopleSoft server.

<i>Location</i>	<i>File Type</i>	<i>Description</i>	<i>PeopleSoft Examples</i>
PS_HOME	Binary	Compiled, non-modifiable executables and libraries.	PSADMIN.EXE PSAPPSRV.EXE PSAESRV.EXE PSAPPENG.DLL PSODBC.DLL

<i>Location</i>	<i>File Type</i>	<i>Description</i>	<i>PeopleSoft Examples</i>
PS_CFG_HOME	ASCII	Text files associated with the configuration and administration of a domain that can be viewed, modified, or generated by the system.	PSAPPSRV.CFG PSAPPSRV.CFX PSTUXCFG APPSRV_<DATE>.LOG TUXLUG.<DATE>

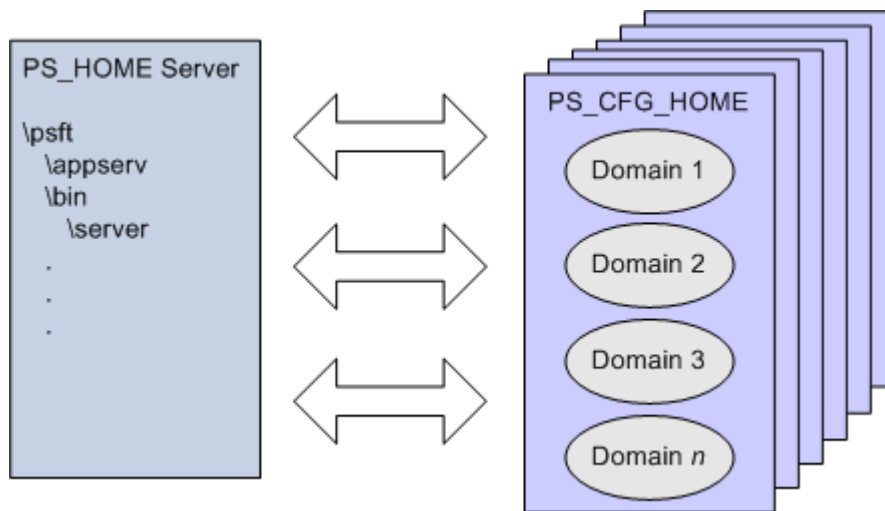
The decoupling of these file types enables system administrators to:

- Streamline and provide more flexible PeopleSoft server installations.
- Apply unique security restrictions to the binary file and configuration file locations.

Note. Although domains always contain their base template, the CFX and UBX templates (such as small, medium, and large) remain in PS_HOME. This means that when you create a new domain the template that you choose comes from PS_HOME\appserv, not PS_CFG_HOME\appserv.

Implementing Flexible Server Installations

With the binary files separate from the domain configuration, you have the option of installing multiple domains on multiple separate servers all leveraging the binary files of a single PS_HOME.



Multiple PS_CFG_HOME locations, containing multiple domains, all referencing the same PS_HOME installation on a remote server

Installing domains on separate servers enables you to:

- Install the PS_HOME binaries a single time.
- Incorporate additional server machines for additional domains per demand and performance requirements.
- Apply updates and upgrades to a single PS_HOME, reducing the time required for upgrading your system.

Applying Security Restrictions

With the server binary files and configuration files in separate locations, you can now apply uniform restrictions per file type. For example, some sites might prefer to have the binary files under read-only security, while providing write access to the configuration files to specific users for administrative tasks and certain server processes, such as logging.

See [Appendix A, "Securing PS_HOME and PS_CFG_HOME,"](#) page 305.

Working with the Default PS_CFG_HOME

This section discusses how to:

- Locate the default PS_CFG_HOME.
- Use PSADMIN with the default PS_CFG_HOME.

Locating the Default PS_CFG_HOME

When you launch PSADMIN, if a PS_CFG_HOME does not exist, the system creates the PS_CFG_HOME directory in the "user" directory of the current user (the owner of the domain). The system assumes the presence of the following environment variables:

<i>Operating System</i>	<i>Required Environment Variable</i>
UNIX/Linux	HOME
Windows	USERPROFILE

For example, depending on the operating system of the server, the system creates PS_CFG_HOME in the following location on the same drive as PS_HOME.

<i>Operating System</i>	<i>PS_CFG_HOME Location</i>
UNIX/Linux	\$HOME/psft/pt/<version>
Windows	%USERPROFILE%\psft\pt\<version>

After you create a domain, the domain exists under \$PS_CFG_HOME\appserv\<domain>.

With a user of *tsawyer*, on UNIX/Linux this would appear as:

```

/home/tsawyer/peopletools/8.50
    peopletools.properties
    /appserv
    /CRM
    /HR
    /PRCS
    /CRM_PPCS
    /HR_PPCS
    /Search
    /ver_dom

```

With a user of *tsawyer*, on Windows this would appear as:

```

C:\Documents and Settings\tsawyer\psft\pt\8.50
    peopletools.properties
    \appserv
    \CRM
    \HR
    \PRCS
    \CRM_PPCS
    \HR_PPCS
    \Search
    pswinsrv.cfg

```

Note. The previous examples show a situation in which CRM, HR, CRM_PPCS, HR_PPCS and ver_dom are all domain directories. They are not in PS_CFG_HOME by default, and appear only after the domains are created.

To display the default PS_CFG_HOME location, you can submit the following command to PSADMIN:

```
psadmin -defaultPS_CFG_HOME
```

Note. These commands are not case sensitive.

Using PSADMIN with the Default PS_CFG_HOME

Launching PSADMIN requires no extra steps or variables defined when domains exist on the same machine as PS_HOME in the default PS_CFG_HOME location.

When you launch PSADMIN, it will either create (if one doesn't exist) PS_CFG_HOME or search for PS_CFG_HOME, based on the current environment settings.

To start PSADMIN, the following conditions need to be fulfilled:

- PS_CFG_HOME must be a valid location. That is, the base directory must exist (UNIX), or the drive must exist (Windows).
- The PS_CFG_HOME must be writeable, and the user running PSADMIN must have write access to the PS_CFG_HOME location.

Working with Alternate PS_CFG_HOME Locations

Domains can exist on the same physical machine or a different physical machine than where the PS_HOME directory resides. That is, multiple domains on different machines can leverage a single installation of the PeopleSoft binary files (PS_HOME) installed on a location accessible through your network.

This section discusses how to:

- Specify alternate PS_CFG_HOME locations.
- Configure domains in alternate locations of PS_CFG_HOME.

Specifying Alternate PS_CFG_HOME Locations

The value of the PS_CFG_HOME environment variable determines where PSADMIN installs a domain. If you accept the default location of PS_CFG_HOME, this environment variable does not need to be specified. However, if you intend to install your domains in a different location, you need to override the default by setting the PS_CFG_HOME environment variable prior to launching PSADMIN.

For UNIX and Linux, you can:

- Set the PS_CFG_HOME environment variable in the psconfig.sh file. However, if you hard code the PS_CFG_HOME environment variable in psconfig.sh other users will not be able to use the same PS_HOME for different PS_CFG_HOMEs. All users of the same PS_HOME would invoke the same psconfig.sh, and therefore see their domains in the same PS_CFG_HOME.
- Set PS_CFG_HOME in the startup script for the platform/shell, resulting in the automatic inheritance of the environment variable when a user signs on.

For Windows, you can set PS_CFG_HOME through the Control Panel in the user variables interface, or issue a SET command prior to starting PSADMIN. For example,

```
C:\>SET PS_CFG_HOME=n:\psftdomains
C:\>cd pt850\appserv
C:\pt850\appserv>psadmin
```

In this case, any domains created during this PSADMIN session would be created in n:\psftdomains\appserv.

Note. If you elect to operate in the same fashion as previous PeopleTools releases (where the configuration files and the binary files exist within the same directory structure) set PS_CFG_HOME = *PS_HOME*.

Using the %V Meta Variable

Use the %V meta variable if you wish to keep all of your PS_CFG_HOMEs together without needing to set PS_CFG_HOME each time you install a new PeopleTools version.

For example, you could set the PS_CFG_HOME environment variable as follows:

```
PS_CFG_HOME=C:\PeopleTools\installs\%V
```

If using the %V meta variable, assume you have installed two versions of PeopleTools: PT8.50 and PT8.51. In this case, PSADMIN automatically maps the %V to the PeopleTools version, creating the following PS_CFG_HOME locations:

```
C:\PT\installs\8.50-00  
C:\PT\installs\8.51-00
```

Configuring Domains in Alternate Locations of PS_CFG_HOME

When using PSADMIN with alternate locations of PS_CFG_HOME, make note of the requirements discussed in this section.

Working with Remote PS_HOME Locations

If you intend to install your domains on a separate machine from where PS_HOME resides, keep these items in mind:

- The network location to where PS_HOME resides must be mapped to from the machine where PS_CFG_HOME resides.
- If a domain references a PS_HOME on a mapped drive, problems may arise if the drive letter or path is changed after initially creating the domain. If a drive mapping changes, the domain definitions may reference invalid path information. In the event of mapped drive changes, shut down the existing domains and reconfigure (or recreate) the domains.
- If PS_HOME resides on a remote server, the operating system of the PS_HOME server and the PS_CFG_HOME server must match. PeopleTools binaries are native to a platform.

Installing Necessary Components

While you can leverage a single, remote installation of PeopleTools, the server on which PS_CFG_HOME resides must have any additional components installed locally, such as Tuxedo, database drivers, ODBC connectivity information, and so on, depending on your implementation.

Ensuring that PS_CFG_HOME is Set Appropriately

Only domains installed within the current PS_CFG_HOME directory can be administered by PSADMIN. That is, the list of domains to administer that PSADMIN displays depends on the value of the PS_CFG_HOME variable. PSADMIN does not aggregate a domain list across multiple locations.

Assume that you have domains installed under these two PS_CFG_HOME directories:

- N:\psftdomains\appserv
- M:\psftfomains\appserv

Assume also that on server N domain 1 and domain 2 are installed, and on server M domain 3 and domain 4 are installed.

If you launch PSADMIN from server N, where PS_CFG_HOME, by default, is set to N:\psftdomains\appserv, you will only be able to view and administer domain 1 and domain 2.

Domains created on machine N, can only be configured on machine N. There are settings in the domain PSTUXCFG file that bind a domain to its host. You can not boot or configure a domain from a different machine.

Note. The user creating and configuring domains in PSADMIN must have write access to the PS_CFG_HOME location.

Managing Domains

When working with decoupled PS_HOME and PS_CFG_HOME directory structures, keep these recommendations in mind:

- Use distinct PS_CFG_HOME locations for each PeopleTools version.
- A domain configured on one machine cannot be copied and run on another machine.
- The Windows service configuration file (PSWINSRV.CFG) will be invalid if it is shared between multiple PS_CFG_HOMES on different network drives.
- As you maintain your system and upgrade domains, keep in mind that dormant domains will consume disk space. Make sure to keep track of where your retired domains reside and take action to remove them when no longer required. It is generally recommended to remove old log files and trace files for both efficiency and security purposes.

Chapter 4

Using the PSADMIN Utility

This chapter provides an overview of PeopleSoft Server Administration (PSADMIN) and discusses how to:

- Start PSADMIN.
- Use PSADMIN.
- Use configuration templates.
- Use the PSADMIN command-line interface.
- Use the Quick-Configure menu.
- Use PSADMIN executables and configuration files.
- Configure the application server to handle cache files and replay files.

Understanding PSADMIN

PSADMIN simplifies the process of configuring and administering all of the servers and features that are available on the application server. For example, you use PSADMIN to configure application server domains, Process Scheduler servers, and search servers.

Note. *PS_HOME* is the directory where you install PeopleTools.

Accessing Network Drives in Microsoft Windows Server

This section applies only if all of the following are true:

- You've upgraded to the current PeopleTools release, including the required Tuxedo version and rolling patch level, from PeopleTools 8.45 or older.
- You plan to administer your application server domains in Microsoft Windows 2003 (or newer) Server.
- One or more PeopleSoft processes need to directly access a mapped network drive. Activities requiring this can include:
 - Using an instance of PSADMIN that was launched from the network drive.
 - Accessing a database on the network drive.
 - Outputting reports to a location on the network drive.

Command	Example	Result of the Example
shutdown	<code>psadmin -c shutdown -d PSDMO</code>	Shuts down the PSDMO application server domain, by using a normal shutdown method. In a normal shutdown, the domain waits for users to complete their tasks and turns away new requests before terminating all of the processes in the domain.
shutdown!	<code>psadmin -c shutdown! -d⇒ PSDMO</code>	Shuts down the PSDMO application server domain by using a forced shutdown method. In a forced shutdown, the domain <i>immediately</i> terminates all of the processes in the domain.
sstatus	<code>psadmin -c sstatus -d PSDMO</code>	Displays the Tuxedo processes and PeopleSoft server processes that are currently running in the PSDMO application server domain.
cstatus	<code>psadmin -c cstatus -d PSDMO</code>	Displays the currently connected users in the PSDMO application server domain.
qstatus	<code>psadmin -c qstatus -d PSDMO</code>	Displays status information about the individual queues for each server process in the PSDMO application server domain.
preload	<code>psadmin -c preload -d PSDMO</code>	Preloads the server cache for the PSDMO domain.
cleanipc	<code>psadmin -c cleanipc -d PSDMO</code>	Cleans the IPC resources for the PSDMO domain.
purge	<code>psadmin -c purge -d PSDMO</code>	Purges the cache for the PSDMO domain.
import	<code>psadmin -c import c:⇒ \ptinstalls\pt85x\...⇒ \psappsrv.cfg` -n NEWSRVR</code>	Imports a domain configuration. See PSADMIN command line help for all possible options.

Using the Process Scheduler Commands

For Process Scheduler administration, PSADMIN has two syntax formats — one for creating new Process Scheduler configurations, and the other for administering existing configurations.

Startup (<i>ps_set</i>) Setting	Description
<i>DBNAME</i>	<p>This is the equivalent of the <i>DBName</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.</p> <p>Note. If you don't include the <i>ps_set</i> parameter, the value of this setting is the same as the database name that you specify in the command.</p>
<i>DBTYPE</i>	This is the equivalent of the <i>DBType</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.
<i>PRCSSERVER</i>	This is the equivalent of the <i>PrcsServer</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.
<i>OPR_ID</i>	This is the equivalent of the <i>UserId</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.
<i>OPR_PSWD</i>	Enter the user password that is associated with the specified user ID. This is the equivalent of the <i>UserPswd</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.
<i>CNCT_ID</i>	This is the equivalent of the <i>ConnectId</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.
<i>CNCT_PSWD</i>	This is the equivalent of the <i>ConnectPswd</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.
<i>SERV_NAME</i>	<p>(Optional) This is the equivalent of the <i>ServerName</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.</p> <p>Important! If you want this setting to be blank, but you can't truncate the string to this point (you still need to specify a value for <i>LOGOUT_DIR</i>), you can specify a value of "_____" (five underscores without the quotes) in this position. PSADMIN interprets this as a blank value.</p>
<i>LOGOUT_DIR</i>	<p>This is the equivalent of the <i>Log/Output Dir</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.</p> <p>Note. If this value contains spaces, it must be in double quotes (" "). For example: "c:\psft app\log_output".</p>
<i>SQRBIN</i>	<p>This is the equivalent of the <i>SQRBIN</i> parameter on the PSADMIN Process Scheduler Quick-Configure menu.</p> <p>Note. If this value contains spaces, it must be in double quotes (" "). For example: "C:\my pt846\bin\sqr\MSS\binw".</p>

Command	Example	Result of the Example
kill	<code>psadmin -p kill -d psdmo</code>	Kills the domain (similar to forced shutdown).

See Also

[Chapter 5, "Using PSADMIN Menus," Using the Process Scheduler Menu, page 69](#)

Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Process Scheduler, "Using the PSADMIN Utility"

Using the Search Server Commands

Use the following syntax to administer an existing search server domain:

```
psadmin -s command -d domain
```

The domain parameter must be the name of the search server domain that you want to administer, for example, PSSRCH. The valid values of the command parameter are as follows:

Command	Example	Result of the Example
boot	<code>psadmin -s boot -d PSSRCH</code>	Boots a search server.
configure	<code>psadmin -s configure -d⇒ PSSRCH</code>	Configures a search server.
shutdown	<code>psadmin -s shutdown -d PSSRCH</code>	Shuts down the domain, by using a normal shutdown method. In a normal shutdown, the domain waits for current transactions to complete and turns away new requests before terminating all of the processes in the domain.
shutdown!	<code>psadmin -s shutdown! -d⇒ PSSRCH</code>	Shuts down the domain by using a forced shutdown method. In a forced shutdown, the domain immediately terminates all of the processes in the domain.
sstatus	<code>psadmin -s sstatus -d PSSRCH</code>	Displays the Tuxedo processes and PeopleSoft server processes that are currently running in the domain.
cstatus	<code>psadmin -s cstatus -d PSSRCH</code>	Displays the currently connected users/clients.

Note. The asterisk that precedes the object name indicates that this object is being used by the current service request.

Chapter 5

Using PSADMIN Menus

This chapter discusses how to:

- Use the Application Server Administration menu.
- Use the PeopleSoft Process Scheduler menu.
- Use the PeopleSoft Search Server menu.
- Set up the PeopleSoft Windows service.

Using the Application Server Administration Menu

This section discusses how to:

- Access the application server options.
- Administer a domain.
- Boot a domain.
- Shut down a domain.
- Perform a normal shutdown.
- Perform a forced shutdown.
- Check the domain status.
- Purge the domain cache.
- Configure a domain.
- Edit configuration and log files.
- Create a domain.
- Delete a domain.
- Configure an application server domain to preload cache.
- Clean domain IPC resources.

Selecting Components for Cache Projects

To select components for preloaded cache:

1. In a browser, select PeopleTools, Utilities, Administration, Select Pre-load Components.
2. Select the Add a New Value tab, and in the Project Name edit box, enter the name of the project that will contain the components you select, and click Add.

Note. All project names used to contain components for preloaded cache, must contain the "PLC_" prefix.

3. On the Preload Comps page, enter a Description, and select the Menu Name, Component Name, and Market for each component you want in the preloaded cache project.
4. Click Save.

Creating Cache Projects

To create a preload file cache project:

1. Select PeopleTools, Utilities, Administration, Create Pre-load Project.
2. Select the Add a New Value tab, and enter a Run Control ID.
3. On the Preload Proj page select the appropriate Project Name (the same project name specified when you selected pre-load components).
4. Click Run.

This invokes an Application Engine program (PTCHPLC_PRJ) that creates the project definition in the database and populates it with the components you selected.

Note. While the cache project can be created manually in Application Designer, the Application Engine program does this automatically,

Deleting Cache Projects

To delete a preload file cache project:

1. Select PeopleTools, Utilities, Administration, Delete Pre-load Project.
2. On the Find an Existing Value page, click the appropriate project name.
3. On the Preload Proj Del page confirm that you have selected the appropriate project and click Delete the pre-load project.

Preloading File Cache

To preload the file cache:

- Delete a Process Scheduler server.
- Edit the Process Scheduler configuration file.
- Use the Process Scheduler options.
- Use Process Scheduler command-line options.
- Clean IPC Resources for the Process Scheduler domain.

Understanding the Process Scheduler Menu

Use the PSADMIN utility to configure and administer PeopleSoft Process Scheduler. PeopleSoft Process Scheduler is used to run batch processes. You only need to configure PeopleSoft Process Scheduler on a server where you intend to run batch processes.

The following sections describe the menus and options within the PSADMIN utility that are related to PeopleSoft Process Scheduler in the order that they appear in the PeopleSoft Process Scheduler Administration menu—not in the order that you would access them the first time you configure the Process Scheduler server. Then, select the option from the PeopleSoft Process Scheduler Administration menu that corresponds to the action that you need to perform.

The following sections explain the options for PeopleSoft Process Scheduler within PSADMIN. Those options that pertain to UNIX only are marked accordingly.

Starting a Process Scheduler Server

To start a Process Scheduler server:

1. Select 1 from the PeopleSoft Process Scheduler Administration menu.
2. To start the Process Scheduler server for a specific database, enter the number in the database list that corresponds to the appropriate database.

Stopping a Process Scheduler Server

You can stop a Process Scheduler server that is running on an application server by using PSADMIN or the Process Monitor.

To stop a Process Scheduler server:

1. Select 2 from the PeopleSoft Process Scheduler Administration menu.
2. To stop the Process Scheduler server for a specific database, enter the number from the database list that corresponds to the appropriate database.

Configuring a Process Scheduler Server

Configuring a Process Scheduler server is similar to configuring application servers and web servers. From the PeopleSoft Process Scheduler Administration menu, you invoke a text-driven interface that prompts you for parameter values. All of the Process Scheduler server configuration information for a specific database is contained in the PSPRCS.CFG file, and the PSADMIN provides an interface for and prompts you to edit the PSPRCS.CFG file.

Note. The PSPRCS.CFG file supports environment variables. For example, the TEMP setting in the Process Scheduler section can look like this: TEMP=%TEMP%.

Although you typically edit the PSPRCS.CFG file through PSADMIN, you can find the PSPRCS.CFG file in the following directory:

- Windows: *PS_CFG_HOME\APPSERV\PRCS\database_name*
- UNIX: *PS_CFG_HOME/appserv/prcs/database_name*

To configure a Process Scheduler server:

1. Select *Configure a Process Scheduler Server* from the PeopleSoft Process Scheduler Administration menu.
2. Select the number in the database list that corresponds to the server that you want to configure.
3. Specify the appropriate values for your site in the following configuration section prompts.

Creating a Process Scheduler Server Configuration

You must add or create a Process Scheduler server before you can configure it.

To add a Process Scheduler server configuration on the application server:

1. Select 4 from the PeopleSoft Process Scheduler Administration menu.
2. Enter the name of the database that the Process Scheduler server will access.
3. Enter *Y* to configure the Process Scheduler.
4. Update the settings as appropriate for your environment. For example, select 9 to change the UserID that the Process Scheduler uses to log on to the database.
5. When all of the settings are correct, select 4 to load the configuration.

Deleting a Process Scheduler Server

To delete a Process Scheduler server configuration:

1. Select 5 from the PeopleSoft Process Scheduler Administration menu.
2. Select the number in the database list that corresponds to the database to which the server has access.

Editing the Service Configuration File

The Windows Services section of PSADMIN modifies the PSWINSRV.CFG file in the *PS_CFG_HOME* \appserv directory. You can edit the file directly by selecting *4 (Edit a Service Configuration File)* from the PeopleSoft Services Administration menu. This opens the PSWINSRV.CFG file in a text editor, where you can enter and save your changes.

The following sections describe each parameter.

Service Start Delay

When a domain resides on the same machine as the database server, consider using the Service Start Delay setting. By using this feature, you can avoid the situation where the database server is booting and is not ready to process requests at the time that the service attempts to boot the domain. In this scenario, without a delay set, the connection fails.

You can configure a Service Start Delay parameter that specifies a delay, in seconds, that elapses before a service attempts to start any domains. This allows the RDBMS enough time to boot and become available to accept requests.

The default is 60 seconds.

Application Server Domains

Specify the names of the domains that you want to start automatically when you boot the application server machine.

If you specify multiple domains, separate each domain with a comma and a space.

Process Scheduler Databases

Enter the databases with which a Process Scheduler server is associated. For each database that you specify, the associated Process Scheduler server starts when you boot the Microsoft Windows server.

If you specify multiple databases, separate each database with a comma and a space.

Search Server Domains

Specify the names of the domains that you want to start automatically when you boot the search server machine.

If you specify multiple domains, separate each domain with a comma and a space.

Chapter 6

Setting Application Server Domain Parameters

This chapter describes all of the configuration options that are related to an application server domain. Generally, the documentation reflects the order in which the configuration sections appear in the PSADMIN interface or the PSAPPSRV.CFG file.

This chapter discusses:

- Startup options.
- Database options.
- Security options.
- Workstation listener options.
- Jolt listener options.
- Jolt relay adapter options.
- Domain settings.
- PeopleCode Debugger options.
- Trace options.
- Cache settings.
- Remote call options.
- PSAPPSRV options.
- PSANALYTICSRV options.
- PSSAMSRV options.
- PSQCKSRV options.
- PSQRYSRV options.
- Integration Broker server processes.
- Simple Mail Transfer Protocol (SMTP) settings.
- Interface driver options.
- PSTOOLS options.

- Integration Broker options.
- Search options.
- Search indexes.
- PSRENSRV options.
- PSPPMSRV options.
- Select server process options.

Note. As a configuration option, you can configure a domain to *spawn* server processes according to the volume of transaction requests. There is no explicit parameter that you must set to enable spawning. In the following configuration section descriptions, some servers enable you to specify a minimum and maximum number of server processes. To enable spawning for these server processes, the maximum value must exceed the minimum value by an increment of at least one. As needed, the domain spawns server processes up to the maximum value. As the volume of transactions decreases, the number of spawned server processes decreases, or decays, until the minimum value is reached. By default, spawning is *disabled*.

Startup Options

Set database sign-in values in the Startup section.

DBName

Enter the PeopleSoft database name, such as FSDMO80 or HRDMO80. This parameter is case sensitive.

DBType

Enter the PeopleSoft database type, such as DB2ODBC, DB2UNIX, INFORMIX, MICROSOFT, ORACLE, or SYBASE. If you enter an invalid database type, PSADMIN prompts you with a valid list.

UserID

Enter the PeopleSoft user ID that is authorized to start the application server. For the application server to boot, the appropriate user ID with the correct authorizations must be assigned to this parameter. This is the ID that the application server passes to the database for authentication and connection. The user ID that you enter here is not related to the actual user (administrator) that carries out the boot command.

The Can Start Application Server permission must be set in the permission list. The authorization to start an application server does not (directly or indirectly) grant any authorizations or privileges beyond the ability to start the application server. Each user who attempts to sign in enters a unique user ID and password, which the application server uses to authenticate each user.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Security Administration, "Setting Up Permission Lists," Setting General Permissions

UserPswd

Enter the password that is used by the specified user ID that will gain access to the database. The value that you enter must be specified in uppercase to simplify administration of the system.

Connect ID

Required for all database platforms. Enter the database-level ID that the PeopleSoft system uses to make the initial connection to the database. This user name must have authority to select from PSACCESSPRFL, PSLOCK, PSOPRDEFN, and PSSTATUS.

Connect Password

Enter the password for the connect ID. For instance, this might be the UNIX user's password (either uppercase or lowercase).

ServerName

Required for Sybase and Informix. Enter the name of the server on which the PeopleSoft database is installed. This value is case sensitive.

Database Options

Use the Database Options section to specify environment variables that may improve the performance of the system. These options do not apply to every database.

SybasePacketSize

Enter a Transmission Control Protocol (TCP) packet size. The minimum value is 512 and the maximum value is 65538. The default packet size is 512. If you change the packet size, make the corresponding changes to the Sybase database server.

See Your Sybase documentation.

Do you want Analytic Servers configured?

Configures analytic servers to run in the domain to process Analytic Calculation Engine requests and to perform optimization processing.

Do you want Domains Gateway configured?

Enable this option if you are configuring a remote, or external, search server to which this domain will send search requests. That is, if you are configuring a Type-3 search option for an application server domain, you need to enable the domains gateway on the application server domain to a communication connection between the application server and its remote search domain.

Chapter 7

Working with Oracle WebLogic

This chapter provides an overview of WebLogic and discusses how to:

- Work With the WebLogic Server Administration Console.
- Start WebLogic.
- Stop WebLogic.
- Use WebLogic Server Administration Console to Monitor PeopleSoft Sessions.
- Set Up Reverse Proxy Servers.
- Set The HTTP Session Timeout.
- Set Authentication Failure Timeout.
- Enable or Disabling HTTP Keep Alive.
- Change WebLogic User Passwords.
- Implement WebLogic SSL Keys and Certificates.
- Work With WebLogic Session Cookies.
- Secure Servlets on WebLogic.
- Adjust the JVM Heap Size.
- Determine the Service Pack Level.
- Enable HTTP Access Log.

Understanding WebLogic

You use the Oracle WebLogic Server as a web server in the PeopleSoft Internet Architecture to deploy PeopleSoft applications. The PeopleSoft Internet Architecture installation on the WebLogic Server provides these primary server configuration options.

- Single server.

This domain configuration contains one server named PIA, and the entire PeopleSoft enterprise application is deployed to it. This configuration is intended for single user or very small scale, non-production environments.

Implementing WebLogic SSL Keys and Certificates

This section provides an overview of Secure Sockets Layer (SSL) encryption with WebLogic and discusses how to:

- Obtain encryption keys.
- Prepare keys and certificates for the keystore.
- Import keys and certificates into the keystore.
- Configure WebLogic SSL encryption keys.

Understanding SSL Encryption with WebLogic

To use SSL encryption with WebLogic and the current PeopleTools release, the WebLogic keystore must contain the following appropriately configured encryption keys:

- The web server's private key.
- The web server's public key, digitally signed by a trusted *certificate authority* (CA).
- The digitally signed public key of the same CA that signed the web server's key.

A public key is transferred and stored as a data element in a digital certificate or a *certificate signing request* (CSR). You can obtain public keys from a variety of sources in several different formats.

When setting up SSL, you need to:

- ensure that the encryption keys are correctly formatted.
- install them in the keystore.
- configure them using the Administration Console.

Note. If you've already installed and configured a set of encryption keys for use with WebLogic in a previous PeopleTools release, you must migrate them as external files to the keystore within the current WebLogic version.

Obtaining Encryption Keys

If you already have a set of existing encryption keys configured as external files, you don't need to obtain new ones. To find the existing keys, refer to the documentation for the PeopleTools and WebLogic releases for which those keys were installed.

The following procedure describes how to obtain new encryption keys, using as an example the trial certificate available from Verisign.

To obtain new encryption keys:

1. At a command prompt, change to the following directory:

PIA_HOME\webserv*domain_name*

Where *domain_name* is the name of the installed PeopleSoft Pure Internet Architecture domain for which you want to obtain encryption keys.

2. Enter the following command:

```
pskeymanager -create
```

Note. Pskeymanager is a script wrapper to Java's keytool, provided by PeopleSoft to manage the WebLogic keystore. For usage information, enter `pskeymanager -help`.

3. Follow the prompts and enter the requested information to create a new private key and a CSR for your web server.
 - Pskeymanager uses the keystore in *PIA_HOME*\webserv*domain_name*\keystore\pskey, with a default password of *password*.
 - Pskeymanager prompts you for an alias for the new keys, for example, *ServerABC*. This is the name you'll use to refer to the keys in the future.
 - Pskeymanager prompts you for distinguished name fields. Enter the appropriate values for your organization.
 - Pskeymanager prompts you for information about the CSR expiration date, key size, key algorithms, and the private key password. All of these fields have default values.

Pskeymanager creates the private key inside the keystore, and creates the CSR as a file called *ServerABC_certreq.txt* in the current directory. You use the CSR to obtain your signed public key certificate and a root certificate from a CA.

4. Decide which trusted CA you want to sign your web server's public key.

You can use any CA that's compatible with Sun's Java JKS standard, such as Verisign.

5. Open your CSR file in a text editor and copy its entire contents, including the first and last lines:

```
-----BEGIN NEW CERTIFICATE REQUEST-----
...
...
-----END NEW CERTIFICATE REQUEST-----
```


Preparing Keys and Certificates for the Keystore

Your encryption keys must be in *privacy enhanced mail* (PEM) format, which is Base64-encoded data. Base64 encoding uses only ASCII characters. A PEM-formatted key or certificate file has an extension of either .pem or .cer. If the file is in the binary *distinguished encoding rules* (DER) format, it has a .der extension. Use the *der2pem* Java utility to convert DER-formatted keys and certificates to PEM format.

For SSL to work, your WebLogic server must present its own public key to each client browser, along with the self-signed public key of a root CA that's also in the browser's keystore, as well as any keys necessary to establish a *chain of trust* between the two. All of these keys must be part of the same certificate file before you can import them into the WebLogic keystore.

If you generated the private key using *pskeymanager* on a WebLogic platform, it's automatically correctly formatted, password protected, and installed in the keystore with no additional steps required. However, if the private key was configured as an external file on an earlier WebLogic platform/version, you must properly format it and incorporate a password, before importing it into the current WebLogic keystore along with the public key certificates.

Converting DER Files to PEM Format

It's important to convert all DER-formatted key and certificate files to PEM format before you work with them further.

To convert DER-formatted key and certificate files to PEM format:

1. At a command prompt, change to the following directory:

`PIA_HOME\webserv\domain_name`

Where *domain_name* is the name of an installed PeopleSoft Pure Internet Architecture domain.

2. Enter the following command:

`setenv.cmd`

This sets the appropriate environment for Java commands.

3. For each DER-formatted key or certificate file, enter the following command:

`java utils.der2pem filename.der`

Make sure that you include the DER file's directory path. A new PEM file by the same name is created in the same location.

If you converted a private key file to PEM format, you must modify the header and footer to be compatible with WebLogic.

To modify the private key file header and footer:

1. Open the PEM-formatted private key file in a text editor.

2. Change the following line:

```
-----BEGIN CERTIFICATE-----
```

To this:

```
-----BEGIN RSA PRIVATE KEY-----
```

3. Change the following line:

```
-----END CERTIFICATE-----
```

To this:

```
-----END RSA PRIVATE KEY-----
```

4. Save and close the private key file.

Establishing the Server Certificate Chain of Trust

Your server certificate must contain, in addition to the web server's public key, any keys necessary to establish a chain of trust that culminates in the self-signed root certificate of a trusted root CA. That CA's root certificate must be in the keystore of any browser that's used to access your web server. Most browsers have an extensive set of trusted root certificates in their keystores.

First append the root certificate of the CA who issued your server certificate to the server certificate file. If you determine that the root certificate is not likely to be in your users' browsers, you must also append to the certificate file a chain certificate that was issued to your CA by another CA, then a chain certificate issued to that CA, and so on, until you append a root certificate that was issued by a trusted CA to itself.

For example, if your server certificate file is `demo_cert.pem` and the CA's root certificate is `ca_cert.pem`, you can open `demo_cert.pem` in a text editor, then insert the contents of `ca_cert.pem` after adding a new line at the end of the file. Make sure that each certificate follows the previous one on the next line, as follows:

```
...
...
DosdDFG256EDHY45yTRH67i345314GQE356mjsdhhjuwbtrh43Gq3QEVe45341tS
YDY6d47lDmQxDs9wGt1bkQ==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
DMICHDCCAcYCEAHSeRkM2guFL+6OvHr4AS0wDQYJKoZIhvcNAQEEBQAwwgAKxjAP
...
...
```

The result is that `demo_cert.pem`, for example, now contains the data from both certificates.

Note. You can also use the *type* command in Windows or the *cat* command in UNIX to combine the certificate files.

Password Protecting the Private Key

Private keys inside the WebLogic keystore are password protected. You can't import an external private key file into the keystore without a password. If it isn't currently password protected, use the WebLogic *wlkeytool* utility to incorporate a password into the private key file.

To password-protect an external PEM-formatted private key file:

1. At a command prompt, change to the following directory:

`WL-HOME\server\native\win\32`

Where `WL_HOME` is the root directory where you installed WebLogic.

2. Enter the following command:

```
wlkeytool insecure_privatekey.pem secure_privatekey.pem
```

Where `insecure_privatekey.pem` is the name of the original private key file, and `secure_privatekey.pem` is the name of the resulting password-protected private key file.

Note. Make sure that you include directory paths for the private key files.

The following message appears:

```
Enter password to unprotect private key:
```

3. Press Enter.

The following message appears:

```
Private key not PKCS8 encoded, trying RSA key
Private key file opened successfully
Enter password to protect private key :
```

4. Enter the password that you want to use for this key.

The following message appears:

```
Verify password to protect private key :
```

5. Enter the password again to confirm it.

The utility creates the password protected private key file that you specified. You import this key into the WebLogic keystore.

Importing Keys and Certificates Into the Keystore

Each WebLogic domain maintains its own keystore in `PIA_HOME\webserv\domain_name\keystore\pskey`, and all servers within a domain can share the same keystore.

These utilities are available for importing keys and certificates into the keystore:

Utility	When to Use
pskeymanager	If you created the private key using the pskeymanager utility on a WebLogic platform, it's already installed in the keystore. You need only use pskeymanager to import your server certificate, which should contain your web server's signed public key, your trusted CA's root certificate, and any public keys necessary to establish a chain of trust between them.

Utility	When to Use
ImportPrivateKey	If the private key was previously configured as an external file on an earlier WebLogic platform, you must import it into the WebLogic keystore along with the server certificate, using the <i>ImportPrivateKey</i> utility. The private key should be password-protected.

Using pskeymanager to Import the Server Certificate

To import the server certificate into the WebLogic keystore:

1. At a command prompt, change to the following directory:

PIA_HOME\webserv*domain_name*\bin

Where *domain_name* is the name of the installed PeopleSoft Pure Internet Architecture domain.

2. Enter the following command:

```
pskeymanager -import
```

Note. PeopleTools provides pskeymanager (a script wrapper to Java's keytool) to help manage the WebLogic keystore. For usage information, enter `pskeymanager -help`.

3. Follow the prompts and enter the requested information to create a new private key and a CSR for your web server.

Keep the following in mind:

- pskeymanager uses the keystore in *PIA_HOME*\webserv*domain_name*\keystore\pskey, with a default password of *password*.
- pskeymanager prompts you for an alias for the server certificate, for example, *ServerABC*. This should be the same alias that you specified for the corresponding private key when you created it.
- pskeymanager prompts you for the name of the server certificate file, for example, *ServerABC-cert.pem*. Include the file path if necessary.

Using ImportPrivateKey to Import an External Private Key File with the Server Certificate

To import a password-protected private key and the server certificate into the WebLogic keystore:

1. At a command prompt, change to the following directory:

PIA_HOME\webserv*domain_name*\bin

Where *domain_name* is the name of an installed PeopleSoft Pure Internet Architecture domain.

2. Enter the following command:

```
setEnv.cmd
```

This sets the appropriate environment for Java commands.

3. Enter the following command:

```
java utils.ImportPrivateKey keystore\pskey store_pass privatekey_alias
privatekey_pass servercert_file privatekey_file
```

The parameters for this command are as follows:

store_pass	Specify the password for the WebLogic pskey keystore. The default password is <i>password</i> .
privatekey_alias	Specify an alias for the private key. This is the name by which the key will be accessible inside the keystore.
privatekey_pass	Specify the password for the private key.
servercert_file	Specify the path and name of the server certificate file that includes the issuing CA's root certificate.
privatekey_file	Specify the path and name of the private key file.

Configuring WebLogic SSL Encryption Keys

This section describes how to configure the SSL encryption keys that you previously imported into the WebLogic keystore in *PIA_HOME*\webserv*domain_name*\keystore\pskey, where *domain_name* is the name of an installed PeopleSoft Pure Internet Architecture domain.

The following procedure applies to a single server configuration of PIA. In a production environment, you would perform these steps for managed server instances of PIA, PIA1, PSOL, RPS, and so on, in a multi-server domain configuration.

To configure WebLogic SSL encryption keys for the PIA server:

1. With the PIA server running, sign in to the Administration Console.
2. Access the keystore configuration pages.
 - a. In the Domain Structure tree, expand Environment, click Servers, and click PIA from the Servers list.
 - b. Select the Keystores tab.
 - c. To change the configuration section, click Lock & Edit.
 - d. Select *Custom Identity and Custom Trust* from the Keystores list.

- Update the fields on the Configure Keystore Properties page as follows:

Field	Value
Custom Identity Key Store File Name	<i>keystore/pskey</i> This should be the relative path and name of the keystore into which you imported your SSL keys.
Custom Identity Key Store Type	<i>JKS</i> Don't change this value.
Custom Identity Key Store Passphrase	<i>password</i>
Confirm Custom Identity Key Store Passphrase	Same as the value of Custom Identity Key Store Pass Phrase.
Custom Trust Key Store File Name	<i>keystore/pskey</i> This should be the relative path and name of the keystore into which you imported your SSL keys.
Custom Trust Key Store Type	<i>JKS</i> Don't change this value.
Custom Trust Key Store Passphrase	<i>password</i>
Confirm Custom Trust Key Store Passphrase	Same as the value of Custom Trust Key Store Pass Phrase.

Warning! The default keystore and private key password is *password*. This should *never* be used in a production environment. You can change a private key's password and a keystore's password using pskeymanager's change password options: -changeprivatekeypassword and -changekeystorepassword, respectively.

- Click Save.
- Access the SSL tab, and update the values on the SSL Private Key Settings as follows:

Field	Value
Private Key Alias	Specify a unique identifier, such as the web server's machine name. This should be the alias that you specified for this server's private key.
Passphrase	<i>password</i>
Confirm Passphrase	Same as the value of Passphrase.

- Click Save and Activate Changes.

You *must* click the Activate Changes button to apply your changes. If you close your browser without clicking Activate Changes, your changes will be lost.

- Restart the WebLogic PIA server.

Working With WebLogic Session Cookies

When a user signs in to a PeopleSoft application, the portal servlet generates a cookie containing the user's HTTP session ID, and sends it to the user's browser to maintain the state of the session. The name of the cookie is fixed for all users accessing that portal.

On a WebLogic portal, the session cookie's name is generated at install time based on the portal hostname and port number, which uniquely identify the portal within your PeopleSoft system. This name is stored in the portal's `weblogic.xml` file.

However, the cookie name must not start with a number, and it must not contain any periods. If your users are experiencing problems signing in to PeopleSoft applications at different URLs from the same browser session, make sure that the session cookie names at those sites are valid.

To ensure valid WebLogic session cookie names:

1. Shut down your WebLogic server.
2. Open the `weblogic.xml` file for your web server in a text editor.

You can find it in `PIA_HOME\webserv\domain_name\applications\peoplesoft\PORTAL\WEB-INF`.

3. Check the value of the session parameter called `CookieName`.

Ensure that the content of the `param-value` element doesn't start with a number or contain any periods. For example, the following session cookie name is invalid:

```
<session-param>
  <param-name>CookieName</param-name>
  <param-value>57.28.208.21-80-WebLogicSession</param-value>
</session-param>
```

You can replace the periods with dashes (-). Following is a valid version of the session cookie name:

```
<session-param>
  <param-name>CookieName</param-name>
  <param-value>c57-28-208-21-80-WebLogicSession</param-value>
</session-param>
```

4. Save and close the file.
5. Restart your WebLogic server.

Securing Servlets on WebLogic

This section describes how to restrict access to a web resource for a single server configuration of PIA. When in production, a multi server configuration would be used to perform these steps to your managed server instances of PIA, PIA1, PIA2, and so on. WebLogic Server provides an optional level of security to restrict access to resources on the web server.

The following steps describe how to restrict access to the PeopleSoft Portal servlet using a WebLogic ID and password. This, for example, could be applied to the report repository servlet.

To restrict access to a servlet:

1. Start the PIA server.
2. Open the Administration Console.
3. Change the security model to "Custom Roles And Policies":
 - a. In the Domain Structure section select Security Realms, myrealm.
 - b. Click Lock and Edit.
 - c. For the Security Model Default field, select Custom Roles And Policies option from the drop down list.
 - d. Click Save.
 - e. Click Activate Changes.
 - f. Select the default security model for the application to Custom Roles and Policies.
4. Enable security policy checks for web applications.
 - a. Edit config.xml under *PIA_HOME*\webserv\<domain>\config.

Note. Backup the file before you make any changes.

- b. Find the <app-deployment> section and add the following line in all of the <app-deployment> (or application deployment) sections:

```
<security-dd-model>CustomRolesAndPolicies</security-dd-model>
```

For example, one of the sections will look similar to the following:

```
<app-deployment>
  <name>peoplesoft</name>
  <target>PIA</target>
  <module-type>ear</module-type>
  <source-path>applications/peoplesoft</source-path>
  <deployment-order>1</deployment-order>
  <security-dd-model>CustomRolesAndPolicies</security-dd-model>
  <staging-mode>nostage</staging-mode>
</app-deployment>
```

- c. Save the file and restart the server so that the changes will take effect.
5. (Optional) Define the WebLogic users that you want to use.

If you want to use one of the provided WebLogic user accounts (system, operator, or monitor) you can skip this step. Otherwise, create a new WebLogic user account:

- a. Under Domain Structure, select Security Realms, myrealm.
- b. Click Users and Groups tab.
- c. Under the Users tab, click New to create a new user.
- d. Enter user name and password and click OK.

6. (Optional) Create a user group, and add user(s).

If you want to create a user group, add your users to that group and in the following steps select Caller is Member of group instead of User name of caller. To create a group:

- a. Under Domain Structure, select Security Realms, myrealm.
- b. Click the Users and Groups tab.
- c. Under the Groups tab, click New to create a new group.
- d. Enter the name of the group and description. Click OK.
- e. Select the Users tab, select your new user, and then click the Groups tab.
- f. Move the appropriate group from the Available box to the Chosen box.

7. Define a security policy for the PeopleSoft Portal web application.

To restrict access to the Portal web application, perform the following in the navigation window on the left:

- a. Under Domain Structure, select Deployments, and select peoplesoft from the list of applications.
- b. Under Overview tab, select the portal module which appears as "/" in the Modules and Components table.
- c. Click the Security tab.
- d. Define a new URL pattern for this web module. Select New and enter the URL pattern as "/*" or specify the URL requiring authentication, and click OK.
- e. Select the URL pattern that you just created from the table and enter a security policy for this URL pattern.
- f. Add the condition to give access to the particular user you want to have access to this URL or any other conditions by clicking Add Conditions.
- g. To restrict access to a specific user, select the policy condition of User name of caller, click Add, and when prompted specify the user name. Repeat this step for additional users, groups, or access times. For access times, the server's local time is used.
- h. Click Finish and go back to the policy page and click Save.

This action does not require a server reboot.

8. Test the configuration.

Test your new security policy by accessing the URL you defined. If the security policy is active, you'll be prompted to sign in using a user ID that you added.

Adjusting the JVM Heap Size

Java options, such as JVM heap size and VM mode, used by the WebLogic server are stored in your WebLogic domain's setEnv script, stored in:

PIA_HOME\websrv\peoplesoft\bin

These options are specified in the script using the `JAVA_OPTIONS_OSplatform` environment variable. If you need to adjust any of the Java options, including changing the JVM heap size, you must manually edit the script.

The Microsoft Windows `setEnv.cmd` script contains the following default setting:

```
SET JAVA_OPTIONS_WIN32=-jrockit -XnoOpt -XXnoJITInline -Xms512m
-Xmx512m -Dtoplink.xml.platform=oracle.toplink.platform.xml.jaxp.JAXPPlatform
```

The UNIX standard `setEnv.sh` script contains the following default settings for supported Linux and UNIX platforms:

```
JAVA_OPTIONS_AIX="-Xms128m -Xmx256m"

JAVA_OPTIONS_HPUX="-server -Xms256m -Xmx256m -XX:MaxPermSize=256m"

JAVA_OPTIONS_LINUX="-jrockit -XnoOpt -XXnoJITInline -Xms512m -Xmx512m
-Dtoplink.xml.platform=oracle.toplink.platform.xml.jaxp.JAXPPlatform
-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0"

JAVA_OPTIONS_SOLARIS="-server -Xms256m -Xmx256m -XX:MaxPermSize=256m"
```

You modify the `Xms` parameter to adjust minimum heap size, and modify the `Xmx` parameter to adjust maximum heap size. For most operating systems, the minimum and maximum values of the heap size are equal, which is recommended for better performance.

In a multi-server domain, the platform-specific versions of the `JAVA_OPTIONS` environment variable that appear in the `setEnv` script apply only to managed servers. The administration server doesn't use any of these variables, but it assumes default JVM heap size values of `"-Xms32m -Xmx64m"`.

To adjust the JVM heap size for the administration server, add the environment variable `JAVA_OPTIONS_ADMINSERVER` following the last entry for `JAVA_OPTIONS_OSplatform`, and set it to your required minimum and maximum values, for example:

```
JAVA_OPTIONS_ADMINSERVER="-Xms64m -Xmx128m"
```

Note. If you're running WebLogic as a Microsoft Windows service and you modify `setEnv.cmd`, you must reinstall the service.

See Also

Appendix B, "WebLogic Managed Server Architecture," Managing JVM Heap Size, page 343

Determining the Service Pack Level

A summary of installed products and their versions and service pack levels is maintained in the `BEA_HOME\registry.xml` file. However, to confirm version information, it's more accurate to check the WebLogic log. A failed service pack install may be indicated in the log, but not found at runtime.

This section discusses how to:

- Check the WebLogic log

- Query WebLogic

Checking Version Information in the WebLogic Log

The WebLogic log is located in:

PIA_HOME\webserv\peoplesoft\servers*<server name>*\logs\weblogic_server_weblogic.log

For version information, look for an entry similar to:

```
####<Mar 19, 2009 4:14:02 AM PDT> <Info> <Management> <PLE-INFODEV-12> <>
<Main Thread> <> <> <>
<1237461242812> <BEA-141107> <Version: WebLogic Server 10.3
Fri Jul 25 16:30:05 EDT 2008 1137967 >
```

Checking Version Information at the Command Line

To query WebLogic for version information from the command line, submit the following arguments to setEnv. For example,

```
C:\PT850\webserv\peoplesoft\bin>setenv.cmd java weblogic.Admin
-url t3://localhost:80 -username <username> -password <password> VERSION

WebLogic Server 10.3  Fri Jun 23 20:47:26 EDT 2009 783464
```

Note. For UNIX, use setEnv.sh.

Enabling HTTP Access Log

This section describes how to change HTTP logging for a single server configuration of PIA. When in production, a multi server configuration would be used to perform these steps to your managed server instance of PIA or PIA1, PIA 2, and so on.

To enable or disable HTTP access log:

1. Make sure the PIA server is running.

See [Chapter 7, "Working with Oracle WebLogic," Starting WebLogic, page 121.](#)

See [Chapter 7, "Working with Oracle WebLogic," Stopping WebLogic, page 123.](#)

2. Log on to the Administrative Console.
3. Open Server's Logging configuration page.
 - a. In the Domain Structure tree, expand Environment, and click Servers.
 - b. Click PIA (or your custom server name) in the Servers list.
 - c. Select the Logging tab, and select the HTTP tab.

4. Enable HTTP access logging.
 - a. Click the Lock&Edit button.
 - b. Select the HTTP access log file enabled check box to turn on the access.log.
 - c. Modify the Log file name field if desired.
 - d. Click Save and Activate Changes.
5. Restart the WebLogic Server.

Chapter 8

Working with IBM WebSphere

This chapter contains an overview and discusses:

- Starting and Stopping WebSphere Application Servers.
- Configuring Reverse Proxy Servers For WebSphere.
- Setting Up SSL For WebSphere.
- Securing The Administrative Console and Applications For WebSphere.
- Setting HTTP Session Timeout.
- Setting Authentication Failure Timeout.
- Working With JVM Heap Size.
- Working with Logging and Tracing Options.

Understanding WebSphere Application Server Within Your PeopleSoft Implementation

This section discusses:

- Deploying PeopleSoft Applications With WebSphere.
- Using the Integrated Solution Console.
- WebSphere Application Server profiles.
- IBM HTTP Server.

Deploying PeopleSoft Applications With WebSphere

The IBM WebSphere Application Server (WAS) is a J2EE application server that PeopleTools uses as a web server to deploy PeopleSoft applications, as an alternative to Oracle WebLogic Server. To install and use WebSphere, download the appropriate version from Oracle E-Delivery site, and install it using the detailed instructions in the Enterprise PeopleTools installation documentation.

Note. WebSphere Base edition and Network Deployment edition (ND) are packaged and installed together. PeopleSoft applications use the Network Deployment edition for application deployment.

See Also

Enterprise PeopleTools 8.50 Installation for your platform

Using The Integrated Solutions Console

To view and configure WebSphere settings, you use the web-based administrative console called Integrated Solutions Console, which is based on the Integrated Solutions Console (ISC) framework, providing a consistent and integrated capability for administering IBM software. The ISC enables you to access settings related to key areas of server administration, including:

- servers
- applications
- security
- environment
- users and groups
- monitoring and tuning

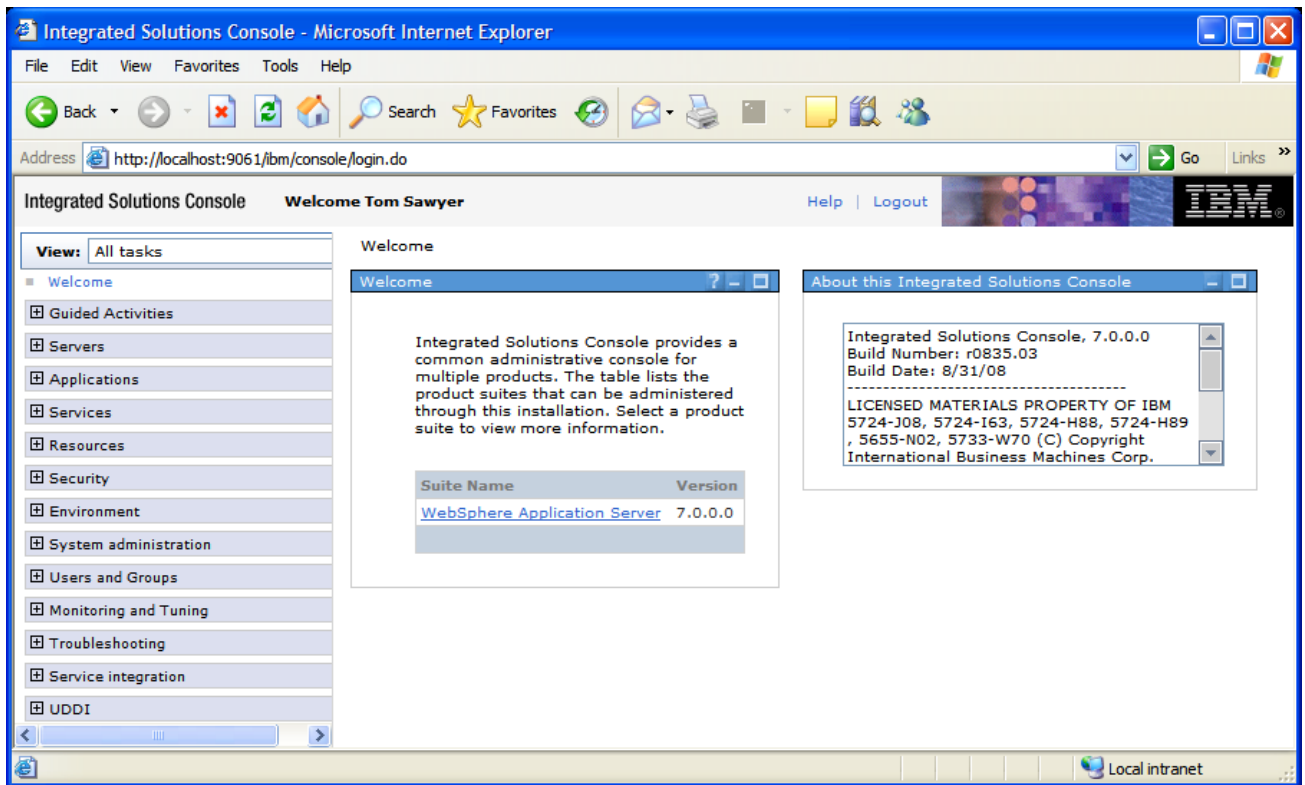
To access ISC:

1. Enter the following URL:

`http://WAS_Hostname:9060/ibm/console`

2. On the ISC login page, enter the appropriate credentials in the User ID field and click Login.

If you have enabled Global Security, add the appropriate user ID and password. If Global Security is not enabled, add any user name in the User ID field (or leave it blank).



Integration Solutions Console

If you have more than one WebSphere server profile installed on a machine, the ISC port number for each installation is different. You can locate the port number corresponding to each profile by viewing the AboutThisProfile.txt file in:

`PIA_HOME\websrv\<profilename>\logs`

A key benefit of the ISC is that it allows administrators to create a custom navigation list of tasks performed more frequently, using the View, My Tasks feature in the upper, left-hand corner.

Note. You can perform numerous administrative tasks using ISC. This guide discusses those that pertain most specifically to deploying PeopleSoft applications. It is assumed that you are familiar with the topics contained in the IBM WebSphere documentation.

Note. In this document, we refer to the ISC as the "administrative console."

See Also

IBM WebSphere documentation

WebSphere Application Server Profiles

Different WebSphere Application Server environments are defined using *profiles*. A profile defines the runtime environment (JVM) for web applications. A profile includes:

- files and settings that the server processes require during runtime.
- a common set of shared WebSphere product binaries.
- a unique directory name containing required files and subdirectories.

PeopleSoft Internet Architecture makes use of the *application server* environment type to deploy the PeopleSoft Enterprise Applications. When you run the PeopleSoft Internet Architecture installation, one of the elements that the system creates are the necessary application server profiles. The number of application server profiles created varies depending on whether you elected to implement a single server or multi server installation.

Note. The location for Application Server profile location differs from the default location when PIA creates Application Server Profile.

Single Server Installation

If you specified the default application name of "peoplesoft" at install time, an application server profile with the same name gets created in *PIA_HOME\websrv\<peoplesoft>*. When the PIA install creates an application server profile, it creates a default server names "server1" (single JVM process) and all of the PeopleSoft web modules are deployed to this single server.

You can view the PeopleSoft modules, such as PORTAL, PSEMHUB, PSIGW, and so on, in the single server profile directory structure.

PIA_HOME\websrv\<peoplesoft>\installedApps

Multi Server Installation

If you specified the default application name of "peoplesoft" at install time, the application server profile is created under *PIA_HOME\websrv*. In the case of a multi server installation, the profile contains the following separate servers.

Profile	Description
server1_peoplesoft	Hosts the PORTAL, PSIGW, and other web modules used for PeopleSoft online transactions.
PSEMHUB_peoplesoft	Hosts the PSEMHUB web module used by the PeopleSoft Environment Management Hub.
PSOL_peoplesoft	Hosts the PSOL web module used by the PeopleSoft Online Library Manager, which facilitates the HTML PeopleBooks installation.

IBM HTTP Server

The IBM HTTP Server is the IBM version of the Apache HTTP Server. It is a separate installation required for IHS and IHS plug-ins. You may use the IBM HTTP Server as a reverse proxy server within your PeopleSoft implementation.

See Also

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.ihs.doc/info/welcome_ihs.html

Enterprise PeopleTools 8.50 Installation, "Installing Web Server Products"

Starting and Stopping WebSphere Application Servers

By default, all of the servers of a WebSphere instance are stopped when you install PIA. You need to start the server in order to access the PeopleSoft components.

To start and stop WebSphere servers, use the delivered .BAT and .SH files located in:

PIA_HOME\webserv\<profilename>\bin

Starting the WebSphere Server

To start the WebSphere server, enter the following command:

<i>Operating System</i>	<i>Command</i>
Windows	startServer.bat <i>server_name</i> -profileName <i>profilename</i>
UNIX	startServer.sh <i>server_name</i> -profileName <i>profilename</i>

Stopping the WebSphere Server

To start the WebSphere server, enter the following command:

<i>Operating System</i>	<i>Command</i>
Windows	stopServer.bat <i>server_name</i> -profileName <i>profilename</i>
UNIX	stopServer.sh <i>server_name</i> -profileName <i>profilename</i>

Configuring Reverse Proxy Servers For WebSphere

This section contains an overview and discusses how to:

- Configure an IBM HTTP server as a reverse proxy server.
- Configure Microsoft IIS as a reverse proxy server.

- Configure Sun Java System Web Server as a reverse proxy server.

Understanding Reverse Proxy Servers With IBM WebSphere

Using reverse proxy servers adds an additional, protective layer between your application and the internet or your end users. A reverse proxy server receives user requests, and sends them to a back end content server, usually behind a firewall. The back end server, in this case your PeopleSoft web server, remains unknown to the user.

For your PeopleSoft implementation, you can configure reverse proxy servers for WebSphere on the following web servers:

- IBM HTTP Server.
- Microsoft Internet Information Server (IIS).
- Sun Java System Web Server.

The communication between your web server and your reverse proxy server is configured using delivered plug-ins. You must install the web server software before you can install a plug-in for the web server. You can install the web server plug-in by itself on a machine where WebSphere Application Server ND has been installed but the plug-in has not. You can also install a plug-in on a remote machine where the HTTP proxy server is already installed.

Web Server Plug-In

Web server plug-ins enable the web server to communicate requests for dynamic content, such as servlets, to the application server. A web server plug-in is associated with each web server definition. The configuration file (plugin-cfg.xml) that is generated for each plug-in is based on the applications that are routed through the associated web server.

WebSphere RPS Plug-in

The RPS plug-in is used to forward HTTP requests from the proxy server to the PeopleSoft web server. The RPS plug-in provides:

- XML-based configuration file.
- Standard protocol recognized by firewall products.
- Security using HTTPS, replacing proprietary Open Servlet Engine (OSE) over Secure Sockets Layer (SSL).

Configuring IBM HTTP Server as a Reverse Proxy Server

To configure the IBM HTTP Server for use as a reverse proxy server, you use the IHS plug-in.

Before you perform the following steps, you need to install the following items:

- IBM HTTP Server.
- Web server plug-ins.

See *Enterprise PeopleTools 8.50 Installation*, Installing Web Server Products, "Installing IBM HTTP Server 7.0 and Web Server Plug-ins"

To configure IHS for reverse proxy:

1. Start WebSphere server and open the Administrative Console window.
2. Navigate to Environment, Virtual Hosts, pia_host, Host Aliases.

The PeopleSoft application is deployed on a virtual host called "pia_host".

3. Create new entries for the required ports.

For example:

Hostname = *, Port =10001 (for web server port)

Hostname = *, Port =10002 (for HTTP Administration Server port)

Hostname = *, Port =10043 (for SSL port assigned to IHS)

4. In a multi server environment, repeat the steps 2. and 3. for the other virtual hosts "psol_host" and "psemhub_host".
5. Click Apply and save the settings to "master".

This updates <PS_HOME>/webserv/profile_name/config/cells/node_name/virtualhosts.xml

6. From the WebSphere Plug-ins installation, copy the configureWebserverDefinition script from the <Plugin_Install_Root>/bin to the directory <PS_HOME>/webserv/profile_name/bin and run it.

This creates the web server definition in WebSphere server.

7. Generate the plugin-cfg.xml by selecting the web server definition in Servers, Web servers.
8. Copy the plugin-cfg.xml from
<PS_HOME>/profile_name/config/cells/cell_name/nodes/node_name/servers/WebserverDefinition to
<Plugin_Install_Root>/config/WebserverDefinition so that IHS can communicate with WebSphere directly and access the PeopleSoft application.
9. Restart the WebSphere server, IBM HTTP Server, and IBM HTTP Administration Server.
10. Verify accessing the PeopleSoft application using the IHS HTTP port.

Configuring Microsoft IIS as a Reverse Proxy Server

This section discusses how to configure Microsoft IIS as a reverse proxy server for WebSphere. Before you perform these steps, the following items need to be installed:

- Microsoft IIS.
- Web server plug-ins.

Microsoft IIS Installation should have the following components already installed in order for the RPS setup to be successful:

- IIS Management Compatibility

- IIS Management Console
- IIS Scripting Tools
- IIS WMI Compatibility
- IIS Metabase compatibility
- ISAPI Extensions
- ISAPI Filters

See

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/topic/com.ibm.websphere.nd.doc/info/ae/ae/tins_webplugins.html.

See *Enterprise PeopleTools 8.50 Installation*, Installing Web Server Products, "Installing IBM HTTP Server 7.0 and Web Server Plug-ins"

See your Microsoft IIS documentation

Installing WebSphere Web Server Plugin for Microsoft IIS

Before installing the plugin, create a new IIS website according to the instructions given in the IBM WebSphere "Installing Web Server Plug-ins" documentation. Alternately, you can use the Default website if it is available. The Plugins install wizard prompts for the location where the WebSphere plugins for IIS need to be installed and also for the location where WebSphere is installed. The wizard also prompts for the name of the web server definition to be created.

Note. The WebSphere image provided by Oracle for your PeopleSoft implementation, comes with 32-bit and 64-bit web server plugins. The 32-bit plugin installer is in CD1 and 64-bit plugin installer is in CD2. Install the appropriate plugin based on your machine architecture and the mode (32-bit or 64-bit) in which your IIS is running.

Configuring the Plugin on Microsoft IIS

To configure the plugin:

1. From the WebSphere plugin installation, copy the configureWebserverDefinition script from the <Plugin_Install_Root>/bin to the directory <PS_HOME>/webserv/profile_name/bin.
2. Start the WebSphere server and run the configureWebserverDefinition script from the PIA profile location.

This creates the web server definition in the WebSphere server.

3. In the Administrative Console select Environment, Virtual hosts, and add new Host Alias entries for the IIS web server HTTP port into following virtual hosts:
 - pia_host
 - psemhub_host
 - psol_host

4. Select Servers, Web servers, and select the server definition and generate the plugin-cfg.xml.
5. Copy the plugin-cfg.xml from
`<PS_HOME>/profile_name/config/cells/cell_name/nodes/node_name/servers/WebserverDefinition` to
`<Plugin_Install_Root>/config/WebserverDefinition`.

This enables the IIS web server to communicate with WebSphere directly and access the PeopleSoft application.

6. Restart the Microsoft IIS web server, and start the IIS website that you have created.
7. Restart the WebSphere server and login to the Administrative Console.
8. (Optional) Enable Administrative Console to manage the IIS web server.

This will show the IIS web server as running. This step is needed only if you want to manage your IIS web server from the WebSphere Administrative Console.

- a. In the Administrative Console select Servers, Server Type, Web servers.
 - b. Click the web server definition to create one for the IIS web server.
 - c. Enter the port number for the IIS web server.
9. Access the PeopleSoft application through the IIS web server HTTP port:

`http://<hostname>:<IIS_HTTPPort>/ps/signon.html`

Note. If you have a Windows 64-bit machine, IIS runs in 64-bit mode by default and requires 64-bit WebSphere plugins to be installed. However, if you have installed 32-bit WebSphere plugins then you can set IIS to run in 32-bit mode by executing the following script at the command prompt:

```
CSCRIPT %SYSTEMDRIVE%\Inetpub\AdminScripts\adsutil.vbs SET
W3SVC/AppPools/Enable32bitAppOnWin64 1
```

Restart the IIS web server following the execution of this script.

Configuring Sun Java System Web Server as a Reverse Proxy Server

The following steps discuss how to configure the Sun Java System Web Server as a Reverse Proxy Server. Before you perform these steps, the following items need to be installed:

- Sun Java System Web Server.
- Web server plug-ins.

Installing WebSphere Plugin for Sun Java System Web Server

The Plugins install wizard prompts for the location where WebSphere and Sun Java System Web Server are installed. The wizard also prompts for the name of the web server definition to be created, the location of the obj.conf file, and the location magnus.conf file. These two configuration files can be found under the Sun Java System Web Server installation in the following directory:

`$SunJava_Home/https-<hostname>/config`

Configuring the Plugin

To configure the plugin:

1. From the WebSphere Plugins installation, copy the configureWebserverDefinition script from the <Plugin_Install_Root>/bin to the directory <PS_HOME>/webserv/profile_name/bin.
2. Start the WebSphere server and run the configureWebserverDefinition script from PIA profile location.

This creates the web server definition in WebSphere server.

3. In the Administrative Console, select Environment, Virtual Host, and add the new Host Alias entries for Sun Java System Web Server HTTP port into following virtual hosts:

- pia_host
- psemhub_host
- psol_host

4. Select Servers, Web servers and generate the plugin-cfg.xml by selecting the web server definition.
5. Copy the plugin-cfg.xml from <PS_HOME>/profile_name/config/cells/cell_name/nodes/node_name/servers/WebserverDefinition to <Plugin_Install_Root>/config/WebserverDefinition,

This enables the Sun Java System Web Server to communicate with WebSphere directly and access the PeopleSoft application.

6. Note the following entries in the Obj.conf file.

After the <Object name=default> tag

```
Service fn="as_handler"
AddLog fn="as_term"
```

7. Note the following entries in the Magnus.conf file.

UNIX:

```
Init fn="load-modules"
    funcs="as_init,as_handler,as_term"
    shlib="/opt/IBM/WebSphere/Plugins/bin/libns61_http.so"

Init fn="as_init"
    bootstrap.properties="/opt/IBM/WebSphere/Plugins/config/webserver1/plugin->
cfg.xml"
```

Windows:

```
Init fn="load-modules"
    funcs="as_init,as_handler,as_term"
    shlib="C:\IBM\WebSphere\Plugins\bin\ns41_http.dll"

Init fn="as_init"
    bootstrap.properties="C:\IBM\WebSphere\Plugins\config\webserver1\plugin->
cfg.xml"
```

8. Restart the Sun Java System Web Server.

9. Restart the WebSphere web server.
10. Access the PeoplesSoft application through the Sun Java System Web Server HTTP port.
`http://<hostname>:<SunJava_HTTPPort>/ps/signon.html`
11. (Optional) Enable Administrative Console to manage the Sun Java System Web Server.

This will show the Sun Java System Web Server as running. This step is needed only if you want to manage your Sun Java System Web Server from the WebSphere Administrative Console.

- a. In the Administrative Console select Servers, Server Type, Web servers.
- b. Click the web server definition to create one for Sun Java System Web Server.
- c. Enter the port number for the Sun Java System Web Server's.

Setting Up SSL For WebSphere

This section provides an overview and discusses how to:

- Generate a certificate using `pskeymanager`.
- Configure the WebSphere container to support SSL.

Understanding WebSphere Key Stores

WebSphere manages keys in key store files. There are two types of files:

- key stores
- trust stores

These store types are very similar, however the trust store contains only trusted signers. The Certificate Authority (CA) certificates and other signing certificates are kept in a trust store. Personal certificates with private keys are stored in a key store.

The `pskeymanager` utility is a wrapper to Java's `keytool`, used to manage the predefined WebSphere keystore located in the following directory:

PIA_HOME\webserv\profilename\installedApps\profilenameNodeCell\peoplesoft.ear\keystore\pskey

Generating a Certificate Using `pskeymanager`

Use the following steps to generate a self-signed certificate for the web container.

To generate a certificate using `pskeymanager`:

1. At a command prompt, change to the WebSphere domain directory, for example:

PIA_HOME \webserv\profilename\installedApps\profilenameNodeCell\peoplesoft.ear

2. Create a new private key and certificate request for your server.

a. Run the following command:

```
pskeymanager.cmd -create
```

- b. Follow the prompts and specify the required information for creating a certificate, such as alias, common name, organizational unit, location, and so on.
- c. Make sure a Certificate Signing Request (CSR) file named *alias_certreq.txt* was created in the *peoplesoft.ear* directory.

You submit this data to a CA for obtaining a public key that you can load into your key store.

3. Decide which CA you wish to use.

You may use an CA that is compatible with Sun's Java JKS standard.

As an example, the following steps indicate how to submit the CSR that you generated to Verisign to obtain a trial certificate.

4. Submit your CSR to a CA.

For example, access Verisign's test cart enrollment site at:

<https://www.verisign.com/products/srv/trial/intro.html>

When prompted, copy and paste the contents of your CSR, provide all necessary contact information, and submit the request.

5. Check your email for the certificate sent from the CA.

The certificate from the CA should look similar to the following:

```
-----BEGIN CERTIFICATE-----
DMICHDCcAcYCEAHSeRkM2guFL+60vHr4AS0wDQYJKoZIhvcNAQEEBQAwwgAKxFjAP
AANVBAAoTDVZlcm1TaWduLCLbAMxRzBFBgNVBAsTPnd3dy52ZXJpc2lnbi5jb20S
VcVwb3NpdG9yeS9UZXXN0Q1ETIEluY29ycC4gQnkgUmVmLiBMaWFiLiBMVEQuMUYF
LIGEc3VyYW5jZXMgKEMpVRMxOSDFertdsfh67TIwNDAwMDAwMFoXDTAwMTIxODIA
ONT1LVoweTELMAKGA1UERhMCVVMxEzARBgNVBAgTCkNhbg1mb3JuaWEeEzARBgNK
VBAUCOBsZWZzYW50b24BEzARBgNVBAoUC1Blb3BsZVNvZnQxZDASBgNVBAsUC1BT
Eb3sZVVvb2xzMRUwEwADVQQDFAxEQ1JpV04xMTE0MDAwXDANBgkqhkiG9w0BAQET
SAALADBEAkeAucfM/GOQhdkk4Q0ZD5i1l4gp6WTYMc4IaReoCYkEAmDKAVcYzY3R
Mdbp4RC8SABd3bjjDOHcoCak9U6oSwwL+HQIDAQABMA0GCSqGSIb3DQEBAUAA0EO
Arm3uf634Md0fqqNxxhAL+e9rbY0ia/X48Axloi17+kLtVI1YPOp+Jy6Slp5iNIFC
DhskdDFH45AjSDAFhjruGHJK56SDFGqwq23SFRfgtjkjyu673424yGWE5Gw4576K
DosdDFG256EDHY45yTRH67i345314GQE356mjsdhhjuwbtrh43Gq3QEVe45341tS
YDY6d47lDmQxDS9wGt1bkQ==
-----END CERTIFICATE-----
```

Copy the entire certificate, including --BEGIN CERTIFICATE-- and --END CERTIFICATE--, and save it as a file named *webservername-cert.pem*.

Note. To save the file, don't use a word processor that inserts formatting or control characters.

Note. If you need to FTP your certificate to UNIX, you must FTP it in ASCII mode.

6. Download the CA root certificate:

For example, download the Verisign trial root CA certificate.

a. Download the Verisign Trial Root CA certificate from:

https://www.verisign.com/support/verisign-intermediate-ca/Trial_Secure_Server_Root/index.html

b. From the specified link, click Select All, and copy the contents of the certificate into the verisignRootCA.cer file and save it to your WebSphere domain directory.

c. Download VeriSign's Trial Intermediate CA certificate from:

<https://www.verisign.com/support/verisign-intermediate-ca/trial-secure-server-intermediate/index.html>

d. Click Select All and copy the contents of the certificate into the verisignInterCA.cer file, and save it into your WebSphere domain directory. You can also append the contents of this Trail Intermediate CA certificate to the Root CA certificate file verisignRootCA.cer.

Note. If you need to FTP your certificate to UNIX, you must FTP it in ASCII mode to your WebSphere domain directory.

7. Import the CA's certificates into your key store.

To import the CA's public certificate into your key store, run:

```
pskeymanager.cmd -import -trustcacerts
```

For example, when prompted for an alias, specify the appropriate name to store the CA as, for example *VerisignTrialCA*. This name is only an alias for this certificate.

When prompted for the certificate file to import, specify the root certificate, such as verisignRootCA.cer file.

If any other certificates (such as the Verisign Intermediate certificate) are saved into a different file, run the command to import that certificate also.

8. Import your certificate into your keystore.

To import your public certificate into your keystore, run the following command from the command prompt

```
pskeymanager.cmd -import
```

When prompted for an alias, specify the same alias you did when you created your private key and certificate request.

When prompted for the certificate file to import, specify your certificate file, *webservername-cert.pem*.

Configuring the WebSphere Container to Support SSL

To complete the SSL configuration, the web container must be modified to use the self-signed certificates you created.

To set up WebSphere Container SSL:

1. Start the Administrative Console, and select Security, SSL certificate and key management, Manage endpoint security configurations.
2. On the Local Topology tab, expand the Inbound tree, and click on the appropriate node, as in *peoplesoftNode*.
3. In the Related Items list on the right, click *Key stores and certificates*.
4. In the resource table, click *NodeDefaultKeyStore* in the Name column.
5. In the Additional Properties list on the right, click *Personal certificates*.
6. Click Import.
7. On the General Properties page, select the Key store file radio button, and complete the following:
 - a. In the Key file name field, enter the fully qualified path to the keystore file containing the certificate to import.

For example,

`C:\PSHCM\webserve\peoplesoft\installedApps\peoplesoftNodeCell\peoplesoft.ear\keystore\pskey`
 - b. From the Type dropdown list, select *JKS*.
 - c. In the Key file password, enter the password you specified when creating pskey.
 - d. Click Get Key File Aliases.

The system searches the key store and should populate the Certificate alias to import list.
 - e. If you want to use a new alias, enter a new value in the Imported certificate alias field, otherwise leave it empty.
8. Click Apply and OK.
9. Save the configuration in the Administrative Console.

Note. To configure Outbound SSL, repeat the same steps within the Outbound tree.

Securing The Administrative Console and Applications For WebSphere

This section provides an overview, and discusses how to:

- Secure the Administrative Console.
- Secure applications (servlets).

Understanding WebSphere Security

This section discusses:

- Registries and repositories.
- Security Domains.

Registries and Repositories

With the IBM WebSphere Application Server, a user registry, or repository, authenticates a user and retrieves information about users and groups to perform security-related functions, including authentication and authorization. WebSphere makes access control decisions based on the information stored in the user registry or repository.

WebSphere supports multiple types of registries and repositories, including:

- Local operating system registry.
- Standalone lightweight directory access protocol (LDAP) registry.
- A standalone custom registry.
- Federated repositories.

Security Domains

WebSphere allows "fine-grained security configuration" enabling security to be configured at the cell, node, server, or cluster level. Also, with WebSphere, you can configure application security separately for administrative security within a cell environment. Administrative security can be enabled through Global Security, while application security can be enabled by creating a new Security Domain and customizing the security configurations specific to the domain.

You configure separate applications to use different security configurations by assigning the servers, cluster, or Service Integration Buses hosting the applications or servlets to appropriate security domains.

Note. For a user to configure multiple security domains, they must be assigned to the administrator role.

For example, administration can be configured to use a federated repository while the applications can be configured to use an LDAP registry. In previous versions of Websphere, administrative and user applications use global security attributes by default, where a user registry defined in global security authenticates users for every application in the cell. Using multiple security domains provides more flexibility and simpler configurations.

This section illustrates how you can configure application security separately by creating a new security domain and assign it to the server level scope. The Administrative Console is secured with global security settings. The primary admin user belongs to the Local Operating System registry realm. The Report Repository Servlet of the PeopleTools application is secured by providing access to a user from another realm.

Note. For simplicity, in this example, both realms are Local Operating System realms. Realms may also be implemented as a Federated Repository or LDAP Registry.

See Also

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.nd.doc/info/ae/ae/csec_locals.html

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0/index.jsp?topic=/com.ibm.websphere.express.doc/info/exp/ae/csec_sec_multiple_domains.html

Securing the Administrative Console

This section discusses how to secure the Administrative Console for the profile associated with your PeopleSoft application. It is assumed that you have already set up user names and passwords on the host machine.

For Windows, the user should be in the Administrators group. On UNIX, use the root user.

This user information (user ID and password) will be used during configuration and authentication.

Configuring Administrative Security

To configure Administrative Security:

1. Open the Administrative Console of the profile hosting the PeopleSoft application.
2. Select Security, Global Security.
3. In the User account repository group box, set Available realm definitions to Local operating system, and click Configure.
4. In the Primary administrative user name field, enter a valid user name.

In this example "ansrivat11" will be the admin user ID. This value is the name of a user with administrative privileges that is defined in the registry. This user name is used to access the Administrative Console or used by wsadmin. On UNIX this should be "root" user.

5. Select the Server User Identity that is stored in the repository radio button, and enter the same user name as mentioned above for Server user ID or administrative user on version WAS 6.x node field, and enter the user's password that corresponds to server identity.
6. In the Custom properties table, click New, and enter these values:

Name: *com.ibm.websphere.registry.UseRegistry*

Value: *local*
7. Click Apply and Save.
8. Select Security, Global Security and click the Administrative user roles link, ensure that the user ID you just specified as the Primary administrative user name appears assigned to the Primary administrative user name role.

9. Return to Security, Global security and set these options:
 - a. Select the Enable administrative security check box.
 - b. Select the Enable application security checkbox only if you want the application security to have the same global security configuration as the admin user. In this example, we illustrate the application security enabled separately within a separate domain.
 - c. Deselect the Java 2 security options. PeopleSoft does not adhere completely to Java 2 security.
 - d. Click Apply.

Configuring SSL

If you are using the default SSL configuration, extract the signer certificate from the WebSphere default Trust Store. If you have set up a customized SSL configuration extract the signer certificate corresponding to that configuration.

To configure SSL:

1. In the Administrative Console select Security, SSL certificates and key management.
2. Under Related Items, click Key stores and certificates.
3. Click NodeDefaultTrustStore.
4. Click Signer certificates.
5. Select Root signer and click Extract.
6. Enter a unique path and filename for the signer, such as c:\temp\rootsigner.arm.
7. Click OK.
8. Add the exported signer certificate to DummyServerTrustFile.jks and DummyClientTrustFile.jks in the <PS_HOME>/webserv/<profile name>/etc directory to enable SSL connectivity, using these steps for both.
 - a. Open the key management utility, iKeyman, for that product version.
 - b. Start ikeyman.bat or ikeyman.sh from <PS_HOME>/webserv/<profile name>/bin.
 - c. Select Key Database File, Open.
 - d. Open <PS_HOME>/webserv/<profilename>/etc/DummyServerTrustFile.jks.
 - e. Enter WebAS for the password. (Do not change this password.)
 - f. Select Add and enter path to the signer certificate you extracted.
9. Log out from the Administrative Console and stop the web server.

Modifying soap.client.props File

In any text editor, open the soap.client.props file located in *PIA_HOME*\webserv\<profile name>\properties. Set securityEnabled to true, and specify the appropriate user ID and password for loginUserID and loginPassword. For example,

```
com.ibm.SOAP.securityEnabled=true

com.ibm.SOAP.loginUserId=<user ID>
com.ibm.SOAP.loginPassword=<password>
```

Use the user ID and password used to access the Administrative Console. If you want to encode the password in this file then run the PropFilePasswordEncoder script located in the folder PS_HOME/webserv/<profileName>/bin. For example,

```
PropFilePasswordEncoder.sh PS_HOME/webserv/<profileName>/properties⇒
/soap.client.props com.ibm.SOAP.loginPassword
```

Testing Administrative Console Security

Successfully starting the WebSphere Application Server and successfully logging into the Administrative Console verifies that Admin Security is enabled.

To test Administrative Console security:

1. Start the server.
2. Launch the Administrative Console.
3. When prompted to log in, submit the user ID and password you configured previously.
4. To ensure no PeopleSoft elements have been changed, sign on to PIA as well.

Configuring Application Security

This section describes how to configure Application Security for WebSphere. In the context of PeopleSoft, the PeopleSoft servlets (PORTAL, Report Repository, and so on) run in the PeopleSoft application. In this discussion, we secure the access to the Report Repository servlet in the PeopleSoft application as the example.

Modifying and Deploying the Delivered EAR file

In this procedure, you modify the delivered peoplesoft.ear file and deploy it using the PeopleSoft install program.

To modify and deploy peoplesoft.ear:

1. Locate the PeopleSoft ear file and move it to a temporary directory.

Copy the PeopleSoft application ear file (peoplesoft.ear) from PS_HOME\setup\PsmPPIAInstall\archives folder into a temporary directory and extract it. In this discussion, C:\temp is used.

2. Modify the application.xml file and add the <security_role> element as shown below.

a. Open C:\temp\META-INF\application.xml.

b. Add the security section shown below:

```
<application>
...
...
...
<module>
<web>
<web-uri> helloportletapp.war</web-uri>
<context-root /helloportletapp</context-root>
</web>
</module>
<security-role>
<description>Role for SchedulerTransfer Servlet</description>
<role-name>SchedulerTransferRole</role-name>
</security-role>
</application>
```

c. Save and close the file.

3. Modify the PORTAL web.xml file.

a. Extract C:\temp\PORTAL.war to C:\temp\PORTAL directory.

b. Open C:\temp\PORTAL\WEB-INF\web.xml.

c. Add the <security-constraint> and <security-role> element after the <welcome-file-list> element, and before the </web-app> element. In this case, the SchedulerTransferRole is mapped to the resource Report Repository servlet.

```
...
</welcome-file-list>
<security-constraint>
<web-resource-collection>
<web-resource-name>SchedulerTransferWebResource</web-resource-name>
<description>SchedulerTransferWebResourceDescription</description>
<url-pattern>/SchedulerTransfer/*</url-pattern>
<http-method>GET</http-method>
<http-method>POST</http-method>
</web-resource-collection>
<auth-constraint>
<description></description>
<role-name>SchedulerTransferRole</role-name>
</auth-constraint>
</security-constraint>
<security-role>
<description></description>
<role-name>SchedulerTransferRole</role-name>
</security-role>
</web-app>
```

d. Save and close the file.

4. Recreate PORTAL.WAR.

- a. Repackage PORTAL.war by running the following command

```
cd C:\temp\PORTAL
jar -cvf ..\PORTAL.war *
```

- b. Delete the C:\temp\PORTAL directory.

5. Repackage the peoplesoft.ear file by running the following command in the temporary directory.

```
jar -cvf ..\peoplesoft.ear *
```

6. Copy the recreated ear file into PS_HOME\setup\PsmPPIAInstall\archives.

7. Deploy the recreated ear file onto WebSphere using the PeopleSoft Internet Architecture installation program.

Create a New Security Domain For Application Security

Creating a separate security domain for applications enables you to assign tailored security attributes to the application.

To create a security domain:

1. Select Security, Security domains, and click New.
2. Provide a name and description for the security domain, and click OK.
3. Define the scope.

For this example, the scope is server1, but other implementations can set the scope at Cell, Clusters, Nodes, and so on.

4. Under Security Attributes, expand Application Security and select the Customize for this domain radio button, and select the Enable application security check box.
5. Expand User Realm, and select Customize for this domain, select a Realm type, and click Configure.

In this example, Local operating system is used, but any of the realm types can be used. The realm selected will need to contain the users who are authorized to access the resource, which in this case is the Report Repository servlet.

For this example, when configuring the realm, we use appRealm for the Provide a realm name field. In the Custom properties table we use com.ibm.websphere.registry.UserRegistry (Name) and local (Value).

Mapping Security Roles to User Groups

To map security roles to user groups:

1. Select Applications, Application Types, WebSphere enterprise applications, and click peoplesoft.
2. Click Security role to user/group mapping.
3. Select SchedulerTransferRole and click on Map Users.
4. Select the User realm as the one created for the application, in this case it is the "appRealm".

5. Click the Search button to display all the available user IDs for this realm.

For this example, we select the user ID `ansrivat12` for the `SchedulerTransferRole`. This ensures that only the user `ansrivat12` is authorized to access the Report Repository servlet.

Testing Application Security

Restart the web server. Test the security of Report Repository servlet (as we configured it in this example) by using the following URL:

`http(s)://<hostname>:<PIA http(s) port>/SchedulerTransfer/ps`

You should be prompted for user ID and Password dialog. If authentication is successful, you should be shown the servlet output.

Setting HTTP Session Timeout

HTTP session timeout controls are accessible on the Security page of the web profile interface. PeopleSoft Internet Architecture ignores any session timeout control set on the web server. At run time, the session timeouts set in the web profile override any HTTP session timeouts set at the web server level.

Setting Authentication Failure Timeout

To limit the effectiveness of DOS attacks on failed authentications, you can use the `psft_failtimeout` Java option. Add this option in the `setEnv` script and assign a value in seconds. By setting the value to 60 seconds, for example, you override the default session timeout of 120 seconds (two minutes) when a user authentication fails or when a user is not yet authenticated.

For example,

```
SET JAVA_OPTIONS_WIN32=-server -Xms32m -Xmx200m -Dpsft_failtimeout=60 -XX:MaxPermSize=128m -Xcomp
```

To determine the proper value for this property, you need to check the time in seconds that it takes to send an `http(s)` request from the browser to the web server and multiply the result by 2.

Working With JVM Heap Size

Adjusting the JVM heap size settings can improve performance in some situations, however, the default settings are typically adequate for most situations.

To access the JVM heap size properties:

1. In the Administrative Console, select Servers, Server Types, WebSphere application servers, and click on your server in the resource list.

2. Select the Configuration tab, and in the Server Infrastructure section expand Java and Process Management, and click Process definition.
3. In the Additional Properties, click Java Virtual Machine.

The JVM heap size properties are:

Property	Description
Initial heap size	The initial heap available to the JVM. If blank, the system assumes the default value of 50 MB.
Maximum heap size	The maximum heap available to the JVM. If blank, the system assumes the default value of 256 MB. IBM recommends that if you determine that garbage collection occurs more than desired, increase the Maximum heap size value.

See Also

http://publib.boulder.ibm.com/infocenter/wasinfo/v7r0//index.jsp?topic=/com.ibm.websphere.base.doc/info/aes/ae/xrun_transport.html

Working with Logging and Tracing Options

In addition to the logging and tracing options PeopleTools provides, WebSphere also offers a variety of tracing options.

See Also

Your IBM WebSphere Application Server and Administrative Console documentation

Enabling HTTP Access and HTTP Error Logging

To enable HTTP access and error logging:

1. In the Administrative Console, select Servers, Server Types, WebSphere application servers, server1, and click the NCSA access and HTTP error logging link.
2. Enable HTTP access logging by selecting the options Enable logging service at server start-up and Enable access logging.

The HTTP access logs will be written to PS_HOME/webserv/profileName/logs/server1/http_access.log.

3. Enable HTTP error logging by selecting Enable error logging.

The HTTP error logs will be written to PS_HOME/webserv/profileName/logs/server1/http_error.log.

Enabling General Logging and Tracing

To access WebSphere logging and tracing options:

1. In the Administrative Console, select Servers, Server Types, WebSphere application servers, and clicking on your server in the resource list.
2. In the Troubleshooting section, click Logging and tracing.

The diagnostic trace, JVM log, and process log output files are directed to a variable, `${SERVER_LOG_ROOT}`, which refers to the following location on a typical PeopleSoft installation:

PIA_HOME\webserv\profilename\logs\server

The IBM Service log output file is directed to a variable, `${LOG_ROOT}`, which refers to the following location on a typical PeopleSoft installation:

PIA_HOME\webserv\profilename\logs

Chapter 9

Configuring Search and Building Search Indexes

This chapter provides an overview of PeopleSoft search indexes and discusses how to:

- Configure PeopleSoft search.
- Work with indexes.
- Build record-based indexes.
- Build file system (spider) indexes.
- Build HTTP spider indexes.
- Administer search indexes.
- Modify the VdkVgwKey key.

Understanding PeopleSoft Search Indexes

This section provides an overview of search indexes and discusses:

- Types of indexes.
- Components of the search architecture.
- Index building.
- Search index limitations.
- User search strategies.

Overview of Search Indexes

A search index is a collection of files that is used during a search to quickly find documents of interest. The process of creating the search index is also called building the search index. The set of files that make up the index is a *collection*. This collection contains a list of words in the indexed documents, an internal documents table containing document field information, and logical pointers to the actual document files.

Fields contain metadata about a document. For example, Author and Title might be fields in an index. *VdkVgwKey* is a special field that identifies each document and is unique to all of the documents in the collection.

The document table is a relational table with one row for each document and columns of fields. Every index can be modified by defining a set of fields for it.

In PeopleSoft search implementations, every search index has a home location where all of the files pertaining to that index are located. This directory is the home directory of the index and is typically located at *PS_CFG_HOME/data/search/INDEXNAME*. Under this directory is another directory named for the database to which the application server or the Process Scheduler is connected. The actual collection files reside in this database directory.

Every search index can be modified by changing the configuration files that are associated with the index. These configuration files are known as *style* files and reside in the style directory. A typical configuration of style files define fields for a particular index.

Types of Indexes

PeopleSoft software supports three types of search indexes:

- Record-based indexes.
- HTTP spider indexes.
- File system indexes.

Record-Based Indexes

Record-based indexes are used to create indexes of data in PeopleSoft tables. For example, if the PeopleSoft application has a catalog record that has two fields (Description and PartID), you can create a record-based index to index the contents of the Description and PartID fields. Once the index is created, you can use the PeopleCode search application programming interface (API) to search this index.

HTTP Spider Indexes

HTTP spider indexes index a web repository by accessing the documents from a web server. You typically specify the starting uniform resource locator (URL). Then the indexer walks through all documents by following the document links and indexes the documents in that repository. You can control to what depth the indexer should traverse.

File System Indexes

File system indexes are similar to HTTP spider indexes, except that the repository that is indexed is a file system. You typically specify the path to the folder or directory. Then the indexer indexes all documents within that folder. HTTP spider indexes and file system indexes are sometimes collectively referred to as *spider* indexes. The indexer recognizes a wide variety of document formats, such as Word or Excel documents. Any document that is an unknown format will be skipped by the indexer.

Components of the Search Architecture

PeopleSoft search architecture uses two main technologies: that provided by the PeopleSoft Portal and that provided by Verity. They are connected by the PeopleSoft search API.

PeopleSoft Portal Technologies

The PeopleSoft Portal search technology contains the following components:

- Search input field.
Captures a query string that is entered by users in the portal header.
- Search API.
Passes the query string that is captured in the search input field to the Verity search engine.
- Portal Registry API.
Applies security to filter the search results.
- Portal registry.
Contains a repository of content references that can be searched.
- Search results page.
Formats and displays search results for the user.
- Search options.
Enables users to personalize search behavior and results.

Note. By default, the PeopleSoft search performs case-insensitive searches.

Verity Technologies

The basic items of the Verity architecture that are incorporated in the PeopleSoft Portal search architecture are:

- Verity collection.
This is the set of files forming a search index. When a user performs a search, the search is conducted against the Verity collection. You can create and maintain your own collections with the Search Design and Search Administration PeopleTools.
- BIF file.
This is an intermediate file that is created in the process of building a Verity collection. The BIF file is a text file that is used to specify the documents to be submitted to a collection. It contains a unique key, the document size (in bytes), field names and values, and the document location in the file system.
- XML file.
This is another intermediate file that is created in the process of building a Verity collection. The XML file is a text file named *indexname.xml* that contains all of the information from the documents that are searchable but not returned in the results list. This information is stored in zones. Zones are specific regions of a document to which searches can be limited.

- Style files.

These files describe a set of configuration options that are used to create the indexes that are associated with a collection.

- mkvdk.

This Verity command-line tool is used to:

- Index a collection.
- Insert new documents into a collection.
- Perform simple maintenance tasks, like purging and deleting a collection.
- Control indexing behavior and performance.

PeopleSoft Search Utilities

To create and administer search indexes for use with PeopleSoft software, use the PeopleTools utilities under PeopleTools, Search Engine. The utilities enable you to administer indexes and to create file system, spider, and record-based indexes.

Index Building

For both HTTP spider and file system indexes, options are available to include or exclude certain documents based on file types and Multipurpose Internet Mail Extensions (MIME) types. The index building procedure is different for record-based indexes and the spider indexes. Typically, the index building procedure is carried out from an Application Engine job that is scheduled by using the process scheduler.

The steps for building record-based indexes are:

1. The data from the application tables is read and two files called *indexname.xml* and *indexname.bif* are created.

indexname.xml contains one XML record for each document that needs to be indexed. The XML record contains all of the data that needs to be indexed. *indexname.bif* contains field information, the VdkVgwKey document, and offsets to denote the start and end of each document in the XML file.

2. The XML and the bulk insert file (BIF) files are typically generated through PeopleCode and reside in the home location of the index. The Verity utility, mkvdk, is called, passing in the BIF file as the argument to build the index.

The steps for building spider indexes are:

1. The Verity utility, vspider, is called.

The vspider utility takes a number of arguments, but the most important ones are the starting URL or directory to spider and the number of links to follow.

2. The vspider utility walks through all of the documents in the repository and builds the index.

Search Index Limitations

Following are the PeopleSoft search index limitations:

- Verity does not run on IBM z/OS.
- Verity collections must reside on the PeopleSoft application server or be accessible from it through a shared drive.

Satisfying this requirement can take several forms, depending on the application server's operating system. On Microsoft Windows, this could be a network drive. On UNIX, this could be an NFS-mounted drive.

- Verity collections are most efficient if you index large groups of data, rather than indexing one or two documents at a time.

Small updates degrade the index and require that you run the Verity cleanup utility.

- Style files are located in the style subdirectory of the index.

To make style changes, apply them to the files in this directory.

- You can have only one language per collection.

Additionally, a number of Verity search index features are limited to certain maximum values, as follows:

<i>Feature</i>	<i>Limitation</i>
Wildcards	Wildcard auto-expansion is limited to 16,000 matches.
Number of collections	The maximum number of physical collections that can be searched at one time is 128.
Documents per collection	The maximum number of documents allowed per collection is 16 million, subject to disk space availability.
Fields per collection	The maximum number of fields allowed per collection is 250.
Field length	The maximum length of any field is 32 kilobytes. Note. The actual number of characters that translates to depends on the character set being used.
Field value length in bulk files	The maximum length of a field value in a bulk file is 32 kilobytes. Note. The actual number of characters that translates to depends on the character set being used.
Zones per document	The number of zones allowed per document is unlimited.
Characters in path	The maximum path size allowed is 256 characters.

Feature	Limitation
Maximum documents with sort specification	The maximum number of documents that are returned when a sort specification is applied is 16,000.
Sort fields per search	The maximum number of fields that can be included in a sort specification is 16.

Refer to the Verity documentation for details about these features.

User Search Strategies

A user submits a search request by entering a search string into the search input form field in the portal header. The "<form action=...>" element in the portal header is generated at runtime to link to a PeopleSoft Internet Architecture page, and a Java script submits the form. The query string is passed to the Search API as a parameter named PortalSearchQuery to find matching results. Those results are filtered for security through PeopleCode by the Portal Registry API. The search results page echoes the original query string and displays a list of content references that match the request. If the user clicks the Go button but does not enter a search query, the search results page displays without any results.

The search results page performs the following steps:

- Changes the case of the entered text to all uppercase characters.

By default, the Verity search engine searches for all mixed-case variations when a query string is entered in all lowercase or in all uppercase. However, search queries that are entered in mixed-case automatically become case sensitive. (For example, a query on *Apple* behaves as if the user had specified *Apple*, which would find only the precise string *Apple*, while a query on *apple* finds *APPLE*, *Apple*, and *apple*.) But the portal makes one important change: It changes the case of the query string to all uppercase, prohibiting users from truly executing case-sensitive searches. This avoids situations where mixed-case searches would otherwise return no results. On the search results page, however, the original case is echoed back to the user.

- Formats the query string to pass to the Search API.

This includes filtering out expired and hidden content reference, and content references that are not valid yet.

- Calls the Search API.

This returns the query results.

- Calls the Portal Registry API.

This is done to apply security filtering to the results. Security is applied in PeopleCode by checking the Authorized property.

- Formats and displays search results.

This completes the user's search request.

Configuring PeopleSoft Search

This section contains an overview and discusses how to:

- Configure search to run natively within the application server (Type-1).
- Configure search to run as a separate process managed by the application server (Type-2).
- Configure a separate Search Server (Type-3).

Understanding PeopleSoft Search Configurations

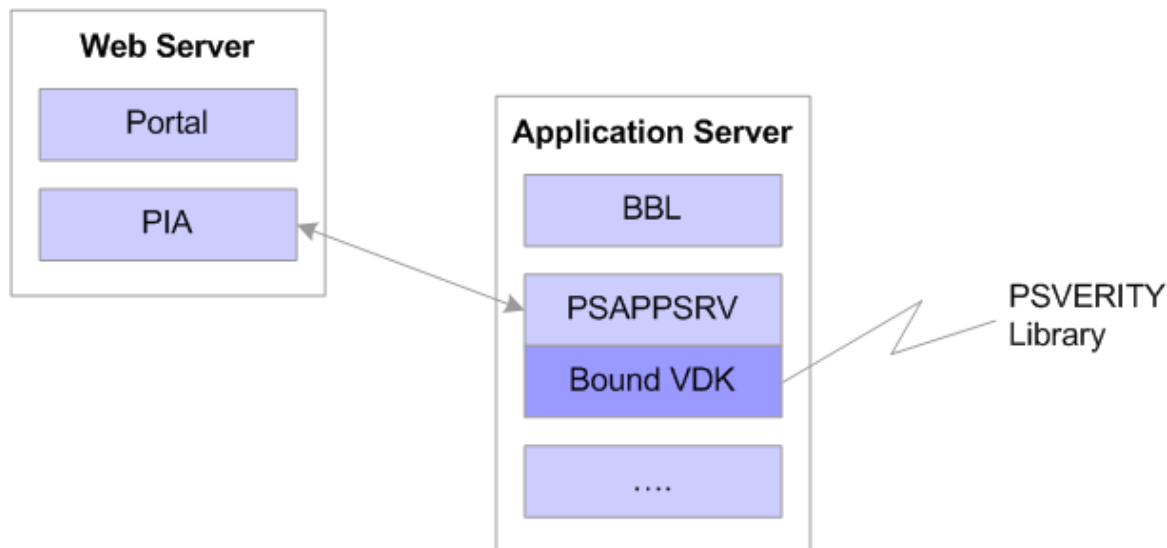
PeopleSoft offers these configuration options for enabling PeopleSoft search:

- Type-1: Verity running within the application server domain.
- Type-2: Verity running within a separate process managed by the application server.
- Type-3: Verity running within a separate search server.

Note. In some cases, the operating system determines which search configuration options can be used. *Always* refer to the *PeopleSoft Hardware and Software Requirements* guide, the Certifications area on Metalink, or customer support for the most recent support information.

Type-1: Verity Running within the Application Server Domain

In this configuration, Verity runs within the application server. Its libraries are linked to the application server. For example, the Verity VDK is bound to the PSAPPSRV server process. When a search request is submitted, the VDK bound to PSAPPSRV processes the request with the PSVERITY library.



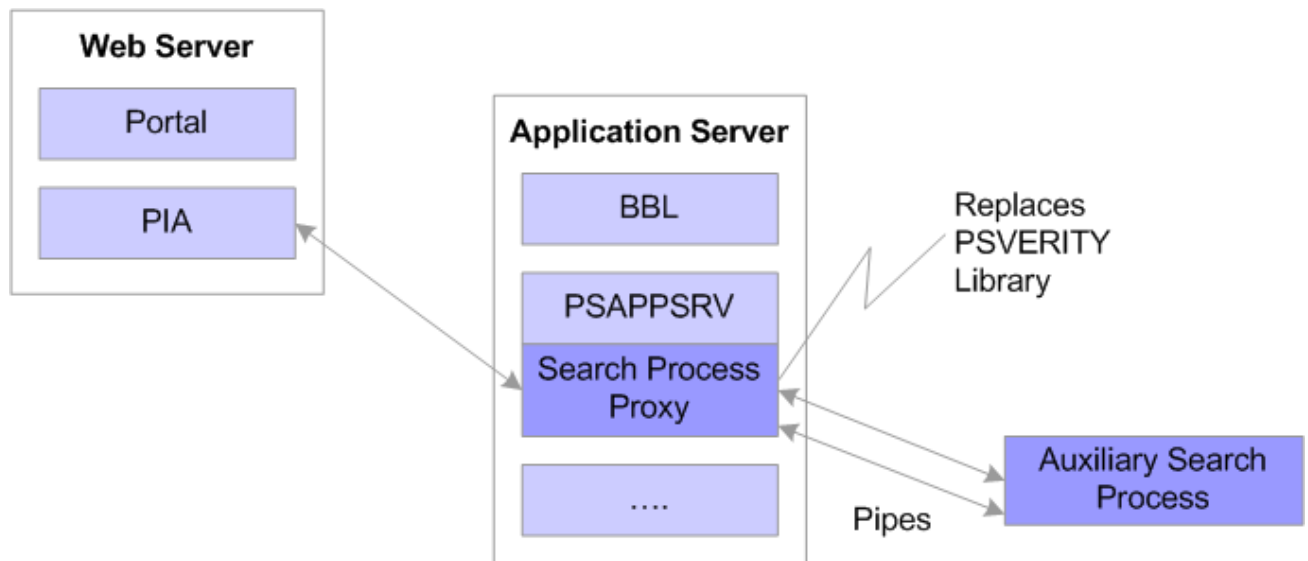
Type 1 search configuration: VDK bound to PSAPPSRV processes the request with the PSVERITY library

Note. This configuration has been used in PeopleSoft applications in all previous releases of the PeopleSoft Internet Architecture.

Type-2: Verity Running as a Separate Process Managed by the Application Server Domain

Having Verity run as a separately managed process enables application server domains running within the 64-bit framework to interoperate with Verity running within the 32-bit framework.

In this configuration, when the first search request is submitted, the PSAPPSRV server process spawns an auxiliary process to run within the application server domain. The spawned process hosts the VDK processing on behalf of the application server domain. A proxy search library within the application server routes search requests from the PeopleSoft Internet Architecture to the auxiliary search process.

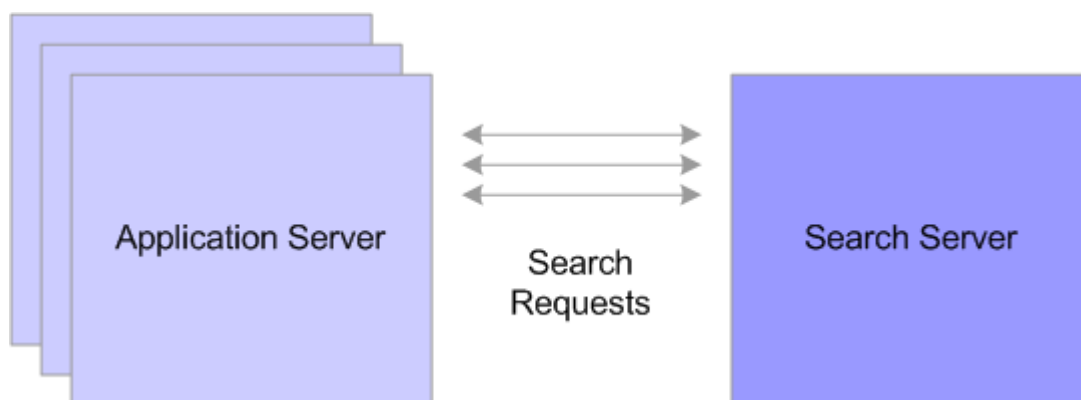


Type 2 search configuration: The spawned process hosts the VDK processing on behalf of the application server domain

The proxy search library and the auxiliary search process transmit data using efficient system resources (anonymous pipes). Having both processes running on the same computer can reduce performance degradation introduced from the extra communication layer of the network in a Type-3 configuration.

Type-3: Verity Running within a Separate Search Server

To centralize the configuration of the search features as well as the maintenance and storage of search indexes, you can implement the Type-3 search configuration.



Type 3 search configuration: application server domain sends search requests to the search server on a separate physical machine

In this configuration, the application server domain routes search requests to the search domain running on a remote search server. Multiple application server domains may use the same search server to execute search requests. In this scenario, the application server domain is the "client," submitting search requests and the search domain is the "server," processing requests and returning results.

Note. Tuxedo must be installed on both the application server machine and search server machine.

Configuring Search to run within the Application Server (Type-1)

This configuration requires the application server to be installed as outlined in the Enterprise PeopleTools Installation guide for your platform. This installation process installs the required application server and Verity software.

In the Search section of PSADMIN, enter 1 for the Deployment Type parameter.

```
Values for config section - Search
  Deployment Type=1
  Application Server Port=
  Remote Search Server Credentials=
```

Note. If you do not assign a value to the Deployment Type parameter, the system assumes the default configuration for your operating system.

Configuring Search to Run as a Separate Process (Type-2)

This configuration requires the application server to be installed as outlined in the Enterprise PeopleTools Installation guide for your platform. This installation process installs the required application server and Verity software.

In the Search section of PSADMIN, enter 2 for the Deployment Type parameter.

```
Values for config section - Search
  Deployment Type=2
  Application Server Port=
  Remote Search Server Credentials=
```

Note. If you do not assign a value to the Deployment Option parameter, the system assumes the default configuration for your operating system.

Configuring a Separate Search Server (Type-3)

Setting up a remote search server to process requests for application server domains requires you to complete configuration steps on the:

- search server.
- application server(s).

Configuring the Search Server Domain

To configure a separate search server:

1. Ensure the environment is set up correctly.

Configuring a search domain is comparable to creating an application server domain on an application server. You need to make sure:

- Tuxedo is installed locally.
- PS_HOME is available (locally or remotely)
- PS_CFG_HOME is set correctly on the search server machine.

2. Launch PSADMIN, and select Search Server from the PeopleSoft Server Administration menu.

```
-----  
PeopleSoft Server Administration  
-----
```

```
1) Application Server  
2) Process Scheduler  
3) Search Server  
4) Service Setup  
q) Quit
```

```
Command to execute (1-4, q): 3
```

3. On the PeopleSoft Search Server Administration menu select 2) *Create a domain*, and enter a name for the search domain.

```
-----
PeopleSoft Search Server Administration
-----
```

```
1) Administer a domain
2) Create a domain
3) Delete a domain
q) Quit
```

Command to execute (1-3, q) : 2

Please enter name of domain to create :SAMPLE

4. Select 1) *search*, for a configuration template.

Configuration templates:

```
1) search
```

Select config template number: 1

5. When prompted to configure the search domain and change any configuration values, enter y to indicate "yes."
6. In the [Startup] section, add the information required for the search domain to connect to the application database.

The values entered should be identical to the connect information in any application server domain connecting to the same database.

Note. The search domain must connect to the same database as the application servers sending requests to the search domain.

7. In the [Database Options] section, select the same options you use for other application server domains in your environment.
8. In the [Domain Settings] section, select the same options you use for other application server domains in your environment, including, for example, Add To Path for specifying database driver locations.

Note. Make note of the unique Domain ID value. It is required when configuring the application server domains using the search server.

9. Modify the options in the [PSSRCHSRV] section.

Min Instances, Max Instances, Service Timeout	<p>These parameters operate the same as PSAPPSRV.</p> <p>See Chapter 6, "Setting Application Server Domain Parameters," PSAPPSRV Options, page 97.</p>
Search Server Port	<p>Enter the port address the search domain will monitor for search requests.</p> <p>The default is 7778.</p>
Application Server Credentials	<p>Enter a list of application server domains that will be using the search domain. The application servers need to be identified by Domain ID, Server ID, and port in the following format.</p> <p><Domain ID> <Server ID>:<port></p> <hr/> <p>Note. Server ID can be an IP Address or a hostname.</p> <hr/> <p>When multiple domains use the same search server, separate the entries by a comma (.). For example, the following illustrates how to enter two different domains running on two different servers.</p> <p>APPDOM1 appsrv_computer1:7777,APPDOM2 appsrv_computer2:7777</p> <hr/> <p>Note. The Domain ID value can be found in the [Domain Settings] section of PSADMIN.</p> <hr/>

Configuring an Application Server Domain to use a remote Search Server

Once you have a remote search domain configured, you then need to modify each application server domain that will use that search server to process search requests.

To configure an application server domain to use a remote search server:

1. Launch PSADMIN, and initiate the configuration interface for the desired application server domain.
2. Modify the [Search] section.

Deployment Type	Enter 3 to indicate Type-3 configuration.
Application Server Port	Enter the port number on which the application server domain will "listen" for responses from the search domain. Make sure this value is the same port number you specified in the search domain in the Application Server Credentials parameter.
Remote Search Server Credentials	<p>Specify the search server domain that will be used by the application server domain. The search server needs to be identified by Domain ID, Server ID, and port in the following format.</p> <p><Domain ID> <Server ID>:<port></p>

3. When prompted to configure Domains Gateway (External Search Server) indicate y for "yes."

Note. The Domains Gateway can also be enabled in the Quick Configure menu.

Setting Up Failover Search Domains

To provide high availability and to compensate for the possibility of issues with the network, a server machine, a search domain, or simply having to shut down a search domain for maintenance, you can configure failover search domains. In these situations, the unavailability of a primary search domain does not affect end users.

Failover search domains only process search requests when the primary search domain is unavailable. If the primary search domain is unavailable, the system seamlessly routes search requests to the next search domain specified in the failover string sequence.

For example, assume an application server domain has the following search domains specified in the search domain failover string in this order:

- SRCH_PRIMARY
- SRCH_FAILOVER1
- SRCH_FAILOVER2

If SRCH_PRIMARY is unavailable, the system checks to see if SRCH_FAILOVER1 is available, and if so, begins routing search requests to SRCH_FAILOVER1. If SRCH_FAILOVER1 is not available, then the system checks the availability of SRCH_FAILOVER2, and so on. When the primary search domain becomes available again, the system begins routing search requests to that search domain.

To set up failover search domains:

1. Install and configure the number of failover search domains you require.

It is recommended that each failover search domain reside on a separate server machine for optimal failover coverage.

2. For each failover search server domain, specify the complete list of application server domains that could potentially use the search domain for failover coverage.

Use PSADMIN, or edit the PSSRCHSRV.CFG manually. Specify the application server domains using the Application Server Credentials parameter in the PSSRCHSRV section.

3. For each application server domain using a particular set of search server domains, modify the Remote Search Server Credentials parameter in the [Search] section to include the connect information for each search domain, with the primary search domain appearing first and a comma (,) separating multiple values.

```
<Domain ID>|<Server ID>:<port>,<Domain ID>|<Server ID>:<port>
```

For example,

```
Remote Search Server Credentials=SRCH_PRIMARY|ts-sun04:7778,SRCH_FAILOVER1|ts->
sun05:7778,SRCH_FAILOVER2|ts-sun06:7778
```

Search Server Administration

While the administrative tasks associated with search servers are similar to your application server or Process Scheduler administration, keep the following items in mind when managing search servers.

Working with Search Domains in PSADMIN

When administering search server domains, you use a subset of PSADMIN menu options.

```
-----
PeopleSoft Search Domain Administration
-----
```

```
Domain Name: search01
```

- 1) Boot this domain
- 2) Domain shutdown menu
- 3) Domain status menu
- 4) Configure this domain
- 5) TUXEDO command line (tadmin)
- 6) Edit configuration/log files menu
- 7) Clean IPC resources of this domain
- 8) Domain Gateway TUXEDO command line (dadmin)
- q) Quit

Command to execute (1-8, q) :

Using these menus is similar to the menus for an application server domain, except that items that are not applicable do not appear. For example, there are no menu options for purging cache, preloading cache, or setting up messaging servers because they do not apply in the context of search servers.

See [Chapter 5, "Using PSADMIN Menus," page 57.](#)

For search servers, the following options differ slightly from application server domain options:

Boot this domain	For application server domains, you have options to boot a domain in serial or parallel mode. Because the number of server processes within a search domain are typically fewer than a large domain, the option of a parallel boot to save time is unnecessary. With search domains, you are not presented with boot options, and the domain boots in serial mode.
Domain Gateway TUXEDO command line (dadmin)	The dadmin is similar to the tadmin interface. dadmin is an interactive command interpreter used for the administration of domain gateway groups defined for a particular Tuxedo application.

Locating Logging Information in Type-3 Search Configurations

The system writes logging information to these files:

- TUXLOG for both the application server and search server domain.
- APPSRV_MMDD.LOG for the application server domain.
- SRCHSRV_MMDD.LOG for the search server domain.

Monitoring Domain Gateway Connections

The domain gateway is a subcomponent of a Tuxedo domain that allows it to communicate with another domain through the network. The domain gateway ensures that the application server and search server domains are successfully connected and able to transmit data. An application server domain and search domain can start independent of one another and do not report any obvious signs of being successfully connected when they start.

When working with search domains and troubleshooting Domain Gateway issues:

- Ensure that the domain gateway is enabled. Check the Tuxedo logs of both the application server and the search server. Both logs should indicate the gateway connection.
- Check machine and port configuration. Failure to connect, or connections with numerous disconnections can be caused by incorrect port and machine address information or another machine using the same port. Use canonical names if you are using a non-numerical IP address.

Use the `dmadmin` command line interface to monitor a Domain Gateway connection between a local and remote domain. Access this interface through the PeopleSoft Search Domain Administration menu in PSADMIN. The following commands can be helpful when working with search domains.

Command	Description
<code>pd -d <LOCAL DOMAIN ACCESSPOINT_ID></code>	<p>Use the <code>pd</code> (print domain) command to confirm whether or not the application server and search server domains are connecting and transmitting data. Confirm successful connection by viewing the 'Connected domains' list.</p> <p>The <code><LOCAL DOMAIN ACCESSPOINT_ID></code> is formed by prepending "SS" to the domain ID. For example, if the domain ID is <code>SRCHSERV</code> the value of <code><LOCAL DOMAIN ACCESSPOINT_ID></code> is <code>SS_SRCHSERV</code>.</p> <p>The following is sample output:</p> <pre>pd -d SS_SRCHSERV Local domain :SS_SRCHSERV Connected domains: Domainid: APPDOM1</pre> <p>If the search domain is not connected to the application server domain you will see output similar to this:</p> <pre>pd -d SS_SRCHSERV Local domain :SS_SRCHSERV Connected domains: Disconnected domains being⇒ retried: Domainid: APPDOM1</pre> <p>This examples show only one application server domain and one search domain. In reality, multiple application server domains would connect to one search domain. The <code>pd</code> command lists the status of each of the application server domains connected to a search domain.</p>

Command	Description
<code>pstats -d <LOCAL DOMAIN ACCESSPOINT_ID></code>	Use the <code>pstats</code> command to extract monitoring statistics from the Tuxedo MIB regarding the domain gateway connection. This can help to identify the amount of requests being processed for application server domain clients.
<code>h</code>	Displays and describes all <code>dmadmin</code> commands.

See Oracle Tuxedo documentation for complete `dmadmin` documentation.

Building Search Indices

For a search server (Type-3 configuration), a Process Scheduler deployed on the search machine should be used for indexing. Because Verity libraries may be available only on the search machine, and because any index would be used by the search server on the search machine, it is recommended to build the indices on the search machine to avoid having to relocate indexes from other machines. A recommended approach is to deploy a Process Scheduler server along side the search server and specify that Process Scheduler server for generating indexes (PeopleTools, Search Engine, Administration, Schedule).

Note. For building search indexes on a Type-3 configuration, it is strongly recommended to use the PSNT Process Scheduler Server running on the same server machine as the Type-3 configuration.

If Verity is not supported on the operating system where your production application server domains run, another option is deploying an application server domain along side the search server. This application server would be accessible through its own web server instance possibly on a different port than the production application server. This provides access to an application server with Verity support, allowing the creation of indices interactively. Also, the indices would be created where the search server can locate them.

Working with Indexes

This section provides overviews of common controls and supported MIME types, and discusses how to:

- Open existing collections.
- Create new collections.

Understanding Common Controls

The following controls appear on the pages that are used for designing record-based, file system, or HTTP spider indexes.

Index	Shows the name of the index that you opened or the name that you gave the index on the Add New Value page.
Build Index	Invokes the collection build program. Before clicking this button, select all of the appropriate options for the collection.

Test Index	After building an index, click to test that the build program assembled the index properly. The Test Index page contains a single text field with a query button. Enter text to search for in the collection and click the [?] button to submit the query. The results return a list of the keys that are stored by Verity in the collection.
Show Logs	View the log files that are produced by the collection build program during execution. This is used mainly for troubleshooting.
Append to Verity Command Line	This control is for PeopleSoft internal use only.

Understanding Supported MIME Types

The following list contains the supported document MIME types. Any document that is not one of these types is ignored during the indexing process.

- application/msword
- application/wordperfect5.1
- application/x-ms-excel
- application/x-ms-powerpoint
- application/x-ms-works
- application/postscript
- application/rtf
- application/x-lotus-amipro
- application/x-lotus-123
- application/x-ms-wordpc
- application/x-corel-wordperfect
- application/x-wordprocessor
- application/x-spreadsheet
- application/x-presentation
- application/x-graphics
- application/x-keyview
- application/x-ms-write
- application/pdf
- application/x-executable
- message/rfc822

- message/news
- text/html
- text/sgml
- text/xml
- text/ascii
- text/enriched
- text/richtext
- text/tab-separated-values
- text/plain
- text/x-empty
- image/gif
- application/x-verity

Opening Existing Collections

To open an existing collection:

1. Select PeopleTools, Search Engine.
2. From the available menus, select the type of collection that you want to open, as in record-based indexes, file system indexes, or HTTP spider indexes.
3. On the Find an Existing Value tab, use the Search for drop-down list box to select the appropriate criteria (begins with or contains).
4. In the edit box to the right, enter the character string that reflects the appropriate begins with or contains criteria.
5. Click Search.

Creating New Collections

To create a new collection:

1. Select PeopleTools, Search Engine.
2. From the available menus, select the type of collection that you want to create, as in record-based indexes, file system indexes, or HTTP spider indexes.
3. Select the Add a New Value page.
4. Enter a name for the collection.
5. Click Add.

6. Specify the appropriate attributes for the collection as described in the following sections.
7. Save your work.

Note. You cannot create indexes of the same name even if they are of different types; for example, record, HTTP, or file.

8. Build the index.

Building Record-Based Indexes

The record-based index extracts data from database tables and inserts the data into BIF and XML files, which are then indexed by Verity. The individual creating the index chooses the records (tables) to be indexed.

Note. The record-based index supports only data that is stored in PeopleSoft databases.

This section discusses how to:

- Modify record-based index properties.
- Add subrecords to search indexes.

Modifying Record-Based Index Properties

Select PeopleTools, Search Engine, Record-Based Indexes to access the Design a Search Index page.

Primary Record

Subrecords

Design a Search Index

Index: TEST

Build Index

Test Index

Show Logs

Index Location: NEW

Key returned in search results:

<pairs/>

Edit Key

Parent Data Record

*Record (Table) Name:

WHERE clause to append:

WHERE

Fields

*How to zone the index:

One zone

[Click here for help with the Fields columns](#)

Fields Included in the Index

	Record	Field Name	Verity Field	Word Index	Has attachment
1			<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

Append to Verity Command Line:

Design a Search Index page (Record-Based)

Parent Data Record

- Record (Table Name)

Enter tables, views, or a PeopleSoft view that contains data. To combine the data from multiple PeopleSoft tables, to create a view on those tables and specify the name of that view here.
- WHERE clause to append

Fine-tune the data that you receive by entering a Structured Query Language (SQL) WHERE clause.

Key returned in search results Use to synthesize the VdkVgwKey, which supports an XML-like syntax enabling you to modify the tag that is returned by Verity.

You have the following options:

- `<pairs/>`: Inserts a string of NAME=VALUE;. One such pair is returned for each key of the record.
- `<row/>`: Inserts the record keys in a SQL-like syntax.
- `<field fieldname='MYFIELD'/>`: Inserts the value of MYFIELD if it exists in the record.
- `<sql stmt='SQL STATEMENT'/>`: Inserts the value that is returned by the SQL statement. The system accepts only the first row that is returned, and PeopleSoft software does not support SQL statements returning more than one column.

Edit Key Click to access the page where you can change the results that are returned by the Key returned in search results functionality.

Fields

How to Zone the Index *One Zone:* Select to put all of the data into one zone. With this option, the collection builds more quickly but the application can't restrict searches to the portions of the index that come from a particular field.

Field Zones: Select to create one zone for each PeopleSoft field on the record. Applications can specify that they want to access that particular zone in their searches.

Field Name After you specify a record name, the fields in that record appear in this grid. Select the following options for each field in the record: Verity Field, Word Index, or Has Attachment (each option is explained in the following sections).

Verity Field Select if the PeopleSoft field should be indexed as a Verity field. In general, PeopleSoft fields that contain a lot of descriptive text, such as description fields, should be indexed as word indexes (See the following definition) and PeopleSoft fields that contain metadata about what is being indexed (such as ProductID) should be indexed as Verity fields.

Word Index Select if this PeopleSoft field should be indexed as a word index. See the preceding Verity Field definition for guidelines on defining a PeopleSoft field as a Verity field versus defining it as a word index.

Has attachment

Enables you to index attachments that are referenced in the field as uniform resource identifiers (URIs). Refer to the PeopleCode Developer's Guide for a description of file attachments. If this field contains the URL to an attachment, select this check box. The indexer downloads the attachment and indexes it as part of the document. This item is enabled only if the corresponding PeopleSoft field contains character data, because numeric fields cannot contain URLs.

To use this field, you need a record that is designed with this feature in mind. In the record, each row has a text field that contains a URI or an empty string.

The text must be a valid File Transfer Protocol (FTP) URI (including the login and password string) of the following form:

- ftp://user:pass@host/path/to/filename.doc.
- A valid record URI of the form record://RECORDNAME/path/to/file.doc.
- A string of the form <urlid name="A_URLID"/>/path/to/file.doc.

The third form references an entry in the URL table (Utilities, Administration, URLs). If the URL ID that is named in the name attribute is valid, the entire URI is rewritten with the part in brackets replaced by the actual URI.

For example, if A_URLID is equal to ftp://anonymous:user@resumes.peoplesoft.com, the entire string in the previous example becomes

ftp://anonymous:user@resumes.peoplesoft.com/path/to/file.doc and is treated like any other FTP URI.

Rows of data with empty strings in the URI field are ignored with no error.

If the string is one of these three valid URI forms and a document can be retrieved at that URI, the document is indexed with the same key as the rest of the row of data and is searchable.

To add subrecords to the index, select the Subrecords tab, and insert the child records that you want to include in the index.

Adding Subrecords to Search Indexes

Select PeopleTools, Search Engine, Record-Based Indexes, Subrecords.

To index more than one record as a single document, the records must be hierarchically related. For example, the record that is specified on the previous page must be a parent of all the others. Formally, this means that the keys of each subrecord named must be a superset of the keys of the parent record. The parent record is the one that you specify in the Record (Table Name) field on the Primary Record page.

To add subrecords to an index:

1. Create and save the index definition.
2. Select PeopleTools, Search Engine, Record-Based Indexes, Subrecords.

- Click the Add a new row button to insert the names of the records that are children of the parent record that is defined on the Primary Record page.

On the Primary Record page, the fields of the child record are added to the Fields grid. When you build the index, data from the child records whose keys match the row in the parent record is included as part of the parent record. When an end user searches for data that is found in the child record, the system returns a reference (VdkVgwKey) for the parent record.

Building File System (Spider) Indexes

You can index file systems that are local to the application server. This refers to any file system on the physical server on which your application server domain runs, and it also refers to any drives that are accessible from the application server machine. File systems might include file servers, report repositories, and so on.

The index is compiled by using vspider. The program descends into the directory structure recursively and indexes the file types that you've selected to be indexed. It indexes only files that Verity supports for collections.

This section discusses how to:

- Set file system options
- Define what to index

Setting File System Options

Select PeopleTools, Search Engine, Filesystem Indexes to access the Design a Search Index page.

Filesystem Options

What to Index

Design a Search Index

Index: TEST



Build Index

Test Index

Show Logs

Index Location: NEW

Filesystem paths

Customize | Find | View All |  

First 1 of 1 Last

List local filesystem paths to spider	Remap Path to This URL		
<input type="text"/>	<input type="text"/>	<div>+ -</div>	

Append to Verity Command Line:

Design a Search Index page (File system)

List local filesystem paths to spider	<p>Specify the network file system path that contains the documents to index. Ensure that the local application server has the proper access to the file systems that you include in the list.</p> <p>For Microsoft Windows, this means the drive mappings must be set up from the applications server. For UNIX, this means the correct network file system (NFS) mappings must be set on the application server.</p> <p>To add a system path to the list, click the plus button. To remove a file system, click the minus button.</p>
Remap Path to This URL	<p>Do not use.</p>

Defining What to Index

Select PeopleTools, Search Engine, Filesystem Indexes, What to Index to access the What to Index page.

Filesystem Options

What to Index

Design a Search Index

Index: TEST

Build Index

Test Index

Index Location: NEW

Show Logs

Mime Types

☒ Index all Mime-types

☐ Index only these Mime-types

☐ Exclude these Mime-types

Mime/Types Allowed:

Filenames

☒ Index all filenames

☐ Index only these filenames

☐ Exclude these filenames

Pathname Globs List:

What to Index page

MIME Types

Index all Mime-types	Select to index all MIME types on a website.
Index only these Mime-types	Select to index only a certain MIME type, and specify the file type in the MIME/Types Allowed list box. Separate multiple MIME types with a space.

Exclude these Mime-types	Select to exclude a set of MIME types, and specify the MIME types to exclude. Separate multiple MIME types with a space.
MIME/Types Allowed	Add a list of MIME types, separated by spaces, if you selected Index only these Mime-types or Exclude these Mime-types.

Filenames

Index all filenames	Select to index all file types.
Index only these filenames	Select to index only a certain file type, and specify the file type in the Pathname Globs List list box.
Exclude these filenames	Select to exclude a set of file types, such as temporary files, but to index all others. Also specify the file types to exclude.
Pathname Globs List	Add the files that you want to incorporate into your index. Separate the entries with spaces. You can use wildcard characters (*) to denote a string and "?" to denote a single character. For example, the string '*.doc 19??.excel' means select all files that end with the ".doc" suffix and Microsoft Excel files that start with 19, followed by 2 characters.

Building HTTP Spider Indexes

HTTP spider indexes are similar to the indexes that the spider functionality compiles for the file system index. When using the spider index on a website, vspider starts at the home page of the site and then follows each link on that page to the next level of the site. For each page at the next level, vspider follows each link on each page. After following a link, vspider indexes all of the data on the target page.

You can specify as many websites as you want, and you can configure the depth, or number of layers of links, that vspider follows into a website and index.

This section discusses how to:

- Define HTTP gateway settings.
- Define what to index.

Defining HTTP Gateway Settings

Select PeopleTools, Search Engine, HTTP Spider Indexes to access the HTTP Gateway page.

HTTP Gateway

What to Index

Design a Search Index

Index: TEST

Build Index


Test Index

Show Logs

Index Location: NEW

Depth of Links to Follow: 1

HTTP URLs


Customize | Find | View All | 


First 1 of 1 Last

List http:// URLs to spider

Stay in Domain

Stay in Host





Proxy Hostname:

Proxy Port: 8080

Append to Verity Command Line:

HTTP Gateway page

Depth of Links to Follow Set the level of detail that you want to index within a certain site. If you enter *1*, vspider starts at the homepage and follows each link on that page and indexes all of the data on the target pages. Then it stops. If you enter *2*, vspider follows the links on the previous pages and indexes one more level into the website.

As you increase the number, the number of links that vspider follows increases geometrically. Do not set this value too high, because it can impact performance negatively. You should not need to set this value higher than 10.

List http://URLs to spider

Click the plus button to add multiple URLs to spider. Click the minus button to remove a URL from the list. If you forget to include the *http://* (scheme) portion of the URL, the system automatically includes it.

URLs should contain only the alphanumeric characters as specified in RFC 1738. Any special character must be encoded. For example, encode a space character as *%20*, and encode a *<* as *%3c*. Additional examples are available.

See <http://www.w3.org/Addressing/rfc1738.txt>.

Stay in Domain

Select to limit spidering to a single domain. For example, suppose that you are spidering *www.peoplesoft.com* and you select this option. If a link points to a site outside the PeopleSoft domain (as in *yahoo.com*), the collection ignores the link.

Stay in Host

Select to further limit spidering within a single server. If you select this option, the collection contains references to content only on the current web server or host. Links to content on other web servers within the domain are ignored. For example, if you are spidering *www.peoplesoft.com* and you select this option, you can index documents on *www.peoplesoft.com*, but not on *www1.peoplesoft.com*.

Proxy Hostname and Proxy Port	Enter a host and port for vspider to use. Enter the same settings that you would use in your web browser if you need a proxy to access the internet.
--------------------------------------	--

Defining What to Index

Select PeopleTools, Search Engine, HTTP Spider Indexes, What to Index. The fields on this page are documented in a previous section.

See [Chapter 9, "Configuring Search and Building Search Indexes," Defining What to Index, page 200.](#)

Administering Search Indexes

After you design and build your search indexes, the Search Administration interface enables you to schedule when and how frequently the indexes must be rebuilt. An important aspect of maintaining the collections involves scheduling PeopleSoft Process Scheduler jobs that, on a regular basis, rebuild the collection completely or incrementally update the index. Search index administration also includes deleting old indexes and building indexes to support additional languages.

This section discusses how to:

- Specify the index location.
- Administer the search index.
- Edit properties.
- Schedule administration.
- Share indexes between application servers and PeopleSoft Process Scheduler.

Specifying the Index Location

By default, search index files are located in:

PS_CFG_HOME/data/search/indexname/db_name/language_code

You can store indexes in different locations, but you need to specify the custom location in the CFG file for an application server, Process Scheduler, or search server. Use the [Search Indexes] section in the PSAPPSRV.CFG, PSPRCS.CFG, or PSSRCHSRV.CFG files to specify alternate search index locations and multiple locations, if necessary.

Note. This procedure assumes that you've already used the Search Index Designer to define, build, and store the search indexes that you will specify in the CFG file.

Note. You must edit the CFG file manually to include the locations. You do not add search index locations through PSADMIN.

To add a search index location:

1. Open the CFG file.
2. Locate the Search Indexes configuration section.

For example:

```
[Search Indexes]
;=====
; Search index settings
;=====
: Search indexes can be given alternate locations if there is an entry here.
; Entries look like: IndexName=fs location (ie EMPLOYEE=c:\temp)
```

3. Add an entry for each search index location that you want to specify by using the following syntax:

index_name=location

For example, to specify the location for search INDEX_A and INDEX_B, your entries would look similar to the following:

```
[Search Indexes]
;=====
; Search index settings
;=====
: Search indexes can be given alternate locations if there is an entry here.
; Entries look like: IndexName=fs location (ie EMPLOYEE=c:\temp)
INDEX_A=c:\temp
INDEX_B=n:\search
```

Note. Make sure that your entries are not commented out with a semicolon (;) appearing before them.

Note. For the Process Scheduler configuration file, PSPRCS.CFG, include the same location as specified in the application server configuration file.

Note. When specifying the index to be generated in a custom location, the directory structure the system builds within the custom location will be slightly different from that built in the default location. The directory structure within custom index locations will not have a directory for the database name.

4. Save the CFG file.

Administering the Search Index

Select PeopleTools, Search Engine, Administration to access the Search Index Admin page.

Search Index Admin

Delete, Modify and Schedule Builds for Indexes

Defined Search Indexes				
	Index	Index Location	Edit Properties	Schedule
<input type="checkbox"/>	1 TEST	C:\Documents and Settings\admin\psft\pt8.50-901-R1\data\search\TEST\QEDMO	Edit Properties	Schedule

[Process Monitor](#)

Deleting the Index Definition also removes the actual collections stored in the filesystem, if any have been built.

Delete checked Indexes

Search Index Admin page

Index	Displays the name of the index so that you can identify specific indexes. To select an index, select the check box to the left of the index name.
Index Location	Displays the current location of the index.
Edit Properties	Click to access the interface for changing the index location and to build indexes to support additional languages.
Schedule	Click to access the interface for scheduling the program that maintains your collection.
Delete checked Indexes	If you have selected indexes to be deleted, click this button to remove them from the system. The deletion process deletes the index definition and the collections that are stored in the file system.

Note. If you attempt to delete a scheduled index, you may see SQL errors on IBM DB2 UDB or Sybase database platforms.

Editing Properties

Select PeopleTools, Search Engine, Administration, Edit Properties.

Index Location	Displays the current location of the index.
Language Code	Select the language for which you want to build an index.
Language to Map	Currently disabled.
Build	After you add the additional indexes, click to create the indexes.

Note. Style files are located in the style subdirectory of the index. To make style changes, apply them to the files in this directory.

Scheduling Administration

Select PeopleTools, Search Engine, Administration, Schedule.

The screenshot shows the 'TEST' window in the PeopleSoft Process Scheduler. At the top, there is a link 'Add a new Recurrence Definition'. Below it is a table with columns: *Type of Build, *Run Recurrence Name, and Server Name. The table has one row with the following values: 1, Rebuild (selected from a dropdown), Daily Search Rebuild, and PSNT. There are search icons and navigation buttons (First, 1 of 1, Last) at the top right of the table.

Scheduling search builds

Add a new Recurrence Definition

In PeopleSoft Process Scheduler, you define run recurrence definitions that enable you to schedule jobs to run at regular intervals, such as monthly, weekly, daily, and so on. The more current you keep the collections, the more accurate your search results will be.

Type of Build

Rebuild: Select to drop the existing collection and rebuild a new collection. This applies to all types of collections.

Increment: Use only for the spider indexes. For record-based indexes, only the *Rebuild* option is available.

Run Recurrence Name

Select the appropriate run recurrence definition for the collection maintenance requirements.

Server Name

Specify the PeopleSoft Process Scheduler server on which you want the build program to run. The PeopleSoft Process Scheduler system must be installed and configured before you can schedule the collection build program to run as a job.

Sharing Indexes Between Application Servers and PeopleSoft Process Scheduler

The index files reside on a file system at the home location and must be accessible to all application servers and process schedulers that will manipulate the index. An application server uses the index for searching while the process scheduler invokes an Application Engine program that builds the indexes. Therefore, if you are running a process scheduler on a different machine than the application server, ensure that the index files are accessible to both. You can do this three ways:

- Make a Microsoft Windows shared drive or NFS file system available for the index.

Specify the index location in both the application server and the process scheduler to point to the shared directory.

- Run an instance of the process scheduler on the application server host and schedule only the building of indexes on this process scheduler.

Because the process scheduler and the application servers are running on the same host, they create and read files from the same location.

- Use an external program such as FTP or Secure Copy (SCP) to copy all of the files and directories in the index home location from the process scheduler host (after the index has been built) to the application server host so that they are available for searching.

Modifying the VdkVgwKey Key

To make the VdkVgwKey more readable and easier to parse, use the following XML-like syntax:

```
<field fieldname='MYFIELD' />
<row/>
<pairs/>
<sql stmt="SELECT 'Y' FROM PS_INSTALLATION"/>
```

- Fieldname and the SQL statement support single and double quotes, as well as no quotes at all (in which case only the first word is considered part of the option).

Using double quotes for the SQL statement is recommended.

- The SQL statement must return only one column.

Multiple rows are ignored. Trying to return more than one column results in a collection-build-time error.

- Currently, the only tag style that is supported is <tag/> with the slash (/) at the end.
- The VdkVgwKey can include any amount of literal text interspersed with the tags.

This text is copied into the VdkVgwKey that goes into the BIF file, unmodified.

- Field names are automatically set in uppercase.

Chapter 10

Using PeopleSoft Configuration Manager

This chapter provides an overview of PeopleSoft Configuration Manager and discusses how to:

- Start PeopleSoft Configuration Manager.
- Specify startup settings.
- Specify display settings.
- Specify Crystal report and JDeveloper settings.
- Specify trace settings.
- Specify workflow settings.
- Specify remote call/AE settings.
- Configure developer workstations.
- Import and export environment settings.
- Configure user profiles.
- Specify command line options.
- Set up the PeopleTools development environment.

Note. PeopleSoft supports a number of versions of UNIX and Linux in addition to Microsoft Windows. Throughout this chapter, we make reference to operating system configuration requirements. Where necessary, this chapter refers to specific operating systems by name (Solaris, HP/UX, Linux, and so forth). However, for simplicity the word UNIX refers to all UNIX-like operating systems, including Linux.

Understanding PeopleSoft Configuration Manager

PeopleSoft Configuration Manager simplifies Windows workstation administration by enabling you to adjust PeopleSoft registry settings from one central location. It contains a variety of controls that let you set up Windows workstations. You can set up one workstation to reflect the environment at your site, and then export the configuration file, which can be shared among all the workstations at your site. You can also define separate profiles for connecting to different PeopleSoft databases. PeopleSoft configuration parameters are grouped on the Configuration Manager tabs according to the function, feature, or tool that they control.

Note. The changes you make within PeopleSoft Configuration Manager do not take effect until the next time a user signs on to PeopleSoft.

See Also

Chapter 10, "Using PeopleSoft Configuration Manager," Setting Up the PeopleTools Development Environment, page 228

Common Elements in PeopleSoft Configuration Manager

OK	Saves your settings and exits PeopleSoft Configuration Manager.
Cancel	Closes PeopleSoft Configuration Manager without saving any changes that you have made.
Apply	Saves your changes without exiting.

Starting PeopleSoft Configuration Manager

You can start PeopleSoft Configuration Manager by one of two methods:

- Double-click the Configuration Manager shortcut in your PeopleSoft program group.
- At a command prompt, enter:

```
PS_HOME\bin\client\winx86\pscfg.exe
```

Important! Due to changes in the registry structure beginning with PeopleTools 8.50, if you intend to run PeopleTools 8.50 and later versions of Configuration Manager and previous versions of Configuration Manager (PeopleTools 8.49, 8.48, and so on), on the same development workstation, pre-PeopleTools 8.50 values stored in the registry will be deleted.

Important! Certain PeopleSoft utilities require setting an environment variable, PS_SERVER_CFG, to run properly. However, PeopleSoft Configuration Manager isn't compatible with PS_SERVER_CFG being set. Before you start Configuration Manager, you must ensure that PS_SERVER_CFG is not set. You can make this convenient by using a DOS batch file to unset PS_SERVER_CFG, launch Configuration Manager, then after Configuration Manager exits, reset PS_SERVER_CFG to its previous value.

Specifying Startup Settings

Select the Startup tab.

Use the Startup tab to customize the default values that appear on the signon screen.

Signon Defaults

Database Type	<p>Select the database type to appear as a default on the PeopleSoft Signon dialog box. Select <i>Application Server</i> to sign in to an application server instead of a database. To enable users to change their database type selection in the signon dialog box, you must select the Database Type option in the User Can Override group.</p> <hr/> <p>Note. When you select <i>Application Server</i> from the Database Type drop-down list, the Server Name and Database Name fields are disabled. The system obtains these values from the application server.</p> <hr/>
Application Server Name	<p>If you selected <i>Application Server</i> from the Database Type drop-down list, specify the application server's name in this field. You must have already configured your application server and registered it on the Profile tab.</p>
Server Name	<p>Enter the name of the default database server. This parameter is only enabled for Informix, Sybase, and Microsoft SQL Server, and refers to the instance to which the user connects.</p> <p>For Informix, enter the server name in lowercase.</p>
Database Name	<p>Enter a default database name. You can choose any valid PeopleSoft database name. As with the database type, you must select the appropriate option in the User Can Override group to enable users to override the default database name selection when they sign in.</p>
User ID	<p>Specify the default user ID to sign in to PeopleSoft.</p> <p>You can use the user ID in conjunction with a PSUSER module containing a user-defined sign-in process. The PSUSER code, if present, can evaluate and modify the user ID value before you attempt to sign in to the selected database.</p>
Connect ID and Connect Password	<p>PeopleSoft uses the connect ID for the initial connection to the database. Use the Connect Password field to define a default connect ID password.</p> <hr/> <p>Note. The connect ID edit box must contain a value, or the user can't sign in to the system in a two-tier environment.</p> <hr/>

See *Enterprise PeopleTools 8.50 PeopleBook: Security Administration*, "Understanding PeopleSoft Security."

See *PeopleTools Installation Guide for Your Database Platform*.

Numeric Keypad - Enter Key Tabs to Next Field

In Microsoft Windows applications, pressing the Enter key in a dialog box selects the default action button. For example, in the PeopleSoft Signon dialog box, pressing Enter is the same as clicking the OK button. Selecting the Numeric keypad check box overrides this default behavior for the Enter key on the numeric keypad; instead of selecting the action button, pressing the Enter key moves the cursor to the next field in the dialog box.

Note. This check box affects the Enter key on the numeric keypad, but not the Enter key on the main keyboard.

User Can Override

Some PeopleSoft sites use multiple database types and names. Using the check boxes in the User Can Override group box, you can enable users to enter a database type, database name, or user ID other than the default provided at sign-in. In most cases, you use these controls to prevent users from attempting to sign in onto any database other than the default.

Database Type	When selected, users can choose a database other than the default. Selecting this check box selects the Database Name and User ID options automatically. You cannot clear Database Name or User ID without first clearing Database Type. When configuring a workstation to connect in both two-tier and three-tier, you must select this box. The user needs to specify a two-tier or three-tier connection from the PeopleSoft Signon dialog box.
Database Name	When selected, the User ID check box is automatically selected, although you can clear it. To clear Database Name, you must clear the Database Type check box.
User ID	Select to enable users to users override only the user ID submitted at when they sign in. You cannot clear User ID if Database Type is selected.

Cache Files

Enter the parent directory that holds your cache file directories. For example, enter *C:\PS\CACHE*.

Note. Cache files store database object information locally and are automatically downloaded the first time you open a PeopleSoft database object. They are also downloaded automatically if the master copy of the object on the database server is changed. One cache file directory stores the cache files for each PeopleSoft database that you use.

Clicking Purge Cache Directories brings up a dialog box with your existing cache file directories.

You can select a single directory and click Delete, or you can click Delete All to remove all directories. If a cache file directory is missing (after you delete it), the system automatically rebuilds it the next time that cache files are downloaded. After you delete the appropriate cache directory, click Close to return to the Startup tab.

Specifying Display Settings

Select the Display tab.

Use the Display tab to configure the appearance of the PeopleSoft graphical user interface. For instance, you can adjust page width and height to fit in with the other elements on your desktop.

Language

In the Language drop-down list box, specify which language you want to display on your PeopleSoft pages. The default setting is US English.

Note. You select from the languages that PeopleSoft delivers. Although you can implement applications to appear in other languages, you cannot switch to custom languages using PeopleSoft Configuration Manager. Switch to these languages by manually changing the registry setting.

Page Display

You can adjust page display size or the page height and width.

Display Size, Width, and Height Specify display size in pixels. This setting affects the default size of the PeopleSoft window as displayed in the corresponding Width and Height fields. Select from:

- *640 X 480*: The default window size is 640 pixels by 448 pixels.
- *800 X 600*: The default window size is 800 pixels by 576 pixels.
- *1024 X 768*: The default window size is 1024 pixels by 744 pixels.
- *Custom*: You can manually set the default window size by specifying width and height values.

Note. Changing these parameters does not affect open windows. If either value is either blank or zero, the values are reset to 640 by 480 pixel resolution.

Page Sizing

Use this field to specify how pages that were designed for a different-size window should be displayed. Select from:

- *CLIP*: Page controls are always displayed in their normal size. If a page is too large for the window, the page information is clipped along the right and bottom edges of the window. Use scroll bars to view the remainder of the page.
- *SCALE*: Pages are scaled to fit the window as necessary. For example, if your display size is set to 640 by 480 pixels, and you open a page designed to display in an 800 by 600 pixel window, the page controls are scaled down so that all page information appears. Conversely, if you open a page designed for 640 by 480 pixel resolution in a larger window, the page controls are scaled to fill the window completely.

Show Page in Navigator Select to see the navigator tree view and the page view at the same time.

Highlight Popup Menu Fields Select to highlight fields with associated pop-up menus. The box is clear by default. In most cases, it's a good idea to indicate which fields contain pop-up menus. Pop-up menus are indicated by a black rectangle surrounding the perimeter of a page control.

Show Database Name Select to display the name of current database in the status bar at the bottom of a PeopleSoft page, in addition to the current page name and the activity. For example, the status bar might read PTDMO, Job Data 1, Add. This feature is useful if you are running multiple instances of PeopleTools.

Note. The database name may be abbreviated to fit on the screen.

Font

Use the Font options to configure the way that text appears on the screen in PeopleSoft applications.

Click the Font button to bring up a standard font selection pop-up menu, as shown in the following example:

Business Process Display

Select from:

- *On:* The navigator appears with each menu group that you open.
- *Off:* You must open the navigator manually.
- *First:* The navigator appears on the first instance of PeopleSoft only. Subsequent instances do not display the navigator.

Specifying Crystal Report, Business Interlink, and JDeveloper Settings

Select the Crystal/Bus.Interlink/JDeveloper tab.

Crystal Options

If you have Crystal Reports installed on a workstation, the Crystal executables path is populated automatically. If Crystal Reports is installed on a network drive, use this field to reflect the location of the Crystal Reports executables. For example, you might enter *n:\hr900\bin\client\winx86\crystal*.

Use the Default Crystal Reports field to specify the default location of reports. If this setting does not apply to your site's Crystal Reports implementation, leave this field blank.

When you select Use Trace during execution, Crystal Reports writes the trace statements to a log file that you specify in the Trace File field.

Business Interlink Driver Options

In the Business Interlink Directory box, enter the complete path to the directory that contains the drivers that PeopleSoft Business Interlinks uses to communicate with external systems.

Note. PeopleSoft Business Interlink is a deprecated product. These options currently exist for upgrade compatibility and transition.

JDeveloper Home Directory

Oracle JDeveloper is used when developing transformations using Oracle XSL Mapper. You need to specify the high-level installation directory as well as the appropriate Classpath string. These settings are discussed in detail in the Integration Broker documentation.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Crystal Reports for PeopleSoft, "Using Crystal Reports 2008," Configuring Crystal Reports 2008

Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Integration Broker, "Applying Filtering, Transformation and Translation," Developing Transforms Using Oracle XSL Mapper

Specifying Trace Settings

Select the Trace tab.

Use the Trace tab to select tracing options for various parts of the PeopleTools system, such as SQL statements, PeopleCode, and Application Engine. If you work on tuning your PeopleSoft system and improving online performance, familiarize yourself with this tab. When you update the Trace tab, the new settings take effect the next time you launch PeopleTools.

Note. The Trace tab in PeopleSoft Configuration Manager traces only Microsoft Windows client (two-tier) interactions. Use these settings only when you require tracing on the client.

You can override some of the trace options on this tab from the Trace SQL and Trace PeopleCode pages in PIA.

See [Chapter 12, "Tracing, Logging, and Debugging," Configuring SQL Trace, page 290](#) and [Chapter 12, "Tracing, Logging, and Debugging," Configuring PeopleCode Trace, page 288](#).

SQL Informational Trace

Select this check box to trace information messages from the Runstats command in DB2 UDB for z/OS executed as a result of an %UpdateStats meta-SQL command.

PeopleTools Trace File

The default filename for the PeopleTools trace file is DBG1.TMP. The system writes the file to the following directories:

- In Microsoft Windows: %TEMP% directory
- In UNIX: \$PS_HOME/log/dbname

Important! The PeopleTools trace file stores elapsed times for PeopleCode and SQL events to a precision of one microsecond (six decimal places). However, due to limitations of the operating system, Windows precision is actually in milliseconds (three decimal places), so the last three digits in a Windows trace will always be zero. Elapsed times in UNIX are accurate to one microsecond.

To specify a different PeopleTools trace file:

1. Click the button on the right side of the PeopleTools Trace File edit box.
A standard Open dialog box appears.
2. Navigate to and select the new trace file.
3. Click Open.

The PeopleTools Trace File field displays the path and filename.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Application Engine, "Tracing Application Engine Programs,"
Setting Options in PeopleSoft Configuration Manager

Specifying Workflow Settings

Select the Workflow tab.

Use the Workflow tab to specify the options and locations related to the PeopleSoft Workflow implementation at your site.

Maximum Worklist Instances	Enter a number to limit the number of worklist instances or entries that appear when viewing worklists. The default value is 250. If you do not want any rows returned, leave the field blank.
SMTP Server	Specify the SMTP settings for email routings.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Workflow Technology, "Defining Worklist Records"

Specifying Remote Call/AE Settings

Select the Remote Call/AE tab.

Some PeopleSoft applications use the Tuxedo Remote Call feature, which invokes data-intensive transactions, such as COBOL processes, on a remote server.

Timeout	Enter the amount of time after which Remote Call terminates the child COBOL process. The default is 50 seconds.
Redirect Output	Select to specify whether the standard out or standard error of the child COBOL process is directed to a file. This check box is clear by default.
Support COBOL Animation	Select to save the COBOL input file so that you can reuse it with COBOL animator. This check box is clear by default.
Normal, Minimized, and Hidden	Specify how the window state of the child COBOL process appears on the desktop. <ul style="list-style-type: none"> • Select Normal to have the window state appear like a DOS window on the desktop. • Select Minimized to have the window state appear as an icon on the task bar. • Select Hidden to have the window state run unseen in the background.
Disable DB Stats (disable database statistics)	Select to turn off the %UpdateStats meta SQL construct. This setting applies to Application Engine programs. See <i>Enterprise PeopleTools 8.50 PeopleBook: Application Engine</i> , "Using Meta-SQL and PeopleCode."

Configuring Developer Workstations

Select the Client Setup tab.

As part of the PeopleSoft installation process, you need to configure developer workstations (also called the PeopleTools development environment) to run successfully with your PeopleSoft system. You use developer workstations for development and administrative tasks that require the use of a Windows workstation. Such tasks include, updating record definitions with Application Designer and running scripts using Data Mover. Development workstations can access the PeopleSoft system using both two-tier and three-tier connections.

Use the Client Setup tab to configure developer workstations and invoke the Client Setup process. Although this tab is specifically for developer settings, all of the PeopleSoft Configuration Manager settings may affect developers, especially the Startup tab and the Process Scheduler tab for the default profile.

See *PeopleTools Hardware and Software Requirements*

Shortcut Links

Here are the various shortcut links:

Application Designer	Adds a shortcut for the main PeopleTools development environment.
-----------------------------	---

Configuration Manager	Adds a shortcut for PeopleSoft Configuration Manager, which enables you to edit registry settings relevant to PeopleSoft.
Data Mover	Adds a shortcut to launch PeopleSoft Data Mover.
Uninstall Workstation	Adds a shortcut for Uninstall Workstation, which uninstalls the most recent client setup.
PeopleTools RPT Converter	Adds a shortcut to a standalone program that converts RPT files from the format PeopleSoft used in previous releases to the currently supported Crystal format. You only need to run this program if you are upgrading from previous versions of PeopleTools.
nVISION	Adds a menu item for PS/nVision to the PeopleSoft 8 menu group in the Microsoft Windows Start menu.

Note. Back up RPT files before you run the converter program, which significantly alters them.

Install Workstation

Select the Install Workstation check box to run the Client Setup process. Only select the check box after specifying all the appropriate selections on all PeopleSoft Configuration Manager tabs. If you do not select this box, the Client Setup process will not run.

After you select this check box, click either OK or Apply.

Enable Dirty-Read for 2Tier Query

(Applies only to DB2 systems.) Select this option if you need to run "dirty read" queries, through PeopleSoft Query during a two-tier connection.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query*, "Creating and Running Simple Queries."

See Also

Chapter 10, "Using PeopleSoft Configuration Manager," Setting Up the PeopleTools Development Environment, page 228

Importing and Exporting Environment Settings

Select the Import/Export tab.

Use this tab to export, or save to file, the specified environment settings, and to import previously exported settings. This feature is useful when you plan to configure multiple workstations with similar settings.

Export to a File

Click to write current configuration settings to a file. A Save dialog box appears. Note the file name that you give the configuration file.

Note. Click Apply before you export a file. This ensures that the exported configuration file reflects the current settings.

Import from a File

Click to import previously saved configurations on another workstation. Importing a configuration file overrides all the current environment settings on the machine that you import to.

When you click this button, an Open dialog box appears. Navigate to the directory containing the appropriate configuration file, select the file, and click Open.

Warning! In addition to overwriting environment settings, this function also overwrites all existing settings made in Application Designer.

Configuring User Profiles

This section discusses how to:

- Define a profile.
 - Specify databases and application servers.
 - Configure process scheduler.
 - Configure nVision.
 - Specify common settings.
-

Note. The term "user profiles" is used here to refer to user configurations for a workstation. These profiles are not to be confused with PeopleSoft security user profiles.

Defining a Profile

Select the Profile tab.

Use this tab to define one or more user profiles, each of which specifies connection parameters and file location information for a PeopleSoft installation.

Many PeopleSoft installations include multiple databases. For example, there may be one database for tracking financial information, such as expense reports, and another database for human resources processes, such as benefits enrollment. Each of these databases has its own set of supporting files, SQR reports, COBOL processes, and so on. PeopleTools locates these files by referring to the Microsoft Windows registry. By defining multiple profiles, you can tell PeopleTools to use different directory paths depending on the database.

When you first open PeopleSoft Configuration Manager, the Profile tab displays a single profile named Default. To set the parameters for this profile, make sure that it's selected, and click the Edit button. The Edit Profile dialog box appears.

Each workstation must have a default profile, which is used when the user signs in to a database or application server that isn't listed in any profile. If the workstation requires only one set of profile settings, you can use the default profile. You can also set up multiple PeopleSoft Configuration Manager profiles. The profiles are set for Microsoft Windows workstations and are specific to each user and not shared by all workstation users.

Note. You can use profiles to easily switch between applications.

Specifying Databases and Application Servers

From the Edit Profile dialog box, select the Database/Application Servers tab.

Use this tab to specify the configured databases and application servers associated with this profile. When a user enters one of these databases or application servers in the PeopleSoft Signon dialog box, PeopleTools uses the registry settings associated with this profile.

Note. You can assign multiple databases and application servers to a single profile. However, each database and application server must be assigned to only one profile. If you try to add a database to a second profile, PeopleSoft Configuration Manager asks you if you want to remove it from the previous profile and add it to the current one.

Note. Before you enter a database or application server on this tab, you should have already installed and configured it as documented in the PeopleSoft installation documentation for your database platform.

Application Server Name

Enter a name for an application server that you have configured. This name will appear in the drop-down list box on the PeopleSoft Signon dialog box. Choose a name that's intuitive for your site.

Note. Application server names cannot exceed 24 characters.

Machine Name or IP Address

Enter the IP address or the resolvable server name of the application server you specified in the Application Server Name field. You specified the IP address in the [Workstation Listener] section of your PSAPPSRV.CFG file when you installed your PeopleSoft application server. For example, you could enter 207.135.65.20 or sp-hp32.

Port Number

Enter the port number for the application server that you specified in the Application Server Name field. You specified the port number when you installed and configured the application server using PSADMIN. A port number is an arbitrary number between 0 and 9999 that is determined by site specifications.

TUXEDO Connect String

Use this field to support dynamic load balancing. You can specify a free-form connect string that allows a client to connect to another application server in case another is either down or being used to full capacity.

Note. The Tuxedo connect string cannot exceed 1000 characters.

When configuring load balancing, you might choose from the following approaches:

- Round robin load balancing.

With this approach, you specify multiple application servers, and each client picks a server randomly. This approach assumes that application server will receive an equal number of connections. To specify round robin load balancing, use the following syntax for the connect string:

(//IP address 1:port 1//IP address 2:port 2//Ip address n:port n)

You can specify the IP address using either dotted notation or by using the server's DNS name. Either way, the slashes (//) preceding the IP address are required.

If the selected application server is unavailable, your connection attempt fails, and the system does not try to connect you to the other application servers defined within the parentheses.

Spaces are not allowed in any part of the connection string. The system automatically removes embedded spaces before storing the value in the registry.

- Round robin with failover.

With this approach, you define a failover connection string. Use the following syntax:

(//IP address 1:port 1//IP address 2:port 2),(//IP address 3: port 3)

If the application server selected from the first group of parentheses (IP addresses 1 and 2) is unavailable, the system automatically attempts to connect to an application server defined in the second group (IP address 3). If that application server fails, the system attempts to connect to the next group to the right, sequentially.

If multiple application servers are defined within any group, the system round-robins between them. If the selected application server fails, the system attempts to connect to the next application server to the right, if any. The following list shows three examples of connect strings that use this approach:

- *(//sp-ibm01:8000//sp-ibm02:8000),(//sp-nt01:8000)*
- *(//208.136.78.88:8000//208.136.78.88:8050//208.136.78.88:8080)*
- *(//sp-sun01:8000),(//sp-sun02:8000),(//sp-sun03:8000)*

Set and Delete Buttons

When you click Set, your application server information is displayed in the grid at the top of the dialog box. You can enter a new application server name and set up a different server if you like.

Note. The settings in the grid are not saved until you click Apply or OK. If you click Cancel without first clicking Apply or OK, you lose all the information in the grid.

To remove an application server configuration, select its application server name in the grid and click Delete.

Configuring Process Scheduler

Access the Process Scheduler tab.

Use this tab to specify the directories that are associated with PeopleSoft Process Scheduler jobs, such as SQR and COBOL directories.

General

PeopleSoft Home Directory	Enter your high-level PeopleSoft directory, such as <i>N:\HR910</i> .
Crystal Reports	Enter the file path to \CRWRTPATH, where Crystal Reports sends your reports.
Output Directory	(Optional) Enter the directory used with the Output Destination field when scheduling a PeopleSoft Process Scheduler request.
Log Directory	Enter the directory for SQR, COBOL, and PeopleSoft Process Scheduler log files.
Temp Directory	Enter the path to your temporary directory, for example, <i>C:\TEMP</i> . This directory stores log files and other output files.
Database Drivers	Enter the path to the directory where your database drivers reside.
Word Executables Directory	Enter the directory containing Microsoft Word executables; for example, <i>N:\Apps\Office200x\Office</i> .
Redirect Output	Select to redirect onscreen COBOL Display statements to a log file. (If this check box is clear, you see the onscreen messages only.) Sending the messages to a log file is useful for debugging purposes. The log file is created in the %TEMP%\PS_HOME\DBNAME directory. In addition to the output generated by COBOL Display statements, the log file contains errors generated by the COBOL runtime system.

Note. To use the Application Engine debug feature, clear Redirect Output.

Application Engine

Debug

Select to enable the Application Engine command-line debugger.

Warning! Select the Debug check box only when you are testing and troubleshooting client-side processes. If you select Debug and submit a process request to the server, the process hangs, waiting for a user command.

Disable Restart

Select to disable the Application Engine restart feature, which lets you restart an abnormally terminated Application Engine program. When selected, Application Engine programs start from the beginning. This option is useful during debugging. Do not select it in a production environment.

SQR

SQR Executables

Enter the path to the directory where SQR executables reside.

SQR Flags

Enter the SQR parameters that PeopleSoft Process Scheduler should pass on the command line to the SQR executables. The following SQR flags are required for launching SQR reports:

- -i: Specifies the path to SQC files.
- -m: Specifies the path to the ALLMAXES.MAX file.
- -f: Specifies the output path.
- -o: Directs log messages to the specified file.
- -ZIF: Sets full path to the and name of the SQR initialization file, SQR.INI.

SQR Report Search 1, SQR Report Search 2, SQR Report Search 3, and SQR Report Search 4

Enter the directory paths that the SQR executable should use to locate SQR reports. SQR Report Search 1 is searched first, followed by SQR Report Search 2, and so on.

COBOL

COBOL Executables

Enter the path to \CBLBIN, where COBOL executables reside.

Configuring nVision

Access the nVision tab.

Use this tab to specify where PS/nVision should look for files and how it should operate. PeopleSoft Query Link, the feature that enables you to send PeopleSoft Query output to a spreadsheet, also uses these settings.

Space between Query Columns

This parameter sets the number of blank Microsoft Excel characters that PeopleSoft Query Link places between query output columns. To eliminate column spacing, set Space between Query Columns to zero.

Directory Paths

Specify the locations of directories associated with PS/nVision jobs.

Customization Macros	Enter the directory containing macros for PS/nVision and PeopleSoft Query Link. It is usually <i>PS_HOME</i> \excel.
Report Layouts	Enter the location of PS/nVision layout fields.
Drilldown Layouts	Enter the location of PS/nVision drilldown files, for example, <i>c:\user\nvision\layout\drilldn</i> .
Report Instance	Enter the directory in which PS/nVision places report instances; for example, <i>c:\user\nvision\instance</i> .
Query Templates	Enter the directory to look for the QUERY.XLT file, which defines the Microsoft Excel styles used to format output. The default is <i>PS_HOME</i> \excel.
Style Sheets	Enter the directory where the NVSUSER style wizard locates nPlosion style sheets.

Trace Level

Indicate whether you want PS/nVision to generate independent trace log files of two-tier activity, and at what level, for each nVision process. Select one of the following values:

- 0: Disable tracing. This is the default value.
- 1: Generate basic high level information.

This setting can be used to check whether nVision has successfully launched and is able to connect to Excel and process the request. Some of the key entries in a level 1 trace log are:

- Command Line Arguments.
- Trace Level.
- Excel Pid.
- Run Control Name.
- Report Id.
- Business Unit.
- Drill Layout.
- Report Id.
- Instance Name.
- 2: Generate level 1 tracing plus high level code flow.
- 3: Generate level 2 tracing plus runtime SQL statements.
- 4: Generate level 3 tracing plus most function calls and output values.

Use this setting to identify problems that are intermittent and hard to predict.

The trace log files are generated in the c:\temp directory, named with the format `psnvs_process_id.nvt`, for example, `psnvs_1024.nvt`. You can view these log files in a text editor.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Process Scheduler*, "Using Process Monitor," Viewing Process Detail Actions.

Note. Extensive tracing will affect PS/nVision performance. Two-tier log files aren't automatically purged by PS/nVision. Users must manually delete them from the temp directory to save disk space.

See Also

Enterprise PeopleTools 8.50 PeopleBook: PS/nVision

Specifying Common Settings

Select the Common tab.

Sybase Packet Size

Specify a TCP packet size. The minimum value is 512 and the maximum value is 65538. The default packet size is 512. If you change the packet size, make sure to make the corresponding changes to the Sybase server. See the material on Sybase administration and tuning on the PeopleSoft Customer Connection website, as well as your Sybase documentation.

See Your Sybase reference manuals.

Application Designer Image Conversion

When you upgrade to newer version of PeopleTools, you'll need to convert images to a new format, which may require more storage space. If the images exceed the record size limit of your platform, you can shrink the images to conform to this limit.

Convert and Shrink Images to Platform Limit	Select to convert and shrink images to fit your selected database platform limit, as shown in the Image Size Limit field.
Convert and Shrink Images to Image Size Limit	If you are upgrading to a different database platform, select this option and specify the correct value in the Image Size Limit field.
Don't Convert, but Shrink Images to Image Size Limit	Select for images that have already been converted, but need to be converted so they meet the platform size limits.

Data Mover Directories

You can control several PeopleSoft Data Mover environment variables through PeopleSoft Configuration Manager.

Input Directory	Enter the directory where PeopleSoft Data Mover should search for its input data files. If no path is specified for SET INPUT in the Data Mover script, Data Mover searches directories for the database file in the following order. <ol style="list-style-type: none"> 1. Specified output directory. 2. C:\TEMP.
Output Directory	Enter the directory where PeopleSoft Data Mover scripts and .DAT files will be created. Data Mover must have write access to the location. The default is <i>PS_CFG_HOME\data</i> .

Log Directory

Enter the location of PeopleSoft Data Mover log files. This location must allow Data Mover write access in the case of a read-only PS_HOME configuration.

The default is C:\Documents and Settings\admin\Local Settings\Temp.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Data Management, "Using PeopleSoft Data Mover"

Specifying Command Line Options

In addition to its GUI interface, PeopleSoft Configuration Manager offers command line options. Syntax for PeopleSoft Configuration Manager command line options is as follows:

`pscfg -command`

For example:

`pscfg -import:n:\config\hr900.cfg`

Import File

To import configuration settings from a named file, enter `-import: filename`.

Export File

To export the current configuration settings, enter `-export: filename`.

Run Client Setup

To run the Client Setup process, enter `-setup`.

Note. You must use the `-setup` command in conjunction with the `-import` command if you are setting up a new workstation.

Run Client Setup Without Displaying Messages

To run the Client Setup process without displaying messages or dialog boxes, enter `-quiet`.

Note. Output messages are written to a log file called `%temp%\PSINSTAL.LOG`.

Install MSS DSN

To install MSS DSN, enter `-dsn`.

Note. For Microsoft SQL Server, the server name value is used to automatically create your ODBC data source name.

Uninstall Workstation

To clear the PeopleSoft settings from the registry or uninstall the PeopleSoft workstation, enter -clean.

The -clean command removes the following items from the workstation:

- PeopleSoft registry settings.
- All cache files from the current \CACHE directory.
- Shortcut links.
- PeopleSoft program group.

Make sure that removing all of these items is acceptable before issuing the -clean command.

Help

To view PeopleSoft Configuration Manager command-line options online, enter -help or a question mark (?).

Setting Up the PeopleTools Development Environment

This section provides overviews of the PeopleTools development environment and the client setup process and discusses how to:

- Verify *PS_HOME* access.
- Verify connectivity.
- Verify supporting applications.
- Use the Configuration Manager pages.
- Run the Client Setup process.

Understanding the PeopleTools Development Environment

Most user workstations are equipped with supported web browsers, but with no special PeopleSoft software installed. The traditional Microsoft Windows client is supported for application developer and administrator use. The PeopleTools development environment runs on a supported version of Windows.

This chapter describes how to configure these Windows-based clients using PeopleSoft Configuration Manager. As before, such clients can connect to the PeopleSoft database directly using client connectivity software (a two-tier connection), or through a PeopleSoft application server (a three-tier connection).

Understanding the Client Setup Process

Before running the Client Setup process, create all the profiles you need.

The Client Setup process installs a PeopleSoft program group on the workstation.

If the Install Workstation check box on the Client Setup tab is selected, these Client Setup functions are performed when you click OK or Apply from PeopleSoft Configuration Manager.

See [Chapter 10, "Using PeopleSoft Configuration Manager," Configuring Developer Workstations, page 217.](#)

Note. Any files installed by the Client Setup process on the workstation from the file server use the paths specified in the default profile.

Verifying PS_HOME Access

To use the PeopleTools development environment, each workstation must have access to the file server *PS_HOME* directory (the high-level directory where PeopleSoft executables were installed) and have a drive mapped to the directory. Workstation users must have read access to the *PS_HOME* directory.

Verifying Connectivity

Database connectivity is required on all Microsoft Windows-based clients that make two-tier connections to the database. A two-tier connection is required if any of the following is true:

- You sign in to the Application Designer in two-tier.
- You run PeopleSoft Data Mover scripts.
- You run COBOL and SQR batch processes on the client.

Verify Supporting Applications

Supporting applications must be installed on any Microsoft Windows-based client on which batch processes are run locally.

SQR

On Microsoft Windows-based clients, you can install SQR locally, or you can map to a copy installed on the file server. Because SQR does not require local registry settings, you can execute SQR from any Windows-based client once SQR has been installed to a shared directory. Installing SQR locally results in improved performance; over a slow network connection, the improvement is significant.

Crystal Reports

Optionally install Crystal Reports on Microsoft Windows-based two-tier clients. As with SQR, you can install Crystal Reports locally, or you can map to a copy installed on the file server. Because Crystal Reports does not require local registry settings, you can run Crystal Reports from any two-tier client once it has been installed to a shared directory. Installing Crystal Reports locally results in improved performance; over a slow network connection, the improvement is significant.

Crystal Reports requires that you install the PeopleSoft ODBC driver on the workstation where Crystal Reports processes run.

Microsoft Office

Install Microsoft Office on any two-tier client that runs PS/nVision or Microsoft Word batch processes. Microsoft Office must be installed locally, because it requires registry settings.

Using the Configuration Manager

PeopleSoft Configuration Manager enables you to manage registry settings on a Windows workstation that control a variety of interface options, such as signon defaults, display settings, and environment profiles.

See Also

Chapter 10, "Using PeopleSoft Configuration Manager," page 209

Running the Client Setup Process

To run the Client Setup process:

1. In Configuration Manager, select the Client Setup tab.
2. In the Group Title text box, enter the name of the program group for the icons you want on the client workstation.
3. Select check boxes to create shortcut links for PeopleSoft applications that you want to access from the workstation.

When you run the Client Setup process, it removes existing shortcuts in the PeopleSoft 8 program group and installs shortcuts for the applications that you have selected. If you later want to install or uninstall shortcuts, you can always run Client Setup again.

4. Select the Install Workstation check box.

Client Setup runs when you click Apply or OK in PeopleSoft Configuration Manager. If this check box is not selected, the Client Setup process creates or updates settings in the registry, but it doesn't set up the PeopleSoft 8 program group or install local DLLs.

5. Click Apply to run the Client Setup process and apply other PeopleSoft Configuration Manager settings.
6. To view a list of the files installed and actions taken by the Client Setup process, open the psinstal.log file in your Temp directory.

See Also

Chapter 10, "Using PeopleSoft Configuration Manager," Configuring Developer Workstations, page 217

Chapter 11

Using PeopleTools Utilities

This chapter provides an overview of the PeopleTools Utilities and discusses how to:

- Use the System Information page.
- Use administration utilities.
- Use audit utilities.
- Use debug utilities.
- Use international utilities.
- Use optimization utilities.
- Use PeopleSoft Ping.

Understanding the PeopleTools Utilities

As you work with the PeopleSoft system, you find that there are some administrative tasks that you only need to perform occasionally. These tasks include such things as maintaining error messages and setting DDL model defaults. The PeopleTools Utilities menu is where you find tools for accomplishing some of these more infrequent tasks.

The documentation of the utilities matches the menu structure of the Utilities interface. For example, the PeopleTools Options utility is under the Administration menu in the Utilities interface; therefore, the documentation for PeopleTools Options is in the Using Administration Utilities section in this chapter. Also, in many cases this book refers to other PeopleBooks for the detailed documentation of a utility.

Using the System Information Page

This section provides an overview of the system information page and discusses how to view the system information page.

With the combination of accessing PeopleSoft applications with a browser, single signon between databases, and the PeopleSoft Portal, users and system administrators need a quick tool to provide orientation information and information regarding the current environment. For this reason, PeopleSoft provides the system information page.

Understanding the System Information Page

With single-signon and the portal, it may not be apparent to all end users just exactly what databases or applications they are currently accessing. Viewing environment information can help end users orient themselves.

In most cases, the administrators use the system help page to aid in troubleshooting. If a user has trouble accessing a particular application, the system administrator can instruct the user to provide the system information that appears in this page so that the administrator can immediately identify the current application server, database, software version, operating system, and so on.

Viewing the System Information Page

To view the System Information help page, you press the CTRL+J hotkey while a PeopleSoft page is active. The following example illustrates the type of information that appears.

Browser	IE/7.0
Operating System	WINXP
Browser Compression	ON (gzip)
Tools Release	8.50-901-R1
Application Release	PeopleTools 8.50.00.000
Service Pack	0
Page	PMN_PRCSLIST
Component	PROCESSMONITOR
Menu	PROCESSMONITOR
User ID	PTDOCGS
Database Name	QEDMO
Database Type	ORACLE
Application Server	//BUFFY:9211
Component Buffer Size (KB)	70

[continue](#)

System Information page

To return to the previous page, click continue.

The following table briefly describes each item:

<i>Item</i>	<i>Description</i>
Browser	The browser version and type, such as Internet Explorer.
Operating System	The operating system that runs on the computer on which the browser is running. For example, this refers to the operating system of the end user's workstation or the operating system running on a kiosk machine. It does not refer to the operating system that runs on the application server, web server, or database server.
Browser Compression	<p>Indicates if browser compression is enabled in the Compress Responses field on the General page of the current web profile. Values are:</p> <ul style="list-style-type: none"> • ON: The flag is on in the web server configuration and the page is compressed. <p>The compression type is either gzip or zip.</p> <ul style="list-style-type: none"> • OFF: The page is not compressed because the flag is cleared in the web profile. • OFF (not supported): The page is not compressed because the browser doesn't support compression, however the flag is turned on in the web profile.
Tools Release	The version of PeopleTools that is currently installed at the site. For example, PeopleTools 8.5, 8.50.01, and so on.
Application Release	The version of PeopleSoft applications that are currently installed at the site.
Service Pack	Typically, updates to PeopleSoft applications arrive in the form of a service pack. This item shows the current service pack that is applied to the applications.
Page	The current page that the user is accessing.
Component	The component to which the current page belongs.
Menu	The name of the menu under which the component appears.
User ID	The user ID of the user that is currently accessing PeopleSoft.

<i>Item</i>	<i>Description</i>
Database Name	The name of the database that the user is currently performing a transaction in.
Database Type	The type of the current database, as in Microsoft, Oracle, DB2 UDB, and so on.
Application Server	The domain name server name or Internet Protocol (IP) address and the JSL port number.
Component Buffer Size (KB)	<p>The component buffer size, which reflects the data buffer size, not including metadata,, such as the record definition or component definition. This metric is the same metric also displayed by the Performance Monitor.</p> <p>Note. The Performance Monitor does not need to be configured for this value to be populated.</p>

You enable and disable the System Information page using the Show Connection Information check box on the Debugging page of the current web profile.

See Also

Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies, "Configuring the Portal Environment," Configuring Trace and Debug Options

Using Administration Utilities

This section discusses:

- PeopleTools Options.
- Message Catalog.
- Spell Check System Dictionary.
- Translate Values.
- Load Application Server Cache.
- Tablespace Utilities.
- Tablespace Management.
- DDL Model Defaults.
- Strings Table.

- XML Link Function Registry.
- Merchant Integration Utilities.
- TableSet IDs.
- Record Group.
- TableSet Control.
- Convert Panels to Pages.
- Update Utilities.
- Remote Database Connection.
- URL Maintenance.
- Copy File Attachments.
- Query Monitor.
- Sync ID Utilities.
- Gather Utility.

PeopleTools Options

Select PeopleTools, Utilities, Administration, PeopleTools Options to access the PeopleTools Options page. Use this page to set a number of options that affect multiple PeopleTools and applications, such as language options and change control settings.

PeopleTools Options

Environment Long Name:	<input type="text"/>	Environment Short Name:	<input type="text"/>
System Type:	<input type="text" value="Undefined Database"/>		

Language Settings

Language Code:	English	*Sort Order Option:	<input type="text" value="Binary Sorting"/>
<input type="checkbox"/> Translations Change Last Update			

Background Disconnect Interval:	<input type="text" value="30"/>	Temp Table Instances (Total):	<input type="text"/>
<input type="checkbox"/> Multi-Company Organization		Temp Table Instances (Online):	<input type="text"/>
<input checked="" type="checkbox"/> Multi-Currency		*Maximum Message Size:	<input type="text" value="10,000,000"/>
<input checked="" type="checkbox"/> Use Business Unit in nVision		Base Time Zone:	<input type="text" value="PST"/>
<input checked="" type="checkbox"/> Use Secure Rep Rqst in nVision		Last Help Context # Used:	<input type="text" value="100222"/>
<input type="checkbox"/> Multiple Jobs Allowed		*Data Field Length Checking:	<input type="text" value="Others"/>
<input checked="" type="checkbox"/> Allow DB Optimizer Trace		*Maximum Attachment Chunk Size:	<input type="text" value="28,000"/>
<input checked="" type="checkbox"/> Grant Access		Upgrade Project Commit Limit:	<input type="text" value="50"/>
<input checked="" type="checkbox"/> Platform Compatibility Mode		*Enable Switch User:	<input type="text" value="All"/>
<input type="checkbox"/> Allow NT batch when CCSID<>37			
<input type="checkbox"/> Save Error is Fatal			
<input type="checkbox"/> Set Focus on Save Button			

*Case Insensitive Searching:	<input type="text" value="On - CaseSensitive Default Off"/>	Max rows in search results	<input type="text" value="300"/>
Style Sheet Name:	<input type="text" value="PSSTYLEDEF"/>	Default rows in search results	<input type="text" value="300"/>
Branding Application Package:	<input type="text" value="PT_BRANDING"/>		
Branding Application Class:	<input type="text" value="BrandingBase"/>		

Tree Manager Options

<input type="checkbox"/> Use Tree Update Reservation
Max Tree Inactivity Period,min: <input type="text" value="20"/>

PeopleTools Options (1 of 2)

Help Options	
F1 Help URL:	<input type="text" value="http://ad-sun12:8000/PSOL/pt850_test2_050609/f1search.htm?"/>
Ctrl-F1 Help URL:	<input type="text"/>
WSRP Display Mode	
WSRP Display Mode	<input type="text" value="Display as Portlet"/>
Database Encryption Algorithm	
Database Encryption Algorithm	<input type="text"/>

PeopleTools Options (2 of 2)

Environment Long Name and Environment Short Name	Enter a long name and a short name for the current PeopleSoft environment. PeopleSoft software update tools use this information to identify the database when searching for updates. For example, enter <i>HR Demo Environment</i> for the long name, and <i>HR Demo DB</i> for the short name.
System Type	Select an appropriate system type from the dropdown list, for example, <i>Demo Database</i> . This information helps to further identify the current environment for the purpose of searching for and applying software updates.

Language Settings

Language Code	<p>The base language of an application is the application's primary language, normally the language that is used most commonly throughout the enterprise. A database can have only one base language. All other language translations that are stored in the database are referred to as nonbase languages (or sometimes as foreign languages).</p> <p>You can't change the Language Code setting on this page. This field is for display purposes only. To change the base language, use the SWAP_BASE_LANGUAGE Data Mover command.</p> <p>The Language Code field box identifies the database's base language.</p>
Translations Change Last Update	<p>If you select the Translations Change Last Update check box, and you use the PeopleTools translate utilities to translate objects, the system updates the Last Updated information of the translated object to the date/time/userid of the translation. If it's turned off, then the date/time/userid of the object does not change when it's translated.</p>

Note. This only applies when you're using the page-based PeopleTools translation utilities; the Translation Workbench always updates the last updated information.

Sort Order Option	<p>Select the sort order that is appropriate for the site.</p> <p>See the Global Technology PeopleBook for descriptions of the options.</p>
--------------------------	---

General Options

Background Disconnect Interval	The value in seconds that you enter here acts as the default for Security Administrator profiles.
Multi-Company Organization	<p>Turn on Multi-Company Organization if more than one company makes up the organization.</p> <p>This option affects how Application Processor displays company-related fields in search dialogs and pages. See the HRMS documentation for more details.</p>
Multi-Currency	<p>The Multi-Currency setting is a systemwide switch that enables automatic formatting of currency amount fields that have associated currency control fields. Another function of this setting is to globally display currency control fields. If you turn off this option, automatic formatting based on currency control fields is no longer active and all currency control fields are thus hidden.</p> <p>When the Multi-Currency setting is on, it also validates user-entered currency data against the currency's defined decimal precision. This validation causes the system to issue an error if a user attempts to enter a decimal precision that is greater than that which is allowed by the currency code definition.</p> <p>Under most circumstances, leave Multi-Currency selected.</p>
Use Business Unit in nVision	Deselect the Use Business Unit in nVision option if you're using an HRMS database. Otherwise, select it.
Use Secure Rep Rqst in nVision	Select this check box if you want the report request in nVision to be secure. The default setting is selected.
Multiple Jobs Allowed	<p>Selecting Multiple Jobs Allowed enables HRMS systems to support employees holding concurrent jobs with more than one set of enrollments.</p> <p>This option affects how Application Processor displays employee-record-number-related fields in search dialogs and pages. See the HRMS documentation for more details.</p>
Allow DB Optimizer Trace	Typically, you turn on this trace only during periods in which you are collecting detailed performance metrics. When you are not tuning your performance, the DB Optimizer trace should be turned off.
Grant Access	When adding a new user by using PeopleTools Security, the system automatically grants the new user select-level access to a set of PeopleTools SQL tables used for security. If you are using a SQL security package and do not want PeopleTools Security to perform any SQL grants, turn off Grant Access.

Platform Compatibility Mode	<p>Enables you to add the capability to set a database compatibility mode as an overall database setting, forcing developers to create applications by using all platforms as the least common denominator. This option enables developers, who create applications for multi-platform deployment, to catch platform-specific issues at design time rather than during testing.</p> <hr/> <p>Note. This option is used mainly by PeopleSoft development teams that need to develop applications to run on all supported database platforms. To support numerous database platforms, PeopleSoft needs to have a tablespace for each physical table record definition.</p> <hr/> <p>If platform compatibility is enabled for a database, the system forces developers to enter a tablespace name when saving a record definition regardless of the current platform. If this option is disabled, you are only prompted for a tablespace name if you are developing on a platform that utilizes tablespaces. This prevents table record definitions being added to the database without a tablespace name.</p>
Allow NT batch when CCSID <>37	<p>Enables you to override non-z/OS COBOL batch restrictions. If the DB2 z/OS database's CCSID is NOT 37, PeopleSoft blocks batch COBOL from running against z/OS Databases on Windows unless you choose this override.</p> <hr/> <p>Note. Even if you choose this override, if you use %BINARYSORT() in the COBOL, the system issues an error on Windows. RemoteCall COBOL can run on Windows and UNIX regardless of this option setting, even if CCSID is NOT 37, but the system issues an error.</p> <hr/>
Save Error is Fatal	<p>Select this option when you have non-repeatable PeopleCode logic in your application's SavePreChange or Workflow. In previous releases, PeopleSoft applications were coded to assume that errors during save are always fatal, but the current PeopleTools release no longer behaves this way. Use this option to ensure predictable behavior with your application without having to modify your older application code.</p> <p>This check box is cleared by default. If you get an error during save processing, the transaction continues and you're allowed to attempt to save again. When this option is selected, if you get an error during save processing the transaction is aborted and all changes are lost. This applies to errors that occur between and including the SavePreChange event to the SavePostChange event. It also includes the component processor save processing. It doesn't include errors from the SaveEdit event.</p> <p>For example, suppose you have some calculations that occur in SavePreChange which are based on the buffers and also modify the buffers. If there's an error during the save and you attempt to save again, the calculations are repeated, but this time based on the buffers that were already modified by the first time the calculations were done. Therefore the second time the calculations are done they will be incorrect, which could lead to incorrect data being saved to the database. In this case you would want to turn on the Save Error is Fatal option, because a fatal error on save is more desirable than incorrect data being put into the database.</p>

Set Focus on Save	<p>If selected, focus is set on the Save button when a user saves a component. If not selected, focus is set on the first control on the page that can assume focus when a user saves a component. By default, the option is not selected.</p> <hr/> <p>Note. This setting has a system-wide effect.</p> <hr/>
Temp Table Instances (Total):	<p>The value that you specify in the Temp Table Instances (Total) edit box controls the total number of physical temporary table instances that Application Designer creates for a temporary table record definition when you perform the Build process.</p> <p>This value indicates the total number of undedicated temporary table instances. The maximum number of temporary table instances that you can specify is 99.</p>
Temp Table Instances (Online)	<p>Enter the available online instance values. When you invoke a process online, PeopleTools randomly allocates a single temporary table instance number to programX for all of its dedicated temp table needs. The higher the number of online instances that is defined, the less likely it is for two online processes to get the same value.</p>
Maximum Message Size	<p>There is practical limit to how large a message can be. Enter the maximum message size; this does not set individual message definition, but defines the size for all application messages.</p>
Base Time Zone	<p>Although you can display time data a number of different ways, PeopleSoft databases store all times relative to a system-wide base time zone. You can adjust the display of the time that an end user sees using the Use Local Time Zone (LTZONE) setting in PeopleTools, Personalizations.</p> <p>This base time zone is the one that the database server uses. In order for PeopleSoft to properly manage time data, the system needs to know which time zone that is. Set the Base Time Zone to the time zone that the database server's clock uses.</p> <hr/> <p>Note. After changing this setting, reboot any application servers that are connected to the database. It is critical for the correct operation of the system that this time zone match the time zone in which the database is operating. Any discrepancy in the base time zone as defined in this page and the time zone in which the database system is operating leads to inaccurate time processing.</p> <hr/>
Last Help Context # Used	<p>This field is no longer used.</p>

Data Field Length Checking

Normally, field length validation is based on the number of characters that are allowed in a field. For example, a field defined as CHAR(10) in Application Designer holds ten characters, regardless of which characters you enter. In a Unicode database, double-byte characters, such as those found in Japanese, are counted the same as single-byte characters, such as those found in the Latin alphabet.

If you create a non-Unicode database, the field length in Application Designer represents the number of bytes that are permitted in the field, not the number of characters. When the non-Unicode database uses a single-byte character set (SBCS), you can only enter single-byte characters, so the number of characters and the number of bytes are the same. However, because double-byte character sets (DBCS) typically allow a mix of single- and double-byte characters, the number of characters that are allowed in a field in a non-Unicode DBCS database varies. This is true for both shifting and non-shifting double-byte character sets.

For example, if a user enters ten Japanese characters into a field that is defined as CHAR(10) in Application Designer, this string needs 20 bytes of storage in a nonshifting double-byte character set and 22 bytes of storage in a shifting double-byte character set. This ten-character input fails insertion into both these databases.

Use the Data Field Length Checking option to ensure field length validation appropriate to the database's character set. Values are DB2 MBCS, MBCS, and Others.

Choose Others if you are using a Unicode-encoded database or a non-Unicode single-byte character set database. This prevents special field length checking. As discussed above, these types of databases do not require such checking.

Choose DB2 MBCS if you are running a Japanese database on the DB2 UDB for z/OS platform. This enables field length checking based on a shifting DBCS character set.

Choose MBCS if you are running a non-Unicode Japanese database on any other platform. This enables field length checking based on a nonshifting DBCS character set.

The non-Unicode DBCS settings are specifically oriented towards Japanese language installations, as Japanese is the only language that PeopleSoft supports in a non-Unicode DBCS encoding. All other languages requiring double-byte character sets are only supported by PeopleSoft by using Unicode encoded databases.

Maximum Attachment Chunk Size

Controls the size of the file attachment chunks that are stored in your database. This allows you to upload a file attachment that is larger than could be stored in a single row of your database. The default is 28,000 bytes.

Upgrade Project Commit Limit

Sets the limit on how many rows can be modified by an upgrade project before the system issues a COMMIT statement.

Enable Switch User

The Enable Switch User option enables you to limit the users who can change identities in a PeopleSoft system. The feature applies only when accessing PeopleSoft using a browser; it has no effect on two-tier or three-tier connections.

Most sites have no reason for individual users to change their PeopleSoft system identity during a session. For those sites that do require this capability, the number of users who need to switch to another user profile typically is fairly small. For example, the users who can switch identities are usually limited to the system's GUEST user (if the system has one) and, perhaps, a few system administrators or power users.

Options for Enable Switch User are:

- *All:* Use this value to indicate that all users are permitted to change their identities during their browser session. This is the default value.
- *None:* Use this value to indicate that no user is permitted to change identities during a browser session
- *Some:* Use this value to indicate that some, selected users are permitted to change their identities during a browser session. When you select this value, the Allow Switch User checkbox appears on the General page of the user profile definition in the PeopleTools Security interface. Select Allow Switch User to indicate that the individual user is permitted to change identities within a PeopleSoft session. User profiles that have the Allow Switch User option unselected will be immediately logged out when the system detects a change in the user's identity.

Once you specify which users can switch identities, if an identity change is attempted (switch user) by an unauthorized user, the system:

- Logs that user off the system, immediately.
- Displays the following message on the signout page: "Illegal Identity switch has been detected by the System. Please re-login."
- Writes an entry in the web server logs indicating that an illegal switch user was attempted.

Note. Assume that the user profile to which a user switches is the "destination" user profile, and the user profile from which a user switches is the "source" user profile. If a user is allowed to switch identities during a session, it is allowed by the source user profile, not the destination user profile. For example if you allow UserA to switch to UserB, it is UserA's user profile that must have the Allow Switch User option selected. The setting for UserB's profile is not relevant in determining whether a switch user from UserA to UserB is allowed. This feature does not control the user profile to which a user can switch; it only addresses which users are permitted to switch identities during a browser session.

See *Enterprise PeopleTools 8.50 PeopleBook: Security Administration*, "Administering User Profiles," Setting General User Profile Attributes.

Case Insensitive Searching

Enables you to provide case-insensitive searching for the PeopleSoft search records when searching for PeopleSoft definitions. The options are:

- *Off* : Enables case sensitive searching and sets it to "on" *without* displaying the Case Sensitive check box on the search page.
- *On - Case Sensitive Default Off*: (Default) Displays the Case Sensitive check box on the search page unselected, providing the user the option of enabling case-sensitive searching for a particular search by selecting the Case Sensitive check box.
- *On - Case Sensitive Default On* : Displays the Case Sensitive check box on the search page selected, providing the user the option of switching to *case-insensitive* searching for a particular search by *deselecting* the Case Sensitive check box.

Note. This option is not associated with the Verity search technology.

Style Sheet Name

All PeopleSoft applications reference the PSSTYLEDEF style sheet by default. You can set the individual style sheets in Application Designer, and these override the general style sheet for the application, which is set here.

Branding Application Package

Specifies the application package that contains the branding application classes to generate the portal headers, footers and menu pagelet icons. The default is the standard PeopleTools branding, PT_Branding. For Enterprise Portal, a different branding application package is specified.

Max rows in search results

Sets the absolute maximum number of rows, system-wide, an end user can specify to be returned on a standard search page. The default value is 300.

At the top of a search page, the following appears:

"Maximum number of rows to return (up to *Max rows in search results*):"

End users specify a custom number of rows on the search results page in the edit box appearing after the above line.

To prevent end users from specifying any custom number of rows returned by the search, set Max rows in search results to 0. If disabled, the number of rows returned by a search is the value from the Default rows in search results setting, and cannot be configured by the end user.

Note. When setting this value to add flexibility to your end users search experience, consider the performance implications of setting this number too high for your system. Perform adequate testing and adjust this value to suit the requirements of your system.

Default rows in search results

Sets the default number of rows, system-wide, to be returned on a standard search page. The default value is 300.

If an end user needs to increase the number of rows returned on a search page, they can specify a number up to the maximum, specified by the Max rows in search results value.

Default grids to scrollable

Sets all grids displayed to the end user as scrollable grids, by default.

Branding Application Class

The main branding application class that generates header, footer, and menu pagelet icons. The default is the standard PeopleTools branding, BrandingBase. For PeopleSoft Enterprise Portal, a different branding application class from a different branding application package is used. It generates different header, footers, and menu pagelet icons dynamically, based on the user role or security.

Help Options**F1 Help URL**

This setting applies only to the Windows environment (such as Application Designer) when the user presses F1 or selects Help, PeopleBooks Help while in PeopleTools.

The F1 Help URL can direct users to any location on the web, such as a custom help system or the website for the company's help desk. It can be a fully qualified uniform resource locator (URL), which is passed literally to the browser, or it can contain one or both of these system variables.

%CONTEXT_ID% is the object name or context ID of the currently displaying page or dialog box.

%LANG_CD% is the three-letter language code for the user's preferred language.

The PeopleBooks context sensitive help system requires that you enter a URL with a specific format, as follows:

```
http://helpwebserver:port/productline/flsearch.htm?=>
ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%
```

For example:

```
http://myhelpwebserver:8080/htmldoc/flsearch.htm?=>
ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%
```

Specify the URL that is needed to link to the correct location in your HTML PeopleBooks. When users click the Help button, the appropriate context-sensitive PeopleSoft documentation should appear. To remove the help link, leave this value blank, and users won't see a Help link on the application page. Construct the URL like this: `http://helpwebserver:port/productline/flsearch.htm?ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%` For example: `http://myhelpwebserver:8080/htmldoc/flsearch.htm?ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%`

Ctrl-F1 Help URL

This setting only applies to the Windows environment (such as Application Designer).

The Ctrl+F1 URL allows you to provide an alternate location for help. For example, you may set the main F1 Help URL to the PeopleBook and the Ctrl+F1 for the company's help site.

WSRP Display Mode

This option determines how WSRP-enabled content appears for users of remote WSRP portals that consume PeopleSoft WSRP content.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies*, "Using WSRP to Consume and Produce Remote Portlets," Setting WSRP Display Mode.

Database Encryption Algorithm


(Applies to Oracle databases only.) Enables you to select the encryption algorithm the system uses to encrypt database fields to additional security.

See *Enterprise PeopleTools 8.50 PeopleBook: Data Management*, "Administering PeopleSoft Databases on Oracle," Implementing Oracle Transparent Data Encryption.



Message Catalog

Select PeopleTools, Utilities, Administration, Message Catalog to access the Message Catalog page.


Message Catalog


Message Set Number: 2
Description: 
Short Description:

Messages Find | View 100 First 1 of 695 Last

Last Update Timestamp: 06/04/1999 8:24AM  





***Message Number:**

***Severity:** 

***Message Text:** 

Description:

The exclamation symbol (!) can only be used as a token within
 PeopleCode as part of a not-equal operator (!=).
 Either make the operator a proper not-equal expression or delete the

Message Catalog page

You add and maintain system messages by using the Message Catalog page. PeopleSoft error messages are stored in the Message Catalog, and organized by message set number. Each message set consists of a category of messages, ranging from PeopleTools Message Bar Items and PeopleCode Runtime Messages to PeopleSoft Payroll and PeopleSoft General Ledger application messages.

Message Set Number Identifies the message set.

Description	The Message Set Description is a reference that is used on reports and pages for easy identification.
Short Description	The Message Set Short Description is a reference that is used on reports and pages for easy identification.
Message Number	Each message set consists of one or more rows of messages that are identified by a message number.
Severity	<p>You assign each message a severity, which determines how the message appears and how the component processor responds after the user acknowledges message. The severity levels are:</p> <p>Cancel: This severity should be reserved for the most severe of messages, as when a critical error occurs and the process must be aborted or a machine needs to be shut down. To indicate how rarely this severity level is appropriate, of all PeopleTools messages only five or so have a severity level of Cancel. In almost all cases, you use one of the other severity levels.</p> <p>Error: Processing stopped, and data cannot be saved until the error is corrected.</p> <p>Message: This is an informational message and processing continues normally.</p> <p>Warning: User can decide to either stop or continue processing despite the error.</p>
Message Text	In the Message Text edit box, you see the message text. Any reference to the characters %n, as in %1 or %2, is replaced by parameter values that the system provides.
Explanation	The Explanation text provides a more in-depth explanation of why the message is generated and how to fix the problem. This text appears below the Message Text when the message appears.

PeopleTools uses some messages, but the applications use the other messages, which get called by the Error, Warning, Message Box, MsgGet, and MsgGetText built-in PeopleCode functions.

Note. You can create messages and message sets to support new or customized functionality in the system. You can also edit the messages that PeopleSoft delivers. In both of these cases, remember that PeopleSoft reserves all message set numbers up to 20,000. If you add a message set or edit a message set with a number that is less than 20,000, it may be overwritten in future upgrades:

To add a message set:

1. Select Utilities, Administration, Message Catalog, and on the search page click Add New Value.
2. Enter the value of the new Message Set Number and click OK.
3. Enter a description and short description of the type of messages that this message set contains.

Try to group the messages logically. For instance, create one message set for the new budgeting application and a different one for the customized billing pages.

4. Add messages.
5. Save your work.

To add a message:

1. Open the desired message set.
2. In the Message Catalog page, click the plus sign button to add a new row.

The Message Number value is automatically set to the next unassigned number in the message set.

3. Select a Severity level, enter message text and a detailed explanation.
4. Save your work.

Spell Check System Dictionary

PeopleSoft PeopleTools provides personal and system-level dictionaries. End users and system administrators can add words to the dictionary for use with the spell check feature. Typically, system administrators add words to the system-level dictionary that are used company-wide; end users add additional role-specific terminology to their personal dictionaries.

Select PeopleTools, Utilities, Administration, System Dictionary to access the system-level dictionary.

Select the All Languages page to enter words that are valid across all languages. Select the Language Specific page for those words that are valid to a specific language:

To add words to the system dictionary by language:

1. Select Spell Check System Dictionary, Language Specific.
2. Select the desired language from the Spell Check Language drop-down list box.
3. Select Session to add a word to the current session's spell check dictionary. After saving this word, the language field refreshes to the current spell check language.
4. Enter the word (maximum 40 characters) that is to be added in the Spell Check Word field.
5. Save your changes.

Case Sensitivity for Spell Check

The words that you add to your personal dictionary are case-sensitive and are validated by the following rules:

1. If the added word is all lower case, such as worklist, then the following are considered valid:
 - Exact match, all lower case (worklist).
 - All uppercase (WORKLIST).
 - Initial capitals (Worklist), regardless of its position in the sentence. Mixed case (WorkList) is considered incorrect.
2. If the added word is all uppercase, such as CRM (customer relationship management), then only an exact match is valid.
3. If the added word is in initial capitals, such as California, then only an exact match and all upper case (CALIFORNIA) are considered valid.

4. If the added word contains an embedded capital letter, such as PeopleSoft, then only an exact match is valid. Therefore, if case is not relevant to the validity of the word, use all lower case.

Table Structure for Word Storage

System and personal words are stored in the database in the PSSCWORDDEFN table with the following fields:

- SCOPRID indicates whether a word is a system word or a user's personal word.
- SCLANG stores the dictionary language for which the word is considered valid. If the system administrator chooses to store the word for all languages, this field is left blank.
- SCWORD stores the actual word, with a maximum length of 40 characters.
- SCNEGWORDFLG is a flag used to determine if a word is negative (incorrect) or not. Values can be 'N' or 'Y'. PeopleSoft does not currently use this feature, so the value should always be set to 'N'.

To load values in bulk into PSSCWORDDEFN:

1. Using the method of your choice (as in a SQL script), issue SQL similar to the following:

```
insert into PSSCWORDDEFN (SCOPRID, SCLANG, SCWORD, SCNEGWORDFLG)
values ('SYSTEM', 'SC00', 'nnn', 'N')
```

Note. For each word you want to add to the library, you need a separate insert command, and the value 'nnn' will be changed in each of those insert statements to be the next value in the list of words you want to add.

2. Add a value (any value) to the Language Specific tab and click Save.

This alerts the runtime system to update the cached version of the PSSCWORDDEFN table.

Note. In the current release, the maximum number of rows in the PSSCWORDDEFN table should not exceed 2,850.

Translate Values

You use the Translate Values interface to maintain the values in the translate table. If it's allowed by site security administrators, power users can now learn to add their own pick lists (translate values) to an application:

Select PeopleTools, Utilities, Administration, Translate Values to access the Maintain Translate Values page.

Use this page to maintain translate values

Field Name: **ANALYSIS_PLATFORM** Length: **3**

Maintain Translate Values						
General		Last Updated		Customize Find View All First 1-5 of 5 Last		
	*Value	Effective Date	Status	*Long Name	*Short Name	
1	ESS	01/01/1900	Active	Hyperion Essbase	Essbase	
2	IAS	01/01/1900	Inactive	PS/ROLAP	PS/ROLAP	
3	MOS	01/01/1900	Inactive	Microsoft OLAP Services	MsftOLAP	
4	PPL	01/01/1900	Active	Cognos PowerPlay	PowerPlay	
5	STS	01/01/1900	Active	Generic Star Schema	StarSchema	

Maintain Translate Values page

Value Enter the value for the translate selection.

Effective Date Specify a date for the value to become active.

Note. If you are adding a second row for the same translate value, you must enter a unique effective date.

Status Specify whether the value is active or not.

Long Name Enter a long description for identification. There is a 30-character limit.

Short Name Enter a shorter description for identification. There is a 10-character limit.

See Also

Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Application Designer Developer's Guide, "Creating Field Definitions," Using the Translate Table

Load Application Server Cache

The Load Application Server Cache page enables you to invoke Application Engine programs that preload the *shared* cache for application server domains. You need to run a cache loading program described in this section only if you intend to implement shared caching on the application server.

To preload *non-shared* cache, you use a separate process.

See [Chapter 5, "Using PSADMIN Menus," Configuring an Application Server Domain to Preload Non-Shared Cache, page 67.](#)

See [Chapter 6, "Setting Application Server Domain Parameters," ServerCacheMode, page 94.](#)

Understanding Application Server Caching

Each server process running within a domain has two types of cache:

<i>Cache Type</i>	<i>Description</i>
Memory cache	Stores application data to improve system performance. Memory cache is always enabled for each server process in a domain.
File cache	Stores PeopleTools definition metadata, such as the metadata for pages, locally on the application server. File caching is controlled by the EnableServerCaching parameter in PSAPPSRV.CFG.

With file caching, you have the option of implementing shared or non-shared file caching.

<i>File Cache Option</i>	<i>Description</i>
Shared	With shared file caching, all of the server processes within a single domain share the same file cache location.
Non-shared	With non-shared file caching, each server processes within a single domain maintains its own file cache. For example, with non-shared caching configured, each PSAPPSRV server process uses its own file cache.

If you intend to implement a shared file cache, then you need to preload the file cache location before users access the system. Oracle provides cache loading programs designed specifically for this purpose.

Note. The delivered cache loading programs focus on file cache, not memory cache.

The cache loading programs retrieve all of the PeopleTools definition metadata from the database and builds a file cache that you can deploy locally on an application server. Running a cache loading program is the equivalent of having a user access every page in the system once so that all the metadata associated with each page is retrieved from the database and stored in the local cache. With all the metadata already stored in cache, the end users experience more predictable performance.

With a preloaded cache, end users don't have to wait for the system to retrieve a definition from the database and cache it, each time a definition is accessed for the first time. Because the cache is preloaded with all the database definitions, the system retrieves all of the required objects from the local cache, not the database. This provides significant performance benefits for first-time and large transactions.

If you elect to implement the shared, preloaded cache, consider the following items:

- You need to run one of the cache loading programs at least once. Once the cache is preloaded, the cache loading programs download only new or changed metadata to the cache.

Note. Items that are outdated in the shared cache are marked invalid but are not deleted. This includes design time changes, upgrades, patches, and so on. However, the system uses only the most current cache definitions.

- The first time that you run a cache loading program, it can take a significant amount of time to complete, for example 2 to 30 hours. The duration of the program run depends on a number of factors, including the number of active languages selected, the size of the database, and the power of the server machine. Subsequent program runs complete in less time if there is already valid cache data in the target cache directory. The programs are designed to update only the changed objects after the staging directory is already loaded.
- You can copy the output of a cache loading program to other machines.

Note. The output is not portable to different operating systems. For instance, if you generate the cached metadata on a Windows server, you can't copy the cache files to a UNIX server.

Warning! If for any reason you update PSSTATUS.LASTREFRESHDTM, the system marks all items in the shared cache as invalid, and you will need to rerun one of the cache loading programs again.

Oracle provides these options for preloading your file cache:

Preloading Cache Option	Process Scheduler Process Type	Description
Serial cache loading	Process	Serial cache loading refers to running an Application Engine program named LOADCACHE. The LOADCACHE program, adds each applicable cache object to the shared cache, sequentially.
Parallel cache loading	PSJob	<p>Parallel cache loading is intended to reduce the time required to build a shared cache. Parallel cache loading splits the work of the LOADCACHE program into two separate Application Engine programs that run simultaneously within the PSJob PLCACHE.</p> <p>The PLCACHE job runs a basic setup program, and then runs these programs:</p> <ul style="list-style-type: none"> • LCACHE_INDEP: Loads all the cacheable objects that are independent of other cached objects. • LCACHE_DEP: Loads all the cacheable objects that are dependent on other cached objects. Objects that are dependent on other cached objects are those objects that, when loaded into cache, cause instances of other object to be loaded as well.

Preloading Shared Application Server Cache

To create and deploy a shared cache:

1. Make sure that the database that the application server runs against produces a clean SYSAUDIT report.
If SYSAUDIT is not clean, the cache loading programs can fail.
2. Access the Load Application Server Cache page, entering an appropriate Run Control ID.

Select PeopleTools, Utilities, Administration, Load Application Server Cache to access the Load Application Server Cache page.

3. In the Output Directory field, enter a valid location.

Note. In almost all situations, the system ignores any custom value entered in the Output Directory and generates output into the default directory: `PS_CFG_HOME\appserv\prcs\ProcessScheduler_domain\cache\stage\stage`. However, if a cache loading program runs through a PSNT Process Scheduler server definition running on a remote drive, *and* the application server cache directory and Process Scheduler cache directory are located on the local drive, then the value entered in the Output Directory field supersedes the default output directory.

4. Select the languages to load.

If not all possible languages are required to be loaded for a domain, you can improve performance of the program run by only selecting those languages that are required.

5. Click Run.

The Process Scheduler Request page appears.

6. From the Server Name dropdown list, select the name of the server that you want to run the process.
7. From the Process List, select *Serial LoadCache* to run the single LOADCACHE Application Engine program, or select *Parallel LoadCache* to run the PLCACHE PSJob to run multiple Application Engine jobs in parallel.
8. If running Parallel LoadCache, ensure that the Process Scheduler server definition is set to run the LOADCACHE process category and that the Max Concurrent value is set appropriately.

Parallel LoadCache is assigned to the LOADCACHE process category. It will run only on servers that are configured to run the LOADCACHE process category.

For the LOADCACHE process category, set the Max Concurrent value greater than 0. The recommended value is 2.

If you modify the server definition, restart the server for your changes to take effect.

Note. When running the cache loading programs, only one server at a time should have the MaxConcurrent value set greater than 0. Otherwise there is no guarantee that the processes running within the Parallel LoadCache job will run on the same server. This could result in each server having an incomplete set of generated cache files.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Process Scheduler*, "Setting Server Definitions."

9. Click OK to launch the cache loading program.

The first time that you run the program, the process may take numerous hours to complete (less time for the Parallel LoadCache option). The cache loading program creates the cache files in the following directory:

`PS_CFG_HOME\appserv\prcs\ProcessScheduler_domain\cache\stage\stage`

Where *ProcessScheduler_domain* is the Process Scheduler domain in which you ran the process.

After you invoke the process, you can use the Report Manager and Process Monitor links to monitor the progress.

10. When the cache loading program completes, and you've determined that the file cache has been successfully loaded, shut down the application server domain(s).
11. Edit the PSAPPSRV.CFG file, and enable shared caching, by uncommenting and setting `ServerCacheMode=1` for all appropriate domains.
12. Reconfigure the domains so that the changes are reflected.
13. Copy the contents of the output directory into the `PS_CFG_HOME\appserv\domain\cache\share` directory for the appropriate application server domain(s).
14. Boot the application server domain(s).

See Also


























Chapter 5, "Using PSADMIN Menus," Configuring an Application Server Domain to Preload Non-Shared Cache, page 67

Tablespace Utilities

Select PeopleTools, Utilities, Administration, Tablespace Utilities to access the Tablespace Utilities page.

To comply with requirements for DB2 UDB for z/OS, the Tablespace Utility now includes both tablespace name and database names when you define a tablespace using the Tablespace Management page. Use the Add/Delete/Rename Tablespace page to change the list of tablespace and database names.

Add/Delete/Rename Tablespace

Tablespaces Defined in the Database						Customize Find View All  		First  1-7 of 14  Last
	*Tablespace Name	Database Name	Type	DB2 Unix Type	Comment			
1	PSIMAGE	PSPTDMO	Regular 	DMS				
2	PSIMGR	PSPTDMO	Regular 	DMS				
3	PTAMSG	PSPTDMO	Regular 	DMS				
4	PTAPP	PSPTDMO	Regular 	DMS				
5	PTAPPE	PSPTDMO	Regular 	DMS				
6	PTAUDIT	PSPTDMO	Regular 	DMS				
7	PTLOCK	PSPTDMO	Regular 	DMS				

Add/Delete/Rename Tablespace

Tablespace Name Enter the name of the Tablespace that you want to add.

Database Name Enter the database name into which you want to add the space.

Type Select the type of Tablespace.

Use the plus/minus buttons to add and delete tablespaces.

Tablespace Management

Select PeopleTools, Utilities, Administration, Tablespace Management to access the Tablespace Management component

These pages enable you to modify the tablespace definition.

Tablespace Defn Page

This page shows the identification values for the tablespace.

Tablespace List Page

This page is where you add records to a particular tablespace. Use the plus and minus buttons to add and delete rows from the list.

Tablespace DDL Page

This page enables you to view and override DDL parameters if needed. View the default DDL in the Default Tablespace DDL list. You override specific parameters, if needed, in the Override Tablespace DDL list. Enter the parameter that you want to override in the Parameter Name column, and enter the override value in the Override column.

DDL Model Defaults

Select PeopleTools, Utilities, Administration, DDL Model Defaults to access the DDL Model Defaults page.

This page is used to view and edit the DDL for creating tablespaces, indexes and tables. Any changes that you make here are global.

DDL Model Defaults

Platform ID:2Oracle

Sizing Set:0

Copy...

DDL

Find | View All

First1 of 5Last

Statement Type:Table

*Model SQL:

CREATE TABLE [TBNAME] ([TBCOLLIST]) TABLESPACE [TBSPCNAME]
STORAGE (INITIAL **INIT** NEXT **NEXT** MAXEXTENTS **MAXEXT**
PCTINCREASE **PCT**) PCTFREE **PCTFREE** PCTUSED

Parameter Count:6

Parameters

Customize | Find | View All

First1-3 of 6Last

DDL Parm	DDL Parameter Value
INIT	40000
MAXEXT	UNLIMITED
NEXT	100000

DDL Model Defaults

- Platform ID

Identify the type of platform of the current database type.
- Sizing Set

Specify multiple Sizing Sets if needed. Sizing Sets are a way to maintain multiple versions of the DDL Model statements for a particular database platform. For example, you could have one sizing set to be used during a development phase, when tables only have test data, and you could have separate sizing set to be used during production, when tables have much more data.
- Copy

Copies information from one sizing set to another.
- Statement Type

Indicates the type of statement that's entered in the Model SQL edit box. Values for this field can be *Table*, *Index*, and *Tablespace*.
- Model SQL

This field displays the model SQL statements, which you can edit. Valid statements are CREATE TABLE, CREATE INDEX, CREATE TABLESPACE, and a platform-specific statement for updating statistics.

Some platforms have all the statements, some do not.
- Parameter Count

The Parameter Count is calculated based on how many nonblank DDL Parm rows that you define.
- DDL Parm

Enter the DDL Parm (parameter) to which you want to assign a default.
- DDL Parameter Value

The DDL Parameter Value field is where you override a DDL parameter's default value with your own for the selected statement type.

Using the DDL Model Defaults page, you can maintain DDL model statements and default parameters for Data Mover and Application Designer when DDL (PSBUILD.SQL, for example) is generated during the build process.

Using this utility, you can:

- Scroll through all the statement types and platforms that are defined in the PSDDLMODEL table.
- Change DDL model statements.
- Add, delete, or change DDL parameters and values.

The Platform IDs are as follows:

<i>Number</i>	<i>Platform</i>
0	SQLBase (no longer supported).
1	DB2.
2	Oracle.
3	Informix.
4	DB2/Unix.
5	Allbase (no longer supported).
6	Sybase.
7	Microsoft.
8	DB2/400 (no longer supported).

Note. There is no validation performed on the Model SQL statement, the DDL Parm syntax, or the relationship between the statement and the parameters.

BOE Integration Administration

This page enables you to manage your BusinessObjects Enterprise installation.

See *Enterprise PeopleTools 8.50 PeopleBook: Crystal Reports for PeopleSoft*, "Getting Started with Crystal Reports for PeopleSoft."















Strings Table

Select PeopleTools, Utilities, Administration, Strings Table to access the Strings Table page.

Strings Table

Program ID: QEGBR09 Report Language: English

String List Find | View All First 1 1-7 of 22 Last

*String Source	*String ID	Default Label	String Text	Width
<div>Text</div>	100%-AREA	<div><input checked="" type="checkbox"/>  </div>	100% Area Chart	15 <div>+ -</div>
<div>Text</div>	100%-BAR	<div><input checked="" type="checkbox"/>  </div>	100% Bar Chart	14 <div>+ -</div>
<div>Text</div>	AREA	<div><input checked="" type="checkbox"/>  </div>	Area Chart	10 <div>+ -</div>
<div>Text</div>	AUGUST	<div><input checked="" type="checkbox"/>  </div>	August	6 <div>+ -</div>
<div>Text</div>	BAR	<div><input checked="" type="checkbox"/>  </div>	Bar Chart	9 <div>+ -</div>
<div>Text</div>	FLOATING-BAR	<div><input checked="" type="checkbox"/>  </div>	Floating Bar Chart	18 <div>+ -</div>
<div>Text</div>	HIGH-LOW-CLOSE	<div><input checked="" type="checkbox"/>  </div>	High-Low-Close	14 <div>+ -</div>

Strings Table page

- String Source

Options are:

RFT Long: Select if you want the long description of the field to be displayed in the column heading as set in Application Designer.

RFT Short: Select if you want the short description of the field as set in the Application Designer to be displayed in the column heading.

Text: Select to enter a custom column heading for the report.
- String ID

Use the browse button to select the string ID that is to be used for the column heading in the SQR report.
- Default Label

The default label is enabled if you select the RFT Long or RFT Short string source, otherwise, the check box is disabled.

Remember that fields can have multiple labels. Select the Default Label option to ensure that the default label is used. If you do not use the field's default label, you must select which of the field's labels to use using the label properties button.
- String Text

Enter the text for the custom column heading, This is the text that is displayed if you set the string source to Text.

Width

The default value is the current width of the string that you enter or select. Be sure to update the width based on the actual space that is available on the report layout to avoid limiting a translator to an artificially short length, which is likely to degrade the quality of the translation.

Lookup Exclusion

Enter the record name for the tables you want to exclude from the auto lookup feature. A prompt or lookup button opens a lookup page in the user's browser populated with up to 300 available values for that field. The user can then either select the desired value or refine their search further. For extremely large tables, the system administrator has the option of excluding that table from auto prompting by adding the table to this list.

See Also

Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Application Designer Developer's Guide, "Creating Page Definitions," Prompt Fields

XML Link Function Registry

The XML Link Function Registry is used exclusively in conjunction with the XML Link technology, which is associated with PeopleSoft Business Interlinks.

Note. PeopleSoft Business Interlinks is a deprecated product. This option currently exists for upgrade compatibility and transition.

Merchant Integration Utilities

There are two utilities that are related to the Merchant Integration technology that are provided for upgrade support only: Merchant Categories and Merchant Profile.

Refer to PeopleSoft documentation from previous releases for information regarding these utilities. These utilities are not intended for any new development purposes.

TableSet IDs

Select PeopleTools, Utilities, Administration, TableSet IDs to access the Tableset ID page.

Use this utility to create Set IDs. Before doing this:

- Add the SETID field (as a key field) to the record definition for that table.
- Define a Set Control Field as the field controlling the assignment of table sets.

SetID

Enter the set ID as defined in the record definition.

Descriptions/Comments Add any descriptions and comments that are necessary for identification and internal documentation.

Record Group

Used to group record definitions for the tables that you want to share, as well as any dependent record definitions.

Select PeopleTools, Utilities, Administration, Record Group to access the Record Group page.

Record Group

Record Group ID: QEDATA02

Description:

DEPT TABLES

Short Description:

DEPT

☐ Force Use of Default SetID

Records in Group

Customize | Find | View All |

First 1-2 of 2 Last

*Record (Table) Name	Record Description		
QE_DEPT_LANG	QE Department Language		
QE_DEPT_TBL	QE Data Department Table		

Record Group page

- Description

The Record Group ID description should provide enough information to encompass a category of related tables, not just the table that you are specifically sharing.
- Short Description

Enter a short description.
- Force Use of Default SetID

This overrides alternate setIDs that are entered so that the default is used.
- Record (Table) Name

This prompt list comes from a SQL view of record definitions that are defined with that Set Control Field that aren't already associated with a record group.
- Record Description

Automatically populated when the Record (Table) Name is selected.

TableSet Control

The following pages are used to control table sets.

Record Group Page

Select PeopleTools, Utilities, Administration, TableSet Control to access the Record Group page.

Used to define which record groups use which table set.

Default SetID	This is the setID that the system uses as you add additional record definition groups to be shared within this tableset.
SetID	Although this database is set up to share only one accounting-related record group, you may have multiple record groups to which you assign default unique Set IDs.

Tree Page

Select PeopleTools, Utilities, Administration, TableSet Control, Tree to access the Tree page.

Used to share Trees as well as tables and views.

Default SetID	The Default setID that you assign to this field value automatically appears. If you create another tableset for sharing trees, you can change this value.
Tree Name	Use the browse button to select from a list of only the tree definitions that are defined with the same Set Control Field.
SetID	Use the browse button to select the appropriate SetID.

Convert Panels to Pages

The following pages are used to convert panels that are used in previous PeopleSoft Windows applications to pages that are used for browser access.

Scope Page

Select PeopleTools, Utilities, Administration, Convert Panels to Pages to access the Scope page.

This utility helps you update panels from PeopleTools 7.5x versions to reflect the pages that are used for the PeopleSoft Internet Architecture.

Project List	Insert projects, containing panels that you want to convert, into this scroll. In addition, if you use the Apply Panel Group Defaults option, any panel group that is contained in projects in this scroll are processed. Note that exceptions may be defined see the task titled, Project Exceptions.
Page List	Insert panels that you want to convert to pages into this scroll.

Project Exceptions If you want to ensure that a group of panels or panel groups is never processed for conversion, you can insert them into an application upgrade project and insert the project name in this scroll.

Page Exceptions Panels that are inserted into this scroll are not be processed.

See PeopleSoft upgrade documentation.

documentation for more information.

Options Page

Select PeopleTools, Utilities, Administration, Convert Panels to Pages, Options to access the Options page.

Specify the options for the conversion process:

Convert Scrolls to Scroll Areas If you select this option, scroll-to-scroll area conversions take place for panels with scroll bars. If this is unchecked, no scroll-to-scroll area conversion takes place.

Convert Scroll Action Buttons to Scroll Areas Some scroll bars may exist with scroll action buttons that are already defined. This option determines whether these scrolls should be converted or ignored. If they are converted, the scroll action buttons are removed before the scroll bar is converted to a scroll area.

If you select this option, scrolls with scroll action buttons are converted. If this options is not checked, scrolls with scroll action buttons are ignored.

Panels with Level 1 Scrolls If you select this option, panels with level 1 scrolls are processed for scroll conversion.

Panels with Level 2 Scrolls If you select this option, panels with level 2 scrolls are processed for scroll conversion.

Panels with Level 3 Scrolls If you select this option, panels with level 3 scrolls are processed for scroll conversion.

Convert Level 1 Scrolls If you select this option, level 1 scrolls are converted to scroll areas.

Convert Level 2 Scrolls If you select this option, level 2 scrolls are converted to scroll areas.

Convert Level 3 Scrolls If you select this option, level 3 scrolls are converted to scroll areas.

Max # Scrolls This parameter is a general scroll count limit for scroll conversion processing. For example, if this is set to 5, any panel with more than five scrolls that are not invisible is ignored. This is a simple way of eliminating complex panels from automatic scroll conversion.

Apply Specific Page Size	<p>This option is used to define whether a specific size should be assigned to a panel. If you select this option, the panel size that is defined in the drop-down list box is applied to the panel. If this is unchecked, no changes are made to the panel size.</p> <hr/> <p>Note. Note. When you select a specific panel size, the panel size is applied to standard panels only (secondary panels and subpanels are not sized automatically).</p> <hr/>
Apply Default Style Sheet	<p>If you select this option, the style sheet that is associated with a panel is updated with a blank value, so that the panel's style sheet appears by default from PSOPTIONS.STYLESHEETNAME ('PSSTYLEDEF').</p>
Apply Frame/Horz/GrpBox Styles	<p>If you select this option, the conversion process looks for frames, group boxes, and horizontal rules that have no styles associated with them, and that appear to be associated with a specific scroll area by virtue of their position within a scroll area. It then assigns level-specific styles, based on the occurs level of the scroll area.</p>
Convert Frames to Horizontal	<p>Horizontal lines were a new page object for PeopleSoft 8. This option applies only for applications upgraded from releases previous to PeopleSoft 8. If you select this option, the conversion process looks for frames on the panel with upper and lower coordinates less than nine grid units apart. These frames are then converted to horizontal lines.</p>
Delete All Frames	<p>If you select this option, the process removes all frames on the converted panel.</p> <hr/> <p>Note. If Convert Frames to Horizontal and Delete All Frames are both checked, the conversion from frame to horizontal takes place first, then any remaining frames are deleted.</p> <hr/>
Turn On Grid 'Odd/Even Style'	<p>This applies to grids that are on a panel being converted. If you select this option, the conversion process determines if grids on the panel have their 'Odd/Even Style' turned on. If it is not turned on, the conversion process turns on this option.</p>
Turn On 'Show Prompt Button'	<p>This option applies to edit box fields that are not invisible and are not display-only. If you select this option, the conversion process turns on the Show Prompt Button option for edit box fields that have it turned off.</p>
Apply Component Defaults	<p>Used to apply standard defaults to component definitions. The defaults that are set are dependent on the Use characteristics of the component. See Application Designer, Component Properties/Use and Component Properties/Internet tabs.</p>
Turn Off 'Show Grid Lines'	<p>Turns off the Show Grid Lines option for grids that have it checked on.</p>
Language Code	<p>Enables you to convert panels whose language code differs from that in PSOPTIONS. Select a language code from the drop-down list box.</p>

Update Utilities

The Update utilities enable you to keep track of the PeopleSoft updates that you apply to the database.

Updates By Release Label

The release label refers to the official release name, such as PeopleTools 8.50.00

Updates By Update ID

The update ID refers to the patch or project name that you apply to the system. The update ID is typically the report ID for a TPRD incident.

Remote Database Connection

Use the Remote Database Connection page to set up remote databases for use with the Remote Data Access (RDA) feature. Select PeopleTools, Utilities, Administration, Remote Database Connection to access the Remote Database Access Management page.

Name	Enter the name of the remote database connection.
Database Type	Available types are Microsoft, DB2 (z/OS), DB2/UNIX, Sybase, Informix, Oracle, and Sybase.
Description	Enter a description of the remote database.
Server	Enter the server name where the remote database resides.
Database	Enter the remote database name.
Local Connect	One connection must be defined as the Local Connect for the current PeopleSoft instance (the local database). Check this to specify which database is the local.
DB Server Port	This value is automatically populated with a default value that is based on the database type. You may need to change this value depending upon the database server configuration.
User ID	Enter the user ID that is needed to connect to the remote database.
Password	Enter the password that is associated with the user ID.
Test Connection	Select this to test the remote database connection.
Connection Type	For Oracle database type only. TNS Names or Specific. TNS Names represent a preconfigured file (tnsnames.ora) that consists of previously defined database connection information. Enter Specific if you want to set up a database that does not already have a TNS entry defined.

TNS Entry For Oracle database type only.

Inf Svr Name For Informix database type only.

Security in Remote Databases

To ensure security and limit the risk of unauthorized access to databases, follow these recommendations:

- The remote system's database administrator should create a user with read-only access to the tables that may be accessed by other systems using PeopleSoft's RDA.

Use this restricted user ID and password in configuring a source RDA node.

- The local system's database administrator should create a user with insert/update access to the RDA destination tables only.

Use this restricted user ID and password in configuring the target RDA node.

URL Maintenance

Select PeopleTools, Utilities, Administration, URLs to access the URL Maintenance page.

Use the URL Table to store URL addresses and to simplify specifying and updating URLs. URLs that are saved here can be referenced from page controls such as a push button/link. The associated URL can be either an internet or intranet link.

URL Maintenance

URL Identifier:	QE_NT4
*Description:	<input type="text" value="QE_NT4"/>
*URL:	<input type="text" value="https://ptsec01.peoplesoft.com:7002/servlets/iclientervlet/psNT4/?cmd=start&"/>
Comments:	<div><div></div><div></div></div>

URL Maintenance page

Description Users can search for URLs by description.

URL	Enter the entire URL.
Comments	This field can be used to make notations and comments and is not displayed elsewhere.

Adding New URLs

To add a new URL entry in the URL table:

1. Click the Add a New Value link.

A new page appears, prompting you to enter the URL Identifier. Enter the name that you want to use to identify the new URL address.

2. Select Add.
3. Enter the Description, URL, and Comment, if any.
4. Select Save.

You must save the page before you can add another URL, or update or display existing URL addresses.

5. Select Add to add another URL.

Viewing and Updating the URL Table

To update or display the URL table:

1. From the URL Maintenance search page, click Search.
2. Select the URL Identifier link that you want to update from the Search Results table.
3. Make changes to the page and save.

Adding Secure FTP (FTPS) URLs

To transfer a file in FTPS mode, PeopleCode expects the URL parameter to be in the form of a URLID, not a string. You create the URLID using the URL Maintenance page and assigning various properties to the URLID using the URL Properties page.

Note. If you intend to implement secure FTP, Oracle recommends you understand the details and behavior of the attachment PeopleCode constructs, such as AddAttachment.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Developer's Guide*, "Using File Attachments and PeopleCode."

URL Properties

URL Identifier DEMOFTPS

Certificate Alias BNG_PSFT632 ▼

Verify Host 2 ▼

Verify Peer Yes ▼

SSL Usage Level 3 - SSL Only ▼

Save Cancel

URL Properties page

1. On the URL Maintenance page, add a URL value containing *ftp://* or *ftps://* in the URL field.
2. Click Save.
3. Click the URL Properties link.

Note. This link is available only when a value containing at least "ftp" has been entered in the URL field.

4. Set the SSL usage values on the URL Properties page.

Certificate Alias	The Certificate Alias must be an alias name of a certificate stored in the database (using the PeopleTools Digital Certificates page).
Verify Host	<p>0: Do not verify the server for host name.</p> <p>1: Checks if there exists any value in the common name field in the server certificate. Does not verify if it matches with what the client specifies.</p> <p>2: (Default) Checks for a match with the host name in the URL with the common name or Subject Alternate field in the server certificate.</p>
Verify Peer	<p><i>False</i>: Do not verify the peer.</p> <p><i>True</i>: (Default) Verify the peer by authenticating the certificate sent by the server.</p>
SSL Usage Level	<p>0 - No SSL: No SSL will be used.</p> <p>1 - Try SSL: Try using SSL, but proceed as normal otherwise.</p> <p>2 - SSL for Control: Require SSL for the control connection.</p> <p>3 - SSL Only: (Default) Require SSL for all communication.</p>

5. Click Save.

Copy File Attachments

Select PeopleTools, Utilities, Administration, Copy File Attachments to access the Copy File Archive page.

Enables you to manage the file attachments that are stored in the database.

Copy File Archive

PeopleTools Copy Attachments

Transfer File Attachments

Source:

Destination:

Copy Files

Description:FTP://YourFTPUser:YourFTPPassword@YourComputerName/YourDirectoryPath

Description: RECORD://PSFILE_ATTDET

Delete Orphan File Attachments

Delete Orphan File Attachme

PeopleTools Copy Attachments page

Transfer File Attachments

Source	When you want to copy the file attachment archive from one location to another, enter the record or directory where the files are currently stored.
Destination	Enter the record or directory where you want to copy the file attachment archive.
Copy	Invokes the PeopleCode function (CopyAttachment) that copies the file attachment archive to another location. For example, you can copy from a file server to a database, and you can copy from a database to a file server.
<div><div>Note. For the file attachment functionality, in specifying the URL for the FTP server, the FTP server's machine name can be more than 30 characters. The length of the full URL is limited to 120 characters.</div></div>	

Removing Orphan File Attachments from the Database

The accumulation of file attachments can consume a significant amount of disk space. Therefore, on a regular basis, it is recommend that you make sure that orphaned file attachments are deleted from the database, reclaiming disk space being used unnecessarily. Click the Delete Orphan File Attachments button to complete this task. This button invokes the CleanAttachments PeopleCode function.

Warning! There is no way to roll back changes made by the CleanAttachments function. Review the CleanAttachments documentation in the PeopleCode Language Reference PeopleBook describing the behavior of CleanAttachments in order to appropriately anticipate how it will behave. Oracle suggests that you perform a database backup before invoking this functionality.

See *Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference*, "PeopleCode Built-in Functions," CleanAttachments

Query Administration

System administrators can use Query Administration to monitor query performance and usage. Some of the conditions that you can monitor include average runtime, number of times run, and the dates last run. Using a predefined search, you can also select queries to review and report on.

See Also

Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Query, "Query Administration"

Sync ID Utilities

The Sync ID Utilities are used exclusively with PeopleSoft Mobile Applications technology.

Important! PeopleSoft Mobile Agent is a deprecated product. These pages currently exist for backward compatibility only.

nVision Report Request Admin

See *Enterprise PeopleTools 8.50 PeopleBook: PS/nVision*, "Setting Up PS/nVision Security," Securing and Sharing Report Requests in PIA.

Analytic Server Administration

See *Enterprise PeopleTools 8.50 PeopleBook: Analytic Calculation Engine*, "Managing Analytic Servers," Administering Analytic Servers.

Upgrade Conversion

Defines upgrade drivers, providing details regarding Application Engine program, section, group, and calling sequence. If you need to specify any of these values, your upgrade documentation will provide the details.

See your upgrade documentation for more information.

Analytic Model Viewer

See *Enterprise PeopleTools 8.50 PeopleBook: Analytic Calculation Engine*, "Viewing and Debugging Analytic Models," Viewing Analytic Model Properties.

Analytic Instance Load/Unload

See *Enterprise PeopleTools 8.50 PeopleBook: Analytic Calculation Engine*, "Managing Analytic Servers," Loading and Unloading Analytic Instances.

Analytic Instance Create/Del/Copy/

See *Enterprise PeopleTools 8.50 PeopleBook: Analytic Calculation Engine*, "Managing Analytic Servers," Creating, Deleting, and Copying Analytic Instances.

Pre-Load Cache Utilities

See Chapter 5, "Using PSADMIN Menus," Configuring an Application Server Domain to Preload Non-Shared Cache, page 67.

Gather Utility

The Gather utility facilitates communications between PeopleSoft and the customer on technical questions or issues. The Global Support Center (GSC) directs the customer to the Gather Utility when problems arise. Customers can also use a self-service website to run this utility and send in relevant information about their problems or issues.

Using a simple command line interface, the Gather utility is a small Java application that can run on any platform to collect various files from the following environments:

- Application Server.
- Web Server.
- Any additional files that the user chooses (SQL Trace files, PeopleCode Trace Files, and so on).

The collected files are placed in a single jar file with psft.jar as the default name, in the temp directory. Subsequently, these files are sent to PeopleSoft.

Note. For this utility to work, the supported version of Java (JRE) must be installed on the target machine.

Getting Started

The following files reside in the starting directory:

- `Gather.class`: The main Java class file
- `Helper.class`: This class file is called by `Gather.class`
- `Runnit.bat`: A MS-DOS batch file that is used by Windows users.

UNIX users have to run the Gather utility manually.

- `Vars.sh`: a UNIX shell script.

Gather calls this automatically if the UNIX operating system is detected.

Windows Users

The following steps are used for Windows:

1. Make sure that you have the `PS_HOME` environment variable set.

This saves the user from having to type it in.

2. Go to `PS_HOME\utility`.
3. Type `runnit`.
4. Follow the directions that are on the screen.

UNIX Users

Use the following steps for UNIX:

1. At a command prompt, run the following command where PeopleSoft is installed:

```
../psconfig.sh
```

2. Go to the `PS_HOME/utility` directory.
3. Change permissions for all files:

```
chmod 777 *.*
```

4. Enter the following to start the utility:

```
java -cp .:$CLASSPATH Gather
```

Note. UNIX is case-sensitive – Gather is spelled with a capital G.

5. Follow the instructions that are on the screen.

Environmental Data

On Windows, both the set and netstat commands are invoked with the results copied to a file that is collected. On UNIX, the same thing is done with the env command.

Application Server Data

The following files are collected from the Application Server:

- PSAPPSRV.CFG
- PSAPPSRV.UBB
- LOGS/*.*—this usually includes all application serv/tuxedo logs, dump, and replay files.

This includes all subdirectories under LOGS .

Web Server Data

The gather utility collects numerous files (log files, configuration files, and so on) from each of the supported web servers. If an analyst only asks for a specific file, send that, but make sure to keep the other collected files in case they are needed.

Additional Files

There is always a need to include files that are not on the above list. These can include PeopleCode Trace files, SQL Trace files, SQL output, and so forth. The command line interface allows you to specify any file that you want to be included in the jar file.

QAS Administration

See *Enterprise PeopleTools 8.50 PeopleBook: Reporting Web Services*, "Accessing PeopleSoft Application Tables," Using QAS Administration.

Oracle Resource Management

See *Enterprise PeopleTools 8.50 PeopleBook: Data Management*, "Administering PeopleSoft Databases on Oracle," Working With Oracle Consumer Groups.

Using Audit Utilities

This section covers the utilities that are used for auditing the system's integrity.

This section discusses how to:

- Use the Record Cross Reference component.
- Perform a system audit.

- Perform database level auditing.

Using the Record Cross Reference Component

Select PeopleTools, Utilities, Audit, Record Cross Reference.

You use the Record Cross Reference component (XREF_PANEL_01) to view where a record is used throughout the application. There are two pages in this page group:

- Pages, Views, Search Records.
- Prompts, Defaults, PeopleCode.

Pages, Views, Search Records

This is a read-only page that shows which Projects, Menus, Pages, and Objects reference a particular record:

Pages, Views, Search Records

Prompts, Defaults, PeopleCode

Record: QEDERVD_ED_SVCS

Referenced in Project(s):

Customize | Find | View All | |

First 1-4 of 4 Last

Project
QEDM084
QEDM084
QEDMOALL
QEDM084

Used as a Search Record on:

Customize | Find | View All | |

First 1 of 1 Last

Menu Name	Item Name	Component

Referenced on Page(s):

Customize | Find | View All | |

First 1 of 1 Last

Page Name
QEORDER_TABLE_WF

Pages, Views, Search Records page

Prompts, Defaults, PeopleCode

On the Prompts, Defaults, PeopleCode page, the group boxes list the components that refer to the record.

Pages, Views, Search Records

Prompts, Defaults, PeopleCode

Record: QEDERVD_ED_SVCS

Used as an Edit Table on:

Customize | Find | View All | | First 1 of 1 Last

Base Record	Field Name

Used as a Default Table in:

Customize | Find | View All | | First 1 of 1 Last

Base Record	Field Name

PeopleCode with Fields from this Record

Customize | Find | View All | | First 1-8 of 8 Last

PeopleCode Reference Name	PeopleCode Fieldname	PeopleCode Recname	PeopleCode Type
QE_TOTAL_PRICE	QE_PRICE	QEORDER_DTL	FieldChange
QE_GRAND_TOTAL	QE_PRICE	QEORDER_DTL	FieldChange
QE_GRAND_TOTAL	QE_PRICE	QEORDER_DTL	RowInit
QE_GRAND_TOTAL	QE_QTY	QEORDER_DTL	FieldChange
QE_GRAND_TOTAL	QE_PRICE	QEORDER_DTL	RowDelete
QE_TOTAL_PRICE	QE_QTY	QEORDER_DTL	FieldChange
QE_TOTAL_PRICE	QE_PRICE	QEORDER_DTL	RowDelete
QE_TOTAL_PRICE	QE_PRICE	QEORDER_DTL	RowInit

PeopleCode referring to this

Customize | Find | View All | | First 1 of 1 Last

PeopleCode Recname	PeopleCode Fieldname	PeopleCode Type

Prompts, Defaults, PeopleCode page

Used as an Edit Table on Lists pages that use the record as edit table.

Used as a Default Table in Lists pages that use the record as a default table.

PeopleCode with Fields from this Record Shows where fields from this record are used in PeopleCode.

PeopleCode referring to this Shows all PeopleCode that references this record.

Performing a System Audit

The System Audit (SYSAUDIT) utility is documented in the Data Management PeopleBook.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Data Management, "Ensuring Data Integrity," Running SYSAUDIT

Performing Database Level Auditing

This utility is used to support database level auditing features, and is documented in the Data Management PeopleBook.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Data Management, "Employing Database Level Auditing"

Using Debug Utilities

This section discusses how to:

- Use the PeopleTools Test Utilities page.
- Use the Trace PeopleCode utility.
- Use the Trace SQL utility.

Note. The Trace Page / Trace Panel page is no longer actively used or maintained.

Using the PeopleTools Test Utilities Page

Select PeopleTools, Utilities, Debug, PeopleTools Test Utilities to access the PeopleTools Test Utilities page:

PeopleTools Test Utilities

Remote Call Test <input type="button" value="Test"/>	Interlink Test <input type="button" value="Interlink Test"/>
PeopleCode/Java Test	
Delivered Class File <input type="button" value="Test 1"/>	External Class File <input type="button" value="Test 2"/>
<input type="text"/>	
File Attachment Test	
FTP Site: <input type="text"/>	
Example: FTP://YourFTPUser:YourFTPPassword@YourComputerName/YourDirectoryPath	
Attached File:	<input type="button" value="Attach"/>

PeopleTools Test Utilities page

Remote Call Test	You use the Remote Call Test button to test the Remote Call configuration.
Delivered Class File	The Delivered Class File button tests Java-PeopleCode integration. It tests to see that Java is being executed correctly through PeopleCode. It tests a Java class shipped and used by PeopleTools.
External Class File	The External Class File button tests Java PeopleCode integration. It tests a Java class that is similar to what a customer may create.
FTP Site	<p>Enter the full path and password for the test file. For example:</p> <p>FTP://YourFTPUser:YourFTPPassword@YourComputerName/YourDirectory/Path</p> <hr/> <p>Note. For the file attachment functionality, in specifying the URL for the FTP server, the FTP server's machine name can be more than 30 characters. The length of the full URL is limited to 120 characters.</p> <hr/>
Attached File	Click this button to attach the file whose path you indicate in the FTP Site.

Replay Appserver Crash

See [Chapter 4, "Using the PSADMIN Utility," Configuring the Application Server to Handle Cache Files and Replay Files, page 53.](#)

Using the Trace PeopleCode Utility

The Trace PeopleCode utility is discussed elsewhere in this PeopleBook.

See Also

Chapter 12, "Tracing, Logging, and Debugging," Setting Up the PeopleCode Debugger, page 285

Chapter 12, "Tracing, Logging, and Debugging," Configuring PeopleCode Trace, page 288

Using the Trace SQL Utility

The Trace SQL utility is discussed elsewhere in this PeopleBook.

See Also

Chapter 12, "Tracing, Logging, and Debugging," Configuring SQL Trace, page 290

Using International Utilities

The following sections cover the utilities that you use in globalization efforts.

This section discusses how to:

- Set international preferences.
- Set process field size.
- Administer time zones.
- Manage languages.

Setting International Preferences

Select PeopleTools, Utilities, International, Preferences.

Used to override the language that you select when you sign in to the database.

Language Preference	Use the International Preferences page to temporarily change the session's language preference that was specified during signon. This change lasts until you exit the PeopleSoft session or change the language preference again. Only languages that are enabled on the Languages page are available for selection.
----------------------------	--

See Also

Enterprise PeopleTools 8.50 PeopleBook: Global Technology, "Controlling International Preferences," Changing the Session Language While Signed In

Setting Process Field Size

Select PeopleTools, Utilities, International, Process Field Size.

If you process currency values that require large numbers, such as Italian lira, that require fields longer than those that are included in the standard application, you can use the International Field Size page to expand amount fields throughout the application.

After you create or select a run control ID, set the appropriate lengths for a list of fields, then click the Run button to launch the batch program that performs the field size changes.

Field Name Use the Browse button to select the field name.

Current Field Size This is a read-only field indicating the current field size as stored in PSDBFIELDS.

Field Size - International Enter the field size to expand (or contract) the field size for foreign fields.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Global Technology, "Controlling Currency Display Format," Resizing Currency Fields by Using the International Field Size Utility

Administering Time Zones

This utility is extensively documented in the PeopleTools Global Technology PeopleBook.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Global Technology, "Setting and Maintaining Time Zones"

Managing Languages

Select PeopleTools, Utilities, International, Languages to access the Manage Installed Languages page.

Manage Installed Languages

Language											
	*Language Code	Language	Enabled	Directionality	*ISO Locale	*Default Character Set	*Verity Locale Mapping	Spell Check Language	*Windows Character Set	*Verity Character Set	
1	ARA	Arabic	<input type="checkbox"/>	Right-to-Left	ar	ISO_8859-6	uni	US and UK English	CP1256	UTF-8	+ -
2	BUL	Bulgarian	<input type="checkbox"/>	Left-to-Right	bg	ISO_8859-5	uni	US and UK English	CP1251	UTF-8	+ -
3	CFR	Canadian French	<input type="checkbox"/>	Left-to-Right	fr-ca	ISO_8859-1	frenchv	French	CP1252	CP1252	+ -
4	CRO	Croatian	<input type="checkbox"/>	Left-to-Right	hr	ISO_8859-2	uni	US and UK English	CP1250	UTF-8	+ -
5	CZE	Czech	<input type="checkbox"/>	Left-to-Right	cs	ISO_8859-2	uni	Czech	CP1250	UTF-8	+ -
6	DAN	Danish	<input type="checkbox"/>	Left-to-Right	da	ISO_8859-1	danishv	Danish	CP1252	CP1252	+ -
7	DUT	Dutch	<input checked="" type="checkbox"/>	Left-to-Right	nl	ISO_8859-1	dutchv	Dutch	CP1252	CP1252	+ -
8	ENG	English	<input checked="" type="checkbox"/>	Left-to-Right	en	ISO_8859-1	englishv	US and UK English	CP1252	CP1252	+ -
9	ESP	Spanish	<input type="checkbox"/>	Left-to-Right	es	ISO_8859-1	spanishv	Spanish	CP1252	CP1252	+ -
10	FIN	Finnish	<input type="checkbox"/>	Left-to-Right	fi	ISO_8859-1	finnishv	Finnish	CP1252	CP1252	+ -
11	FRA	French	<input type="checkbox"/>	Left-to-Right	fr	ISO_8859-1	frenchv	French	CP1252	CP1252	+ -
12	GER	German	<input type="checkbox"/>	Left-to-Right	de	ISO_8859-1	germanv	German (new)	CP1252	CP1252	+ -
13	GRK	Greek	<input type="checkbox"/>	Left-to-Right	el	ISO_8859-7	uni	Greek	CP1253	UTF-8	+ -
14	HEB	Hebrew	<input type="checkbox"/>	Right-to-Left	he	ISO_8859-8	uni	US and UK English	CP1255	UTF-8	+ -
15	HUN	Hungarian	<input type="checkbox"/>	Left-to-Right	hu	ISO_8859-2	uni	Hungarian	CP1250	UTF-8	+ -
16	ITA	Italian	<input type="checkbox"/>	Left-to-Right	it	ISO_8859-1	italianv	Italian	CP1252	CP1252	+ -
17	JPN	Japanese	<input type="checkbox"/>	Left-to-Right	ja	Shift_JIS	japanb	US and UK English	CP932	Shift_JIS	+ -
18	KOR	Korean	<input type="checkbox"/>	Left-to-Right	ko	CP949	koreab	US and UK English	CP949	CP949	+ -
19	MAY	Bahasa Malay	<input type="checkbox"/>	Left-to-Right	ms	ISO_8859-1	uni	US and UK English	CP1252	UTF-8	+ -

Manage Installed Languages page

Use this page as a central utility to manage language information for the currently enabled languages.

Language Code

Use the search prompt to select the PeopleSoft language code from the PSXLATITEM table. The language description appears to the right of the code field.

Enabled

When you select this check box, PeopleSoft Internet Architecture enables you to log in with the language.

ISO Locale

Use the search prompt to select the ISO locale code from the PSLOCALEDEFN table. Consists of an ISO 639 language code, optionally followed by an ISO 3166 country code.

Default Character Set

Use the search prompt to select the character set from the PSCHARSETS table. Determines the default encoding for input and output files.

Verity Locale Mapping

Select the Verity locale code from the PSVERITYLOCALE table. Determines the locale to use for building search collections and searching data.

Spell Check Language

Select the spell check language from the PSXLATITEM table. This enables you to select the language of the spell check dictionary that is associated with a given language code.

Windows Character Set

Select the Microsoft codepage that is associated with the given language. This defines the codepage to use with certain Microsoft applications.

Verity Character Set Select the character set that the Verity engine uses for its internal encoding in the given language. You should not modify the value in this field under normal circumstances.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Global Technology, "Adding New Languages," Managing Languages in the PSLANGUAGES Table

Using Optimization Utilities

The Optimization utilities are documented extensively in the Optimization Framework PeopleBook.

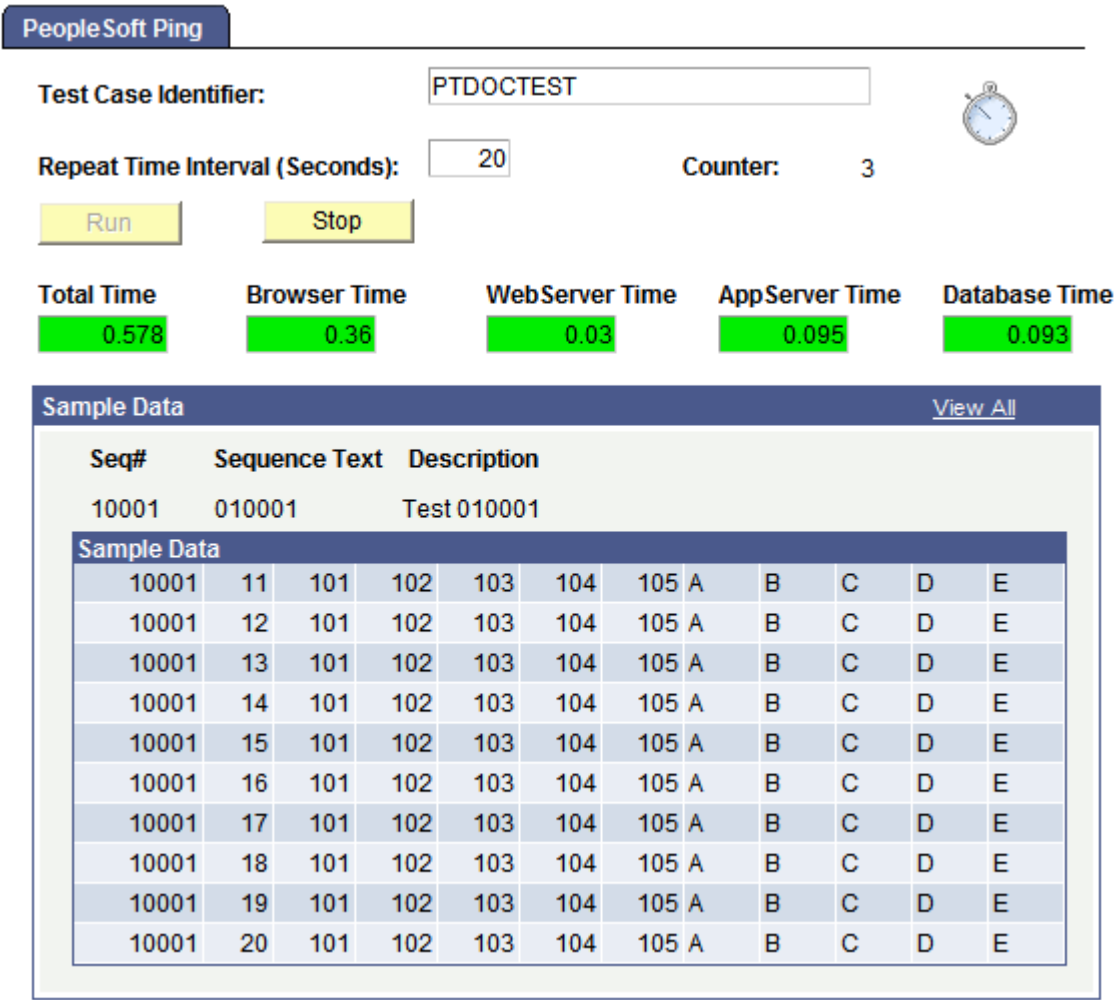
See Also

Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Optimization Framework, "Designing Analytic Type Definitions"

Using PeopleSoft Ping

The PeopleSoft Ping utility collects timestamps by sending a specific page to different tiers of the PeopleSoft system, starting at the browser, then going to the web server, the application server, the database and back. The timestamps that are collected are total time elapsed for the round trip, and arrival and departure time at each of the tiers.

To use the PeopleSoft Ping feature, select PeopleTools, Utilities, PeopleSoft Ping.



PeopleSoft Ping page

Enter a Test Case Identifier to uniquely group each set of metrics. For Repeat Time Interval , enter an increment for the ping to run. To avoid creating unnecessary traffic and overhead to the PeopleSoft system, set the Repeat Time Interval to a relatively high value, such as 600 to 1800 seconds, during normal operations. You may need to increase the Session Timeout value accordingly.

PeopleSoft Ping Chart

Select PeopleTools, Utilities, PeopleSoft Ping, PeopleSoft Ping Chart.

PeopleSoft Ping includes a charting utility to zoom in to a specific time interval from the ping test.

You can change the displayed time interval to a subset of the full ping test period. Edit the start time and end time values, and click Redraw to refresh the chart display with the new time interval.

Click Query Viewer to query the database for the ping data. A new browser window opens, displaying the ping data for the full test period in a table.

PeopleSoft Ping Delete

To delete a ping page test case:

1. Select PeopleTools, Utilities, PeopleSoft Ping, PeopleSoft Ping Delete.

The Delete page lists the current test case identifiers.

2. Select the check box next to the identifier(s) you want to delete.
3. Click Delete.

PeopleSoft Ping Options

Select PeopleTools, Utilities, PeopleSoft Ping, PeopleSoft Ping Options.

The PeopleSoft Ping Options page enables you to set targets for each tier as well as overall completion time. If the ping process exceeds your targets, this affects the color-coding on the PeopleSoft Ping interface, using green, yellow, and red. Green is any time under the yellow and red targets, yellow is any time over the yellow target yet under red, and red is anything exceeding the red target. The metrics you enter depend on the typical conditions and expectations at your site.

Chapter 12

Tracing, Logging, and Debugging

This chapter discusses how to:

- Set up the PeopleCode Debugger.
- Enable PeopleCode tracing.
- Enable SQL tracing.
- Enable IDDA logging.

See Also

Chapter 6, "Setting Application Server Domain Parameters," Trace Options, page 90

Chapter 10, "Using PeopleSoft Configuration Manager," Specifying Trace Settings, page 215

Setting Up the PeopleCode Debugger

This section discusses how to:

- Debug for a two-tier connection.
- Debug for a three-tier connection.
- Use the PeopleCode Debugger.

Note. PeopleCode debugging is not supported on z/OS.

You can debug the PeopleCode program configurations of a two-tier connection to the database or a three-tier connection to the database.

Note. When you debug PeopleCode with an application server, Application Designer should be run in three-tier mode. PeopleCode debugging by using a two-tier PSIDE and an application server is not supported on multi-homed (multiple Internet Protocol address) workstations.

Debugging for a Two-Tier Connection

Debugging in two-tier connections involves connecting directly to the database, not through the application server. Use this method to debug two-tier Windows applications.

Note. By default, the port number that the PeopleCode debugger uses is 9500. Unless this port number is being used by another application, you do not need to alter any environment settings, and after you sign on to the database, you are able to debug PeopleCode.

If you need to change the PeopleCode Debugger port settings, complete the following procedure.

To change the default PSDBGSRV listener port number:

1. Open PeopleSoft Configuration Manager.
2. Select the Trace tab.
3. Locate the PeopleCode Debugger section, and make sure that the default value for the Local PSDBGSRV Listener Port is suitable for the system.

For example, make sure that no other applications are configured to listen on the default port number (9500). If so, you must assign a port number that is not being used.

Note. If you're using a personal firewall, you must configure it to enable data packets to flow through the PSDBGSRV listener port. If you can't configure your firewall appropriately, you must shut it down while performing PeopleCode debugging.

Debugging for a Three-Tier Connection

Use three-tier debugging to debug three-tier Windows applications and PeopleSoft Internet Architecture (PIA) applications. For three-tier debugging, use PSADMIN to ensure that the following items are set:

- The appropriate PSDBGSRV listener port is specified in the PeopleCode Debugger section of PSADMIN.
- At least two PSAPPSRV processes are configured to boot in the domain with the service timeout parameter set to zero.
- Enter y for yes at the Enable PSDBGSRV Server Process prompt at the end of the PSADMIN interface.

Debugging on a Multi-Homed System

If you're debugging on a multi-homed (multiple IP address) system, you must explicitly specify an IP address in the Workstation Listener section of the PSADMIN configuration, rather than %PS_MACH%. The address you specify must be one by which the application server identifies the machine on which you're doing the debugging. This ensures that the workstation listener monitors requests from the correct location.

See [Chapter 6, "Setting Application Server Domain Parameters," Workstation Listener Options, page 81.](#)

Setting the PSDBGSRV Listener Port

In the PeopleCode Debugger section of PSADMIN make sure that the value that is assigned to the PSDBGSRV listener port is not already in use by another application or listener on the application server. The default value is 9500. If the default is not acceptable, assign a suitable value to the parameter. If it is acceptable, no changes are required.

For example,

```
Values for config section - PeopleCode Debugger
  PSDBGSRV Listener Port=9500
```

Do you want to change any values (y/n)? [n]:

Consider the following when debugging PeopleCode:

- If multiple application server domains are running on a single, physical machine, each domain needs to use different debugging port numbers.

Otherwise, there is contention for the PSDBGSRV listener port value. This is the same principle that requires each application server domain on a server to have unique workstation listener port numbers.

- When you are not debugging, turn off (set to 0) the Enable Debugging parameter.

The debugging mode results in an unavoidable amount of overhead, which can degrade performance.

- Regarding performance, do not perform debugging on a production domain.

Debugging should be performed on a designated testing domain only.

Enabling Multiple PSAPPSRV Server Processes

The minimum requirements for PeopleCode debugging are:

- Two PSAPPSRV server processes are configured to boot in the domain.
- The Service Timeout value in the PSAPPSRV configuration section must be set to 0.

For the debugger to work, it has to run in parallel with the application that it's debugging. Suppose that the domain has only one PSAPPSRV server process running. In this case, the PSAPPSRV can process the requests of only one component at a time, and therefore debugging is not possible. Debugging involves two items, the debugger (PSDBGSRV) and the PSAPPSRV server process that is running the application PeopleCode.

Provided that you have two PSAPPSRV server processes configured; one PSAPPSRV handles the debugger program, while the other handles the application that you're stepping through with the debugger. In this case, the two programs run in parallel, which enables interactive debugging.

The configuration templates that PeopleSoft delivers all have at least two PSAPPSRV processes. However, if you are using a custom template, make sure that you configure the domain to start two PSAPPSRV processes prior to debugging. To do this, in PSADMIN set the Min Instances parameter in the PSAPPSRV section to 2.

The following example shows a sample PSAPPSRV section properly configured for debugging PeopleCode:

```
Values for config section - PSAPPSRV
  Min Instances=2
  Max Instances=2
  Service Timeout=0
  Recycle Count=0
  Allowed Consec Service Failures=0
  Max Fetch Size=5000
```

Do you want to change any values (y/n)? [n]:

When configuring the PeopleCode debugger:

- PeopleSoft recommends using the Developer configuration template because this template, by default, provides two PSAPPSRV server processes and has service timeout set to zero.

- PeopleSoft recommends using a simple configuration where you are assured that the server that Application Designer connects to is the same server that the application you are debugging is running on.

Note. If you do not set the settings for PSAPPSRV correctly (at least two PSAPPSRV processes), PSADMIN automatically sets these values to comply with the minimum requirements when you enable PeopleCode Debugging (as discussed in the next section).

Note. When you enable the PeopleCode Debugger (PSDBGSRV), service timeout settings for server processes are set to zero (0) by the system, overriding any previous settings you may have made in PSADMIN. For example, if the service timeout settings for the PSAPPSRV service process are set to 300 prior to enabling the debugger, after you enable the debugger, the service timeout value will be zero (0). After using the debugger, you need to reset your service timeout settings to the desired values. Before enabling the debugger, it is recommended that you make a backup of your configuration file or make note of your service timeout settings.

Requesting a PSDBGSRV Server Process

After you specify the settings by using PSADMIN, the system prompts you with a series of options, such as setting up messaging server processes, enabling Jolt, and so on.

When you're prompted to enable the PSDBGSRV, enter y. Y appears in the Developer template by default.

Using the PeopleCode Debugger

After the system is configured properly, using the PeopleCode debugger is just a matter of signing on to the PeopleSoft system and entering the PeopleCode Debugger mode in Application Designer.

Note. You must use a unique user ID when you're performing PeopleCode debugging, as opposed to using a shared user ID, such as those that PeopleSoft delivers, for example QEDMO, PS, or VP1. Shared IDs are likely to be used by others that are connecting to the same test database, which can affect debugging.

Configuring PeopleCode Trace

Select PeopleTools, Utilities, Debug, Trace PeopleCode to access the Trace PeopleCode page.

You use this page to change the PeopleCode tracing options while online. This page does not affect trace options that are set in PeopleSoft Configuration Manager. Use Trace PeopleCode to create a file displaying information about PeopleCode programs processed from the time that you start the trace.

Trace Evaluator Instructions	Select to show a line-by-line trace of the program
List Evaluator Program	Select to show the code of the PeopleCode program.
Show Assignments to Variables	Select to show variable assignments.
Show Fetched Values	Select to show values that are from PeopleCode Fetch call.

- Show Stack** Select to display the PeopleCode evaluator's stack after each PeopleCode (internal) instruction.
- Trace Start of Programs** Select to show the starting and ending points of the program.
- Trace External Function Calls** Select to show calls to application written functions.
- Trace Internal Function Calls** Select to show the calls to PeopleTools built-in function calls.
- Show Parameter Values** Select to show function parameter values.
- Show Return Parameter Values** Select to show function return parameter values.
- Show Each** Select to trace each statement in the program.

Note. The Trace PeopleCode Utility decreases system performance because of the overhead that occurs during the monitoring and recording of all PeopleCode actions.

The check boxes on this page correspond to the options on the Trace tab in Configuration Manager. However, the selections that appear on this page do not necessarily reflect those that are made in Configuration Manager. While the Configuration Manager settings are stored in the Windows registry and used at each signon, the settings in the Utilities page only apply to the current online session, and, once set, they override the Configuration Manager's settings.

The benefit of using this page to control PeopleCode tracing is that you can turn it on and off without having to restart PeopleTools, and without resetting the Configuration Manager settings. Keep in mind, though, your selections are not enabled until you save the page.

To enable/disable PeopleCode tracing while on line

1. Select PeopleTools, Utilities, Debug, Trace PeopleCode.

The Trace PeopleCode page appears.

2. Select/deselect the desired Options.
3. Save the page.

If you selected any of the check boxes, the system starts writing to the trace file.

See Also

Chapter 10, "Using PeopleSoft Configuration Manager," Specifying Trace Settings, page 215

Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference, "PeopleCode Built-in Functions," SetTracePC

Configuring SQL Trace

Select PeopleTools, Utilities, Debug, Trace SQL to access the Trace SQL page.

You use this page to change the SQL tracing options while you're online. Your Configuration Manager settings are not affected:

Trace SQL Statement	Select to show the SQL statement.
Trace SQL Bind	Select to show bind values for SQL statements that have parameter markers.
Trace SQL Cursor	Select to show connect, disconnect, commit and rollback calls.
Trace SQL Fetch	Select to show fetch call for Select Statement.
Trace SQL API	Select to show other API calls (Execute, Describe, and so on.)
Trace SQL Set Select Buffer	Select to show Binds for Select columns.
Trace SQL -- Database Level	Select to specify low-level tracing at the database API (ODBC, ct-lib, and so on.)
Trace SQL -- Manager Level	Select to show calls for Cache calls.

The check boxes on the Trace SQL page correspond to options on the Trace tab in the Configuration Manager. However, the selections that appear on this page do not necessarily reflect those that are made in the Configuration Manager. The displayed page selections are not enabled until you save the page.

To enable or disable SQL tracing while online:

1. Select or deselect the desired trace options.
2. Save the page.

If you select any of the check boxes, the system starts writing to the trace file.

See Also

[Chapter 10, "Using PeopleSoft Configuration Manager," Specifying Trace Settings, page 215](#)

Enterprise PeopleTools 8.50 PeopleBook: PeopleCode Language Reference, "PeopleCode Built-in Functions," SetTraceSQL

Enabling IDDA Logging

This section contains an overview and discusses how to:

- Enable IDDA logging.
- Work with IDDA functional categories.
- Configure logging options.
- Viewing log results.

Understanding IDDA Logging

System administrators typically use numerous monitoring and logging utilities to diagnose system issues and internet applications. Such logging utilities include:

- TCPMON
- ieHttpHeaders
- Access log
- Heap dump
- Thread dump

All of these utilities provide different information, but each helps an administrator gain insight and detailed information related to specific system behavior. PeopleSoft provides a variety of logging and tracing mechanisms as well, enabling you to gather vital information at various levels of the architecture, including database server, application server, web server, and so on.

The PeopleSoft Instrumented Development Diagnostic Aid (IDDA) logger, enables you to gather specific information about various areas within the PeopleSoft Internet Architecture and PeopleSoft Portal, including:

- PeopleSoft Internet Architecture processing.
- Integration Broker.
- Reporting, Report Repository.
- Portal.
- Caching.
- Security, authentication.
- Performance Monitor.
- WSRP.
- Jolt.

Typically, administrators only run IDDA traces when instructed to do so by Oracle support contacts, looking for specific information. However, it can also be a useful troubleshooting tool.

Note. When ever tracing or logging is enabled, you should always expect a certain degree of performance degradation as the system incorporates the overhead involved with logging. Disable logging when you finish troubleshooting.

Enabling IDDA Logging

To enable IDDA logging:

1. Select PeopleTools, Web Profile, Web Profile Configuration, and open the current web profile.
2. Select the Custom Properties page.
3. Add a new row, and enter these values:

Column	Value
Property Name	<i>IDDA</i>
Validation Type	<i>Number</i>
Property Value	The sum of the bit values of the functional area(s) you want to log. For example, if you wanted to log PIA (1) and Portal (8), you enter 9.

4. Click Save.
5. Restart the PeopleSoft site.

Working with IDDA Functional Categories

The IDDA logger, gathers information for a wide range of technology within the PeopleSoft Internet Architecture. Depending on the needs of your troubleshooting, you can select different areas to log.

The different areas of technology are referred to as functional categories in the context of the IDDA logger. Each functional category is assigned a bit value (in a 32-bit space). When you enable IDDA logging, you enter specific bit values *or* the sum of the bit values of different areas.

The IDDA functional categories are:

Bit Value	Functional Category
1	PeopleSoft Internet Architecture
2	Integration Broker
4	Report repository
8	Portal
16	Web server caching
32	Unassigned

Bit Value	Functional Category
64	Authentication
128	Performance Monitor
256	Web Services for Remote Portlets (WSRP)
512	Jolt

The type of information included in the log message differs per category. The log message is free format and can display any information that helps troubleshooting for that particular area, such as error codes, exception messages, and so on.

Configuring Logging Options

Once you've enabled IDDA logging, you can modify configuration options in the logging.properties file, which you can find in these locations:

Web Server	Location
Oracle WebLogic	<i>PS_HOME</i> /webserv/ <i>domain name</i> /applications/peoplesoft
IBM WebSphere	<i>PS_HOME</i> /webserv/ <i>profile name</i> /installedApps/ <i>node&cell name</i> /peoplesoft.ear/

The relevant configuration properties are:

Property	Description
.level	<p>Sets the global logging level.</p> <ul style="list-style-type: none"> • OFF: Disables all IDDA logging. • SEVERE: Displays server issues, such as premature termination and failure. • WARNING: Displays less severe issues, such as configuration issues. • INFO: Displays basic operational information, such as starting and stopping. • FINE, FINER, FINEST: Displays internal non-critical informational messages. • ALL: Enables all logging levels. <p>Default value is <i>INFO</i>.</p> <p>WARNING and SEVERE messages are always logged, unless IDDA logging is set to OFF.</p> <p>INFO or FINE, FINER, FINEST messages are only logged if the IDDA value is greater than zero.</p>

Property	Description
java.util.logging.FileHandler.pattern	<p>Sets the naming style for the log file and the output directory location.</p> <p>Default value is: <code>./servers/PIA/logs/PIA_servlets%u.log</code></p> <p>If you are running a multi-server, distributed environment (for clustering and failover purposes), the default value for <code>java.util.logging.FileHandler.pattern</code> is not applicable. You must set this property to point to a valid location in order to collect logging messages for a particular server.</p> <p>Logging output is tab-delimited.</p>
java.util.logging.FileHandler.limit	<p>Limits the size of the output file in bytes. When set to 0, there is no size limit.</p> <p>Default value is 0.</p>
java.util.logging.FileHandler.count	<p>Sets the total number of log files. Default value is 5.</p> <p>If the value is greater than 1, the system writes to a rotating set of log files. When the file reaches a given size limit or the web server restarts, the system ends writing to the current file and begins writing to a new file. The system names each file in the sequence it is saved by adding "0", "1", "2", (and so on) into the base filename.</p>

Viewing IDDA Logging Output

The output log files contain:

- Machine header information
- Log message information

Working with Machine Header Information

Each log file displays the following machine environment information at the top of each log file.

Header Entry	Description
Timestamp	Date, time, time zone.
PeopleTools Release	PeopleTools version number.
os.name	Operating system.
os.version	Operating system version.
os.arch	Operating system architecture (such as x86 for Windows servers).
java.version	Java version number.

Header Entry	Description
java.vendor	Java vendor.
java.vm.info	Mode of the Java virtual machine (JVM), such as "compiled mode."
java.home	Java installation directory.
user.dir	The value of the user.dir property.
java.class.path	CLASSPATH setting on the server.
.level	Logging level, such as INFO, SEVERE, WARNING, and so on.
java.util.logging.FileHandler.pattern	Logging output directory and file naming convention.
Trace	Current IDDA functional group(s) being logged (integer reflecting the sum of the bit values assigned to each group).

Working with Log Message Information

Each entry in the log file contains this information.

Log Message Data	Description
Timestamp	Date, Time and Time zone. For example, 2/7/09 4:00:39 PM PST
Sequence	Tracks the sequential order of the messages. Starting at 1, the system increments by 1 per each log message.
Thread ID	Java thread ID.
Logging group	Bit value representing the functional grouping, as in 1 for PeopleSoft Internet Architecture.
Source class	Class name of where the message is logged. For example, <code>psft.pt8.auth.PSAuthenticator</code>
Source method	Method name of where the message is logged. For example, <code>SetCookie</code>
Log message	The actual log message.

Viewing Log Contents

The log file is a tab-delimited text file and can be opened in any standard text editor, such as Notepad or Textpad. Because the output is tab-delimited, you can also use a spreadsheet application, such as Microsoft Excel, for more efficient analysis. For example, viewing the output within a spreadsheet enables you to apply filters to columns and only view specific log messages, which can be helpful with large files.

Chapter 13

Working with Jolt Configuration Options

This chapter provides overview information and discusses how to:

- Configure Jolt failover and load balancing.
- Configure Jolt session pooling.
- Configure JRLY.
- Configure JRAD.
- Run Jolt Relay.

Configuring Jolt Failover and Load Balancing

This section discusses how to:

- Configure weighted load balancing.
- Configure Jolt failover.

Configuring Weighted Load Balancing

With weighted load balancing, you can set the "weight" of the load, or amount of requests, being directed to a particular server. Weight values are integers 1–10, with 1 being low and 10 being a heavy load. Servers that can handle extra work can take heavy loads, while servers that are either less powerful or are being used in other capacities can take lower loads. You specify weighted load balancing by modifying the server values in the `psserver` property in the `configuration.properties` file, using the following format.

```
psserver=Host1:Port1#Wt,Host2:Port2#Wt
```

For example,

```
psserver=appserver1:9000#3,appserver2:9010#1
```

In this case, `appserver1` would receive 3x more requests than `appserver2`.

Configuring Jolt Failover

You can also specify strict failover assignments with weighted load balancing, with the following options:

- Strict failover with weighted backup.
- Strict failover with sequential backup.

You add the failover string within curly brackets at the end of the server entry.

```
psserver=<host>:<port>#wt{failover servers}
```

With the failover string, you can set weighted backup by separating failover server with a comma (.).

```
psserver=Host1:Port1#Wt{Host3:Port3#Wt,Host4:Port4#Wt},Host2:Port2#Wt
```

In this case, Host 3 and Host 4 are failover servers when Host 1 is down. You can assign weighted load balancing to the backup servers just as you would a primary server.

You can also set a sequential backup with your failover string. To set sequential backup, you separate multiple backup servers using a semicolon (;).

```
psserver=Host1:Port1#Wt{Host3:Port3;Host4:Port4},Host2:Port2#Wt
```

In this case, the system assigns Host 4 the requests when both Hosts 1 and 3 are down.

Configuring Jolt Session Pooling

Jolt session pooling is enabled by default. Jolt session pooling enables web server connections to be shared between user sessions, which reduces the usage of system resources, such as threads and file descriptors.

You control session pooling by modifying the `joltPooling` property in the `configuration.properties` file per site.

```
joltPooling=true
```

To enable Jolt session pooling, set the property value to *true*, and to disable Jolt session pooling set the property value to *false*.

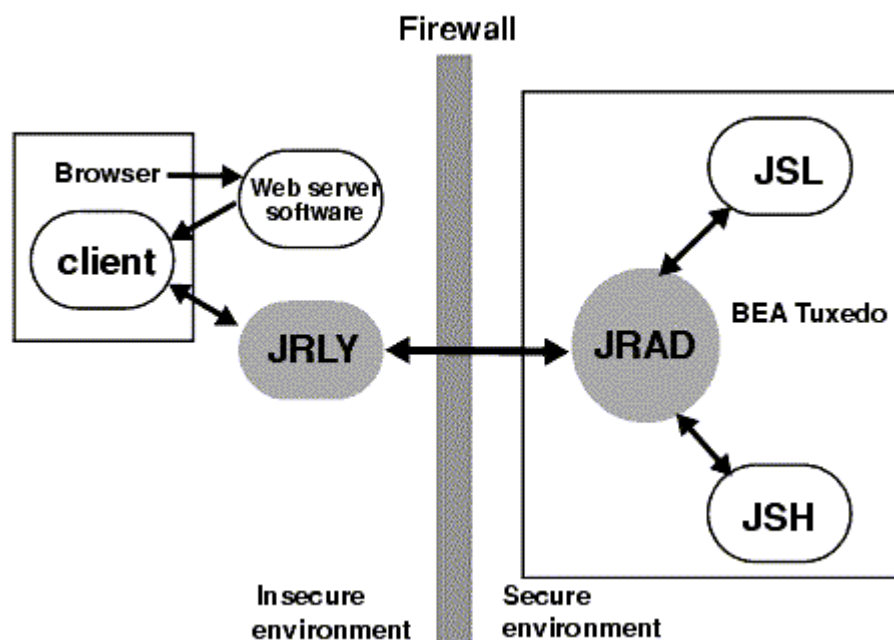
Understanding Jolt Internet Relay

This section discusses:

- Jolt Internet Relay architecture.
- A Jolt Internet Relay example.
- Implementation considerations.

Jolt Internet Relay Architecture

Jolt Internet Relay provides an environment in which the PeopleSoft web server and application server can be further decoupled. This provides greater security at sites where security is of utmost importance. Jolt Internet Relay routes messages from a Jolt client to a Jolt Server Listener (JSL) or Jolt Server Handler (JSH), and eliminates the need for the JSL, JSH, and Tuxedo application to run on the same machine as the web server. Communication takes place between the JRLY and JRAD elements rather than between the Jolt client and JSL/JSH processes. Traditionally an application server domain opens between 2 and 6 ports for such communications. The use of Jolt relay restricts this to one port per domain on the application server machine. This enables an administrator to open just one port on the application server machine. The following diagram illustrates this feature:



Jolt Internet Relay Architecture showing the Jolt Relay sending messages through a firewall to the Jolt Relay Adapter

Jolt Internet Relay consists of two elements: Jolt Relay (JRLY) and Jolt Relay Adapter (JRAD). It's important to understand the difference between these two elements.

JRLY consists of a standalone program and configuration file; the program runs on the same machine as the web server. JRLY receives Jolt messages from a PeopleSoft web application and routes those messages to JRAD on the application server. It receives the Jolt message through one port, the **LISTEN** port, and connects to the JRAD by using another port, the **CONNECT** port. JRLY is sometimes referred to as a front-end relay.

JRAD runs on the same machine as the application server. It's configured on the application server domain as part of the PeopleSoft PSADMIN domain configuration procedure. JRAD listens for JRLY messages on its **LISTEN** port and transfers the message to the JSL or JSH. JRAD is sometimes referred to as a back-end relay.

Note. Implementing Jolt Relay can impede performance. Always perform testing with typical production system load to ensure it will meet your service level requirements.

Implementation Considerations

Keep the following points in mind as you configure the Jolt Internet Relay components:

- The jrly binary and its corresponding jrly.config file must exist in the same directory. To start multiple Jolt Relays on a machine, copy the jrly binary and jrly.config into each subdirectory, modify the parameters in the jrly.config file, and start Jolt Relay. On Windows, you can define multiple Jolt Relay services on a machine.
- You can start the JRLY process before or after you start JRAD. The JRLY attempts to connect to JRAD on the client request. If the JRLY is unable to connect to the JRAD, the client is denied access and disconnected. The connection will be retried upon the first use of PeopleTools..
- If you're installing Jolt Internet Relay on UNIX and anticipate a large number of concurrent connected clients, increase the file descriptors limit before running the JRLY executable.
- At runtime, if you get the following message:

```
[FRI JUN 1 6 20:25:11 1997] JRLY:accept():accept failed,=>
err no: 23, strerror: File table overflow
```

PeopleSoft recommends that you increase the MaxUSERS kernel parameter and regenerate the kernel.

- If you're unable to connect, make sure that you check the following items:
 - Port numbers do not match.
Print out the jrly.cfg file and the psappsrv.cfg file and compare the port numbers that you specified.
 - Make sure that the application server is running.
 - Make sure that JRLY is running.
- Jolt Internet Relay can be installed on an intermediate machine rather than the web server machine if necessary. This extra level of indirection can cause performance degradation.
- Make sure that JRAD is running on the application server and that you configure JRAD using PSADMIN.

Configuring JRLY

Configuring JRLY is identical on UNIX and Windows.

To configure JRLY, navigate to *TUXDIR\udataobj\jolt\relay* and open jrly.config in a text editor.

Important! On UNIX, you can edit this configuration file by using VI or an equivalent editor. However, on Windows, you must edit the file using an editor that preserves the file's UNIX line feeds. WordPad is valid for this purpose, but Notepad is not.

Modify the parameters in the configuration file to reflect the site specifications, as follows:

<i>Parameter</i>	<i>Description</i>
LOGDIR	<p><i>LOGDIR</i> specifies the directory where JRLY creates access and error log files. This directory must exist; the JRLY program does not start if it can't find this directory. The path that you specify for LOGDIR should be an absolute path (starting from / on UNIX systems, starting from a drive letter on Windows systems). The JRLY accepts relative path names, but LOGDIR is relative to the directory from which the JRLY program is started, unless you specify it as an absolute.</p>
ACCESS_LOG	<p><i>ACCESS_LOG</i> specifies the name of the file where JRLY records access information. This log file is created in <i>LOGDIR</i>. If the log file already exists, the most recent information is appended to it.</p> <p>This parameter can be any valid file name. Everything after the equals sign (=) to the end of the line is considered as part of the file name, but leading and trailing blanks are ignored.</p> <p>Note. If the JRLY program can't create this file or open it for appending, the program exits.</p>
ERROR_LOG	<p><i>ERROR_LOG</i> specifies the name of the file where JRLY records error information. This file follows all the rules that apply to the <i>ACCESS_LOG</i> parameter. JRLY_error_log is created in /tmp.</p>
LISTEN	<p><i>LISTEN</i> specifies the host and port on the current machine (that is, the machine where you're installing Jolt Relay). JRLY listens for client connections. The following formats are acceptable:</p> <pre>LISTEN=192.9.100.100:9000 LISTEN=//192.9.100.100:9000 LISTEN=sp-ibm02:9000 LISTEN=//sp-ibm02:9000</pre> <p>Specify the port number in decimal; it must match the port number that is specified by the <i>psserver</i> parameter in the configuration.properties file for the PIA web application.</p> <p>Note. If a machine has multiple network interfaces, you should use the IP address notation, because specifying the hostname could be ambiguous (the result is OS dependent). If the JRLY program can't establish a network listening end-point at the host and port specified, it prints an error and exits.</p> <p>The hostname that's specified for this parameter must be the name of the host on which the program is running.</p> <p>Note. You can create multiple configuration files to run multiple instances of JRLY. Each configuration file must specify a different port number for this parameter.</p>

<i>Parameter</i>	<i>Description</i>
CONNECT	<p><i>CONNECT</i> specifies the location of the JRAD machine and process port on the application server machine to which the JRLY program connects. A JRLY program communicates only with a single JRAD. The address you specify for this parameter must match the JRAD listener address that's on the application server machine (check the PSAPPSRV.CFG file in <i>PS_CFG_HOME/appserv/domain</i>). The JRAD doesn't have to be running when you start the JRLY program. JRLY attempts to connect to the JRAD when it first starts, and if the JRAD is not available, JRLY tries again whenever a new client connects to it. You can use any of the following formats for this parameter:</p> <pre>CONNECT=192.9.100.100:9100 CONNECT=//207.135.44.91:9105 CONNECT=sp-hp06:9105 CONNECT=//sp-hp06:9105</pre> <p>Note. PeopleSoft has found that machine address formats are operating system and environment dependent. If one fails to connect to the application server, try another format.</p>
SOCKETTIMEOUT	<p><i>SOCKETTIMEOUT</i> specifies the duration (in seconds) for which the Jolt Internet Relay Windows service blocks the establishment of new socket connections to allow network activity (new connections, data to be read, closed connections) to complete. It's valid only on Windows machines.</p> <p><i>SOCKETTIMEOUT</i> also affects the Service Control Manager (SCM). When the SCM requests that the service stop, the SCM needs to wait at least the number of seconds specified by this parameter.</p>

Configuring JRAD

The JRLY connect port connects to the JRAD listener port that is specified on the application server machine. JRAD then routes the message to Jolt, either using the JSL for initial connection from a web client, or using the JSH for all subsequent connections from a web client. The return message follows the same path in reverse.

To configure JRAD:

1. Launch the PSADMIN utility.
2. Navigate to the PeopleSoft Domain Administration menu and select *Configure this domain*.
3. In the Quick Configure menu, select the number for the Jolt Relay option, to enable Jolt Internet Relay.
4. Select the JRAD Port option, and enter the appropriate port number for the JRAD Port.

Note. The JRAD (listener) port number must match the JRLY connect port that you previously configured.

See Also

[Chapter 4, "Using the PSADMIN Utility," Using the Quick-Configure Menu, page 38](#)

Running Jolt Relay

This section discusses how to:

- Use the JRLY administration program.
- Run Jolt Relay on Windows.
- Run Jolt Relay on UNIX.

Using the JRLY Administration Program

You use the `jrly` command located in `TUXDIR\udataobj\jolt\relay` to administer Jolt Relay on all platforms. You can use the following `jrly` command options at any time:

- `jrly -version`
Display the current version of the JRLY binary.
- `jrly -help`
Display a summary of command-line options with brief descriptions.

Running Jolt Relay on Windows

On Windows, you set up Jolt Relay to run as a service. On other platforms you must run Jolt Relay directly.

See [Chapter 13, "Working with Jolt Configuration Options," Running Jolt Relay on UNIX, page 304.](#)

If you want to install multiple Jolt Relay services, you must specify a string to be used as a *display suffix* that uniquely identifies each additional service you install. You subsequently use the suffix to identify each service it commands. An additional service with the suffix `MyJoltRelay`, for example, is called *Oracle Jolt Relay_MyJoltRelay*, but you refer to it using only the suffix. You can omit the suffix when installing only one of these services, which becomes the default Jolt Relay service, called *Oracle Jolt Relay*.

Note. All administrative commands in the following table except `-start` and `-stop` require that you have write access to the Windows registry. The `-start` and `-stop` commands require that you have Windows service control access. These requirements are based on Windows user restrictions.

Command	Description
<code>jrly -install [display_suffix]</code>	Install JRLY as a Windows service.

Command	Description
<code>jrly -remove [<i>display_suffix</i> -all]</code>	Remove one instance, all instances, or the default JRLY Windows service.
<code>jrly -set [-d <i>display_suffix</i>] -f <i>config_file</i></code>	Update the registry with the full path of a new configuration file for the specified JRLY service. Note. You can run multiple Jolt Relay services by specifying a different display suffix along with the name of a different configuration file for each installed service. Each configuration file must contain a unique value for the LISTEN parameter that specifies a different port. This is essential to avoid port clashes when running the services concurrently. You must run this command before the service starts.
<code>jrly -manual [<i>display_suffix</i>]</code>	Set the start/stop to manual. This command sets the specified JRLY service to be manually controlled, using either the command-line options or the Service Control Manager (SCM).
<code>jrly -auto [<i>display_suffix</i>]</code>	Set the start/stop to automatic. This command sets all the operations for a specified JRLY service to be automatically started when the OS boots and stopped when the OS shuts down.
<code>jrly -start [<i>display_suffix</i>]</code>	Start the specified JRLY service.
<code>jrly -stop [<i>display_suffix</i>]</code>	Stop the specified JRLY service.

Running Jolt Relay on UNIX

This section discusses how to start and stop Jolt Relay directly from a command line on UNIX.

To start Jolt Relay on UNIX:

1. Change directories to the Jolt Relay directory within your Tuxedo installation:

```
cd $TUXDIR/udataobj/jolt/relay
```

2. Run the following command:

```
jrly -f jrly_config &
```

Where *jrly_config* is the name of a Jolt Relay configuration file.

You can run multiple instances of Jolt Relay by using a different port for each instance. You run JRLY once for each instance, and specify a different configuration file each time. Each configuration file must contain a value for the LISTEN parameter that specifies a different port.

The & causes JRLY to run in the background.

To shut down Jolt Relay on UNIX, use the UNIX kill -9 command.

Appendix A

Securing PS_HOME and PS_CFG_HOME

This chapter provides an overview of PS_HOME and PS_CFG_HOME security and discusses how to:

- Secure PS_HOME.
- Secure PS_CFG_HOME.

Understanding PS_HOME and PS_CFG_HOME Security

With the separation of the PS_HOME and PS_CFG_HOME directories, system administrators can implement more secure PeopleSoft deployments by restricting access within each of these directory structures.

This section describes the procedures and considerations involved in configuring these additional security options.

Note. Each site can elect to implement these security measures as needed according individual security policies.

Note. Each PeopleSoft application you have licensed may have specific instructions regarding the implementation of these security measures. Always check your application-specific documentation for any information you need to consider to ensure both a secure environment and a properly functioning application.

Understanding PS_HOME Security

Because the configuration files, by default, do not reside in PS_HOME, the PS_HOME installation can be locked down to prevent unauthorized access, by user or system process. By making the PS_HOME directory 'Read-Only', processes running in the domain cannot write to PS_HOME or any of the subdirectories therein. Likewise, any users with malicious intent are unable to delete or modify executable files in PS_HOME.

Securing PS_HOME involves making the directory read-only, yet making sure that the following system elements have sufficient access.

<i>System Element</i>	<i>Description</i>
Application server	Application server domains need read access to the executable and binary files of PS_HOME to process requests and run application logic.

System Element	Description
Process Scheduler	<p>Process Scheduler domains need read access to the executable and binary files of PS_HOME to run batch processes. Plus, keep in mind these points:</p> <ul style="list-style-type: none"> • The user creating a Process Scheduler domain on Windows needs read and write access to the Windows Registry. • The restricted OS user needs to have full privilege access to the psreports folder (and its children). Process Scheduler and the Report Distribution agent inherit the restricted user's security settings they will need to create folders in psreports. • Oracle ProcMGR (Tuxedo) should be started with the restricted OS user ID. • Configure Process Scheduler with an Admin OS user for Windows. While logged into Windows with the full privilege OS user ID, create and configure the Process Scheduler domain, so that ODBC and NVision DLLs register properly. <p>See <i>Enterprise PeopleTools Installation</i> for your platform</p> <p>See <i>Enterprise PeopleTools 8.50 PeopleBook: PeopleSoft Process Scheduler</i></p>
File server/Windows workstations	The file server and/or Windows workstations running Application Designer need read-only access to PS_HOME to facilitate three-tier connections.

Note. These instructions do not apply to the PS_HOME residing on the web server for PIA or the PS_HOME on the database server.

Note. When implementing a read-only PS_HOME, consider that environment locations to which processes write files can't be in a read-only location. Settings for "temporary" directories and "output" directories should not be located within the PS_HOME directory structure. For example, the default temporary directory locations are c:\TMP (Windows) and %root%\TMP (UNIX).

Note. All elements of your PeopleSoft implementation, such as Process Scheduler and SQR can operate within a secure PS_HOME configuration.

Understanding Minimum Access Required by The User Starting Domains

The bare minimum that needs write access at the time a domain boots includes:

- The domain directory: it must be possible to write content to the domain directory although most of the configuration files in this directory can be read-only.
- The domain LOGS directory: by default this is the LOGS directory beneath the domain directory. If this location is overridden in the configuration file, the relevant location must also be read-write.

- The .adm directory: this subdirectory within the domain (if present) must also be read-write. This is required by Oracle Tuxedo.
- The Archive directory: located within the domain directory, a copy of the .cfg is archived to this directory each time it is updated. This directory is also used by the Purge Cache PSADMIN option for application servers.

All other files in the PS_CFG_HOME directory tree can be made read-only to the user starting the domain.

Understanding PS_CFG_HOME Security

Some administrators may want to implement additional security and restrict access to PS_CFG_HOME. For example, in some cases you may want take further steps to limit privileges of the user starting a domain, or lock down configuration files to prevent unintended configuration changes during runtime.

Securing PS_HOME on UNIX

The UNIX operating system lends itself to a read-only configuration for PS_HOME because of the way that Inter-process Communication (IPC) resources are allocated and managed. UNIX was designed to allow multiple users concurrent access to the same physical hardware and file system while enforcing a strong privileges model.

Note. It is necessary to have access to at least two user accounts in order to setup a true and complete read-only environment on UNIX.

To illustrate the procedure, two user accounts are used.

<i>User Account</i>	<i>Description</i>
InstallAdmin	User responsible for installing PeopleTools.
DomainAdmin	User responsible for creating, configuring, and booting domains. Note. It is under this account that domain processes will run and therefore should have the most stringent permissions.

To setup read-only PS_HOME on UNIX:

1. Install PeopleTools using the InstallAdmin account.

2. Verify that PS_HOME is read-only.

After installing PeopleTools, attempt to delete both a directory and file from PS_HOME using the DomainAdmin account.

If it is not read-only to the DomainAdmin account, login as the InstallAdmin account and use the chmod command to make PS_HOME read-execute to the world.

If the DomainAdmin account is a member of the same group as the InstallAdmin account you will need to apply the read-execute restriction to the group also. For example,

```
chmod -R 755 $PS_HOME
```

3. Sign in as the DomainAdmin account, open a shell, and change directory to PS_HOME.
4. Invoke psconfig.sh to set the environment.
5. Create and configure a new domain.

This can be an application server, search server, or Process Scheduler domain.

6. Start the new domain and verify that all of the domain processes have started.

For application server domains, ensure that you can sign in through PIA and make successful page requests.

Managing a Secure PS_HOME on UNIX

When deploying a secured PS_HOME environment on UNIX, keep the items in this section in mind.

Working with User Accounts

The user account that boots the domain must be the same user who configures the domain. This is a Tuxedo requirement, not a PeopleTools requirement. This means that the user account under which the domain processes will run must have read-write access to the domain directory.

The owner of the domain processes is the user account who starts the domain. This is different from Microsoft Windows, where the domain processes are booted by the account that starts the Oracle ProcMGR service. If you use both Windows and UNIX servers to deploy PeopleSoft, keep this subtle distinction between the two operating systems.

Configuring Partial PS_HOME Access

In some cases, user accounts may need to access specific parts of the PS_HOME directory tree. This is recommended through the addition of a "hybrid" user to the same group to which the "InstallAdmin" account (the user who installed PeopleTools) belongs. The InstallAdmin can then choose to allow group access to the specific parts of the PS_HOME directory tree to which the hybrid user is permitted read-write access.

For example, consider a scenario where you have installed PeopleTools at your site, but have hired a consultant to help with various implementation tasks. The InstallAdmin only wants to allow the consultant access to specific parts of the PS_HOME directory tree. The account that the consultant uses is therefore a hybrid account. It is has read-write access to PS_HOME, but only to the specific subdirectories deemed necessary.

To achieve this hybrid privilege model, allow group access to those specific directories under PS_HOME to which the consultant requires write access.

Securing PS_HOME on Windows

When securing PS_HOME on Windows, you have these options:

- Multiple administrator user accounts.
- Local user accounts.

Multiple Administrator User Accounts

This method of securing PS_HOME on Windows offers the ability for many administrator user accounts to share the same PS_HOME while managing separate PS_CFG_HOMEs. This method is most appropriate for a production environment.

In this configuration, you install PeopleTools on a server machine, and share the PS_HOME installation location as read-only. Domain administrators may then map to this network drive, and invoke PSADMIN to create and start domains.

Once you have set up a secure PS_HOME, domain creation and the various prerequisites for setup are the same as before. For example, database connectivity must be available on the machine on which the domain will boot. The server where PS_HOME is located acts as a read-only file server for the domains.

To illustrate this procedure, two user accounts are used.

User Account	Description
User1	An administrator on Machine, having read-write access to PS_HOME.
User2	An administrator on Machine2, having read-only access to PS_HOME. A restricted user.

In this scenario, one administrator installs PeopleTools, and a second user, with a more limited set of privileges, creates and administers domains.

The steps for User2 on Machine2 can be applied to multiple users. Any number of users can map to a single read-only PS_HOME.

Note. This information applies to PeopleTools installations on drives assumed to be formatted as NTFS.

To install PS_HOME and restrict privileges:

1. While logged in as User1 on Machine1, run the PeopleTools installation program.
2. Choose Application Server and Batch Server installation options.

File Server is optional, depending upon whether or not it is required.

3. Ensure that you install PeopleTools such that PS_HOME is not the top most directory on the drive.

For example,

D:\PTInstalls\PT8.50.

Note. This is essential if you plan on accessing PS_HOME as a mapped drive.

4. After the installation has completed, set privileges on the PS_HOME directory tree, using Windows File Sharing.
 - Using Windows Explorer, select the high-level directory (as in D:\PTInstalls), right-click, and select Sharing and Security.
 - On the Properties, Sharing tab, click Share this folder.
 - Click Permissions, and for the Group of Everyone, check the Allow checkbox for Read, make sure the Allow checkbox for Change is not selected, and click Apply and/or OK.
5. Verify that the folder has been shared by making sure the 'hand' icon appears on the folder in Windows Explorer.

To set up access to a secure PS_HOME:

1. Sign in as User2 on Machine2.
2. Map a network drive to the shared PS_HOME.
3. Verify that you cannot add, modify, or delete any content below the mapped location.

This ensures that PS_HOME is read-only. If you can delete or change content in the mapped location, it is possible that User2 is an administrator on Machine1. User2 must not be an administrator on Machine1 for these security measures to be effective.

If User2 cannot see the shared location there may be a problem with the share or the local network. Make sure the machine can be pinged.

4. Configure Oracle ProcMGR service to allow the User2 account to access PS_HOME.

This is necessary because by default the Oracle PRocMGR service uses the Local System account.

- a. Select Start, Programs, Administrative Tools, Services, and double-click on the Oracle ProcMGR service.
- b. On the General tab, stop the service, and set the Startup type to Manual.
- c. On the Log On tab, select the This account radio button, and enter the logon information for User2.
- d. Click OK.

Note. Do not start the service yet.

5. Set the TM_TUXIPC_MAPDRIVER user variable for User2.

This environment variable must be set to contain any mapped drives required by the domain processes, such as the drive where PS_HOME is located. Use the following format:

```
drive1:=\\machine_name1\dirpath1[;drive2:=\\machine_name2\dirpath2[...]]
```

For example,

```
N:\\10.100.200.300\PTInstalls
```

If multiple mapped drives are required, use a semicolon to separate the values, similar to the way directories are expressed in the PATH environment variable.

Note. Depending on your network, use either the DNS name or the IP address to specify the machine name.

Note. If the Oracle ProcMGR needs to run unattended, where no user is signed in, set the TM_TUXIPC_MAPDRIVER environment variable as a system environment variable instead of a user environment variable.

6. Start the Oracle ProcMGR service.

Once started, you can start PSADMIN as you normally would.

Local User Accounts

Using local user account to secure PS_HOME is a machine-bound solution that you may consider during an initial demo, development, or testing environment, where PS_HOME and PS_CFG_HOME reside on the same machine. In this method, only one machine and one domain account is required.

In this scenario, PeopleTools is installed by a user with administrative privileges. In the context of this scenario, this refers to the network domain user, a user that is a member of an existing network domain of users.

Application server domains are administered by a second user with a more limited set of privileges. This second user is created on the local system and only has access to resources on that machine.

In the following procedure, these user types are represented as:

User	Description
COMPANY\User1	User1 is an administrator on Machine1 and belongs to the network domain named <i>COMPANY</i> .
LOCAL_MACH\Guest2	Guest2 does not have administrative privileges on Machine1 but can sign on to Machine1. Guest2 is a Restricted User.

Setting up a secure PS_HOME and restricting privileges:

1. While logged in as User1 on Machine1, install PeopleTools to the server using the install program.

Choose Application Server and Batch Server installation options. File Server is optional depending upon whether or not it is required.

Ensure that you install PeopleTools such that PS_HOME is not in the top directory level on the drive. For example, an ideal location would be C:\PTInstalls\PT8.50.

2. While logged in as User1 on Machine1, create the Guest2 user account.

Create a new user with at least the following attributes:

- User name
- Password
- Select Restricted user on the Group Membership tab.
- De-select the User must change password at next logon checkbox.

See Microsoft Windows documentation for more information related to creating users on Microsoft Windows.

3. Verify that the user is a Restricted User by highlighting the user ID and clicking on the Properties tab.

You may need to exit and re-enter the User Accounts dialog to see the new user added.

4. Make the PS_HOME directory tree read-only.

- a. Open Windows Explorer and navigate to and select the PS_HOME directory location.
- b. Open the Properties dialog and click on the Security tab.
- c. Deny Write access to all Local Users.

To configure access PS_HOME:

1. Setup the Oracle Proc MGR Service to allow the IPC resources to be created using the restricted user ID (Guest2).

This is necessary because by default the Oracle PProcMGR service uses 'Local System' account which provides greater access to the PS_HOME than desired.

- a. Select Control Panel, Administrative Tools, Services.
- b. Double click on the Oracle ProcMGR and change the Startup Type to Manual.

2. Change the user and password with which the service is started to match the new local user that you created earlier (Guest2).

Note. Do not start the service, just click OK.

3. Log off and sign on to Machine1 as Guest2.

In most cases, any error messages that you see when re-signing on can be ignored as they are associated with signing on with restricted permissions.

4. Verify that you cannot delete, update, or add any content to PS_HOME.

5. Before creating a domain your database connectivity must be setup within the restrictions of the guest2 local user account that you have created.
6. Start PSADMIN, and create a new domain, and confirm that the domain boots.
7. Install PIA on a separate machine, and verify that you can signon through PIA.

Because PIA is not within the scope of the secure PS_HOME, PIA should be installed on an additional machine. This is necessary because PIA needs write access to various locations within PS_HOME.

Managing a Secure PS_HOME on Windows

When deploying a secured PS_HOME environment on UNIX, keep the items in this section in mind. Depending upon your domain configuration and usage pattern, you may need to unlock specific subdirectories of PS_HOME.

Working With Mapped Drives, UNC Paths, and TM_TUXIPC_MAPDRIVER

This page explains mapped drives (Windows share drives) and the use of UNC paths for PeopleTools application server, Process Scheduler server, and search servers. When a PeopleTools domain is started on a Windows machine, it runs under the user for whom the ProcMGR Windows service has been configured. As such, the domain processes inherit the privileges of that user and not the user logged on to the system.

By default, the ProcMGR runs as the Local System account. While the Local System account has most privileges on the local host, it can't usually access UNC paths or mapped drives.

Note. ProcMGR is the Windows service that is responsible for allocating resources to Tuxedo domains.

Accessing UNC Paths and Mapped Drives

To allow PeopleTools domain processes to access UNC paths or mapped drives, it is necessary to start the ProcMGR service using an account that has access to these resources. This is typically a Windows *domain account*. A domain account refers to an account that logs a user ID on to both a machine and the corporate network. This account has been created by the network administrator. An example of such an account would be BIGCOMPANY\TSawyer.

The ProcMGR should be configured to start as the domain account. On the Log On tab of the service configuration dialog, click This account:, and enter the credentials of the domain account.

UNC Paths

If you plan to use UNC paths to access PS_HOME you must start PSADMIN using a UNC Path. For example:

```
\\ptinstalls\pt850\APPSERV\psadmin.exe
```

With the ProcMGR service set to a domain account, you can use PSADMIN to create and configure domains as if PS_HOME were on the local file system.

Mapped Drives

Additional steps are required if you plan on using a mapped drive for your PS_HOME or PS_CFG_HOME. These additional steps are required because Windows services do not recognize mapped drives. However, Oracle Tuxedo provides a mechanism by which the ProcMGR service is permitted to access network drives, which involves defining any mapped drives using an additional environment variable, TM_TUXIPC_MAPDRIVER. If this environment variable has not been set, the domain processes will be unable to recognize the network drives.

To make sure that the TM_TUXIPC_MAPDRIVER variable is visible to the ProcMGR service, it is necessary to set it globally, as a System environment variable. For example, set TM_TUXIPC_MAPDRIVER to:

```
N:==\\10.233.238.123\PTInstalls
```

Important! The mapped drive cannot point *directly* to PS_HOME. The mapping must point to the parent directory above PS_HOME. For example, if PS_HOME is N:\\10.233.238.123\\PTInstalls\\pt850, TM_TUXIPC_MAPDRIVER should point to N:\\10.233.238.123\\PTInstalls. PTInstalls is the parent directory of \\pt850, which would resolve to N:\\pt850.

Note. When mapping network drives to the PS_HOME is located, make sure to select Reconnect at logon.

Working With Oracle ProcMGR Service

Recall that the Oracle ProcMGR service to 'Manual' instead of 'Automatic'. Failure to do so may result in your domain account becoming locked. If set to 'Automatic' the service may continually attempt to start with an expired password causing the network to lock out the domain user account due to successive failed retries.

Note. On Windows servers, it is recommended to have the Windows user logged in running PSADMIN be the same user that runs the ProcMGR service.

Managing TM_TUXIPC_MAPDRIVER

The TM_TUXIPC_MAPDRIVER environment variable needs to be maintained consistent with the mapped drives upon which you access PS_HOME. If the drive mappings change, then you need to make sure the new values are specified in TM_TUXIPC_MAPDRIVER. If the drive mapping represented by the TM_TUXIPC_MAPDRIVER does not exist, the ProcMGR service will fail to start.

Resolving Initialization Timeout Issues

If the PS_HOME used by your domains is on a network drive, you may notice a delay with starting a domain. This is a result of the binaries being loaded from across the network versus from the local disk. This can cause an initialization timeout.

If you notice domain start failures, check for the following message in the Tuxedo log for the domain:

```
tmboot.16020.15792.-2: CMDTUX_CAT:1859: ERROR: Server process ID 12668 failed to
initialize within 60 seconds
```

In such cases, increase the timeout values in the domain's psappsrv.env file to accommodate for the slower start time. For example,

```
TM_BOOTTIMEOUT=300
TM_RESTARTSRVTIMEOUT=300
```

Note. Changes to the .ENV file are overwritten when the domain is reconfigured. So that these modified values persist after you reconfigure the domain, add the modified values to the .UBX file's *PS_ENVFILE section before reconfiguring the domain. The .UBX file is located in the domain's directory.

Implementing PS_CFG_HOME Security

The steps in this section describe techniques for applying more stringent access to a PeopleTools environment by restricting access to PS_CFG_HOME. If you intend to secure PS_CFG_HOME, it is assumed that you have also secured PS_HOME. Securing PS_CFG_HOME enables you to prevent malicious access to content and configuration files located in PS_CFG_HOME and in domain directories.

These steps describe a security implementation where you configure a user account(s) that can create and configure domains, and user account(s) for domain administrators, who can start and stop domains.

It is possible to limit the permissions to PS_CFG_HOME, such that the domain administrator account can:

- Create files and sub-directories in PS_SERVDIR.

This is necessary for creating log files and temporary files. Tuxedo also requires read-write access to the domain directory.

- Read (but not change or delete) any existing configuration or template files in PS_SERVDIR.

These files include .cfg, .ubb, .ubx, .val files, and so on.

Note. To apply these permissions, you must do so after the domain has been configured but before it has been started.

Note. Once these permissions have been implemented, only a user account with the appropriate privileges can reconfigure the domain.

Securing PS_CFG_HOME on UNIX

You have these approaches when implementing a secure PS_CFG_HOME on UNIX:

Security Approach	Description
Use a root account, or use <i>super user do</i> (sudo), instead, to perform root-level operations with a non-root account:	This technique involves locking a user account's access to a file from the root account.

Security Approach	Description
Use two administrator user accounts:	<p>The two administrator accounts approach involves using two user accounts that are members of the same group.</p> <p>This approach permits all users in the account in the group read access to the domain configuration.</p> <p>This approach works best if the number of users in the group is kept to a minimum.</p> <p>Because there are multiple users involved, it is necessary to override the default PS_CFG_HOME environment variable such that both users will see the same domains.</p>

To set up a secure PS_CFG_HOME using sudo:

1. Create and configure the domain.

For this procedure, assume the user who does this is DomainAdmin.

2. Use `sudo` to restrict write access to the sensitive configuration files.

For example, with the `sudo` command include:

```
chmod 555 <filename>
```

In this case, only `sudo` can change the configuration files or restore write access to DomainAdmin.

3. Log in as DomainAdmin, and verify that none of the restricted files can be changed or deleted by the DomainAdmin session.
4. Start the domain as DomainAdmin.

To set up a secure PS_CFG_HOME using two administrator accounts:

1. Create and configure a domain.

For this procedure, assume the user who does this is DomainAdmin.

2. As the DomainAdmin user, change the permissions on the domain configuration to allow write access to only those files and directories needing to be written to by the user starting the domain.
3. Signon as the DomainBootAdmin user, start PSADMIN, navigate to the Domain Administration menu, and re-configure the domain without making any changes.

This results in only the TUXCONFIG file being updated.

4. Star the domain as the DomainBootAdmin user.

Securing PS_CFG_HOME on Windows

To secure PS_CFG_HOME on Windows:

1. Ensure that your PS_CFG_HOME is created in a location that is accessible to the user account that needs to start the domain.

By default, the PS_CFG_HOME location will not be visible to the user account starting the domain because it is created in the user home of the user account creating the domain. Use one of these options to make the PS_CFG_HOME visible to the user account that needs to start the domain:

- Override the default location for PS_CFG_HOME, by manually setting the PS_CFG_HOME environment variable to a custom value.
 - Enable the restricted user to view the PS_CFG_HOME in the domain creator's user home. Set the PS_CFG_HOME as a system-level environment variable so that the restricted user will be able to see it when logged on.
2. While signed in as the domain administrator, create and configure your new application server, Process Scheduler, or search server domain.
 3. While signed on as the domain administrator, apply the necessary read-write restrictions.

It is recommended to apply read-only privileges to the entire directory path to the domain.

- a. Using Windows Explorer select the domain directory and open the Properties dialog.
 - b. Select the Security tab. If the restricted user is not displayed, click Add to append the user to the list.
 - c. Once added, highlight the restricted user ID and ensure that the following actions are checked in the Allow column: Read & Execute, List Folder Contents, Read, and Write.
4. Apply read-only privileges to the sensitive domain files that should be protected from read-write at runtime.

This typically includes the Tuxedo binary configuration (PSTUXCFG and PSBDMCFG), ASCII configuration files and templates (*.cfg, *.ubb, *.env, *.ubx, *.lst).

Using Windows Explorer, select each of these files in the domain directory and open the Properties dialog.

Note. At a minimum, it must be possible for the user starting the domain to write to the .adm and LOGS directory within PS_SERVDIR. Additionally, this user must be permitted to create new files in the domain directory.

5. Sign in as the restricted user, boot the domain, and verify that it is not possible to delete or modify any of the restricted domain files.

Note. Sign in using the same user account as the one entered in the Oracle ProcMGR service.

Note. If subsequent configuration changes are required it is necessary to configure the domain while signed in as the administrator account that created and configured the domain.

Appendix B

WebLogic Managed Server Architecture

The PeopleSoft Internet Architecture running on the WebLogic Server takes advantage of WebLogic's managed server architecture. This appendix provides overview discussion and discusses:

- Administering a WebLogic server life cycle.
- Tuning performance and monitoring resources.
- Changing configuration settings.
- Applying an example single-server configuration.
- Applying an example multi-server configuration.

See Also

[Chapter 7, "Working with Oracle WebLogic," page 119](#)

Clustering and High Availability for PeopleSoft on My Oracle Support

PeopleSoft Internet Architecture Servlets and Applications

PIA is packaged as a J2EE Enterprise Archive and is comprised of a collection of J2EE web applications, commonly referred to as webapps or servlets. For the most part, in the context of PeopleSoft, the term 'servlets' is used. The PeopleSoft servlets are:

PORTAL	PeopleSoft Portal
PSIGW	Integration gateway
PSOL	PeopleSoft On-line Library
PSEMHUB	PeopleSoft Environment Management Framework
PSINTERLINKS	PeopleSoft Business Interlinks

Note. PeopleSoft Business Interlinks is a deprecated product. These options exist for upgrade compatibility and transition.

In addition, these servlets are added when you install PIA on a WebLogic server machine. These elements are not part of the PeopleSoft Enterprise Archive, but instead are defined individually.

HttpProxyServlet	Reverse Proxy Server – Proxy to a single content server per URL. Each URL can provide unique content.
HttpClusterServlet	Reverse Proxy Server – Proxy to multiple WebLogic servers. All content servers provide access to the same content for load balancing.
Console	Administrative console for WebLogic Server.

WebLogic Domain Types

This section provides an overview of Weblogic domain types and discusses:

- Single-server domain.
- Multi-server domain.
- Distributed managed server.
- Common default settings.
- Single-server and multi-server/distributed server analogy.
- Domain topology.

Understanding WebLogic Domain Types

During PIA setup, you can choose between two different WebLogic domain configurations: a single-server domain and a multi-server domain. A multi-server domain can be expanded across multiple machines using the *distributed managed server* option. So, a distributed managed server implementation is a variation of the multi-server domain. Each of these domain configurations has a specific purpose but is fully customizable beyond that purpose.

This section discusses:

- single-server domains.
- multi-server domains.
- distributed managed servers.

Single-Server Domains

In a single-server configuration, the WebLogic domain's administration console and the J2EE components of PIA are all provided on a single instance of WebLogic Server. This configuration is intended for single-user or very small scale, non-critical production environments. It can be used as a starting point for you to familiarize yourself with WebLogic Server.

In a single-server domain, the resources used to administer WebLogic Server and your PeopleSoft application are not isolated from one another, therefore you don't administer each element individually. While each element must complete for the same resources, the low resource requirements of this configuration make it ideal for small scale and non-production usage.

The single-server domain in a PeopleSoft implementation consists of only one server: PIA.

Single-Server Deployment

Some of the servlets deployed in a single-server domain configuration must be accessed using a modified URL:

`http://server:port/servlet_name/...`

The single-server domain configuration deploys servlets as follows:

Application	Deployed to Server	Servlet Name in URL
PORTAL	PIA	(not needed)
PSIGW	PIA	PSIGW
PSOL	PIA	PSOL
PSEMHUB	PIA	PSEMHUB
PSINTERLINKS	PIA	PSINTERLINKS
Console	PIA	console
HttpProxyServlet	Defined but not deployed.	(not needed)
HttpClusterServlet	Defined but not deployed.	(not needed)

Single-Server Domain Specific Settings

To configure the single-server domain specific settings, launch the Administration Console.

In the console, expand the Environment tree and select Servers. Click on the PIA server. Select the Configuration tab, and the General sub-tab. The default web application for the PIA server is PORTAL. The single-server domain specific default settings for the PIA server are as follows:

Setting	Default Value
Listen address	* (all local IPs).
Listen Port	80 (set during PIA setup).
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
SSL Listen Port	443 (set during PIA setup).

Note. To configure SSL, you must also define SSL certificates.

See [Chapter 7, "Working with Oracle WebLogic," Implementing WebLogic SSL Keys and Certificates](#), page 138.

Example: Single-Server Domain

The following illustrates sample contents of a typical single-server domain:

Server	Process Type	WebLogic and PIA Elements
Single server machine	WebLogic administration	Weblogic Administration Server
	Server instance	PIA
	Servlet or application	Administration Console PORTAL PSEMHUB PSIGW

Multi-Server Domains

The multi-server domain configuration is intended for production environments. This configuration takes advantage of WebLogic's administration server and managed server architecture. In a multi-server configuration, multiple instances of WebLogic server are used, each contributing a specific function. The WebLogic console is provided on the domain's administration server, WebLogicAdmin, and the J2EE components of PIA are provided on individual or shared WebLogic managed servers.

A production application warrants process and resource pool isolation for greater stability and optionally tighter security controls, which this configuration provides. In a multi-server configuration, the resources used for WebLogic domain administration and monitoring are isolated from similar resources used to support the PIA application. A server process named *WebLogicAdmin* performs nothing but WebLogic administration, which includes domain administration and monitoring. Continuing that separation, the individual PIA servlets are (usually) isolated from each other. The PIA servlets are targeted and deployed across a portion of the six remaining server definitions, all of which are classified as *managed servers*, which are delivered in the multi-server configuration.

A multi-server domain in a PeopleSoft implementation creates the following servers:

WebLogicAdmin	Administration server for WebLogic domain administration.
PIA	Server for the PeopleSoft Portal and integration gateway.
PIA1	Server for the PeopleSoft Portal and integration gateway.
PIA2	Server for the PeopleSoft Portal and integration gateway.
PSOL	Server for the PeopleSoft Online Library (PeopleBooks) application.

PSEMHUB Server for the PeopleSoft Environment Management Framework application.

RPS Server for WebLogic reverse proxy server applications.

Multi-Server Servlet Deployment

Some of the servlets deployed in a multi-server domain configuration must be accessed using a modified URL:

`http://server:port/servlet_name/...`

The multi-server domain configuration deploys servlets as follows:

<i>Application</i>	<i>Deployed to Server, Cluster (members)</i>	<i>Servlets Name in URL</i>
PORTAL	PIA, PeopleSoftCluster (PIA1, PIA2)	(not needed)
PSIGW	PIA, PeopleSoftCluster (PIA1, PIA2)	PSIGW
PSOL	PSOL	PSOL
PSEMHUB	PSEMHUB	PSEMHUB
PSINTERLINKS	PIA, PeopleSoftCluster (PIA1, PIA2)	PSINTERLINKS
Console	WebLogicAdmin	console
HttpProxyServlet	RPS	(not needed)
HttpClusterServlet	Defined but not deployed.	(not needed)

Multi-Server Domain Specific Default Settings

To configure the multi-server domain specific settings, launch the Administration Console. In the Domain Structure tree, expand Environment, select Servers and click on the appropriate servlet, server, or application. Then select the Configuration tab, and the General sub-tab.

The domain specific default settings for the WebLogicAdmin server are as follows:

<i>WebLogicAdmin Setting</i>	<i>Default Value</i>
Listen Address	* (all local IPs)
Listen Port	9999
SSL Listen Port Enabled	Disabled

The domain specific default settings for the PIA server are as follows:

PIA Setting	Default Value
Listen Address	* (all local IPs)
Listen Port	80 (set during PIA setup)
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
HTTPS Listen port	443 (set during PIA setup)

The domain specific default settings for the PIA1 server are as follows:

PIA1 Setting	Default Value
Listen Address	Locally determined hostname.
Listen Port	80 (set during PIA setup)
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
SSL Listen Port	443 (set during PIA setup)

The domain specific default settings for the PIA2 server are as follows:

PIA2 Setting	Default Value
Listen Address	127.0.0.1
Listen Port	80 (set during PIA setup)
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
SSL Listen Port	443 (set during PIA setup)

The domain specific default settings for the PSOL server are as follows:

PSOL Setting	Default Value
Listen Address	* (all local IPs)
Listen Port	6001
SSL Listen Port Enabled	Disabled

The default web application for the PSEMHUB server is PSEMHUB. The domain specific default settings for the PSEMHUB server are as follows:

PSEMHUB Setting	Default Value
Listen Address	* (all local IPs)

PSEMHUB Setting	Default Value
Listen Port	8001
SSL Listen Port Enabled	Disabled

In the console, navigate to Environments, RPS, Configuration, General to configure the RPS server. The default web application for the RPS server is HttpProxyServlet. The domain specific default settings for the RPS server are as follows:

RPS Setting	Default Value
Listen Address	* (all local IPs)
Listen Port	8080 (set during PIA setup)
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
SSL Listen Port	8443 (set during PIA setup)

Note. To configure SSL, you must also define SSL certificates.

See [Chapter 7, "Working with Oracle WebLogic," Implementing WebLogic SSL Keys and Certificates, page 138.](#)

Example: Multi-Server Domain

The following illustrates the elements running within a typical multi-server domain:

Server	Process Type	WebLogic and PIA Elements
Single server machine	WebLogic administration	WebLogic Administration Server (WebLogicAdmin) WebLogic Node Manager
	Managed Server	PIA 1, PIA 2 for PeopleSoft Portal
	Managed Server	PIA for PSIGW
	Managed Server	RPS for httpproxyservlet to proxy content for PIA 1 and PIA 2

Distributed Managed Servers

The *distributed managed server* configuration, although listed alongside the single-server and multi-server domain types, is not a *true* domain type. It's an optional extension for an existing multi-server configuration. As with the multi-server domain type, this configuration takes advantage of WebLogic's managed server architecture. The distributed managed server configuration is intended for production environments encompassing multiple machines.

A distributed managed server configuration enables you to spread a *logical* WebLogic domain configuration *physically* across multiple machines in a heterogeneous network. One server machine might act as the domain administration server, while the other server machines act as distributed managed servers, running various managed servers, such as PIA1, PIA2, and so on. Typically, you would also take advantage of the WebLogic Node Manager, which enables a system administrator to control the managed servers remotely.

See your WebLogic Node Manager documentation.

In a distributed, multi-server configuration you can have multiple machines, multiple collections of resources and program files, multiple web server processes, and a replicated domain configuration file. In this model, if any of system resources or server instances becomes unavailable, work shifts to the next instance of that resource. In the multi-server configuration, an increase in PeopleSoft Portal usage, for example, can be accommodated by configuring an additional WebLogic server instance or machine to also serve the PeopleSoft Portal application. Using distributed managed servers provides flexibility in shaping your hardware allocation to meet your system demands.

Benefits of the Distributed Managed Server

A distributed managed server configuration provides the same benefits as a multi-server configuration with the added benefit of hardware isolation. This option requires a multi-server installation to be performed to some other location, which will contain the configuration for this distributed managed server.

The fundamental benefits of a multi-server configuration include:

- **Dedicated service providers:** Web servers can be dedicated to providing PeopleSoft Portal and are insulated from other portions of PIA such as PeopleSoft Integration Gateway or PeopleBooks.
- **Redundant service providers:** Multiple web servers can be used to serve different aspects of PIA, providing load balancing and failover support.
- **Distributed resources:** Multiple web server machines can be used, each capable of serving different or redundant aspects of PIA.
- **Centralized and replicated configuration:** Master domain configuration is centralized and distributed server information is replicated locally.

The server configuration settings for a distributed managed server are maintained by that domain's administration server are stored locally on that administration server. Configuration settings are replicated to a managed server during its startup, but are only maintained as a read-only backup copy for that individual managed server in the event that the administration server isn't available the next time this particular managed server needs to be started.

Note. Only one managed server can be run per distributed managed server domain directory. If you intend to run multiple distributed managed servers on a single machine, perform the PIA install and create unique distributed managed server domain directories, one for each distributed managed server that you intend to run on that machine.

Example: Distributed Managed Server

The following illustrates the multiple servers, their roles, and the various elements running on the multiple server machines within a sample distributed managed server configuration:

Server	Role	WebLogic and PIA Elements
Server 1	Domain Administration Server	WebLogic Administration Server (WebLogicAdmin)
Server 2	Distributed Managed Server	PIA1 (for PeopleSoft Portal) PIA2 (for PeopleSoft Portal) WebLogic Node Manager
Server 3	Distributed Managed Server	PIA3 (for PeopleSoft Portal) PIA4 (for PeopleSoft Portal) WebLogic Node Manager
Server 4	Distributed Managed Server	PIA (for Integration Gateway) WebLogic Node Manager
Server 5	Distributed Managed Server	PSOL (for PeopleBooks) RPS (for httpproxyservlet) WebLogic Node Manager

Common Default Settings

Single-server and multi-server domain configurations have many settings in common.

Domain Defaults

Many of these common settings can be configured in the WebLogic Server Console, but some are configured in other environments. Default values are listed when available.

Setting	Default Value	Where To Configure In the Administration Console
SSL functionality	Enabled with demonstration self-signed digital certificates.	Environment, Servers <i>server name</i> , Configuration, Keystores tab and SSL tab. Command line: pskeymanager
Server logs	<i>Weblogic domain</i> \logs\server_name_*.log	Environment, Servers, <i>server name</i> , Logging.
HTTP access log	Disabled	Environment, Servers, <i>server name</i> , Logging, HTTP.
HTTP keep-Alive	30 seconds	Environment, Servers, <i>server name</i> , Protocols, HTTP.
HTTPS keep-Alive	60 seconds	Environment, Servers, <i>server name</i> , Protocols, HTTP.

Setting	Default Value	Where To Configure In the Administration Console
Low memory settings	various	Environment, Servers, <i>server name</i> , Configuration, Tuning.
Stuck Thread Max Time	600	Environment, Servers, <i>server name</i> , Configuration, Tuning.
Stuck Thread Timer Interval	60	Environment, Servers, <i>server name</i> , Configuration, Tuning.
Enable Administration Port	Disabled	Click the <i>domain name</i> (such as peoplesoft) in the Domain Structure section, then select Configuration, General.
PORTAL HTTP session monitoring	On (applies only to servers running PORTAL)	Deployments, Applications, peoplesoft, PORTAL, Monitoring.
System administrator user ID	system (set during PIA setup)	Security Realms, myrealm, Users and Groups, Users.
System administrator password	password (set during PIA setup)	Security Realms, myrealm, Users and Groups, Users.
System operator user ID	operator	Security Realms, myrealm, Users and Groups, Users.
System operator password	password	Security Realms, myrealm, Users and Groups, Users.
System monitor user ID	monitor	Security Realms, myrealm, Users and Groups, Users.
System monitor password	password	Security Realms, myrealm, Users and Groups, Users.

Script and Environment Defaults

Modify these settings by editing a setEnv script or applying command line parameter overrides to WebLogic control scripts.

The following settings specify the names and structure of various directories on the web server machine.

Setting	Default Value	Description/Override
PS_HOME	(none)	PeopleSoft home directory (set during PIA setup).
BEA_HOME	(none)	High-level install directory (set during PIA setup). Where Tuxedo and WebLogic may be installed.

Setting	Default Value	Description/Override
WL_HOME	(none)	WebLogic home directory (set during PIA setup).
DOMAIN_NAME	peoplesoft	Name of this WebLogic domain (set during PIA setup).
JAVA_HOME	(Depends on the operating system platform.)	Location of Java. Set during PIA setup or with a call to WebLogic's CommEnv script.

Note. You configure Java VM options including JVM memory size using the `JAVA_OPTIONS_OSplatform` parameter, during PIA setup.

The following are miscellaneous settings.

Setting	Default Value	Description/Override
HOSTNAME	<i>Local hostname</i>	Set during PIA setup.
PRODUCTION_MODE	TRUE	Enable WebLogic production mode (set during PIA setup).
DISCOVERY_MODE	FALSE	Disable auto detection of unregistered applications. Script: setEnv
WLS_USER	Operator	Use to stop WebLogic with stop scripts and run it as a Windows service.
WLS_PW	Password	Use to stop WebLogic with stop scripts and run it as a Windows service.
ADMINSERVER_PROTOCOL	HTTP	Protocol used for managed server to connect to administration server (not used in single-server domain).
ADMINSERVER_HOSTNAME	Single-server: <i>local hostname</i> . Multi-server: <i>local hostname</i> . Distributed server: (none — set manually).	Administration server's hostname that managed servers attempt to connect to by default when started. Set during PIA setup (except distributed server).
ADMINSERVER_PORT	Single-server: <i>HTTP port of PIA server</i> . Multi-server: 9999. Distributed server: (none — set manually).	Administration server's Listen port that managed servers attempt to connect to by default when started. Set during PIA setup (except distributed server).

Setting	Default Value	Description/Override
ADMINSERVER_SERVERNAME	Single-server: PIA. Multi-server: WebLogicAdmin. Distributed server: WebLogicAdmin.	WebLogic server instance name of this domain's administration server, used for stopping and starting the server.
WL_VERSION	<i>Detected major WebLogic version.</i>	WebLogic major version, such as 10.
WL_SERVICEPACK	<i>Detected minor WebLogic version.</i>	WebLogic service pack level.
WL_PATCH	<i>Detected WebLogic patch version.</i>	WebLogic patch level.
BACKGROUND_PROCESS	TRUE	Run WebLogic server as a background process. On UNIX you can force foreground execution using the start script's -foreground option.

The following are debugging output settings.

Setting	Default Value	Description/Override
SET CAPTURE_STDOUT_STDERR	FALSE	(Windows only) Capture standard output and standard error of a WebLogic server running as a foreground process. You can also set this with the start script's –capture option.
ENABLE_JDPA_DEBUG	FALSE	(PeopleSoft development only) Enable JDPA debug support. You can also set this with the start script's –debug option.
ENABLE_VERBOSE_GC	FALSE	Enable verbose output of Java's garbage collector. You can also set this with the start script's –verbose:gc option.
ENABLE_VERBOSE_SSL	FALSE	Enable SSL debug support. Produces verbose SSL output. You can also set this with the start script's –verbose:ssl option.
ENABLE_VERBOSE_WL	FALSE	Enable verbose output for the core WebLogic server (not verbose output of PIA). You can also set this with the start script's –verbose:wl option.
MAX_FILE_DESCRIPTOR	4096	The number of open file descriptors set for any Weblogic server process.

The following are HTTP forward proxy support settings.

Setting	Default Value	Description/Override
ENABLE_HTTP_PROXY	FALSE	Enable the use of the forward http proxy.
HTTP_PROXY_HTTPHOST	(none)	IP address or hostname of the forward HTTP proxy server for HTTP requests.
HTTP_PROXY_HTTPPORT	(none)	HTTP Port number of the forward HTTP proxy server for HTTP requests.
HTTP_PROXY_HTTPSHOST	(none)	IP address or hostname of the forward HTTP proxy server for HTTPS requests.
HTTP_PROXY_HTTPSPORT	(none)	HTTP Port number of the forward HTTP proxy server for HTTPS requests.
HTTP_PROXY_NONPROXY_HOSTS	<i>localhost, local hostname, and domainname.</i>	Host names and domain names of content servers that will not be proxied.

WebLogic Domain Directory Structure and Files

This section discusses:

- WebLogic domain directory structure.
- WebLogic domain file listing by type.
- J2EE application files.

WebLogic Domain Directory Structure

In a PeopleSoft implementation, the WebLogic domains are installed into your PS_HOME directory. The major components of the directory structure layout of a PIA install on WebLogic Server are:

Directory	Description
<i>PIA_HOME</i> \webserv	Parent, high-level WebLogic domain directory.
<i>PIA_HOME</i> \webserv\peoplesoft	Default WebLogic domain directory.
<i>PIA_HOME</i> \webserv\peoplesoft\applications\peoplesoft	PeopleSoft application directory.
<i>PIA_HOME</i> \webserv\peoplesoft\bin	WebLogic domain's bin directory, containing numerous administration scripts.
<i>PIA_HOME</i> \webserv\peoplesoft\config	WebLogic domain's configuration directory.
<i>PIA_HOME</i> \webserv\peoplesoft\servers\PIA\logs	WebLogic domain's log directory.

Directory	Description
<i>PIA_HOME</i> \webserv\peoplesoft\keystore	WebLogic key store location for storing keys for configuring SSL.

WebLogic Domain File Listing by Type

This section lists the WebLogic domain files installed by the PeopleSoft installation, organized by file type. Where necessary, each table includes columns that indicate whether a given file is used in a single-server, multi-server, or distributed server configuration.

This listing does not include Java classes or PIA configuration files. On UNIX an equivalent Bourne shell script is provided where a Windows script is listed.

The following table lists WebLogic server administration scripts. All the life cycle scripts are stored in *PIA_HOME*\webserv\domain\bin.

Script	Single-Server	Multi-Server	Distributed Server	Description
setEnv.cmd	X	X	X	Use this script to set required environment variables for the WebLogic server, for example: CLASSPATH, PATH, UNIX Library Path, and JVM options.
startPIA.cmd	X			Use this script to start the WebLogic domain's administration server (the PIA server) in a single-server configuration. On Windows this starts WebLogic as a foreground process. On UNIX this starts WebLogic as a background process. Run the script with <code>-help</code> for usage.
startWebLogicAdmin.cmd		X		Use this script to start the WebLogic domain's administration server (the WebLogicAdmin server) in a multi-server configuration. On Windows this starts WebLogic as a foreground process. On UNIX this starts WebLogic as a background process. Run the script with <code>-help</code> for usage.
startManagedWebLogic.cmd		X	X	Use this script to start a WebLogic managed server. All WebLogic servers in a WebLogic domain other than the administration server are WebLogic managed servers. Run the script with <code>-help</code> for usage.

Script	Single-Server	Multi-Server	Distributed Server	Description
stopPIA.cmd	X			Use this script to stop the WebLogic PIA server. Run the script with –help for usage.
stopWebLogic.cmd		X	X	Use this script to stop WebLogic servers. Run the script with –help for usage.
InstallNTservicePIA.cmd	X			(Windows only) Use this script to install the WebLogic PIA server as a Windows service. The service name is <i>WebLogic_domain-PIA</i> . Run the script with –help for usage.
InstallNTservice.cmd		X	X	(Windows only) Use this script to install a WebLogic server as a Windows service. The service name is <i>WebLogic_domain-server_name</i> . Run the script with –help for usage.
uninstallNTServicePIA.cmd	X			(Windows only) Use this script to uninstall the WebLogic PIA server Windows service. Run the script with –help for usage.
uninstallNTService.cmd		X	X	(Windows only) Use this script to uninstall a WebLogic server Windows service. Run the script with –help for usage.
pskeymanager.cmd	X	X	X	Use this script to manage the JKS keystore used by WebLogic Server, which is in <i>WebLogic_domain\keystore\pskey</i> . SSL certificates for WebLogic Server are stored in this keystore. PeopleSoft Integration Gateway can also share this keystore. Run the script with –help for usage.
startWebLogicBuilder.cmd	X	X	X	Use this script to start WebLogic Builder, which is used to change local application deployment descriptors.
createThreadDump.cmd	X	X	X	Use this script to create a JVM Thread dump. Run the script with –help for usage.

The following table lists WebLogic server configuration files stored in PS_HOME/webserv/domain/config folder under

File	Single-Server	Multi-Server	Distributed Server	Description
config.xml	X	X		This file stores the WebLogic domain configuration, including information about server names, ports, IP addresses, webapps, and SSL. Edit these settings using the WebLogic administration console: http://webserver:port/console .
msi-config.xml		X	X	This is a version of config.xml that's copied for use with a distributed managed server configuration. It's automatically replicated from the original config.xml after a managed server successfully starts.
boot.properties	X	X	X	This file contains the WebLogic system ID and password used for administering the WebLogic domain.
fileRealm.properties	X	X	X	This file is used by WebLogic's internal LDAP server for system administration.
DefaultAuthenticatorInit.ldif	X	X	X	This file is used by WebLogic's internal LDAP server for system administration.
DefaultRoleMapperInit.ldif	X	X	X	This file is used by WebLogic's internal LDAP server for system administration.
SerializedSystemIni.dat	X	X	X	This file is used by WebLogic's internal LDAP server for system administration.

The following table lists PeopleSoft J2EE application scripts, which are all used with Integration Broker, and can be used with every WebLogic server configuration.

Script	Description
BatchProjectExecutor.bat	Use this script for Integration Broker batch EIP testing.
HashKeyGenerator.bat	Use this script to generate a hash key used for Integration Gateway playback.
MessageExport.bat	Use this Integration Broker script for extracting transaction data from request and response data.
StartSendMaster.bat	This is a Integration Broker test utility.

The following table lists miscellaneous files, which can be used with every WebLogic server configuration.

File	Description
Businterlink.txt	This file is used by PeopleSoft's Business Interlinks servlet for loading PeopleSoft libraries when needed.

File	Description
piaInstallLog.xml	This is the PIA install log.
PSCipher.bat	Use this script for encrypting Integration Broker passwords.

See Also

Enterprise PeopleTools 8.50 PeopleBook: Integration Broker

J2EE Application Files

In addition to WebLogic domain configuration files, application descriptors are installed with the PeopleSoft J2EE enterprise application. The following table lists these descriptor files. The path shown for each file is relative to *PIA_HOME*\webserv\WebLogic_domain\applications\.

File	Description
peoplesoft\META-INF\MANIFEST.MF	Use this script to set required environment variables for the WebLogic server, for example: CLASSPATH, PATH, UNIX Library Path, and JVM options.
peoplesoft\META-INF\application.xml	This file contains a list of the webapps that comprise the PeopleSoft J2EE enterprise application.
peoplesoft\PORTAL\WEB-INF\web.xml	This file is the web application descriptor for the PORTAL webapp. It lists all of the servlets deployed as part of that application.
peoplesoft\PORTAL\WEB-INF\weblogic.xml	This file is the PORTAL web application extension descriptor. It specifies, among other things, the HTTP session cookie name, optional cookie domain, and context path of this application.
peoplesoft\PSIGW\WEB-INF\web.xml	This file is the web application descriptor for the PeopleSoft Integration Gateway (PSIGW) webapp. It lists all of the servlets deployed as part of that application.
peoplesoft\PSIGW\WEB-INF\weblogic.xml	This file is the PSIGW web application extension descriptor. It specifies the context path of this application.
peoplesoft\PSEMHUB\WEB-INF\web.xml	This file is the web application descriptor for the PeopleSoft Environment Framework (PSEMHUB) webapp. It lists all of the servlets deployed as part of that application.
peoplesoft\PSEMHUB\WEB-INF\weblogic.xml	This file is the PSEMHUB web application extension descriptor. It specifies the context path of this application.

File	Description
peoplesoft\PSOL\WEB-INF\web.xml	This file is the web application descriptor for the PeopleSoft Online Library (PSOL) webapp (PeopleBooks). It lists all of the servlets deployed as part of that application.
peoplesoft\PSOL\WEB-INF\weblogic.xml	This file is the PSOL web application extension descriptor. It specifies the context path of this application.
peoplesoft\PSINTERLINKS\WEB-INF\web.xml	This file is the web application descriptor for the PeopleSoft Business Interlinks (PSINTERLINKS) webapp. It lists all of the servlets deployed as part of that application.
peoplesoft\PSINTERLINKS\WEB-INF\weblogic.xml	This file is the PSINTERLINKS web application extension descriptor. It specifies the context path of this application.
HttpProxtServlet\WEB-INF\web.xml	This file is the web application descriptor for the WebLogic Server Reverse Proxy Server (RPS) webapp that's used to proxy content from a single WebLogic server. It lists all of the servlets deployed as part of that application.
HttpProxyServlet\WEB-INF\weblogic.xml	This file is the single-server RPS web application extension descriptor. It specifies the context path of this application.
HttpClusterServlet\WEB-INF\web.xml	This file is the web application descriptor for the WebLogic Server Reverse Proxy Server (RPS) webapp that's used to proxy content from a cluster of WebLogic servers. It lists all of the servlets deployed as part of that application.
HttpClusterServlet\WEB-INF\weblogic.xml	This file is the multi-server RPS web application extension descriptor. It specifies the context path of this application.

PIA Install and Reinstall Options

The PeopleSoft Internet Architecture (PIA) installer enables you to create a new WebLogic server domain or update a valid existing WebLogic domain. A valid domain is a domain built by the PIA installer in the *PS_HOME* directory that you specify.

Depending on which option you select, you're prompted for additional information relevant to that selection. When creating a new domain, you're prompted to select from three configuration types: Single-server, multi-server and distributed managed server. If you select to update an existing domain, you're prompted to indicate which domain you would like to update and what type of update you would like to perform. These options are described in detail in your Enterprise PeopleTools installation documentation.

See Also

Enterprise PeopleTools 8.50 Installation for your platform: "Setting Up the PeopleSoft Pure Internet Architecture in GUI Mode"

Administering a WebLogic Server Life Cycle

This section provides an overview of the WebLogic server life cycle and discusses how to:

- Start and stop single-server processes.
- Start and stop multi-server processes.
- Start and stop a distributed managed server.

See Also

[Chapter 7, "Working with Oracle WebLogic," Starting WebLogic, page 121](#)

[Chapter 7, "Working with Oracle WebLogic," Stopping WebLogic, page 123](#)

Understanding the WebLogic Server Life Cycle

You control a WebLogic server's life cycle primarily using a collection of scripts provided in that server's WebLogic domain directory. Each instance of a WebLogic server runs in an isolated Java Runtime Environment (JRE), regardless of whether you're testing with a single-server configuration or implementing a multi-server configuration for production. All scripts must be launched from the WebLogic domain directory; and provide usage syntax if run with `–help`.

Starting and Stopping Single-Server Processes

In a single-server configuration, there's only one server to administer: PIA. You can control the life cycle of the PIA server using scripts or in the WebLogic console.

Scripts

For all platforms:

startPIA Use this script to start the WebLogic server locally.

stopPIA Use this script to connect to a locally running WebLogic server and issue a shutdown command through WebLogic APIs.

Note. When you shut down the server, a warning is displayed since the shutdown command uses a non-SSL http connection to connect to the WebLogic Server. This shutdown command can be changed to use the SSL connection by editing the `stopPIA.sh` script. To use the SSL connection the shutdown command will be the following.

For Windows only:

- installNTservicePIA

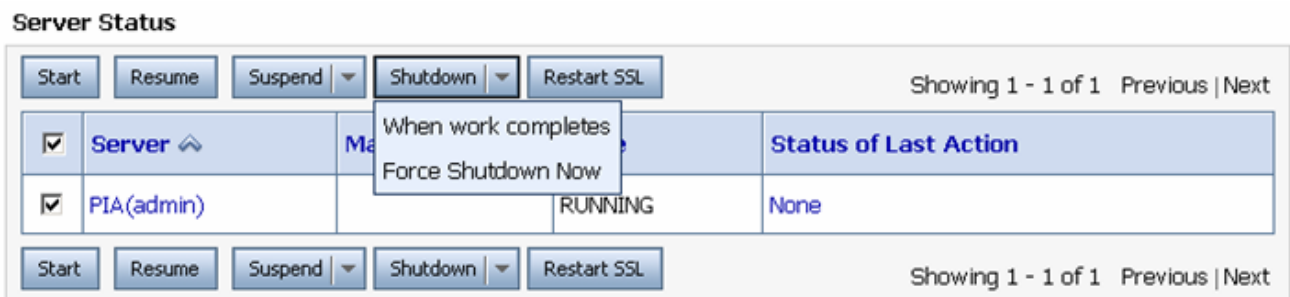
Use this script to register the PIA WebLogic server as a Windows service that runs as a background process. the service is named as *WebLogicDomainName-PIA*, for example: peoplesoft-PIA.
- uninstallNTservicePIA

Use this script to deregister the PIA Windows service.

WebLogic Console

A WebLogic server can also be shut down from the Administration Console. Sign in to the console at `http://webserver:port/console` and perform either of the following.

Note. Before you perform any action in the WebLogic console, you have to click the Lock and Edit button and then the Activate Changes button after the changes are done.



Shutting down from the console

In the navigation tree on the left, expand your domain, click Environment, Servers. Click PIA and select the Control tab. Select the check box for the server that you would like to shutdown, and click Shutdown. You have these options:

- When work completes

This option enables transactions in progress to complete before shutting down the server. To terminate all HTTP sessions immediately, you can first select Ignore Sessions During Shutdown.
- Force Shutdown Now

Immediately terminate all HTTP sessions and transactions in progress, and shut down the server.

Starting and Stopping Multi-Server Processes

In a Multi-server configuration, as the title implies there are multiple servers to administer. Controlling the life cycle of these servers can be done via scripts, the WebLogic console and the WebLogic Node Manager.

Scripts

For all platforms:

- startWebLogicAdmin

Use this script to start the WebLogicAdmin server.

startManagedWebLogic Use this script to start a WebLogic managed server. All of the servers defined in a multi-server domain, except the WebLogicAdmin server, are controlled as managed servers. For example, to start PIA1 as a managed server run `startManagedWebLogic PIA1`.

stopWebLogic Use this script to connect to a locally running WebLogic server and issue a shutdown command using WebLogic APIs. A remote distributed managed server can be shut down using a local administration server.

For Windows only:

installNTservice.cmd Use this script to register a WebLogic server as a Windows service that runs as a background process. The service is named as *WebLogicDomainName-ServerName*. For example, to define the PIA1 managed server as a Windows service, run `installNTservice PIA1`. To define the WebLogicAdmin server as a Windows service, simply run `installNTservice`.

UninstallNTservice.cmd Use this script to deregister a WebLogic server that's defined as a Windows service.

Consider the following when using scripts with managed servers:

Operating System	Consideration
All	<p>When starting a WebLogic managed server it will attempt to connect to its administration server. A managed server's administration server is specified either as a command line parameter when starting the managed server, or using the three administration server environment variables in <code>setEnv</code>, specifically <code>ADMINSERVER_PROTOCOL</code>, <code>ADMINSERVER_PORT</code>, and <code>ADMINSERVER_HOSTNAME</code>. The first time a managed server starts, it <i>must</i> connect to its administration server. If on subsequent startups the administration server is not available, the managed server starts up in Managed Server Independence (MSI) mode by using its locally replicated <code>msi-config.xml</code>. A managed server running in MSI mode can't be administered from a console, so this situation should only be encountered when it is imperative that the managed server be started even though the administration server is not running. Once the administration server is back online, running managed servers that were not previously known by the administration server to be running may be rediscovered using WebLogic's command line utility <code>java weblogic.Admin DISCOVERMANAGEDSERVER</code>, or you can just restart the managed server.</p> <p>To use WebLogic's java command line utility classes run <code>setEnv</code> to set up your environment, then run <code>java weblogic.Admin</code> for usage.</p>

Operating System	Consideration
Windows	<p>When running a WebLogic managed server as a Windows service, the managed server's administration server <i>must</i> be running. When installing a managed server as a Windows service, the managed server service can be configured to be dependent on its local administration server. To configure a managed server service to be dependent on its local admin server service use the – <i>depends</i> option of installNTservice.cmd when defining the Windows service for the managed server. In addition, when the administration server is also a Windows service, you must define it using the following command:</p> <pre>installNTservice.cmd –delay <i>interval</i></pre> <p>Where <i>interval</i> is a period in milliseconds, for example <i>6000</i>. This allows the administration server sufficient time to start before the managed server starts.</p>

WebLogic Server Console

A WebLogic server can also be shut down from its administration console in the same way as single server environments.

WebLogic Node Manager

The WebLogic Node Manager provides the ability to start a WebLogic managed server from the WebLogic Server Console. In addition, the console provides a way to automatically restart a failed server. As with all WebLogic servers, the WebLogic Node Manager runs isolated in its own JRE, and on Windows it can also run as a Windows service. The WebLogic Node Manager binds to a unique IP address and port at startup and accepts lifecycle commands from a WebLogic administration server.

Multiple WebLogic domains running on a single machine can have its managed servers administered by a shared WebLogic Node Manager, as long as each WebLogic domain uses the same version of WebLogic.

The following table lists the WebLogic Node Manager files that are provided with WebLogic server, not the PIA install. These files are located in *BEA_HOME\wlserver 10.3\server\bin*, not your WebLogic domain directory created within the PeopleSoft directory structure.

File	Description
startNodeManager.cmd	Use this script to start the WebLogic Node Manager as a foreground process.
installNodeMgrSvc.cmd	Use this script to define the WebLogic Node Manager as a Windows service that runs as a background process. The service is called WebLogic Platform NodeManager.
uninstallNodeMgrSvc.cmd	Use this script to uninstall the WebLogic Node Manager as a Windows service.
nodemanager.properties	This is the WebLogic Node Manager configuration file.

BEA_HOME\wlserver 10.3\common\nodemanager\NodeManagerLogs\ is the default logs directory for WebLogic Node Manager.

Additional configuration files are located in ...\\common\\nodemanager.

To configure Node Manager:

1. Start the Admin Server.
2. Signon to the Administration Console.
3. Select Environment, Machine.

Create one local machine (as in MachineLocal) for all of the managed servers running in the local server and one remote machine (as in MachineRemoteX) for each of the managed servers running in remote servers.

4. Configure machines.

Click each of the machines created. Under the Configuration tab, select the Node Manger tab.

Choose *Plain* from the Type drop down list.

For Listen Address, enter the IP address of the server on which Node Manager is running.

Enter the Node Manger port number into the Listen Port text box. The default port number is 5556. This can be changed by updating the NodeManager.properties file.

5. Add managed servers.

Under the Servers tab, add the managed servers into the machine. For example, you may have two managed servers running locally, with *PIA1* and *RPS* assigned to *MachineLocal*.

6. Start the Node Manager.

Start the Node Manger from your local WebLogic install directory (BEA Home for the WebLogic Sever on which your PeopleSoft domain is referring to). Node Manager can be started by startNodeManager.cmd/.sh script under BEA_Home\\wlserver_10.3\\server\\bin directory.

After the Node Manger has been started for the first time, a file called nodemanager.properties appears under the BEA_Home\\wlserver_10.3\\common\\nodemanager directory.

7. Modify the nodemanager.properties file.

Stop the Node Manger and open the nodemanager.properties file in a text editor, and make the following changes and save the file.

SecureListener=false

ListenAddress= the IP of the box where the Node Manager is running

Note. In the same file you can see the default port number is set as ListenPort=5556. Changing the port setting here and restarting the Node Manager will set the Node Manager to be listening on the newly configured port number. The port setting here must match with the one set in the machine configuration.

8. Restart the Node Manager.

9. Once the Node Manager is up and running, it should be reachable from the machine configured to listen to the local Node Manager (as in, MachineLocal). This can be confirmed by the Node Manager Status tab under the Monitoring tab.
10. In the remote box hosting the managed server, follow the previous steps to configure the Node Manager.
11. From the bin directory under your PS_Domain, run `setEnv.cmd/sh` to set up the environment.
12. Start WLST using the command `java weblogic.WLST`.
13. Connect the remote server to the Admin Server running on the local box.

Following is the syntax of the `connect` command:

```
connect('username','password','t3://Local Machine's IP:port on which the admin
server is running on Local machine')
```

14. Enroll the remote domain and Node Manager into the Admin Server running on the local box.

The command is `nmEnroll` with the following syntax:

```
nmEnroll('full path to the distributed PeopleSoft Domain
','BEA_Home\wlserver_10.3\common\nodemanager')
```

For example:

```
nmEnroll('D:\PT850-809R2\webserve\Dis4103Installed603','D:\Wls103Installed603⇒
\wlserver_10.3
\common\nodemanager')
```

15. Once the domain and Node Manager have been enrolled, check the Admin Console to confirm that the Node Manager can be reached from the server that was configured to listen to the remote Node Manager.

Once a Node Manager is reached from a "machine," the Admin Console will be able to start and stop the managed server assigned to that "machine."

The Servers tab shows the list of servers. The Control tab grants the control of those servers to the Admin Console.

Starting and Stopping a Distributed Managed Server

In a multi-server configuration, a distributed managed server is simply a managed server that isn't started from the same physical location as its domain's administration server. You can control the life cycle of these servers using scripts, the WebLogic Server Console and the WebLogic Node Manager.

See [Appendix B, "WebLogic Managed Server Architecture," Starting and Stopping Multi-Server Processes, page 338.](#)

Tuning Performance and Monitoring Resources

Monitoring the performance of a WebLogic instance is primarily performed in the Administration Console. This section discusses how to:

- Manage JVM heap size and execute thread usage.

- Monitor HTTP session count for PeopleSoft portal.

See Also

Chapter 7, "Working with Oracle WebLogic," Using WebLogic Server Administration Console to Monitor PeopleSoft Sessions, page 124

Managing JVM Heap Size

This section discusses how to:

- Monitor JVM heap.
- Change JVM heap size.

Monitoring JVM Heap

The JVM heap size is the amount of memory that a particular JRE (Java Runtime Environment) gives to the JVM (Java Virtual Machine) that it creates. The `java.exe` command on Windows, `java` on UNIX and `beasvc.exe` when running WebLogic as a Windows service is the JRE, and the JVM exists within the JRE's memory space. The primary sources for monitoring the amount of memory that is in use within a JVM are the Administration Console and the WebLogic logs.

To monitor the amount of the JVM heap size available and in use:

1. Sign on to the Administration Console by entering the following URL in a browser:

`http://webserver:9999/console`

Where *webserver* is the hostname of the WebLogic server.

2. Traverse the following in the navigation tree on the left:
 - a. Expand your WebLogic domain (for example, peoplesoft).
 - b. Expand Servers.
3. Click the server you intend to monitor (for example, PIA).
4. Select the Monitoring tab, and the Performance sub-tab.

Changing the JVM Heap Size

If you need to adjust any of the Java options, most commonly the JVM heap size, you must manually edit that WebLogic domain's local `setEnv` script. The parameters, `-Xms` and `-Xmx`, control the JVM memory minimum and maximum heap size respectively.

Following are examples of the JVM heap size as specified in `setEnv` using the `JAVA_OPTIONS_OSplatform` environment variable. You only need to set the variables that correspond to the operating system where the WebLogic server is running.

- `JAVA_OPTIONS_WIN32="-server -Xms256m -Xmx256m -XX:MaxPermSize=128m"`

- `JAVA_OPTIONS_AIX="-Xms128m -Xmx256m"`
- `JAVA_OPTIONS_HPUX="-server -Xms256m -Xmx256m -XX:MaxPermSize=128m"`
- `JAVA_OPTIONS_LINUX="-jrockit -Xms256m -Xmx256m"`
- `JAVA_OPTIONS_SOLARIS="-server -Xms256m -Xmx256m -XX:MaxPermSize=128m"`

Note. If you do adjust any of the Java options, most commonly the JVM heap size, you must restart WebLogic for these changes to take effect.

If you're running WebLogic Server as a Windows service you must rerun the `installNTservice` script to propagate this change into the Windows registry.

The WebLogic Node Manager does not use the Java options set in `setEnv`, but instead uses Java options set from the WebLogic console.

To modify the Java options for a WebLogic instance started via the WebLogic Node Manager:

1. Sign on to the WebLogic Server Console by entering the following URL in a browser:

`http://webserver:9999/console`

Where *webserver* is the hostname of the WebLogic server.

2. Expand your WebLogic domain (for example, `peoplesoft`) and click Environment, then Servers.
3. Select the managed server you intend to modify.
4. Select the Configuration tab, and the Server Start sub-tab.
5. Update the Arguments field.
6. Click Save.

See Also

[Chapter 7, "Working with Oracle WebLogic," Adjusting the JVM Heap Size, page 149](#)

Monitoring HTTP Session Count for PeopleSoft Portal

In addition to memory and thread usage, it's also possible to monitor the number of established HTTP sessions used in conjunction with the PeopleSoft PORTAL application. This number, although not necessarily directly related to current performance, is an indicator of the following performance factors:

- JVM memory used to store HTTP session data.
- Current number of logged on clients.
- Peak number of logged on clients.
- Idle time of logged on clients.

To monitor the total number of HTTP sessions:

1. Sign on to the Administration Console.
2. In the Domain Structure section, click Deployments and select the check box next to peoplesoft.
3. Select the Monitoring tab.

Changing Configuration Settings

This section provides an overview of the WebLogic server configuration files, and discusses how to:

- Change the WebLogicAdmin server's listen ports.
- Change application and server deployment targets.

Understanding the WebLogic Server Configuration Files

WebLogic server configuration settings are stored in a collection of files, primarily these include: script, config.xml, and the web.xml and weblogic.xml for each webapp.

<i>Configuration File</i>	<i>Description</i>
setEnv script	SetEnv contains statically and dynamically defined environment variables. It's called from all of the WebLogic administration scripts to assist in building the Java command line. You modify this file using a text editor.
config.xml	Config.xml contains server runtime settings, such as the HTTP port. You modify this file using the Administration Console.
web.xml weblogic.xml	Located in the WEB-INF directory of each servlet, providing web application descriptors and settings relevant to their application.

Changing the WebLogicAdmin Server's Listen Ports

In the multi-server configuration, several parameters are set based on the environment detected and delivered defaults. One such parameter is the HTTP port of the WebLogicAdmin server. By default the WebLogicAdmin server's HTTP listen port is 9999.

To change this value:

1. Start the WebLogicAdmin server using the startWebLogicAdmin script.
2. Sign on to the WebLogic Server Console by entering the following URL in a browser:

`http://webserver:9999/console`

Where *webserver* is the hostname of the WebLogic server.

3. Navigate to Servers, WebLogicAdmin, Configuration, General.
4. Modify the value of the Listen Port field.
5. Click Apply.
6. Restart the WebLogic server.

If you can't initially start the server due to a port conflict, you can manually edit the value of the ListenPort parameter in that domain's config.xml file. Creating a backup of config.xml is recommended before manually changing this file.

After changing the ListenPort value in your domain's config.xml, either directly or using the console, you must also update your setEnv script. Update the ADMINSERVER_PORT environment variable to reflect the new HTTP port. This setting is used by the stopWebLogic and startManagedWebLogic scripts as the default administration server HTTP port.

Changing Application and Server Deployment Targets

With WebLogic, J2EE applications are targeted to any combination of WebLogic servers and WebLogic clusters. A WebLogic cluster is a logical grouping of servers, generally all providing the same application, though that's not a requirement. To change the servers or clusters that to which an application is targeted and deployed, sign on to the Administration Console and update the application's target assignments. You can view and modify application and server target assignments on the Deployments, Applications tabs, and on the Targets tab for each server.

Following is an example of how to change the target assignments of the PeopleSoft Integration Gateway (PSIGW) web application so it's the only application targeted to the PIA server, and is the sole application on that instance.

To change the target assignments of the PeopleSoft Integration Gateway web application:

1. Sign on to the WebLogic Server Console.
2. In the Domain Structure section:
 - a. Expand peoplesoft.
 - b. Expand Deployments.
 - c. Expand Applications.
 - d. Expand peoplesoft.
3. Select PSIGW.
4. Select the Targets tab.
5. In the Clusters section, clear the peoplesoftCluster check box.
6. Click Apply.
7. In the navigation tree, select PORTAL.
8. Select the Targets tab.

9. In the Independent Servers grid, clear the check box for targeting the PORTAL to this server.
10. Click Apply.
11. Repeat steps 7 to 10 for the PSEMHUB, PSINTERLINKS and PSOL.

To deploy an application to a cluster, target the server to the cluster and target the application to the cluster.

Appendix C

PeopleSoft Timeout Settings

This appendix discusses:

- Web server timeouts.
- Application server timeouts.
- Process Scheduler timeouts.
- PIA timeouts.

Web Server Timeouts

You specify web server timeouts using the Web Profile Configuration component (WEB_PROFILE). To access these settings in PIA, select PeopleTools, Web Profile, Web Profile Configuration, then select the appropriate page.

The following table provides basic information about the web server timeout settings, which are more completely documented in the *Internet Technology PeopleBook*.

Page Element	Page Name	Description	Default
Inactivity Warning	Security	<p>Specify how long the portal should wait before warning users that their browser session is about to expire. They can continue with their current session by clicking the OK button in the message.</p> <p>If a user doesn't respond, the session ends and the expired connection page appears.</p> <p>Suppress this warning by setting this value to be greater than the sessionTimeout value.</p>	1080 seconds (18 minutes)
Inactivity Logout	Security	<p>Specify the inactivity timeout interval of the PeopleSoft application for which the user is currently authenticated. When the interval passes with no user activity, the user's browser displays the page specified by the Expire Page - Page field on the Web Profile Configuration - Look and Feel page.</p> <p>Note. Depending on the application implementation, authenticated users might also experience an HTTP session inactivity timeout.</p>	1200 seconds (20 minutes)

Page Element	Page Name	Description	Default
Authenticated Users - HTTP Session Inactivity	Security	Specify the HTTP session inactivity timeout interval that applies to authenticated users. When the interval passes with no user activity, the web server discards all session information, including cached page states. The next time the user submits a request, the web server creates a new HTTP session. If not set, the HTTP interval for an authenticated user is the same value as the inactivity logout.	0 seconds for all profile types.
Public Users - HTTP Session Inactivity	Security	Specify in seconds the inactivity timeout interval that applies to public users. When the interval passes with no user activity, the web server discards all session information, including cached page states. The next time the user submits a request, the web server creates a new HTTP session. Unlike authenticated users, public users are not signed out of their PeopleSoft application when this interval expires. However, PIA releases their application states from memory. If users click a link, they regain access to the application at the search dialog. This setting prevents an overload of web server resources for inactive public users.	DEV, KIOSK profile: 1200 seconds (20 minutes). TEST, PROD profile: not set.
Disconnect Timeout	Security	Specify the amount of time to wait before disconnecting the Jolt connection. A value of 0 seconds (the default) means no limit. This means that the client connection must be retained throughout the session. If the connection becomes invalid (due to one of the other timeouts) the session will be expired. Note. If you specify 0 seconds, the Jolt client attempts to connect the Jolt Server Handler (JSH) in RETAINED mode. If any positive value is specified, the Jolt client attempts to connect the JSH in RECONNECT mode.	0 seconds
Send Timeout	Security	Specify the maximum time permitted between the sending of the Jolt Request by the client servlet and its full receipt on the application server. Note. You might need to increase this value where a large amount of data is being sent to the application server, or the network is slow.	50 seconds

Page Element	Page Name	Description	Default
Receive Timeout	Security	<p>Specify how long the client servlet should wait after issuing a Jolt Request for a response from the application server.</p> <p>This value should be considerably larger than the Send Timeout. Make sure that this value is also greater than your application server online service timeouts, such as the Service Timeout setting for PSAPPSRV that appear in the PSAPPSRV.CFG configuration file on the application server.</p> <p>Note. Ideally, this timeout should also be greater than the Tuxedo SANITY_SCAN setting (BLOCKTIME * SCANUNIT).</p>	1300 seconds

See Also

Enterprise PeopleTools 8.50 PeopleBook: PeopleTools Portal Technologies, "Configuring the Portal Environment," Configuring Portal Security

Session-Timeout

You specify the web server *session-timeout* using the Inactivity Logout property and HTTP Session Inactivity property in the web profile.

Web Server Default System Timeout

PeopleSoft portal technology normally depends on a content reference timeout setting to determine how long to wait for a pagelet to load before it considers the pagelet to be unavailable. However, if the remote server is unavailable, the content reference timeout setting is ignored. If the portal can't establish a connection to the remote host, it uses the default system timeout.

The default system timeout defaults to 20 seconds. If you expect the remote server to be slow or down for longer than 20 seconds, you should specify a longer default system timeout, by configuring your web server to set the defaultConnectTimeout JVM environment variable to an appropriate value using one of the following procedures.

For example,

```
SET JAVA_OPTIONS_WIN32=-server -Xms32m -Xmx200m
-XX:MaxPermSize=128m
-Dsun.net.client.defaultConnectTimeout=default_timeout
```

Where *default_timeout* is the number of milliseconds that the portal should wait to establish the connection to the host.

See Your web server documentation for instructions on modifying this JVM environment variable.

Application Server Timeouts

All configurable settings for the application server require modification in PSADMIN:

Name	In This File	Description	Default
JOLT Listener/Client CleanupTimeout	psappsrv.cfg	<p>Specify the inactivity interval permitted for the server-side JoltSession.</p> <p>Specifying too low a value can cause unnecessary reinstantiation of resources for clients who surpass this inactivity interval. However, specifying too high a value can keep unnecessary server-side resources allocated.</p> <p>Note. This setting doesn't affect the user experience, but it has an impact on server-side performance.</p>	10 minutes
JOLT Listener/Init Timeout	psappsrv.cfg	<p>Specify the amount of time that's allowed for the JSL process to start.</p> <p>Note. It's not necessary to adjust this setting.</p>	5 minutes
Workstation Listener/Client Cleanup Timeout	psappsrv.cfg	<p>Specify the inactivity interval permitted for the server-side Workstation Listener Session.</p> <p>Specifying too low a value can cause unnecessary reinstantiation of resources for clients who surpass this inactivity interval. However, specifying too high a value can keep unnecessary server-side resources allocated.</p> <p>Note. This value is required only for three-tier connections.</p>	60 minutes
Workstation Listener/init Timeout	psappsrv.cfg	<p>Specify the amount of time that's allowed for the WSL process to start.</p> <p>Note. It's not necessary to adjust this setting.</p> <p>This value is required only for three-tier connections.</p>	5 minutes

Name	In This File	Description	Default
Spawn Threshold	psappsrv.cfg	<p>Applies only if spawning is enabled.</p> <p>Specify the rates at which PSAPPSRV processes spawn and decay.</p> <p>The spawn rate is determined by the last two numbers, and the decay rate is determined by the first two numbers.</p> <p>Using the default value as an example, for the spawn rate of <i>1,1</i> an extra PSAPPSRV process is spawned if there is at least 1 outstanding service request on the application server request queue for 1 second or more. This spawning will continue until the PSAPPSRV Max Instances value is reached.</p> <p>For the decay rate of <i>1,600</i> a server process is decayed if less than 1 service request is in the application server request queue for 600 seconds (ten minutes) or more.</p> <p>Note. This parameter applies only if, for PSAPPSRV, the value of <i>Max Instances</i> is greater than that of <i>Min Instances</i>.</p>	1,600:1,1
Service Timeout	psappsrv.cfg	<p>Each server process has its own instance of this setting in its section of the psappsrv.cfg file.</p> <p>Specify the maximum interval for services to run in a given process. If a service has not completed within the specified interval, Tuxedo terminates the server processing and restarts the server process.</p> <p>For each server process, specify the longest time that any service is expected to take.</p> <p>Note. A value of 0 produces an indefinite timeout for any service.</p>	PSAPPSRV: 300 seconds (5 minutes) PSSAMSRV: 300 seconds PSQCKSRV: 300 seconds PSQRYSRV: 1200 seconds (20 minutes) PSBRKHND_dflt: 1200 seconds PSSUBHND_dflt: 1200 seconds PSPUBHND_dflt: 1200 seconds
Restart Period (PSBRKDSP_dflt, PSSUBDSP_dflt, PSPUBDSP_dflt)	psappsrv.cfg	Specify how long each dispatcher should wait before redispersing a message if the associated handler has not started processing it.	120 seconds

Name	In This File	Description	Default
TM_RESTARTSRV TIMEOUT	psappsrv.ubx (which is the template for psappsrv.env)	Specify the time period that a domain server process (for example, PSAPPSRV, PSWATCHSRV, PSSAMSRV) is permitted to remain in REStarting mode before it is killed by Tuxedo. This setting resolves processes hanging during restart. Note. To modify this setting, you must change the value in the .UBX template file, then recreate your domain.	60 seconds (one minute)

Process Scheduler Timeouts

All configurable settings for PeopleSoft Process Scheduler require modification through domain configuration within PSADMIN:

Name	In This File	Description	Default
Process Scheduler/Reconnecti on Interval	psprcs.cfg	Specify the interval between attempts to reconnect to the database when the connection is lost.	300 seconds (5 minutes)
Process Scheduler/Authentica tion Timeout	psprcs.cfg	Specify how long PeopleSoft Security has to authenticate a process that's released by PeopleSoft Process Scheduler The timer starts when Process Scheduler initiates the request.	5 minutes
RemoteCall/RCCBL Timeout	psprcs.cfg	Specify the maximum interval for a remote call from an Application Engine program to run before it's terminated. This is similar to a general Tuxedo service timeout.	300 seconds (5 minutes)

For Spawn Threshold, see the application server timeout settings.

Search Server Timeouts

The following are the configurable timeout settings for the search server.

Name	File	Description	Default
Domain Settings / Spawn Threshold	pssrchsrv.cfg	<p>Applies only if spawning is enabled.</p> <p>This is the rate at which PSSRCHSRV processes will spawn and decay. The spawn ratio is determined by the last two digits. The decay ratio is determined by the first two digits. Using the default value as an example, we see that an extra PSSRCHSRV process will be spawned if there is at least 1 outstanding service request on the request queue for one second or more. This spawning will continue until Max Instances is reached. For the decay rate of 1, 600, if less than 1 service request is on the request queue for ten minutes (600 seconds), a server process is decayed. Note: This value is only relevant if $\text{PSSRCHSRV} / \text{Max instances} > \text{PSSRCHSRV} / \text{Min Instances}$.</p>	1,600:1,1
PSSRCHSRV / Service Timeout	pssrchsrv.cfg	This parameter indicates the duration in seconds to run a Search service within a Search domain.	300 secs
TM_RESTARTSRVTIMEOUT	pssrchsrv.ubx (and then UBBGENned into pssrchsrv.env)	The time period that a domain server process PSSRCHSRV, is allowed to remain in Restarting mode before it is killed by the BBL. This resolves processes hanging during restart. This setting is defaulted in the \$ <i>PS_CFG_HOME</i> /appserv/Search/*.UBX files. If this value needs to be changed, you must change the value in the UBX template file and then recreate your domain.	60 secs

PIA Timeouts

A number of additional timeouts may be set through PIA. These settings reflect changes at the database level that may pertain to different groups of users.

Note. The timeout settings in PIA are optional and are not required to run PIA. However, an understanding of how these settings can contribute to a user's session duration is important in the context of other timeout values that appear in this appendix.

Name	Navigation Path	Description	Default
Authentication Token expiration time	PeopleTools, Security, Security Objects, Single Signon	Specify the interval during which the system can trust a single signon token (PS_TOKEN) from the same or another content provider. Note. As long as users remain signed in, the expiration of PS_TOKEN does not affect them. This setting is relevant only for the GetCertificate request during single signon.	720 minutes (12 hours)
Permission List - Time-out Minutes	PeopleTools, Security, Permissions & Roles, Permission Lists	Specify an interval during which a given permission list applies. The interval starts for a user to which the permission list is assigned when that user signs in. When the timeout period elapses, the user's online session is terminated. If a user belongs to multiple permission lists, the largest timeout value from among those permission lists is applied to the user's session during signon. The permission list timeout is effective only if its value is less than the web server session-timeout. This means that all of the permission list timeouts for a given user must be less than the web server session-timeout to be effective. However, the Inactivity Warning timeout still applies. Note. A value of 0 produces an indefinite timeout.	0 minutes

See Also

[Appendix C, "PeopleSoft Timeout Settings," Web Server Timeouts, page 349](#)

Appendix D

Troubleshooting Server Issues

This section describes the solutions to some common issues related to the configuration and administration of your PeopleSoft application server environment.

Uploading Files Using Non-Latin Characters

When uploading an attachment file with a filename that uses non-Latin characters, such as Hebrew or Japanese, the filename on the server gets renamed with the non-Latin characters converted to strings of numbers, and often also leads to getting the error "Attachment Filename is too long (maximum is 64 characters)."

Solution For UNIX

To resolve this issue on UNIX servers:

1. Make sure that the corresponding language locale UTF-8 character set is installed on the operating system level.

Use "locale -a" to get the list of installed locales.

2. Use the LANG environment variable to set for the session that runs the application server a valid locale with the UTF-8 character set.

For example, en_US.utf8, iw_IL.utf8, ja_JP.utf8, and so on. See your "locale -a" output.

3. Use PSADMIN to set the parameter Character Set=utf8 in PSAPPSRV.CFG, and reboot the application server.

The character set of the shell locale (LANG) and PSAPPSRV.CFG (Character Set) must agree, and PSAPPSRV.CFG requires utf8.

Note. If the shell locale is set to something other than the default of C, Tuxedo will issue warnings when the application server is booting, so in such a case add a symlink from \$TUXDIR/prod/locales/<your_shell_locale> to \$TUXDIR/prod/locales/C to resolve that issue.

Solution For Windows

On a Windows-based application server, make sure the server operating system has the corresponding language installed.

WebSphere: Port Set In the Host Header of a Request Returned Incorrectly

This issue applies only to configurations involving the IBM WebSphere web server and either of the following reverse proxy servers:

- Microsoft IIS
- Sun Java System Web Server

Scenario

This issue appears in the following situation (or similar). Assume there is PIA Portal deployed on WebSphere 1, say Site A, which accesses a pagelet in a PIA Portal deployed on WebSphere 2, say Site B. In front of Site A, there is reverse proxy server (Sun Java System Web Server or IIS) configured.

In such a scenario, when a HTTP request is made to the PIA Portal on Site A through an RPS port, Site A returns the HTTP request to fetch the remote pagelet from Site B. The HTTP response from Site B contains the RPS port number instead of the HTTP port of Site A. This is due to the default behavior of WebSphere, which reads the port number from the Request URL instead of from the HTTP Header (HOST field).

See <http://www-01.ibm.com/support/docview.wss?uid=swg1PK55330>.

Solution

This default behavior can be altered by modifying the web container custom properties in WebSphere on Site B.

To modify the custom properties:

1. Open the Administrative Console and select Servers, Server Types, WebSphere application servers, server1, Web Container Settings, web container.
2. Set these custom properties:
 - trusted = false
 - trustheaderport = true
 - com.ibm.ws.webcontainer.extractHostHeaderPort = true
3. Restart the web server.

Index

Numerics/Symbols

%V variable 31

A

analytic server framework

See PSANALYTICSRV, PSANALYTICSRV

analytic servers

understanding 2

Apache HTTP, configuring as an RPS 134

Application Designer 288

Application Engine

activating program tracing 92

running LOADCACHE 251

application server 11

crash replay 277

domain configurations 27

domains 12

processes 12

Application Server Profiles (WebSphere) 155

application servers

configuring 21

configuring for cache/replay files 53

configuring PeopleCode debugging 89

enabling multiple server processes for
debugging 287

loading the cache 251

logical separation 21

physical separation 23

setting database options 79

setting domain parameters 77

setting Integration Broker options 114

setting interface driver options 109

setting JRAD options 86

setting messaging server options 105

setting PSANALYTICSRV options 100

setting PSAPPSRV options 97

setting PSPPMSRV options 115

setting PSQCKSRV options 102

setting PSQRYSRV options 103

setting PSRENSRV options 115

setting PSSAMSRV options 101

setting PSTOOLS options 109

setting remote call options 96

setting security options 81

setting server process options 116

setting startup options 78

setting up the PeopleSoft Windows service
73

setting workstation listener options 81

sharing indexes with Process Scheduler 206

spawning 78

specifying cache settings 94

specifying domain settings 86

specifying search index file location 114

specifying search options 114

specifying SMTP settings 105

specifying timeouts 352

specifying trace options 90

specifying workstation settings 220

using PSADMIN 35

using PSADMIN menus 57

using Tuxedo connect string 221

architecture

application server 11

options 24

PeopleSoft 9

architecture

database 10

auditing

database-level utilities 276

enabling database-level auditing 80

record cross-references 274

authentication timeout 135

B

batch environment 17

BOE

administration 258

browsers

understanding 21

browsers, enabling compression 235

BusinessObjects

administration 258

C

cache files, workstation settings 212

caching

configuring application servers for cache
files 53

enabling 94

file cache 96

loading application server caches 251

memory cache 96

non-shared 67

running LOADCACHE 253

setting cache file location 95

setting memory maximum 96

setting server caching mode 94

specifying settings 94

character sets

checking data field length 243

setting for SMTP servers 106

setting for trace-log files 89

setting for Verity engines 280

setting the codepage for Microsoft
applications 280

setting the default 280

specifying for data processing 111

client

setting up 217

setting up in Configuration Manager 230

setup in Configuration Manager 229

collections

- creating 194
 - opening 194
- column headings, customizing 259
- command-line
 - configuring domains 40
 - creating domains 40
 - specifying options 227
 - understanding the PSADMIN interface 39
 - using options for PeopleSoft Process Scheduler 72
 - using PSADMIN options 39
- components, applying defaults 264
- compression
 - enabling for browsers 235
 - setting for Tuxedo 83
 - setting the Jolt compression threshold 85
 - setting the message compression threshold 114
- configuration files
 - archiving application server 52
 - configuring JRLY 300
 - editing for PeopleSoft Process Scheduler 72
 - editing PSWINSRV.CFG 76
 - using PSADMIN 50
- Configuration Manager
 - running client setup 229, 230
 - setting signon defaults 210
 - specifying command line options 227
 - specifying display settings 212
 - specifying startup settings 210
 - starting 210
 - understanding 209
 - using shortcut links 217
 - using the Client Setup tab 217
 - using the Common tab 226
 - using the Crystal/Bus Interlink tab 214
 - using the Database/Application Server tab 220
 - using the Display tab 212
 - using the Import/Export tab 218
 - using the nVision tab 223
 - using the Process Scheduler tab 222
 - using the Profile tab 219
 - using the Remote Call/AE tab 216
 - using the Trace tab 215
 - using the Workflow tab 216
- Configure Keystores page 146
- configuring a workstation 209
- connect ID, default password 211
- connectivity, verifying 229
- cookies
 - authentication 21
- Copy File Archive page 269

D

- database
 - database-level auditing 276
 - displaying the name 212, 214
 - enabling database-level auditing 80
 - entering a name 211
 - initiating local connection to PeopleSoft
 - database on the same machine 80
 - selecting the type 212
 - setting environment variables 79
 - setting maximum cursors 80
 - setting sign-in values 78
 - setting Sybase TCP packet size 79
 - setting the default type 211
 - setting up remote database connections 265
 - specifying connect IDs 79
 - specifying name 78
 - specifying type 78
 - starting Process Scheduler servers
 - automatically 76
 - validating signon 81
- database server 10
- database servers
 - configuring 21
- Data Mover, workstation settings 226
- DB2 z/OS, defining tablespaces 255
- DDL 256
- debugging
 - configuring for PeopleCode 89
 - enabling for PeopleCode 117
 - enabling multiple PSAPPSRV server processes 287
 - entering Debugger mode 288
 - reproducing crashes 93
 - requesting PSDBGSRV server processes 288
 - setting the PSDBGSRV listener port 286
 - setting up PeopleCode Debugger 285
 - tracing PeopleCode 288
 - tracing SQL 290
 - using three-tier connections 286
 - using two-tier connections 285
 - using utilities 276
- Design a Search Index page 195
- development environment 228
- Development Environment 16
- dirty-reads, using 104
- display
 - adjusting page sizes 213
 - specifying settings 212
 - using the Font options 214
- distributed managed server 325
- domains 12
 - administering 58
 - allowing dynamic changes 88
 - booting 53, 60
 - checking status 61
 - cleaning IPC resources 69
 - configuring 51, 64
 - configuring via command-line 40
 - creating 66
 - creating via command-line 40
 - deleting 66
 - entering network tracing level 89
 - forcing shutdown 61
 - handlers 14
 - listeners 14
 - loading configurations 52
 - logging 89
 - managing 33
 - monitoring 53
 - purging the cache 62
 - queues 14
 - restarting server processes 88
 - server processes 12
 - setting application server domain parameters 77
 - shutting down 60, 61
 - specifying IDs 86
 - specifying settings 86
 - specifying the spawn threshold 87
 - specifying trace-log character set 89

- specifying your database connectivity
 - software directory 86
- starting automatically 76
- starting PSPPMSRV servers 117
- stopping 53
- understanding the PeopleSoft domain 119
- using PSADMIN 35

Domains Gateway, configuring 118

E

- encryption
 - enabling for Jolt listener 84
 - enabling for workstation listener 82
- environment variables
 - accessing class libraries 110
 - improving system performance 79
- errors
 - logging 93
 - notifying by mail 93
 - suppressing application error messages 112
 - suppressing SQL error messages 113
- event notifications 117
- export Configuration Manager 218

F

- failover 221
- file system indexes
 - building 180, 199
 - defining what to index 200
 - setting options 199
 - understanding 178
- Filesystem Options page 199
- functions, XML link function registry 260

G

- Gather utility
 - application server data 273
 - environmental data 273
 - getting started 272
 - including additional files 273
 - in UNIX 272
 - in Windows 272
 - understanding 271
 - web server data 273
- globalization *See* international settings

H

- handler 14
- handlers
 - specifying Jolt handler quantity 84
 - specifying maximum clients per Jolt handler 84
 - specifying maximum clients per workstation handler 82
 - specifying maximum Jolt handler quantity 84

- specifying maximum workstation handler quantity 82
- specifying workstation handler quantity 82
- help
 - setting PeopleTools options 246
 - viewing the Sytem Information help page 234
- high availability 221
- HTTP Gateway page 201
- HTTP Keep-Alive 135
- HTTP servers
 - configuring Apache HTTP as an RPS 134
- HTTP spider indexes
 - building 180, 201
 - defining gateway settings 201
 - defining what to index 203
 - understanding 178

I

- IBM HTTP Server 156
- IBM HTTP Server plug-in 158
- IBM Websphere
 - authentication failure timeout 173
 - JVM heap 173
 - logging 174
 - session timeout 173
 - starting 157
 - tracing 174
- IBM WebSphere
 - application security 170
 - Application Server profiles 155
 - container SSL 165
 - HTTP server 156
 - IHS plug-in 158
 - IIS plug-in 159
 - Integration Solutions Console 154
 - reverse proxy servers 157
 - RPS plug-in 158
 - security 166
 - SSL 163
 - stopping 157
 - Sun Java System Web Server 161
 - usage with PeopleSoft 153
 - web server plug-in 158
- images
 - converting 226
 - creating 93
- ImportPrivateKey *See* SSL
- indexes search indexes
- Informix, specifying server names 79
- installation
 - setting up the development environment 228
 - using workstations 217
- Integration Broker
 - setting message size compression threshold 114
 - thread pool size 114
- Integration Solutions Console
 - securing 166
 - using with PeopleSoft 154
- interface driver 109
- international settings
 - administering time zones 279
 - managing multiple languages 279
 - preferences 278

- sizing process field 279

IPC resources

- cleaning 69

J

J2EE web applications *See* web applications

Java

- adding to CLASSPATH 110
- adjusting JVM heap size 149, 343
- servlets *See Also* Java servlets

Java servlets

- understanding 18

Java VM options 110

Jolt 11

- configuring 117
- listener *See* Jolt listener
- transmitting requests and data 20

Jolt Internet Relay *See Also* JRLY

- implementation considerations 300
- understanding the architecture 299

Jolt listener

- assigning a port 83
- configuring 117
- configuring Jolt 85
- enabling encryption 84
- setting client connection modes 84
- setting client connection request binding time 84
- setting the address 83
- setting the compression threshold 85
- setting timeout 84
- specifying handler quantity 84
- specifying maximum clients per handler 84
- specifying maximum handler quantity 84
- understanding 83

Jolt Relay *See* JRLY, JRLY

Jolt Relay Adapter JRAD

Jolt server handler (JSH) 17

Jolt server listener (JSL) 16

JRAD

- configuring 117, 302
- setting the listener address 86
- setting the listener port 86
- understanding 86, 298

JRLY *See* JRLY

- assigning the JRAD listener port 86
- configuring 300
- configuring JRAD 117
- running 303
- running on UNIX 304
- running on Windows 303
- understanding 298
- using the administration program 303

JVM heap

- monitoring 343
- sizing 149, 343

K

Keep-Alive, HTTP 135

keys

- modifying VdkVgwKey 207
- updating keystore properties 146

keystores, updating properties 146

L

language preference 213

languages

- managing 279
- specifying PeopleTools settings 239
- using the spell check system dictionary 249

listener 14

Load Application Server Cache page 253

load balancing 129

LOADCACHE 253

logging

- editing log files 65
- enabling/disabling HTTP access log 151
- enabling for analytic servers 92
- setting severity level for PSRENSRV process 115
- setting the level for SQL tracing for all clients 90
- setting the level for SQL tracing for individual clients 90
- tracing email details to log file 107
- viewing crash log information 93
- writing error information 93

look-up pages 100

M

Manage Installed Languages page 279

MCF servers, configuring 117

menus

- using PSADMIN 57
- using Quick-Configure 38

merchant integration 260

Message Catalog page 247

messaging

- adding/maintaining system messages 247
- setting server options 105

metadata caching 252

Microsoft Internet Information Server (ISS) 126

Microsoft Windows services

- configuring the PeopleSoft service 74
- editing PSWINSRV.CFG 76
- monitoring the executables 75
- understanding 73

MIME indexing 180, 193, 200

Mobile Applications 270

MS IIS, configuring as an RPS 126

Multipurpose Internet Mail Extensions

- See* MIME indexing

N

navigator 214

network tracing 89

non proxy hosts

- specifying for PSTOOLS 111

notifications

- configuring event 117
- designing real time event notification (REN)

- 115
- logging error 93
- nVision
 - administration 270
- nVision workstation settings 223

O

- operator
 - specifying signon settings 210
 - using overrides 212
- optimization
 - utilities 281
- Options page, converting panels to pages 263
- Oracle
 - initiating local connection to PeopleSoft database 80
 - Jolt 11
 - Transparent Data Encryption (TDE) 247
 - Tuxedo 11
- Oracle WebLogic *See* WebLogic

P

- pages
 - converting from panels 262
 - displaying 213
 - displaying in navigator 213
- panels, converting to pages 262
- passwords
 - changing for WebLogic users 136
 - specifying for connect IDs 79
 - specifying for users 79
- PeopleCode
 - configuring debugging for 89
 - enabling debugging 117
 - setting up the Debugger 285
 - using the debugger 288
- PeopleCode trace settings 215
- PeopleSoft
 - architecture 9
- PeopleSoft Application Server
 - preloading cache 67
- PeopleSoft application server process
 - See* PSAPPSRV
- PeopleSoft Internet Architecture
 - browsers *See Also* browsers
 - configuring 21
- PeopleSoft Mobile Applications
 - See* Mobile Applications, PeopleSoft
- PeopleSoft Ping Chart page 282
- PeopleSoft Ping page 281
- PeopleSoft Process Scheduler
 - cleaning IPC resources 73
 - configuring servers 71
 - creating servers 71
 - deleting servers 71
 - editing the configuration file 72
 - running servers as standalone or Tuxedo-controlled 72
 - setting up the PeopleSoft Windows service 73
 - sharing indexes with application servers 206
 - starting servers 70, 72

- starting servers automatically 76
- stopping servers 70, 72
- timeouts 354
- using command-line options 72
- using menu options 69
- PeopleTools
 - adding system messages 247
 - administering Query 270
 - converting panels to pages 262
 - copying file attachments 269
 - debug utilities 276
 - grouping records 261
 - implementation options 24
 - international utilities 278
 - maintaining system messages 247
 - maintaining URLs 266
 - optimization utilities 281
 - setting general options 240
 - setting help options 246
 - setting up remote databases 265
 - specifying language settings 239
 - using administration utilities 236
 - using audit utilities 273
 - using DDL Model Defaults page 256
 - using Gather utility 271
 - using merchant integration utilities 260
 - using PeopleSoft Ping 281
 - using Sync ID utilities 270
 - using tablespets 260, 261
 - using Tablespace Utility 255
 - using the spell check system dictionary 249
 - using translate values 250
 - using update utilities 265
 - using XML Link Function Registry 260
 - utilities 233
- PeopleTools Test Utilities page 276
- performance collators, configuring 117
- performance issues
 - servers, configuring application/database 21
- Performance Monitor
 - disabling the agent 109
 - enabling the agent 93
 - starting PSPPMSRV servers 117
- permission lists
 - adding user IDs 78
 - applying timeouts 356
- PIA
 - installing on WebLogic 336
 - starting and stopping on WebLogic 337
- pinging
 - charting 282
 - using PeopleSoft Ping 281
- plug-ins
 - using Apache HTTP server 134
- popup menus 213
- portals
 - monitoring HTTP session count on WebLogic 344
 - portal servlets *See Also* portal servlets
 - search technology 179
- portal servlets
 - understanding 19
- ports
 - assigning for Jolt listener 83
 - assigning the JRAD listener port 86
 - assigning the REN servers HTTP port 115
 - assigning the REN servers HTTPS port 115
 - assigning the workstation listener port

- number 82
 - setting the PSDBGSRV listener port 286
 - specifying for failover mail servers 106
 - specifying for mail servers 106
 - specifying the proxy server port 111
 - using PeopleCode Debugger 285
- Process Scheduler
 - See* PeopleSoft Process Scheduler
 - batch environment 17
- Process Scheduler workstation settings 222
- proxy servers
 - configuring Apache HTTP as an RPS 134
 - setting up 125
 - specifying for PSTOOLS 110
 - specifying the port 111
- PS_CFG_HOME 27
 - %V variable 31
 - alternate locations 31
 - location of 29
 - managing 33
 - PSADMIN 30
 - setting 32
 - UNIX/Linux 29
 - Windows 29
- PS_HOME 27
 - decoupled 27
 - managing 33
 - read only 29
 - remote PS_HOME 32
 - security 29
- PS_HOME Access 229
- PSADMIN
 - administering application servers 58
 - administering domains 58
 - archiving configuration files 52
 - booting domains 53, 60
 - checking domain status 61
 - configuration templates 37
 - configuring application servers 58
 - configuring domains 51, 64
 - configuring PeopleCode debugging 89
 - configuring the PeopleSoft service 74
 - creating domains 66
 - deleting domains 66
 - editing configuration and log files 65
 - forcing domain shutdown 61
 - loading configurations 52
 - monitoring domains 53
 - normal domain shutdown 61
 - preloading cache 67
 - PS_CFG_HOME 30
 - purging the domain cache 62
 - setting application server domain parameters 77
 - setting database options 79
 - setting database sign-in options 78
 - setting Integration Broker options 114
 - setting interface driver options 109
 - setting JRAD options 86
 - setting messaging server options 105
 - setting PSANALYTICSRV options 100
 - setting PSAPPSRV options 97
 - setting PSPMSRV options 115
 - setting PSQCKSRV options 102
 - setting PSQYRSRV options 103
 - setting PSRENSRV options 115
 - setting PSSAMSRV options 101
 - setting PSTOOLS options 109
 - setting remote call options 96
 - setting security options 81
 - setting server process options 116
 - setting workstation listener options 81
 - shutting down domains 60
 - specifying cache settings 94
 - specifying domain settings 86
 - specifying search index file location 114
 - specifying SMTP settings 105
 - specifying trace options 90
 - starting 36
 - stopping domains 53
 - understanding 1, 35
 - using 37
 - using command-line options 39
 - using executables and configuration files 50
 - using menu options 57
 - using Process Scheduler menu 69
 - using Quick-Configure menu 38
 - Windows drives 35
- PSANALYTICSRV 13
 - setting server process options 100
 - setting the idle timeout 100
 - setting the maximum number of analytic server instances 100
 - setting the minimum number of analytic server instances 100
- PSANALYTICSRV, configuring 118
- PSAPPSRV 13
 - enabling prompting on look-up pages 100
 - moving services into PSQCKSRV 116
 - percentage of memory growth 98
 - setting PSQCKSRV options 102
 - setting the maximum fetched row storage 100
 - setting the maximum number of servers 98
 - setting the number of startup servers 97
 - setting the service failure threshold for server process restarts 99
 - setting the service request threshold for server termination/restart 98
 - setting the service request wait time 98
 - setting the Tuxedo queue size 100
 - understanding 97
- PSDBGSRV 14
- Pskeymanager *See* SSL
- PSMCFLOG 13
- PSPMSRV 14
 - setting options 115
 - starting 117
- PSPRCS.CFG, editing 72
- PSQCKSRV 13
 - moving PSAPPSRV services into 116
 - setting server process options 102
 - setting the maximum fetched row storage 103
 - setting the maximum number of servers 102
 - setting the number of startup servers 102
 - setting the request wait time 102
 - setting the service failure threshold for server process restarts 103
 - setting the service request threshold for server termination/restart 102
- PSQYRSRV 13
 - enabling reading uncommitted data 104
 - moving long-running queries into 116
 - setting options 103
 - setting the maximum number of servers 103
 - setting the maximum result set size 104
 - setting the number of startup servers 103

- setting the request wait time 103
 - setting the service failure threshold for server
 - process restarts 104
 - setting the service request threshold for
 - server termination/restart 104
- PSRENSRV 13
 - assigning the REN servers HTTP port 115
 - assigning the REN servers HTTPS port 115
 - configuring event notification 117
 - setting the application server domain name 115
 - setting the log severity level 115
 - setting the TCP buffer size 115
 - understanding 115
- PSSAMSRV 13
 - setting server process options 101
 - setting the maximum fetched row storage 102
 - setting the maximum number of servers 101
 - setting the number of startup servers 101
 - setting the request wait time 101
 - setting the service failure threshold for server
 - process restarts 101
 - setting the service request threshold for
 - server termination/restart 101
- PSTOOLS
 - accessing class libraries 110
 - disabling the Performance Monitor agent 109
 - enabling/disabling %UpdateStats 112
 - setting advanced configuration parameters 109
 - specifying character sets 111
 - specifying Java VM options 110
 - specifying proxy servers 110
 - specifying servers for direct connection 111
 - specifying the proxy server port 111
 - suppressing application error messages 112
 - suppressing SQL error messages 113
- PSUQSRV 13
- PSWATCHSRV 13
- pub/sub servers
 - booting 116
 - configuring 116

Q

- queries
 - moving long-running queries into PSQRYSRV 116
 - setting PSQRYSRV options 103
- Query
 - maintaining 270
- queue 14

R

- RDA 265
- real time event notification (REN) 115
- record-based indexes
 - adding subrecords 198
 - building 180, 195
 - modifying properties 195
 - understanding 178
- records
 - building record-based indexes 195
 - grouping 261
- registry settings 209
- remote call
 - options 96, 216
- remote data access (RDA) 265
- remote databases
 - ensuring security 266
 - setting up connections 265
- REN 115
- reports
 - report repository servlet 20
- reverse proxy servers *See* RPS, RPS WebSphere 157
- RPS
 - configuring Apache HTTP 134
 - configuring MS IIS 126
 - configuring Sun Java System Web Server 131
 - configuring WebLogic 128
 - load balancing 129
 - setting up 125

S

- Scope page, converting panels to pages 262
- SCP server 109
- search
 - configuring 114
- searches *See* search indexes
- search indexes
 - administering 203, 204
 - building and maintaining 177
 - common controls 192
 - creating collections 194
 - editing properties 205
 - file system *See Also* file system indexes
 - HTTP spider HTTP spider indexes
 - MIME types 193
 - modifying VdkVgwKey 207
 - on z/OS 181
 - opening collections 194
 - portal technologies 179
 - record-based *See Also* record-based indexes
 - record-based indexes 195
 - scheduling administration 206
 - search architecture 178
 - search utilities 180
 - sharing between application servers and
 - Process Scheduler 206
 - specifying file locations 114
 - specifying locations 203
 - types 178
 - understanding 3, 177
 - understanding limitations 181
 - understanding searches 182
 - verity technologies 179
- search server domains
 - starting automatically 76
- security
 - adding user IDs to permission lists 78
 - implementing SSL on WebLogic 138
 - setting sign-in options 81
 - using remote databases 266
 - validating signon with database 81
- server process

- services 14
 - server processes 12
 - servers
 - setting process options 116
 - setting PSPMSRV options 115
 - setting up the PeopleSoft Windows service 73
 - specifying names 79
 - using PSADMIN 35
 - service packs
 - determining the level 150
 - viewing system information 235
 - services
 - application server 14
 - web 18
 - service start delay 76
 - servlets
 - integration gateway 19
 - Java *See Also* Java servlets
 - portal portal servlets
 - report repository 20
 - servlet engine 18
 - understanding Jolt and Tuxedo 20
 - session cookies
 - WebLogic name format 147
 - session timeouts *See* timeouts
 - shortcut links 217
 - signon
 - defaults 211
 - setting defaults 210
 - using connect ID 211
 - using default application server 211
 - using default database name 211
 - using default database server 211
 - using default operator ID 211
 - using operator overrides 212
 - Simple Mail Transfer Protocol *See* SMTP
 - SMTP
 - client certificate 108
 - delivering TriggerBusinessEvent email via messaging system 107
 - enabling send times for messages 107
 - further considerations 109
 - specifying character set of sender's machine 106
 - specifying DLL for translating mail 106
 - specifying failover mail server host name and IP address 106
 - specifying failover mail server port 106
 - specifying mail server host name and IP address 105
 - specifying mail server port 106
 - specifying reply internet address for BlackBerry email 106
 - specifying sender's internet address 106
 - specifying sender's source machine 106
 - specifying settings 105
 - SSL Port 108
 - timeout retries 108
 - time to wait for result 108
 - tracing email details 107
 - user name for authentication 107
 - user name for failover 107
 - user password for authentication 107
 - user password for failover 107
 - use SSL 108
 - spawning application servers 78
 - setting the threshold 87
 - spell checking 249
 - case sensitivity 249
 - table structure for word storage 250
 - Spell Check System Dictionary page 249
 - spider indexes *See Also* HTTP spider indexes
 - SQL
 - specifying trace settings 215
 - tracing 290
 - SQRs
 - customizing column headings 259
 - SSL
 - configuring WebLogic keys 145
 - converting keys and certificates for WebLogic use 141
 - implementing certificates 138
 - implementing keys 138
 - importing keys and certificates to the WebLogic keystore 143
 - importing the server certificate using ImportPrivateKey 144
 - importing the server certificate using Pskeymanager 144
 - obtaining encryption keys for WebLogic 138
 - preparing keys and certificates for WebLogic 141
 - server certificate chain of trust 142
 - used with WebLogic 138
 - WebLogic private key 142
 - WebSphere 163
 - strings table 259
 - Structured Query Reports *See* SQRs
 - Sun Java System Web Server
 - configuring as an RPS 131
 - Supply Chain Planning (SCP) server 109
 - supporting applications, verifying 229
 - Sybase
 - setting TCP packet size 79
 - specifying server names 79
 - SyncRequest
 - setting thread pool size 114
 - System Information page 233
- ## T
- tables
 - maintaining URLs 266
 - sharing 261
 - storing words 250
 - translate 250
 - tablesets
 - controlling 261
 - creating IDs 260
 - trees 262
 - tablespaces
 - adding 255
 - deleting 255
 - managing 256
 - renaming 255
 - using Tablespace Utility 255
 - templates, configuration 37, 66
 - test utilities 276
 - three-tier workstation settings 220
 - timeout, authentication 135
 - timeouts
 - application servers 352
 - PIA 356
 - Process Scheduler 354

- search server 354
- setting analytic instance idle time 100
- setting client connection idle time 83
- setting client connection request binding time 83
- settings 349
- setting the PSAPPSRV wait time for server requests 98
- WebLogic HTTP sessions 135
- web servers 349
- Time Zones utility 279
- Trace PeopleCode page 288
- Trace SQL page 290
- tracing
 - activating for Application Engine programs 92
 - activating for pages 92
 - configuring PeopleCode 288
 - enabling for email details 107
 - enabling memory image creation for crashes 93
 - enabling Performance Monitor agent 93
 - logging error reports 93
 - logging for analytic servers 92
 - mailing error notifications 93
 - reproducing crashes 93
 - setting in Configuration Manager 215
 - setting the level for all client-generated activity 90
 - setting the logging level for all clients 90
 - setting the logging level for individual clients 90
 - specifying options 90
 - specifying trace-log character set 89
 - specifying trace options to be written to the trace file 91
 - SQL 290
 - using page processor activity 91
 - viewing crash log information 93
- Translate Values page 250
- translating languages 239
- Tuxedo 11
 - booting domains 53
 - restarting server processes 88
 - running the Process Scheduler server 72
 - setting compression 83
 - setting the queue size 100
 - setting the spawn threshold 87
 - setting timeouts 83
 - transmitting requests and data 20
- Tuxedo connect string 221

U

- uninstall workstation 228
- Update utilities 265
- upgrade, image conversion 226
- URLs
 - adding new entries 267
 - maintaining 266
- user ID 212
 - specifying 78
 - specifying the default 211

V

- variables *See* environment variables
- Verity
 - search index limitations 181
 - search technologies 179
 - setting the character set 280
- vspider
 - building file system indexes 199
 - building HTTP spider indexes 201
 - building indexes 180

W

- web applications 319
 - changing server targets on WebLogic 346
 - descriptor files 335
 - using Console 320
 - using HttpClusterServlet 320
 - using HttpProxyServlet 320
- web container 18
- WebLogic
 - accessing the server console 120
 - adjusting JVM heap size 149, 343
 - authentication timeout 135
 - changing multi-server listen ports 345
 - changing user passwords 136
 - common default domain settings 327
 - configuration files 345
 - configuring as an RPS 128
 - configuring SSL keys 145
 - converting SSL keys and certificates 141
 - determining the service pack level 150
 - distributed managed server 325
 - domain directories and files 331
 - domain types 320
 - enabling/disabling HTTP access log 151
 - enabling/disabling HTTP Keep-Alive 135
 - enabling RPS load balancing 129
 - HTTP session timeouts 135
 - implementing SSL 138
 - importing SSL keys and certificates to the keystore 143
 - importing the server certificate using ImportPrivateKey 144
 - importing the server certificate using Pskeymanager 144
 - in a single-server domain 320
 - installing PIA 336
 - managed server architecture 319
 - monitoring Execute thread pool size 343
 - monitoring portal HTTP session count 344
 - monitoring resources 342
 - monitoring sessions via WebLogic Server console 124
 - multi-server domain 322
 - obtaining SSL encryption keys 138
 - password protecting the private key 142
 - preparing SSL keys and certificates 141
 - server certificate chain of trust 142
 - server life cycle 337
 - session cookie name format 147
 - setting up an RPS 125
 - starting 121
 - starting and stopping a distributed managed

- server 342
 - starting and stopping multiple servers 338
 - starting and stopping PIA 337
 - starting on UNIX 123
 - starting on Windows 121
 - stopping 123
 - tuning performance 342
 - understanding SSL 138
 - understanding the PeopleSoft domain 119
 - web applications 319
- web servers
 - browsers *See Also* browsers
 - Gather utility 273
 - Jolt *See Also* Jolt
 - PeopleSoft servlets 19
 - setting PSRENSRV options 115
 - software elements 18
 - timeouts 349
 - Tuxedo *See Also* Tuxedo
 - understanding 3
 - WebLogic 319
 - working with WebLogic 119
 - working with WebSphere 153
- web services 18
- WebSphere *See* IBM WebSphere
- What to Index page 200
- Windows services *See* Microsoft Windows services
- workflow Configuration Manager settings 216
- workstation handler (WSH) 16
- workstation listener
 - assigning the port number 82
 - enabling encryption 82
 - setting client connection request binding time 83
 - setting the address 81
 - setting timeout 83
 - setting Tuxedo compression 83
 - specifying handler quantity 82
 - specifying maximum clients per handler 82
 - specifying maximum handler quantity 82
 - understanding 81
- workstation listener (WSL) 16
- workstations
 - importing and exporting settings 218
 - installing 217
 - setting up 209
 - uninstalling 228
- WSL
 - configuring 117

X

XML Link Function Registry 260

Z

- z/OS
 - defining passwords/user IDs 81
 - PeopleCode debugging 285