



PeopleTools 8.12 PeopleSoft
Internet Architecture Administration
PeopleBook

SKU MTASr8SP1B 1200

PeopleBooks Contributors: Teams from PeopleSoft Product Documentation and Development.

Copyright © 2001 by PeopleSoft, Inc. All rights reserved.

Printed in the United States of America.

All material contained in this documentation is proprietary and confidential to PeopleSoft, Inc. and is protected by copyright laws. No part of this documentation may be reproduced, stored in a retrieval system, or transmitted in any form or by any means, including, but not limited to, electronic, graphic, mechanical, photocopying, recording, or otherwise without the prior written permission of PeopleSoft, Inc.

This documentation is subject to change without notice, and PeopleSoft, Inc. does not warrant that the material contained in this documentation is free of errors. Any errors found in this document should be reported to PeopleSoft, Inc. in writing.

The copyrighted software that accompanies this documentation is licensed for use only in strict accordance with the applicable license agreement which should be read carefully as it governs the terms of use of the software and this documentation, including the disclosure thereof.

PeopleSoft, the PeopleSoft logo, PeopleTools, PS/nVision, PeopleCode, PeopleBooks, Vantive, and Vantive Enterprise are registered trademarks, and *PeopleTalk* and "People power the internet." are trademarks of PeopleSoft, Inc. All other company and product names may be trademarks of their respective owners.

Contents

About This PeopleBook

Audience	ix
Before You Begin.....	x
Related Documentation	x
Documentation on the Internet.....	x
Documentation on CD-ROM	x
Hardcopy Documentation	xi
Typographical Conventions and Visual Cues.....	xi
Comments and Suggestions.....	xiii

Chapter 1

PeopleSoft Architecture

Component Overview	1-1
Web Browser.....	1-2
Web Server.....	1-3
Application Server	1-3
Database Server.....	1-3
Batch Server.....	1-3
The Physical Architecture.....	1-4
Web Browser.....	1-4
Web Server.....	1-4
Server Components	1-4
PeopleSoft Servlets	1-5
Jolt.....	1-5
Application Server	1-6
Domains	1-6
Listeners, Handlers, and Queues.....	1-7
PeopleSoft Server Processes	1-10
Services	1-11
BEA Products.....	1-12
Database Connectivity	1-12
Batch Server Environment	1-12
Batch Server Support	1-13

Process Scheduler and the Application Server.....	1-13
Database Server.....	1-14
Configuration and Deployment Options.....	1-14
Configuration Options.....	1-14
Physical Application Server Configuration.....	1-14
Logical Application Server Configuration.....	1-15
Deployment Options	1-16
Connecting through PIA.....	1-16
PeopleSoft Portal.....	1-17
Connecting from the Development Environment	1-18
Integration Technologies.....	1-18
Architecture Terminology.....	1-18
Application Server Domain.....	1-19
Jolt Server Handler (JSH)	1-19
Jolt Server Listener (JSL).....	1-19
Jolt.....	1-19
Messaging Servers	1-19
PeopleSoft Portal.....	1-19
Page Servlet.....	1-20
PSADMIN.....	1-20
PSAPPSRV	1-20
PSQCKSRV	1-20
PSQRYSRV	1-20
PSSAMSRV	1-20
Server Process	1-20
Service.....	1-21
Servlet	1-21
Tuxedo	1-21
Web Server.....	1-21
Windows Workstation/Development Environment	1-21
Workstation Listeners (WSL)	1-21
Workstation Handlers (WSH).....	1-22

Chapter 2

The PSADMIN Utility

Starting PSADMIN.....	2-1
Using PSADMIN.....	2-2
Menu Interface	2-2
Selecting Menu Options.....	2-3
Navigating to Previous Menu.....	2-4

Menu Structure.....	2-5
Application Server	2-6
Process Scheduler	2-7
Web Components.....	2-8
Service Setup.....	2-8
Configuration Templates	2-9
Command Line Options.....	2-10
Syntax.....	2-10
General Administrative Commands	2-10
Command Line Create and Configure	2-11
Overview	2-11
Including Unique Values.....	2-12
Default Values.....	2-13
Valid Values.....	2-14
Quick Configure	2-17
Executables and Configuration Files	2-19
Configuring a Domain.....	2-20
Loading a Configuration	2-22
Booting a Domain	2-23
Stopping a Domain.....	2-23

Chapter 3

Application Server

Accessing the Application Server Options	3-1
Administering a Domain.....	3-2
Booting a Domain	3-3
Shutting Down a Domain.....	3-4
Normal shutdown	3-4
Forced shutdown	3-4
Checking Domain Status.....	3-4
Server status	3-5
Client status.....	3-6
Queue Status	3-6
Configuring a Domain.....	3-7
The Edit configuration/log files Menu	3-9
Edit PSAPPSRV.CFG.....	3-10
Edit APPSRV.LOG.....	3-11
Edit TUXLOG.....	3-11
Edit PSAPPSRV.tracesql	3-11
Edit PSSAMSRV.tracesql.....	3-11

Creating a Domain	3-11
Deleting a Domain	3-12

Chapter 4

Domain Parameter Reference

Startup	4-1
Database Options	4-3
Security	4-4
Workstation Listener	4-5
JOLT Listener	4-7
JOLT Relay Adapter	4-9
Domain Settings	4-10
Trace	4-12
Cache Settings	4-16
Remote Call	4-17
PSAPPSRV	4-17
PSSAMSRV	4-19
PSQCKSRV	4-20
PSQRYSRV	4-21
Messaging Server Processes	4-23
SMTP Settings	4-23
Interface Driver	4-25
PSTOOLS	4-25
Select Server Process Options (Final Prompts)	4-28

Chapter 5

Process Scheduler

Starting a Process Scheduler Server	5-3
Stopping a Process Scheduler Server	5-4
Configuring a Process Scheduler Server	5-5
Creating a Process Scheduler Server Configuration	5-6
Deleting a Process Scheduler Server	5-7
Editing the Process Scheduler Configuration File	5-8
Process Scheduler Options	5-9
Process Scheduler Command Line Options	5-9
Start the Process Scheduler	5-9
Stopping the Process Scheduler	5-9

Chapter 6

Web Components

Web Server Administration.....	6-3
Start Tuxedo administrative web server.....	6-3
PSWEBSRV.INI.....	6-3
Tuxedo Web Monitoring Facility Administration.....	6-4
Start Tuxedo Web Monitor wlisten process.....	6-5
Edit pswebgui.ini	6-5
PeopleSoft Starter Page Administration.....	6-6
Launching the Starter Web Page.....	6-6
Edit psstarter.html	6-8
Generate Web Components configuration files	6-10

Chapter 7

PeopleSoft Service Setup

Overview.....	7-1
Configuring the PeopleSoft Service	7-2
Monitoring the Executables	7-6
PeopleSoft Services Administration Reference	7-7
Editing the PSNTSRV.CFG File Manually	7-8

Chapter 8

Web Server Administration

Server Components.....	8-1
PeopleSoft Servlets.....	8-2
Page Servlet.....	8-3
Portal Servlet.....	8-3
Report Repository Servlet	8-3
Application Messaging Gateway Servlet	8-3
Java Message Agent Client (JMAC).....	8-3
Transmitting Requests	8-4
Multiple Servlets.....	8-5
Configuration Files	8-6
PeopleSoft Configuration Files	8-6
Non-PeopleSoft Configuration Files.....	8-7
Apache Configuration Files	8-7
JServ Configuration Files.....	8-8
Modifying PeopleSoft Web Configuration Files	8-8
configuration.properties variables	8-8
pstools.properties	8-15
Setting Timeout Intervals	8-16
Setting Jolt Failover	8-19

Linux Shell.....	8-19
------------------	------

Chapter 9

Administration Considerations

System Information Page.....	9-1
Overview	9-1
Viewing the System Information Page.....	9-1
User Specifications	9-3
Setting Up Jolt Internet Relay.....	9-3
Overview	9-4
Installing Jolt Internet Relay (JRLY)	9-5
Configuring Jolt Relay (Front-End) on UNIX or Windows NT	9-8
Configuring Jolt Relay Adapter (JRAD).....	9-10
Starting Jolt Relay	9-11
Stopping Jolt Relay	9-12
Jolt Relay Notes	9-12
Development Environment.....	9-13
Windows Client.....	9-13
Specifying Application Server Names	9-13
Machine Host Name Versus IP Dotted Notation	9-14
How to Determine if Connected in Three-Tier Mode.....	9-14
Help About.....	9-14
Hourglass	9-15
Remote Call.....	9-15
Crystal and nVision in Three-Tier Configurations.....	9-15
Timeouts.....	9-16
Client Connection	9-16
Wait in Queue	9-16
Maximum In-Database Time	9-16
Debugging PIA Applications: Set up the PeopleCode Debugger	9-16
Setting up the Debug Broker Class and Debug User Account.....	9-16
Setting up the Application Server for Debugging.....	9-20

Index

ABOUT THIS PEOPLEBOOK

This book contains information related to the PeopleSoft Internet Architecture (PIA). Although the Installation and Administration book provides procedures for installing and configuring the components, this book offers information that you'll use over time, not just at installation.

Audience

This book is intended for technical users, system administrators, and programmers who will be implementing, maintaining, or developing applications for your PeopleSoft system. To take full advantage of the information covered in this book, we recommend that you have a basic understanding of how to use PeopleSoft applications, system administration, and basic client/server and Internet architecture. You should know how to navigate through the system and how to add, update, and delete information using PeopleSoft tables and pages. You should also have a basic familiarity with relational database concepts and SQL. We also recommend that you read the BEA documentation delivered with PeopleSoft, as mentioned in the Installation and Administration book.

PeopleSoft Architecture covers all of the components that comprise the PeopleSoft architecture including application servers, web servers, connection options, and so on.

The PSADMIN Utility covers the interface with which you configure application server domains and Process Scheduler Servers.

Application Server covers the PSADMIN menus that are specific to configuring application server domains.

Domain Parameter Reference contains information for every parameter within the PSADMIN utility.

Process Scheduler covers the menus that are specific to configuring the Process Scheduler Server Agent.

Web Components discusses the menus used for setting up Tuxedo web-based documentation and Tuxedo's web-based monitoring tool. The web components discussed here are not involved with the PeopleSoft Internet Architecture.

PeopleSoft Service Setup describes how to set up your Process Scheduler servers and application servers to be Windows NT Services.

Web Server Administration describes the components running on the web server as well as some of the more popular configuration options. This chapter discusses the web server components that comprise the PeopleSoft Internet Architecture.

Administration Considerations describes some of the options you can take advantage of in certain situations, such as configuring Jolt Relay, setting up Windows Clients, and so on.

Before You Begin

To benefit fully from the information covered in this book, you need to have a basic understanding of how to use PeopleSoft applications. We recommend that you complete at least one PeopleSoft introductory training course.

You should be familiar with navigating around the system and adding, updating, and deleting information using PeopleSoft windows, menus, and pages. You should also be comfortable using the World Wide Web and the Microsoft® Windows or Windows NT graphical user interface.

Related Documentation

To add to your knowledge of PeopleSoft applications and tools, you may want to refer to the documentation of the specific PeopleSoft applications your company uses. You can access additional documentation for this release from PeopleSoft Customer Connection (www.peoplesoft.com). We post updates and other items on Customer Connection, as well. In addition, documentation for this release is available on CD-ROM and in hard copy.



Important! Before upgrading, it is *imperative* that you check PeopleSoft Customer Connection for updates to the upgrade instructions. We continually post updates as we refine the upgrade process.

Documentation on the Internet

You can order printed, bound versions of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM. You can order additional copies of the PeopleBooks CDs through the Documentation section of the PeopleSoft Customer Connection Web site:
<http://www.peoplesoft.com/>

You'll also find updates to the documentation for this and previous releases on Customer Connection. Through the Documentation section of Customer Connection, you can download files to add to your PeopleBook library. You'll find a variety of useful and timely materials, including updates to the full PeopleSoft documentation delivered on your PeopleBooks CD.

Documentation on CD-ROM

Complete documentation for this PeopleTools release is provided in HTML format on the PeopleTools PeopleBooks CD-ROM. The documentation for the PeopleSoft applications you have purchased appears on a separate PeopleBooks CD for the product line.

Hardcopy Documentation

To order printed, bound volumes of the complete PeopleSoft documentation delivered on your PeopleBooks CD-ROM, visit the PeopleSoft Press Web site from the Documentation section of PeopleSoft Customer Connection. The PeopleSoft Press Web site is a joint venture between PeopleSoft and Consolidated Publications Incorporated (CPI), our book print vendor.

We make printed documentation for each major release available shortly after the software is first shipped. Customers and partners can order printed PeopleSoft documentation using any of the following methods:

Internet

From the main PeopleSoft Internet site, go to the Documentation section of Customer Connection. You can find order information under the Ordering PeopleBooks topic. Use a Customer Connection ID, credit card, or purchase order to place your order.

PeopleSoft Internet site: <http://www.peoplesoft.com/>.

Telephone

Contact Consolidated Publishing Incorporated (CPI) at **800 888 3559**.

Email

Email CPI at callcenter@conpub.com.

Typographical Conventions and Visual Cues

To help you locate and interpret information, we use a number of standard conventions in our online documentation.

Please take a moment to review the following typographical cues:

`monospace font`

Indicates PeopleCode.

Bold

Indicates field names and other page elements, such as buttons and group box labels, when these elements are documented below the page on which they appear. When we refer to these elements elsewhere in the documentation, we set them in Normal style (not in bold).

We also use boldface when we refer to navigational paths, menu names, or process actions (such as **Save** and **Run**).

Italics

Indicates a PeopleSoft or other book-length publication. We also use italics for *emphasis* and to indicate specific field values. When we cite a field value under the page on which it appears, we use this style: ***field value***.

We also use italics when we refer to words as words or letters as letters, as in the following: Enter the number *0*, not the letter *O*.

KEY+KEY Indicates a key combination action. For example, a plus sign (+) between keys means that you must hold down the first key while you press the second key. For ALT+W, hold down the ALT key while you press W.

Jump links Indicates a jump (also called a link, hyperlink, or hypertext link). Click a jump to move to the jump destination or referenced section.

Cross-references The phrase For more information indicates where you can find additional documentation on the topic at hand. We include the navigational path to the referenced topic, separated by colons (:). Capitalized titles in *italics* indicate the title of a PeopleBook; capitalized titles in normal font refer to sections and specific topics within the PeopleBook. Cross-references typically begin with a jump link. Here's an example:

For more information, see Documentation on CD-ROM in *About These PeopleBooks*: Related Documentation.

• **Topic list** Contains jump links to all the topics in the section. Note that these correspond to the heading levels you'll find in the Contents window.



Name of Page or
Dialog Box

Opens a pop-up window that contains the named page or dialog box. Click the icon to display the image. Some screen shots may also appear inline (directly in the text).



Text in this bar indicates information that you should pay particular attention to as you work with your PeopleSoft system. If the note is preceded by **Important!**, the note is crucial and includes information that concerns what you need to do for the system to function properly.



Text in this bar indicates For more information cross-references to related or additional information.



Text within this bar indicates a crucial configuration consideration. Pay very close attention to these warning messages.

Comments and Suggestions

Your comments are important to us. We encourage you to tell us what you like, or what you would like changed about our documentation, PeopleBooks, and other PeopleSoft reference and training materials. Please send your suggestions to:

PeopleTools Product Documentation Manager
PeopleSoft, Inc.
4460 Hacienda Drive
Pleasanton, CA 94588

Or send comments by email to the authors of the PeopleSoft documentation at:

DOC@PEOPLESOFT.COM

While we cannot guarantee to answer every email message, we will pay careful attention to your comments and suggestions. We are always improving our product communications for you.

CHAPTER 1

PeopleSoft Architecture

The topics contained in this chapter will give you an overview of the individual components of the PeopleSoft Internet Architecture (PIA). It's important to understand the role of each component before you begin configuring your implementation. With this information you can decide which configuration will work best at your site.

Configuring PIA is not just about enabling Internet application deployment through a browser. PIA enables you to take advantage of all of the PeopleSoft intranet and internet solutions, as well as the PeopleSoft integration technologies, such as Application Messaging (publish/subscribe).



The PeopleSoft Internet Architecture is also referred to by the acronym, PIA. In most cases within this PeopleBook the term PIA appears.

After reading this chapter, you should be familiar with the essential PeopleSoft components, the niche that each component fills within the architecture, your configuration options, and deployment options.



With PIA, PeopleSoft introduces an entirely new architecture. Because our technology continually evolves, PeopleSoft provides an additional document called *PIA Answer Book*. This is where you go to find late-breaking information from the field, tips, and, most important of all, clarification to some of the more subtle aspects of implementing PIA. You download *PIA Answer Book* from the Continuous Documentation site in Customer Connection.

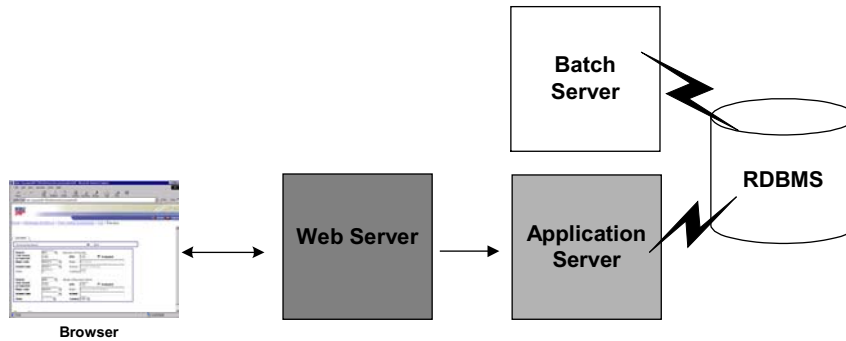
Component Overview

The PeopleSoft Internet Architecture is comprised of a variety of components ranging from the browser to the database server, including the following:

- Web Browser
- Web Server
- Application Server
- Batch Server

- Database Server

The following example illustrates, at a high level, the physical relationship between the PeopleSoft components.



High-Level PeopleSoft Architecture

Each component fulfills a unique niche within the system. The following topics briefly introduce you to the components. The following section provides more details for each component.

Web Browser

The web browser is the primary means by which end users and even administrators access PeopleSoft applications and administrative tools.

You need to make sure that on each workstation a currently supported browser is installed. Other than that, you don't need to install anything else, such as applets or connectivity software, on the workstation running the browser.



For more information on supported browsers refer to the PeopleSoft Platforms database on Customer Connection.



There is no "traditional" client involved in PIA. The system sends pure HTML to a supported browser interface. All processing occurs on the server level. PeopleSoft does support the Development Environment, which is intended for applications developers and system administrators who need access to PeopleTools. These users need machines running Windows.



For more information on the Development Environment, see Administration Considerations.

Web Server

A Java-enabled web server is required to support browser transaction requests and the application messaging technology. You install on the web server a collection of PeopleSoft Java servlets designed to handle a wide range of PeopleSoft transactions.

Application Server

The application server is the core of the PeopleSoft Internet Architecture; it executes business logic and issues SQL to the database server. The application server consists of numerous PeopleSoft services and server processes that handle transaction requests.

The application server is responsible for maintaining the SQL connection to the database for the browser requests as well as the Windows Development Environment.

PeopleSoft uses TUXEDO to manage database transactions, and Jolt, TUXEDO's counterpart, to facilitate transaction requests issued from the Internet. Both TUXEDO and Jolt are products of BEA Systems.

Database Server

The database server houses your database engine and your PeopleSoft database, which includes all of your object definitions, system tables, application tables, and data. The database server must be running one of the supported RDBMS/operating system combinations.

The relationship between a database server and an application server is a one-to-many model. That is, a single database server can have multiple application servers connecting to it. The database server simultaneously handles the application server connections, Development Environment connections, and batch programs running against it.



Using the Windows Development Environment, you can connect directly to the database or indirectly through an application server.

Batch Server

The batch server, or batch environment, is where you have Process Scheduler installed and configured, and it is the location where many of your batch programs run, such as Application Engine programs. In most situations this is also where you have your COBOL and SQR executables installed.

The Physical Architecture

Now that you have been introduced to the main components of PIA, let's take a look at some important details to consider regarding each component.

Web Browser

The browser uses the HTTP protocol. The browser sends requests to the web server, which forwards the request to the application server. A servlet installed on the web server facilitates all browser connections.

The browser does not download any applets to complete a transaction. The application server sends only the following to the browser:

- HTML
- Javascript
- Cookies

With the browser only having to process the items in the previous list, the result is a lightweight deployment of PeopleSoft applications that does not burden the client workstation with unnecessary processing responsibility. PIA is a server-centric architecture.

PIA leverages web browser cookies to store a unique access token for each user when they are initially authenticated. When the user connects to another PeopleSoft system, the token in the browser cookie is used to re-authenticate the user so they don't have to go through the signon process again. It should be noted that the browser cookie is an in-memory cookie and is never written to disk. The cookie is also encrypted to prevent snooping and check-summed to prevent tampering.

Web Server

The following topics explain the software that needs to be installed on the web server, the PeopleSoft Servlets, and how the servlets relay transaction requests to the application server.

Server Components

The web server must be Java-enabled so that it can run servlets. During the PeopleSoft install, a variety of PeopleSoft Java servlets are installed on the web server. So, in addition to the web server software, you also need to have a supported servlet engine installed.

The following list presents the software that runs on the PIA web server:

- **Web services.** The program installed on the host system that manages the web server, such as Apache.
- **Servlet Engine.** The environment in which servlets run. This component is tied to the web services software, but you install it separately. A common servlet engine is JServ.

- **Java Servlets.** A platform-independent programming language used widely for web-based programs. Servlets are Java programs that run on the web server. The Java executables are required for the servlet engine to operate.



For more information on the required software for your web server and the supported versions, see the Platforms Database on Customer Connection.

PeopleSoft Servlets

The following list contains the PeopleSoft servlets that reside on the web server.

- **Page Servlet.** The Page Servlet handles all inbound and outbound transaction requests for the browser. It is a thin servlet acting as a relay between the browser and back-end services. Also, the Page Servlet formats the pure HTML that the application server generates and presents it in the browser. If a user is not using the PeopleSoft Portal for navigation, the URL that the user enters into the browser to access a PeopleSoft application points to the Page Servlet. A PeopleTools Java Servlet that handles all inbound and outbound HTTP requests for PeopleSoft transactions and queries.
- **Portal Servlet.** The Portal servlet handles all of the requests and formatting for the users accessing PeopleSoft through the PeopleSoft Portal. It also manages all aspects of the PeopleSoft Portal such as search, content management, and home page personalization.



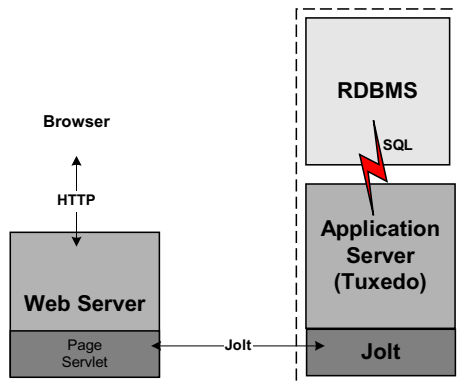
For more information on the PeopleSoft Portal see Portal Technology.

- **Application Messaging Gateway Servlet.** The Application Messaging Gateway servlet transmits publish/subscribe messages between message nodes. The gateway handles PeopleSoft-to-PeopleSoft messages, PeopleSoft-to-third party messages, and third party-to-PeopleSoft messages.
- **Report Repository Servlet.** The Report Manager servlet enables users to easily access and distribute the output of batch reports, such as Crystal and SQR, run through Process Scheduler over the Internet. This servlet retrieves the report output in the Report Repository and serves it to the browser.

Jolt

The PeopleSoft servlets on the web server transmit requests and data by way of a connection to Jolt, which runs on the application server. Jolt extends Tuxedo's capabilities to the Internet; it is the communication layer between the web-based environment and the C++ environments. You configure the servlets to direct requests from the web server to a pre-defined Jolt port on the application server.

Jolt is a companion product that must coexist with Tuxedo on the same application server machine. Jolt is *not* a standalone product; it can't function without Tuxedo.



BEA Jolt

The browsers do not connect directly to the application server. Instead they send HTTP requests to the Page Servlet running on the web server that translates the HTTP request into a Jolt request that is sent to a specified Jolt port. Then the application server, leveraging Tuxedo, runs the appropriate SQL against the database.

Application Server

Just as there are different components that comprise the physical environment in which the application server operates, such as database servers and web servers, there are a variety of components that operate on the application server allowing it to respond effectively to a multitude of transaction requests.

It's a good idea to have a general knowledge of how the application server operates "under the covers" before you spend too much time configuring and tuning it. We also recommend as you read further that you refer to the Glossary for brief descriptions of the terminology related to the application server.



For more information see Architecture Terminology.

The following topics describe the major components that run on the application server to enable transaction processing, system scaling, browser requests, and so on. After reading this section you will have an understanding of an application server domain, server processes, listeners, and handlers.

Domains

An application server domain, is the collection of Server Processes, supporting processes, and resource managers that enable connections to the database. 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 located in the PS_HOME\appserv directory on the application server.

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 will 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 database.

For example, suppose you have installed three databases, HRDMO1, HRDMO2 and HRDMO3, and you want to enable browser requests to each database. In this case, you need to configure at least three application server domains—one for each database. Keep in mind that as demand increases you may need to configure multiple application server domains per database for redundancy/failover and to maintain a certain level of performance.

You can configure multiple application server domains under a single PeopleSoft home directory, or PS_HOME. In this context, PS_HOME refers to the PeopleSoft high-level directory on the application server, not the file server. Your PS_HOME is the directory to which you installed the PeopleSoft application server files during the Server Transfer process.

PSADMIN creates a directory beneath PS_HOME/appserv for each application server domain you configure. Using the previous HRDMO example, let's say you decided to name the application server domains the same name as the database to which they connect. In this case, PSADMIN creates sub-directories \HRDMO1, \HRDMO2, and \HRDMO3 beneath the PS_HOME directory on the application server. In this scenario, the directory structure should look similar to the following:

```
\PS_HOME
  \APPSERV
    ...
    \HRDMO1
    \HRDMO2
    \HRDMO3
    ...
```

When you boot an application server domain it starts the set of Server Processes associated with that domain, such as PSAPPSRV, PSQCKSRV, PSSAMSRV and so on. Each Server Process establishes a persistent connection to a PeopleSoft database, and this connection acts as generic, SQL pipeline that the Server Process uses to send and receive SQL.

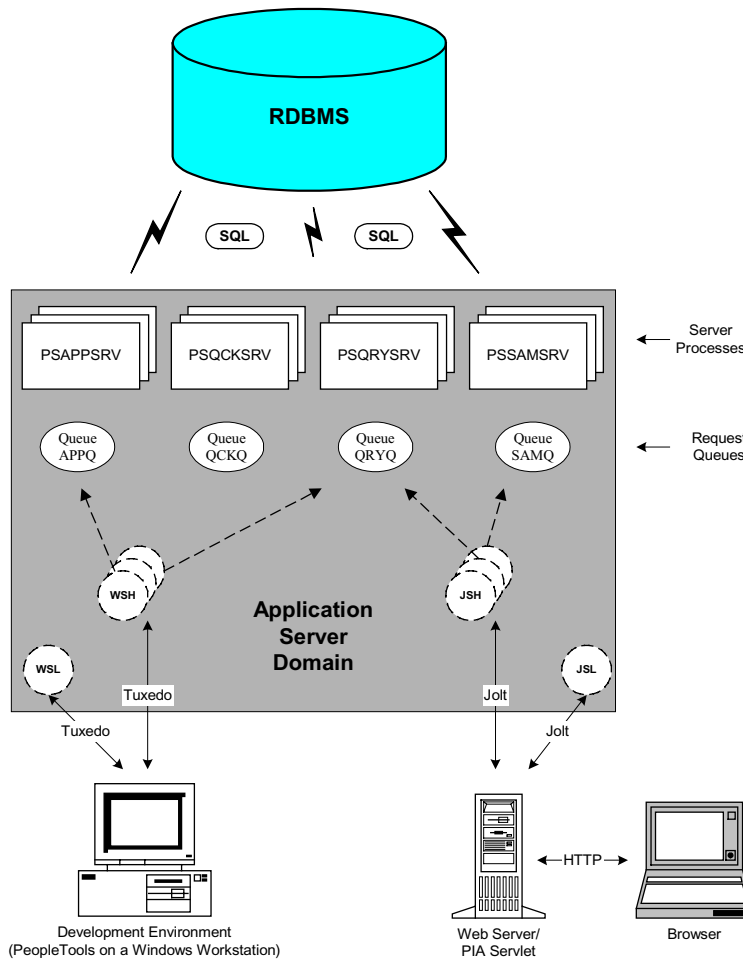
Each Server Process uses the same SQL connection to facilitate requests from multiple sources. From the RDBMS perspective, each Server Process within a domain represents a connected user. Server Processes are discussed in more detail in the following sections.

Listeners, Handlers, and Queues

This section reveals the different services that comprise the basis of the application server functionality. Understanding the “big picture” and the role of each sub-component within is essential when you configure and tune your application server. For instance, although it is

important to know how to configure multiple Java Server Handlers; it is equally important, if not more important, to know *why* and *when* to perform this task to address your site's needs.

The following example illustrates all of the processes that comprise the application server.



Application Server Components and Server Processes



When discussing the mechanics of the PeopleSoft architecture the term *server*, unavoidably, becomes overused. Although this is described later in this document, the following statement may help to illustrate the relationship between the components discussed in this chapter. An application *server* domain calls *Server Processes*, such as PSAPPSRV, which in turn invoke *Services*, such as MgrGetObject, on the database.

The following table provides a brief description of each component depicted in the previous example.

Item	Description
Workstation Listener (WSL)	The WSL monitors the Tuxedo ports for initial connection requests sent from the Windows Workstation/Development Environment. Once the Workstation Listener accepts a connection from a workstation, it directs the request to a Workstation Handler. From that point, the workstation interacts with the Workstation Handler to which it is assigned.
Workstation Handler (WSH)	The Workstation Handler processes the requests it receives from the WSL. A unique port number identifies a WSH. The port numbers for WSH are selected randomly (and internally by Tuxedo) from a given range of numbers. You can configure multiple WSHs to handle demand increases; new processes "spawn" as other processes become overloaded.
Jolt Server Listener (JSL)	Only applies to browser requests. The JSL monitors the Jolt port for connection requests sent from the browser. Once 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 WSL and the WSH discussed previously.
Jolt Server Handler (JSH)	Only applies to browser requests. Processes the requests it receives from the JSL. 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 servers of the same type (as in PSAPPSRV on APPQ, PSQCKSRV on QCKQ). The WSL and JSL insert requests into the appropriate queue, then the individual Server Processes complete each request in the order it appears in the queue.

<i>Item</i>	<i>Description</i>
Server Processes	The Server Processes act as the heart 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. The following section explores the numerous PeopleSoft server processes that run on the application server.

PeopleSoft Server Processes

The following topics provide more details on the collection of PeopleSoft Server Processes that provide the core functionality on the application server.

Understanding Server Processes

Multiple, individual server processes run in an application server domain. 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).

Server Processes invoke *Services* to perform application logic and issue SQL to the RDBMS. Each application server process, as in PSAPPSRV, PSQCKSRV, PSQRYSRV, PSSAMSRV, establishes and maintains its own connection to the database.

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 yet another transaction request.

Some of the server processes are optional, so you only need to configure those server processes that your implementation requires per domain. The minimum server processes that a domain requires are PSAPPSRV and PSSAMSRV. Each server process, its function and configuration parameters, is discussed in detail later within this document.

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. Furthermore, Tuxedo is able to dynamically spawn incremental server processes to handle increasing numbers of transaction requests. The capability to configure multiple server processes and spawn incremental server processes contributes to the application server's scalability.

Identifying the Server Processes

The following is a complete list of all the possible server processes included in an application server domain. Keep in mind that depending on the configuration options you choose, not all of the server processes will necessarily be a part of every domain.

The basic PeopleSoft Server Processes are:

- **PSAPPSRV.** Performs the functional requests, such as building and loading components (which are also known as panel groups in previous releases). It also provides the memory and disk-caching feature for PeopleTools objects on the application server. The PSAPPSRV is required to be running in any domain.
- **PSQCKSRV.** It performs quick, read-only SQL requests. This is an optional process designed to improve performance by reducing the workload of PSAPPSRV.
- **PSQRYSRV.** Designed to handle any query executed by PeopleSoft Query. This is an optional process designed to improve performance by reducing the workload of PSAPPSRV.
- **PSSAMSRV.** The SQL Access Manager: this process handles the conversational SQL that is mainly associated with Application Designer. This process is required to be running on any domain.

For application messaging, there is a separate set of server processes dedicated to that functionality. Keep in mind that that your messaging domain must also contain PSAPPSRV and PSSAMSRV, the required server processes.

- PSMSGDSP (Required for application messaging only).
- PSMSGHND (Required for application messaging only).
- PSPUBDSP (Required for application messaging only).
- PSPUBHND (Required for application messaging only).
- PSSUBDSP (Required for application messaging only).
- PSSUBHND (Required for application messaging only).



Note. You can examine servers by using the `ps -ef` command in UNIX or the Task Manager in Windows NT. The PeopleSoft configuration utility, PSADMIN, also offers a monitoring utility.

Each of the server processes is discussed in detail in subsequent chapters in the appropriate context.

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 + 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. You can see the association between the many services and server processes by reviewing the PSAPPSRV.UBB file.



For more information on the PSAPPSRV.UBB file, see Executables and Configuration Files.

BEA Products

The PeopleSoft application server uses BEA Systems' transaction monitor, Tuxedo, to handle the PeopleSoft transaction processing. The other BEA software we use, Jolt, which is the communication layer between the PeopleSoft servlets on the web server and the application server. Both Tuxedo and Jolt are required for PIA.

Although we document the BEA components with respect to their function within the context of a PeopleSoft environment, we do not duplicate the documentation provided by BEA. As such, we strongly encourage all system administrators involved with application server-related projects to become familiar with the BEA documentation that we ship along with our product. The BEA documentation provides an extensive error catalog that serves as an *invaluable* source of information when troubleshooting.



For more information on installing the BEA documentation refer to the PeopleSoft *Installation and Administration* PeopleBook.

Tuxedo doesn't actually perform the processing on the application server; it schedules *PeopleSoft Server Processes* to perform the transactions.

Database Connectivity

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.

Once 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 and therefore requires no connectivity software.



For more information on the most current versions of connectivity software, always consult the PeopleSoft Platforms database on Customer Connection.

Batch Server Environment

In a multi server environment, you have options regarding the decision as to where your site's batch server environment resides. In the context of PeopleSoft, the batch server, or batch environment, refers to your Process Scheduler environment. Typically, this is the server on which your Application Engine and other batch programs run. This section contains topics to help clarify the relationship between the application server and the batch server. In short, you

have the option to install the batch server on a separate server, or it can run on either the application server or the database server.



For more information on installing and configuring Process Scheduler see your *Installation and Administration* book and Process Scheduler.

Batch Server Support

You can install your batch server (also known as the PeopleSoft Process Scheduler) on any supported application server, on any supported database server, or on any supported batch server. Where you choose to run your batch environment is entirely up to you, but it is very important that you choose a location that's supported in the PeopleSoft environment. For all database platforms you will have at least two options for your batch environment location.

In cases where you install Process Scheduler on a machine that *is* a supported database machine, but not a supported application server machine, you are able to configure and administer Process Scheduler using PSADMIN.

Unfortunately, we can't remove PSADMIN's application server menu options based on the operating system upon which it runs. So, keep in mind that just because particular options may appear on the screen, as in the Application Server menu, it does not necessarily indicate that you can configure an application server on that machine.

Process Scheduler and the Application Server

PeopleSoft uses PSADMIN, the PeopleSoft Server Administration menu interface, to configure and administer both the application server and Process Scheduler Server (also known as Batch Server). The Process Scheduler setup procedure exists in PSADMIN to provide a menu-driven interface to configure Process Scheduler parameters and administer the Process Scheduler Server Agent.

Even though the application server and Process Scheduler (batch server) have PSADMIN as a common interface and share the directories under the <PS_HOME> on the application server, they are separate entities. For instance, you boot, configure, and shut down the application server and the Process Scheduler server separately.

The application server uses Tuxedo to schedule transaction requests and send transaction results, and it issues SQL requests to the database server using the persistent connections of a collection of PeopleSoft application server processes.

In contrast, Process Scheduler is a separate facility that is designed to poll the PeopleSoft database table, PSPRCRQST, for inserted requests and initiate Application Engine, COBOL, SQR, and other batch processes.

Database Server

The PeopleSoft Database is the repository for all information managed by PeopleSoft applications. Not only is application data stored in the database, but the PeopleSoft metadata is also maintained in the database. The Application Designer enables you to define and maintain this metadata that the system uses to drive the runtime architecture. The application server executes business logic based on the PeopleSoft metadata.

You use Application Designer to define and configure PeopleSoft applications. You can create dozens of different types of application objects, such as Fields, Records, Pages, Messages, and so on. When an application developer saves an application object, the Application Designer saves this definition to the metadata repository in the PeopleSoft database.

At execution time, the application server fetches the most recent application object definitions from the metadata repository, compiles and caches the application object into memory, and executes the business rules based on the definition.

Configuration and Deployment Options

The following topics present your options for deploying PeopleSoft across your enterprise. Because the PeopleSoft Internet Architecture is flexible, this means you have many options to consider.

Configuration Options

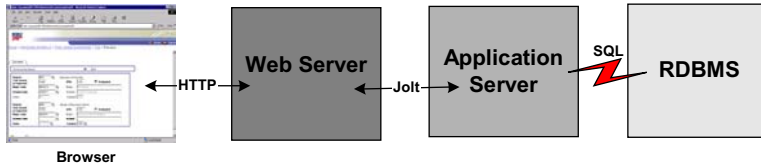
You can configure your environment to support either a “physical” or a “logical” application server configuration. While both configurations are viable options, in some cases, the PeopleSoft standard installation procedure recommends one or the other depending on the combination of database and operating system combination at your site.

Generally, PeopleSoft recommends that you have your application server on the same physical machine as the database server. This would be a logical separation between the application server and database server. If the application server and database server do not reside on the same machine, then we recommend that the application server and the database server are connected to the same, high-performance, backbone network. This ensures optimum performance.

The following sections provide additional details regarding these configuration options.

Physical Application Server Configuration

A *physical* application server configuration means that each component resides on a separate machine. In this configuration the application server and the database server are physically separated. The following example depicts a physical application server configuration.



Physical Application Server Configuration

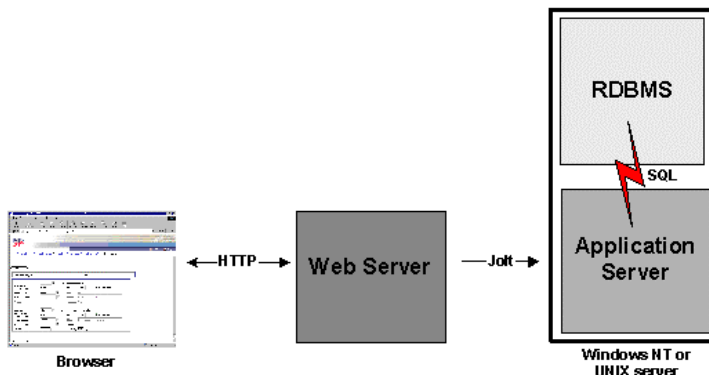
Logical Application Server Configuration

A *logical* application server environment means that only two, server machines exist in the configuration. In this configuration, the application server and database server are not physically, but logically split.

For example, you could have the following:

- Web server
- Server on which both the application server and the database server reside

The following example depicts a logical configuration.



Logical Application Server Configuration

The solid line surrounding the application server and the database server represents one physical machine. In the case of the application server and database server, a logical application server configuration is only possible in situations where both the database server and the application server are supported on a particular operating system. Ultimately, this example is intended to illustrate that certain components can share the same machine.

The previous example depicts the application server and the database server sharing the same machine, but theoretically the web server *could* also reside on the same machine with both the application server and the database server. The only requirement being that each component must be supported by the underlying operating system. With this approach (all servers on the same machine), there are some items to consider, one item being security. If you are deploying PeopleSoft applications to the Internet, you will most likely want your web server outside of your network firewall and not residing on the database server.

How you set up your configuration (logical or physical) depends entirely on your site's requirements.



Note. For demonstration purposes, you can have all components on the same Windows NT or UNIX machine. This configuration can be very useful for development, testing, and training.

Deployment Options

There are a variety of end user deployment options as well as some third party integration solutions that you should consider implementing.

Connecting through PIA

With PIA, end users have direct access to PeopleSoft applications. There is no CAB or JAR file to download containing an applet. The application server handles all browser service requests as well as generates all of the HTML.

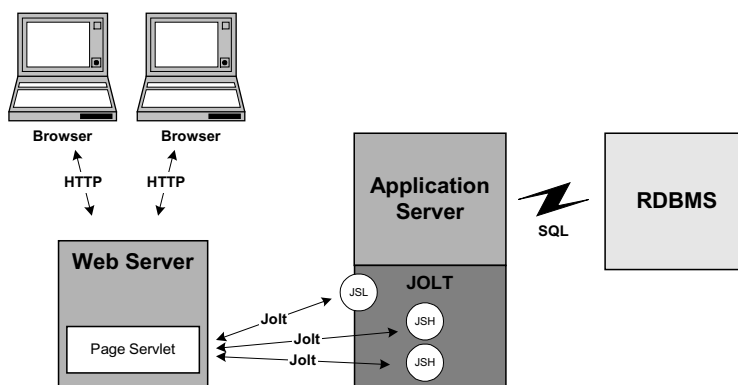
For a PIA connection, the following items need to be installed:

- A supported browser on the workstation.
- A supported, Java-enabled web server with the Page Servlet installed.
- An application server domain with Jolt configured.



For more information on the installing the Page Servlet see your *Installation and Administration* PeopleBook.

The following example depicts the architecture involved in deploying PIA.



Connecting with a Browser through the Page Servlet

The workstation only requires a supported browser. All a user needs to do is enter the correct URL of the Page Servlet or click a link in a page assembled by the PeopleSoft portal. All processing occurs on the Web server and application server, and only pure HTML arrives at the browser creating a thin, low bandwidth connection to PeopleSoft.

On the Web server is the Page Servlet to which the browser connects. The Page Servlet is responsible for formatting the HTML generated on the application server for presentation in the browser as well as managing connections to the JSL.

Since the PIA connections originate from a web-based environment, Jolt is required to act as the communications layer between the Page Servlet and the application server. The Page Servlet routes the browser request to the JSL (Jolt Server Listener) that, in turn, routes the Browser request to an available JSH (Jolt Server Handler). The JSH passes the request on to the application server, which transmits the request to the database using SQL.



Note. For every number of browsers sending requests you will have an equal number of connections to the JSHs. Some sites prefer to have only a single connection to an application across a firewall. BEA offers a method by which you can create a single connection to the Jolt port by using the Jolt Internet Relay feature. However, keep in mind that our testing reveals that using Jolt Internet Relay does not offer optimum performance. The single connection tends to create bottlenecks when the volume of transaction requests increases.



For more information on Jolt Internet Relay see Administration Considerations.

On the application server, the PSAPPSRV server process generates the HTML required to present the interface to the browser. PSAPPSRV receives the incoming requests and it routes the results to the browser through the same JSH from which the request came.

As far as the database is concerned, the request from a browser appears in SQL just like a typical client/server or batch program transaction request. Therefore, it handles browser requests no differently and offers no restrictions for browsers.

PeopleSoft Portal

The PeopleSoft Portal is another option that you have for deploying PeopleSoft applications to a browser. The portal enables you to integrate PeopleSoft content with the content from other data sources. The PeopleSoft portal can stand on its own or you can integrate it with any enterprise portal you may already be using.

The PeopleSoft Portal technology consists of the Portal Servlet and an application server. These two components work together to provide common portal processing such as page assembly, search ability, content management, navigation, and home page personalization.

The portal technology is an important integration technology because with it you combine content from a wide variety of data sources and deliver the result to end users in a unified, simple-to-use interface.



For more information on the portal see Portal Technology.

Connecting from the Development Environment

Although the majority of your end users connect using their browser, your applications developers and system administrators, who need access to PeopleTools, need to use Windows workstations. Application Designer and various other PeopleTools are not accessible through a browser.

With the Development Environment you can connect directly to the database (two-tier), or you can connect through an application server (three-tier).



For more information on Jolt Internet Relay see Administration Considerations.

Integration Technologies

Although the PeopleSoft integration solutions don't exactly deploy an interface to an end user, they do enable you to share information with third party systems and other PeopleSoft databases. To take advantage of these integration solutions, you must configure PIA.

- **Application Messaging.** Publish/subscribe messaging architecture for asynchronous integration and data synchronization. The messaging technology handles inbound and outbound messages.
- **Component Interfaces.** Object-oriented, request/reply, component architecture that enable third-party applications to synchronously invoke PeopleSoft business logic.
- **Business Interlinks.** Plug-in framework that enables PeopleSoft applications to easily invoke third-party, Application Program Interface (API) over the Internet.

Developing for and implementing the integration solutions is beyond the scope of this administration document.



For more information on integration options, see the PeopleTools Integration documentation suite.

Architecture Terminology

This section contains a list of important terms related to the PeopleSoft architecture with a brief description of each. This may be useful as you proceed through the following chapters.



Note. The terminology associated with our messaging architecture does not appear in the following list. It appears in the application messaging discussion.



For more information on the terminology associated with our messaging architecture see Application Messaging Architecture.

Application Server Domain

The collection of server processes, services, and associated resource managers defined by a single PSTUXCFG configuration file. Each application server domain is configured to connect to a single database. When you boot a domain, multiple processes start, such as WSL, JSL, PSAPPSRV, PSQCKSRV, and PSSAMSRV.

Jolt Server Handler (JSH)

The JSH handles the transaction for a transaction request originating from the Internet. The JSH manages network connectivity, making service requests from the Jolt Repository, and translating Tuxedo buffer data into the Jolt buffer.

Jolt Server Listener (JSL)

The JSL handles the work of the browser request just like the WSL does for the Windows workstation. It tracks (or listens) for browser requests and then hands off connections to an available JSH.

Jolt

Jolt is a BEA Tuxedo companion product that runs on an application server domain. It is designed to listen for browser requests and transfer them to the application server for processing. Jolt is the communication layer between Tuxedo and the Internet.

Messaging Servers

The collection of application server processes designed to facilitate the PeopleSoft Application Messaging system, which is an asynchronous, publish/subscribe approach to data synchronization across an enterprise.

PeopleSoft Portal

The PeopleSoft Internet portal technology enables you to manage content, integrate PeopleSoft interfaces with other data sources, and provide easy navigation for PeopleSoft applications.

Page Servlet

The PeopleSoft Java program that resides on a web server and enables the browser to transmit requests to the Jolt port. It also formats the HTML produced by PSAPPSRV so that it appears appropriately within the browser.

PSADMIN

A PeopleSoft command-line utility that provides a simple interface to create, configure, and administer application server domains and Process Scheduler Servers. The PSADMIN utility looks the same on Windows NT and UNIX operating systems.

PSAPPSRV

PSAPPSRV is the main server process running within a domain. PSSAPPSRV performs the functional requests, such as building and loading components. It also manages the memory and disk-caching for PeopleTools objects on the application server. Each PSAPPSRV process maintains its own memory and disk cache.

It provides authentication services for incoming users. For instance, it checks the PeopleSoft OPRID against the directory server or PSOPRDEFN table.

PSQCKSRV

Essentially, PSQCKSRV is a copy of the PSAPPSRV. It performs quick, read-only SQL requests. It is an optional Server Process designed to improve performance by handling items in the PSAPPSRV transaction request queue.

PSQRYSRV

Like the PSQCKSRV server process, PSQRYSRV is designed to alleviate the workload of PSAPPSRV. PSQRYSRV is designed to specifically handle all user-generated queries submitted by PeopleSoft Query (PSQED.EXE). This server process is designed to improve overall application server performance whether or not you have PSQCKSRV configured. It is specifically, and exclusively designed to process PeopleSoft Query transactions, which can be very SQL intensive.

PSSAMSRV

It processes conversational SQL transactions primarily for Application Designer.

Server Process

A Server Process is executable code, written by PeopleSoft that receives and processes incoming requests. Examples of PeopleSoft server processes are PSAPPSRV, PSQCKSRV, and PSSAMSRV, to name a few. The server process carries out a transaction request by making calls to a Service.

Service

Application programmers develop Services to perform a particular task of the application.

Examples of services are MgrGetObj, SQLAccess, RemoteCall, and so on. When a PeopleSoft application sends a request to the application server, it sends a service name and a set of parameters, such as “MgrGetObject + parameters”.

Servlet

A program written in Java that runs on the web server. PeopleSoft provides a variety of servlets that you install onto your web server and each one handles a different aspect of our Internet and integration technology.

Tuxedo

BEA’s middleware product used to manage transaction queues, Server Process initiation, system administration, time-outs, data encryption, compression, and logging.

Web Server

An HTTPS, Java enabled server used in the PeopleSoft architecture to deploy the PeopleSoft applications to browsers and act as the hub for the PeopleSoft Application Messaging technology. The web server has a variety of PeopleSoft Servlets installed that handle the individual tasks involved in completing transactions.

Windows Workstation/Development Environment

PeopleTools, such as Application Designer and PSQuery, require a Windows workstation. These PeopleTools are programmed with C++ and require a considerable amount of RAM and system resources to run effectively. These programs are not lightweight tools intended for casual users. Only application developers, system administrators, and a handful of power users need to be configured for the Development Environment.

Workstation Listeners (WSL)

This process is the initial point of contact for Windows workstations. After it receives the transaction request, it passes the request on to the Workstation Handler.

At that point, the workstation only “communicates” with its assigned Workstation Handler.

The Workstation Listener also monitors the distribution of connections to Workstation Handlers and when the number of connections increases past a defined limit, the JSL starts, or spawns, new handlers to meet the demand.

Workstation Handlers (WSH)

Receives the transaction request from the WSL and “handles” the workstation transaction. You can increase the pool of available WSHs to dynamically accommodate workload as demand increases.

CHAPTER 2

The PSADMIN Utility

The term “PSADMIN” is short for PeopleSoft Server Administration. PeopleSoft developed PSADMIN to simplify the process of configuring and administering all the servers and features available on the application server. For example, to configure your application server domains, Process Scheduler Servers, or Windows NT Services, you use PSADMIN. The PSADMIN provides a friendlier interface to Tuxedo’s native configuration and administration command line interface.

PSADMIN performs the following:

- Acts as a front-end to BEA/Tuxedo’s native configuration process and provides a friendly interface to create and build the “UBB” file.
- Provides the same interface on Windows NT and UNIX.
- Simplifies the execution of Tuxedo operations such as boot, shutdown, and checking status.

Starting PSADMIN

This section assumes that you already have your PeopleSoft Application Server installed and configured as described in the *PeopleSoft Installation and Administration* book for your RDBMS platform.

To start the PSADMIN utility

1. Launch your command interface on Windows NT or UNIX.
2. Change directory to the \appserv directory beneath the high-level PeopleSoft directory on the application server.

For example, if your application server runs on Windows NT, enter:

```
cd ps_home\appserv
psadmin
```

And, if your application server runs on UNIX, enter:

```
CD $PS_HOME/APPSESV
psadmin
```

3. Choose the server that you want to configure, administer, or monitor from the **PeopleSoft Server Administration** menu.

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

- ```

1) Application Server
2) Process Scheduler
3) Web Components
4) Service Setup
q) Quit

```

Command to execute (1-4, q):

## Using PSADMIN

Using the PSADMIN merely involves selecting the number of the menu item that reflects the action you want to take place, entering the correct number on the command line, and pressing ENTER. However, in some cases, you may want to take advantage of the command line options that PSADMIN offers. After reading this section you will be familiar with PSADMIN's menu options, where to find them, and how to select them.

---

### Menu Interface

Each PSADMIN menu presents the same look and feel. The menus all contain the following components:

- **Menu Title.** In the following example, PeopleSoft Server Administration.
- **Menu Item.** These are the numbered items beneath the menu title that you select to invoke a particular action or menu.
- **User Prompt.** This is where you enter your menu option selection. This is the **Command to execute** prompt.

```

PeopleSoft Server Administration

```

- 1) Application Server
- 2) Process Scheduler
- 3) Web Components
- 4) Service Setup
- q) Quit

Command to execute (1-4, q):

### Selecting Menu Options

To select a menu item, you just need to enter the corresponding number at the prompt and press ENTER. For example, if you wanted to view the menus associated with the application server, you would enter *1* on the command line, and press ENTER.



---

Just enter the digit for the menu item. Don't enter *01*, for example or any text; if you do, an error message will appear.

---

For example,

```

PeopleSoft Server Administration

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

Command to execute (1-4, q): **1**

Entering *1* invokes the following menu:

```

```

```

PeopleSoft Application Server Administration

```

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

```

Command to execute (1-3, q) :

```

### Navigating to Previous Menu

To return to the previous menu enter *q*, for **Quit**, on the command line, as shown, and press ENTER:

```

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

```

Throughout all the menus in this administrative utility, selecting *q* and pressing ENTER will return you to the previous menu (or the previous *layer* of menus depending on how you look at it). So, if you've entered the wrong number and find yourself at the wrong menu, you can backtrack to your previous decision point.

The **PeopleSoft Server Administration** menu is the “main” menu in the PSADMIN, and it is also the first menu you see upon launching the PSADMIN utility from the command line. In that sense it is the first layer of menu options. When you select *q* and press ENTER from this menu, you will exit PSADMIN entirely.

In some situations, if you accidentally enter the wrong number on the command line and press ENTER, you can return to the previous menu by entering *q* and pressing ENTER or by just pressing ENTER. For example, if you accidentally select 3 from the **PeopleSoft Application Server Administration** menu, you can enter *q* or just press enter on the **Select domain number to delete** command, as shown in the following example. By doing so, you arrive at the previous menu.

For example,

```

PeopleSoft Application Server Administration

1) Administer a domain
2) Create a domain

```



3) Delete a domain

q) Quit

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

The following appears:

Tuxedo domain list:

1) ps800dmo

Select domain number to delete:

Press ENTER, or enter *q* then press ENTER.

```

PeopleSoft Application Server Administration

```

1) Administer a domain

2) Create a domain

3) Delete a domain

q) Quit

Command to execute (1-3, q) :

---

## Menu Structure

Some of the menu items that you select in the PSADMIN utility lead to multiple sets, or layers, of additional menu items. Rather than hunting through the structure on your own using a trial-and-error technique, you can use the following diagrams to make yourself familiar with the menu structure within the PSADMIN utility.

This will save you time as you become acquainted with PSADMIN. Certainly, after you've used the PSADMIN for a short time, you'll need to refer to these charts less and less.

The following sections are divided according to menu items that appear on the **PeopleSoft Server Administration** menu—the PSADMIN “home” menu.

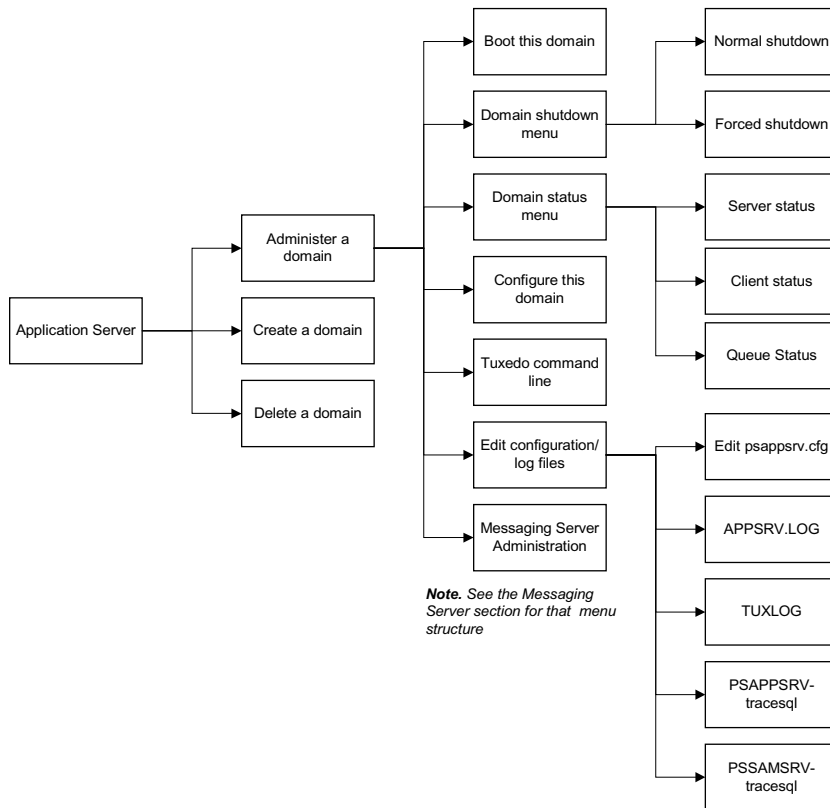
-----  
PeopleSoft Server Administration  
-----

- 1) Application Server
- 2) Process Scheduler
- 3) Web Components
- 4) Service Setup
- q) Quit

Command to execute (1-4, q):

## Application Server

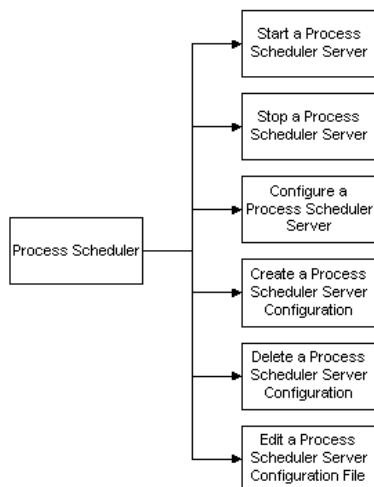
The following diagram illustrates all the menu items that stem from the *Application Server* menu item on the **PeopleSoft Server Administration** menu.



PSADMIN Application Server Menu Structure

## Process Scheduler

The following diagram illustrates all the menu items that stem from the *Process Scheduler* menu item on the **PeopleSoft Server Administration** menu.



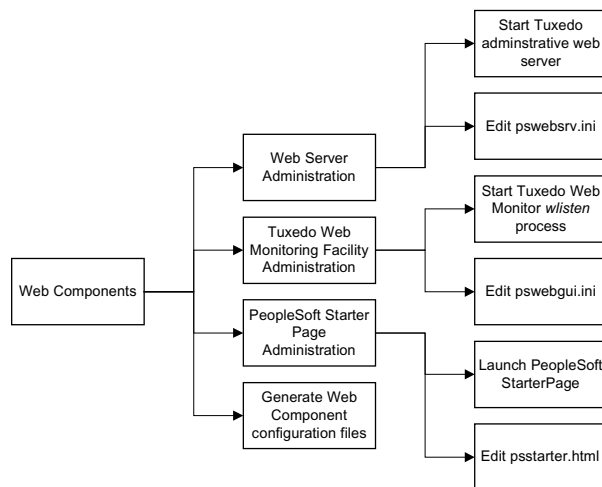
PSADMIN Process Scheduler Menu Structure

## Web Components

The following diagram illustrates all the menu items that stem from the *Web Components* menu item on the **PeopleSoft Server Administration** menu.



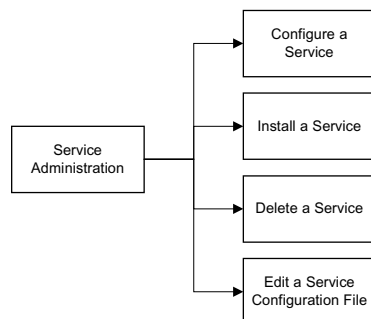
This set of menu options is intended for use with the Web Client, an internet solution from previous releases, and for supporting the Tuxedo monitoring utility. The Web Components menu *is not* what you use to configure the internet architecture.



PSADMIN Web Components Menu Structure

## Service Setup

The following diagram illustrates all the menu items that stem from the *Service Setup* menu item on the **PeopleSoft Server Administration** menu



PSADMIN Service Setup Menu Structure

## Configuration Templates

The initial values that you will see in PSADMIN are derived from the configuration template that you choose when you create your domain. The templates provide a starting point for your further configurations. The delivered templates provide a range of possible implementations. The delivered templates are as follows:

- **Small.** This template is designed for a range of users in the range of 1-100.
- **Medium.** This template is designed for a range of users in the range of 100-500.
- **Large.** This template is designed for a range of users in the range of 500-1000.
- **Developer.** The developer template is intended for development and demonstration environments only.



As mentioned previously, these templates provide a starting point for configuring domains. Performance may vary due to increased transaction volume and various other factors. It's important for you application server administrator to monitor the performance of the domains and make the appropriate adjustments as necessary.

Each configuration template includes a number of server processes, such as PSAPPSRV, that is sufficient for its intended use. Keep in mind that you can easily modify and create your own configuration templates to fully include your site's needs. The configuration templates are .CFX files that you can locate in the PS\_HOME\appserv directory on you application server.

You can modify the CFX files using any text editor, such as Notepad.

```

developer.cfx - Notepad
File Edit Search Help
[Startup]
;=====
; Database Signon settings
;=====
DBName=
DBType=MICROSFT
OprId=PTDMO
OprPswd=PTDMO
ConnectId=
ConnectPswd=
ServerName=

[Database Options]
;=====
; Database-specific configuration options
;=====
SybasePacketSize=
; Please see Chapter "Tuning and Administration", in
; Oracle Installation and Administration Guide for details
UseLocalOracleDB=0
;ORACLE_SID=
EnableDBMonitoring=0

```

Modifying a CFX file

Use the Save As option to create your own template.



For more information on the Developer configuration template, see Configuring PSMBSRV and PSMBHND.

## Command Line Options

In some cases you may want to use the PSADMIN command line options rather than launching the PSADMIN interface and navigating to a particular menu. The command line options save you time and offer a direct method of executing select tasks on the application server.

Before you begin using the PSADMIN commands, we recommend that you become generally familiar with PSADMIN and the components it controls.

### Syntax

The syntax to which you need to adhere is as follows:

```
psadmin -c <command> -d <domain/database> -t <template if applicable>
```

For example, if you wanted to boot a domain, you would enter the following:

```
psadmin -c boot -d ps800dmo
```

### General Administrative Commands

The following table contains the commands that you can submit on the command line and bypass the PSADMIN utility.

| <b>Command</b>            | <b>Example</b>                                                                 | <b>Result</b>                                                                      |
|---------------------------|--------------------------------------------------------------------------------|------------------------------------------------------------------------------------|
| <b>Application Server</b> |                                                                                |                                                                                    |
| boot                      | psadmin -c boot -d ps800dmo                                                    | Boots an application server.                                                       |
| shutdown                  | psadmin -c shutdown -d ps800dmo                                                | Shuts down an application server domain using a “normal” shutdown method.          |
| shutdown!                 | psadmin -c shutdown! -d ps800dmo                                               | Shuts down an application server domain using a “forced” shutdown method.          |
| create                    | psadmin -c create -d ps800dmo -t small<br>-s <startup_string> -p <port_string> | Creates an application server configuration file with specified template. Where -t |

| <b>Command</b>           | <b>Example</b>                   | <b>Result</b>                                                                                                                                                                                                      |
|--------------------------|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| configure                | psadmin -c configure -d ps800dmo | specifies the template to use: small, medium, or large.<br><br><b>Note.</b> The <code>-s</code> and <code>-p</code> parameters are discussed in the following section.<br><br>Invokes the configuration interface. |
| <b>Process Scheduler</b> |                                  |                                                                                                                                                                                                                    |
| create                   | psadmin -c create -d hrdmo -t nt | Creates a new Process Scheduler Server. Where <code>-d</code> specifies the database, and <code>-t</code> specifies the template to use, as in <i>nt</i> , <i>unix</i> , or <i>vms</i> .                           |
| start                    | psadmin -c start -d hrdmo        | Starts a Process Scheduler.                                                                                                                                                                                        |
| stop                     | psadmin -c stop -d hrdmo         | Stops a Process Scheduler.                                                                                                                                                                                         |
| <b>General</b>           |                                  |                                                                                                                                                                                                                    |
| help                     | psadmin -h                       | Displays command help and syntax.                                                                                                                                                                                  |
| version #                | psadmin -v                       | Displays version number, as in <b>Version 8.10</b> .                                                                                                                                                               |
| environment              | psadmin -env                     | Displays your environment variables.                                                                                                                                                                               |

---

## Command Line Create and Configure

You can create and configure an application server domain directly from the command line. We added this functionality to simplify the task of creating numerous domains that will use default server settings.

### Overview

We've extended the PSADMIN command line functionality to allow you to fully create and configure a simple application server domain from the command line. By "simple" we mean a domain that, for the most part, reflects the default settings packaged in the domain Configuration templates, as in small or medium.

Recall that submitting the following command to PSADMIN created a domain referencing the chosen template.

```
psadmin -c create -d <domain_name> -t <template>
```

With the `-s` and the `-p` option, you can include two strings of values to supply the PSAPPSRV.CFG configuration file with the minimum settings required to create and configure an application server domain. So, now you can create *and* configure a domain from the command line. The two strings of values are the Startup string and the Port string.

The syntax when including the Startup string and the Port string on the command line is

```
psadmin -c create -d <domain_name> -t <template> -s <startup_string> -p
<port_string>
```



**Note.** Keep in mind, you can always enter `psadmin -h` on the command line for syntax help.

---

## Including Unique Values

You need to separate the values that you include in the Startup and Port strings with a forward slash (/). The values may not contain spaces, nor can a previous value be left blank while providing a value intended for a following parameter.

For example, the Startup string includes the following values:

- Database name (DBNAME)
- Database type (DBTYPE)
- Operator ID/User ID (OPR\_ID)
- Operator Password/User Password (OPR\_PSWD)
- Domain Name (DOMAIN\_ID)
- Add to Path (ADD\_TO\_PATH)
- Connect ID (CNCT\_ID)
- Connect Password (CNCT\_PSWD)
- Server Name (SERV\_NAME)

On the command line, it needs to appear in the following order:

```
psadmin -c create -d domain -t template -s DBNAME/DBTYPE/
OPR_ID/OPR_PSWD/DOMAIN_ID/ADD_TO_PATH/CNCT_ID/CNCT_PSWD/SERV_NAME
```

The Domain Name and Add to Path values reside in the Domain Settings section of the PSAPPSRV.CFG, and the rest of the Startup values reside in the Startup section of the PSAPPSRV.CFG.

Similarly, the Port string contains the following values:



- Workstation Listener Port (WSL\_PORT)
- Jolt Port (JSL\_PORT)
- Jolt Internet Relay Adapter Port (JRAD\_PORT)

If you chose to include the Port string on the command line, it would appear as:

```
psadmin -c create -d domain -t template -s DBNAME/DBTYPE/
OPR_ID/OPR_PSWD/DOMAIN_ID/ ADD_TO_PATH/CNCT_ID/CNCT_PSWD/SERV_NAME
-p WSL_PORT/JSL_PORT/JRAD_PORT
```

The values in the Port string control the Port parameter in the Workstation Listener, Jolt Listener, and Jolt Relay Adapter sections of the PSAPPSRV.CFG.

### Default Values

The only "required" value is DBNAME. After that, the individual values in the strings may be truncated from right-to-left. If you do not specify a value, PSADMIN uses a default value.

The defaults are:

- DBTYPE (MICROSFT)
- OPRID (PTDMO)
- OPRPSWD (PTDMO)
- DOMAIN\_ID (same as DBNAME)
- ADD\_TO\_PATH (c:\apps\db\mssql7\bin)
- CONNECT\_ID (blank)
- CONNECT\_PSWD (blank)
- SERVER\_NAME (blank)

As for the -p string, we introduced it to allow you to set port settings without having to specify all of the other values.

The defaults are:

- WSL\_PORT (7000)
- JSL\_PORT (9000)
- JRAD\_PORT (9100)



**Note.** If the default values contained in the delivered CFX files are not acceptable, keep in mind that you can easily modify the CFX files in the <PS\_HOME>\appserv directory to reflect your environment.

## Valid Values

The following table provides more information regarding the valid parameters and string values associated with the create and configure command line option.

| <b>Parameter/Option</b> | <b>String Values</b> | <b>Description</b>                                                                                                                                                                                                                                                                    |
|-------------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| -c create               |                      | As with the other command line parameters you need to enter the initial command. In this case, use create.                                                                                                                                                                            |
| -d <domain>             |                      | Enter the name of the new domain that you want to create. For example, HR800DMO                                                                                                                                                                                                       |
| -t <template>           |                      | Enter the template for your new domain: small, medium, or large.                                                                                                                                                                                                                      |
| -s <startup string>     |                      | For the -s flag, you need to enter the following values in the exact order and include the forward slash (/) between each value. After you enter all of the values required by your RDBMS, this will comprise your startup string.                                                    |
|                         | /DBNAME              | Enter the name of the database name to which the application server will connect. (From the Startup section in PSAPPSRV.CFG).                                                                                                                                                         |
|                         | /DBTYPE              | Enter your database type, as in MICROSOFT or INFORMIX. (From the Startup section in PSAPPSRV.CFG).                                                                                                                                                                                    |
|                         | /OPR_ID              | Add the Operator ID, such as PTDMO, that you will use to connect to the database. (From the Startup section in PSAPPSRV.CFG).                                                                                                                                                         |
|                         | /OPR_PSWD            | Add the operator password, such as PTDMO, that is associated with the specified Operator ID. (From the Startup section in PSAPPSRV.CFG).                                                                                                                                              |
|                         | /DOMAIN_ID           | Enter a Domain ID, such as TESTSRV1, TESTSRV2, and so on. This does not need to match the domain name. This name is important only in that the Tuxedo Web Monitor uses it to identify application server domains on each machine. (From the Domain Settings section in PSAPPSRV.CFG). |

| <b>Parameter/Option</b> | <b>String Values</b> | <b>Description</b>                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|-------------------------|----------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
|                         | /ADD_TO_PATH         | Add the directory that contains your connectivity software, or database drivers. (From the Domain Settings section in PSAPPSRV.CFG).                                                                                                                                                                                                                                                                                                                          |
|                         | /CNCT_ID             | Connect IDs are required for all platforms. (From the Startup section in PSAPPSRV.CFG).                                                                                                                                                                                                                                                                                                                                                                       |
|                         | /CNCT_PSWD           | Specify the password associated with the Connect ID. (From the Startup section in PSAPPSRV.CFG).                                                                                                                                                                                                                                                                                                                                                              |
|                         | /SERV_NAME           | If your RDBMS requires that you specify the Server Name on which the database resides, enter the appropriate Server Name. (From the Startup section in PSAPPSRV.CFG).                                                                                                                                                                                                                                                                                         |
| -p <port string>        |                      | The -p command line parameter supports an optional set of values that you can specify for your domain. The values you do specify comprise your port string. Typically, you'll only specify values here if you have more than domain on the same application server machine, or if for some reason, you need to specify a specific value due to your environment or testing needs. Otherwise, we recommend that accept the defaults for ease of configuration. |
|                         | /WSL_PORT            | This controls the Workstation Listener port setting. If you need to change the Workstation Listener port to reflect a unique value, enter that value. For example, enter 7100. (From the Workstation Listener section in PSAPPSRV.CFG).                                                                                                                                                                                                                       |
|                         | /JSL_PORT            | This controls the Java Station Listener port setting. If you do not intend for a domain to support browser deployment, you do not need to specify a value for the JSL_PORT. (From the Jolt Listener section in PSAPPSRV.CFG).                                                                                                                                                                                                                                 |
|                         | /JRAD_PORT           | This controls the Jolt Relay Adapter port setting. You only need to specify a value here if you intend to support browser deployment <i>and</i> your web server resides on a separate machine than the application server. (From the Jolt Relay Adapter section in PSAPPSRV.CFG).                                                                                                                                                                             |

Whether or not you use these command line options depends on why you're creating and configuring a domain. If you creating a domain that requires tuned variables based on the needs of your site, then you'd want to use the PSADMIN menu interface to enter the tuned variables per configuration section.

However, there are occasions where a systems administrator or a developer needs to set up a simple application server domain for which the defaults contained in the packaged templates are sufficient. The extended command line options provide a convenient alternative to stepping through each PSADMIN configuration prompt.

To create and configure an application server domain from the command line

1. Change directories to the PS\_HOME\appserv directory on the application server.

For example,

```
C:\cd hr800\appserv
```

2. Enter psadmin on the command line and submit the command line parameters for -c <command> -d <domain>, -t <template>, -s <startup>, and -p <port>.

The following example shows the basic syntax you'll need to use when you configure a domain using this command line option.

```
C:\hr800\appserv\psadmin -c create -d <domain> -t <template> [-s
<startup_string> [-p <port_string>]]
```

The following example shows the syntax for the startup string, which you enter after -s.

```
DBNAME/DBTYPE/OPR_ID/OPR_PSW/DOMAIN_ID/ADD_TO_PATH/CNCT_ID/CNCT_PSWD/SERV_NAME
```

The following example shows the syntax for the optional port string, which you enter after -p.

```
WSL_PORT/JSL_PORT/JRAD_PORT
```

Your final command line may look similar to the following:

```
C:\hr800\appserv\psadmin -c create -d HR800DMO -t small -s
HR800DB1/MICROSFT/PS/PS/TESTSRV2/c:\apps\db\mssql7\bin -p 7100/9010
```

You only need to include the parameters that apply to your RDBMS.

3. Press ENTER.

In your command screen, you should see messages that resemble the following.

```
Copying application server configuration files...
```

```
copying [small.cfx] to [HR800DMO\psappsrv.cfg]
```

```
Copying Jolt repository file...
```

```
Domain created.
```

```
Loading UBBGEN configuration utility with "-s HR800DB1/MICROSFT/PS/PS/TESTSRV2/c
:\apps\db\mssql7\bin -p 7100/9010"...
```

```

setting DBName=HR800DB1

setting DBType=MICROSFT

setting OprId=PS

setting OprPswd=PS

setting ConnectId=

setting ConnectPswd=

setting ServerName=

setting Port=7100

setting Port=9010

setting Listener Port=9100

setting Domain ID=TESTSRV2

setting Add to PATH=c:\apps\db\mssql7\binn

New CFG file written with modified Startup parameters

Log Directory entry not found in configuration file.

Setting Log Directory to the default... [PS_SERVDIR\LOGS]

PSAUTH Spawning disabled because Max Instances <= Min Instances.

Configuration file successfully created.

CFG setting changes completed, loading configuration...
```

## Quick Configure

Immediately after you create a domain, you will be prompted to configure it using the Quick configure menu.

```
Would you like to configure this domain now? (y/n) [y] :
```

If you enter *y*, then the Quick-configure menu appears.

```

Quick-configure menu -- domain: PT810QC

Features Settings
```

```

=====
1) Pub/Sub Servers: No 8) DBNAME : [PT8]
2) Quick Servers : No 9) DBTYPE : [MICROSFT]
3) Query Servers : No 10) OPRID : [PTDMO]
4) Jolt : Yes 11) OPRPSWD : [PTDMO]
5) Jolt Relay : No 12) DomainID : [PT8]
 13) AddToPATH : [C:\Apps\Db\Mssql17\Binn]
 14) ConnectID : [psft]
 15) ConnectPswd: [psft8]
 16) ServerName : []
Actions
=====
17) WSL Port : [7000]
18) JSL Port : [9000]
6) Load config as shown 19) JRAD Port : [9100]
7) Custom configuration
h) Help for this menu
q) Return to previous menu

```

HINT: Enter 8 to edit DBNAME, then 6 to load

Enter selection (1-19, h, or q):

The Quick-configure menu is not intended to replace the series of configuration sections in the PSADMIN interface. In most cases, your site will require the custom parameters and tuning options that are only available through the full PSADMIN menu. For this reason, the Quick-configure menu is provided for situations where you are setting up a demonstration domain for testing or for development needs.

This Quick-configure menu shows which features are currently set for the newly created domain. The menu contains the values most commonly changed when setting up a demonstration or test domain. If you happen to be responsible for setting up numerous domains on your site for testing or demonstration, the Quick-configure menu could save you a significant amount of time.

To change the value of a parameter under Features, just enter the number corresponding to the feature and that will toggle the feature on or off.

To change the value of a parameter under Settings, enter the number corresponding to the setting and enter the new value at the prompt. For example, if you wanted to change name of the

database, just enter 8 for DBNAME. Then on the following prompt, enter the name of the database to which the application server should connect.

```
Enter new setting for DBNAME [PT8]:NEWDB
```

To perform any of the items under the Actions menu, just enter the number corresponding to the action. If you want to leave the Quick-configure interface and perform a “traditional” domain configuration using the PSADMIN interface, just enter 7 and respond positively when prompted by the following:

```
Do you want to change any config values (y/n)? [n]:
```

If you need help while using Quick-configure, enter *h* to invoke online descriptions of the interface.

## Executables and Configuration Files

You can create, configure, and boot an application server domain all from the PSADMIN interface or through its command line options. In fact, you can use PSADMIN to control or configure just about any aspect of an application server domain. Because of this, it is not *essential* to become familiar with the executables and configuration files that comprise the operation of the application server domain.

However, for troubleshooting and general knowledge we recommend that you read the following section to gain an appreciation of the main files and executables associated with an application server domain.

In this topic we’ll discuss the main executables and configuration files that allow you to configure and boot a PeopleSoft Application Server domain.

The executables that will be addressing are:

- **PSADMIN.EXE.** This PeopleSoft executable resides in PS\_HOME\appserv.
- **UBBGEN.EXE.** This PeopleSoft executable resides in PS\_HOME\bin\server\winx86.
- **TMLOADCF.EXE.** This BEA executable resides in TUXDIR\bin.
- **TMBOOT.EXE.** This BEA executable resides in TUXDIR\bin.
- **TMSHUTDOWN.EXE.** This BEA executable resides in TUXDIR\bin.

The configuration/data files on which the executables rely all reside in the following directory PS\_HOME\appserv\<domain\_name>. Each domain has its own set of these files. Namely, these files are:

- **PSAPPSRV.CFG.** This is the catchall configuration file that contains the entire collection configuration values for a given application server domain.
- **PSAPPSRV.UBX.** This is the template or model file for PSAPPSRV.UBB.
- **PSAPPSRV.UBB.** This is the file that stores and passes all of the domain values to the Tuxedo

load configuration program (tmloadcf.exe).

- **PSAPPSRV.PSX.** This file is the template or model file specifically for the messaging server configuration sections, such as PSBRKRSRV, PSSUBSRV, and so on.
- **PSAPPSRV.ENV.** This file contains environment information such as the PS\_HOME and connectivity locations on the application server machine.
- **PSAPPSRV.VAL.** The VAL file contains valid values for selected parameters. This helps to prevent administrators from entering invalid values.
- **PSTUXCFG.** This file contains PeopleSoft and Tuxedo information regarding the location of executables, files, as well as command lines for server processes. This file is required to boot a domain.

There are more files and executables associated with Tuxedo and a Peoplesoft Application Server Domain. This general discussion is limited to the mentioned files. After reading this section you will have an understanding of what occurs “under the covers” as you use PSADMIN to execute the basic application server functions.



For more information on all of the executables and binaries that comprise BEA Tuxedo and Jolt. You should consult the BEA Online Documentation.

---

---

## Configuring a Domain

As you read further regarding PSADMIN, you will see that you are not required to use the PSADMIN interface to create and configure a domain. As stated in the previous section, you can use the command line option. You can also manually edit the PSAPPSRV.CFG instead of prompting through the PSADMIN menus.

Regardless of how you choose to specify domain values, ultimately you need to run PSADMIN to generate some necessary files that will include your custom values. As you'll see in the following example, PSADMIN calls another PeopleSoft executable UBBGEN, which is designed to read the values and format stored in the PSAPPSRV.CFG, VAL, and UBX files and write to the PSAPPSRV.UBB and ENV files.



```

PeopleSoft Domain Administration

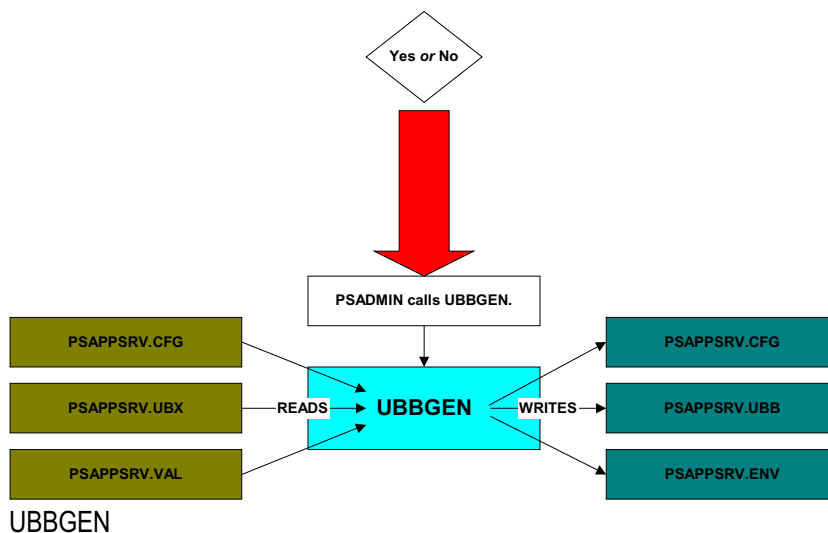
Domain Name: PSXXDEMO

1) Boot this domain
2) Domain shutdown menu
3) Domain status menu
4) Configure this domain
5) TUXEDO command line (tmadmin)
6) Edit configuration/log files menu
7) Messaging Server Administration menu
q) Quit

Command to execute (1-7, q) :4

Removing any existing configuration...
Generating new configuration...
Loading validation table...
Do you want to change any config values (y/n)? [n]:

```



At the point where you see **Do you want to change any config values? (y/n)** regardless of what you enter, ultimately PSADMIN calls UBBGEN.

If you already entered values manually in the PSAPPSRV.CFG file and enter *n*, UBBGEN will read those and write to the necessary files.

If you enter *y*, you'll see the PSADMIN prompt interface, which is actually a wrapper to UBBGEN. UBBGEN reads the previous values stored in the PSAPPSRV.CFG, and presents those values and allows you to change them. It presents the values in the format derived from reading the PSAPPSRV.UBX file and it validates selected values based on criteria stored in the PSAPPSRV.VAL file.



**Note.** In the previous example, it appears that UBBGEN both reads from and writes to the PSAPPSRV.CFG file. It reads the previous values or defaults and, if any values are modified, it writes the new files to the “new” PSAPPSRV.CFG.

So, here are the scenarios by which you can configure a domain:

- **PSADMIN.** Launch PSADMIN, and enter values at all the prompts. This generates all of the necessary files automatically.
- **EDIT PSAPPSRV.CFG.** If you feel compelled *not* to use PSADMIN you will need to complete the following tasks in order:
  - From the command line create a domain based on a particular template.
  - Edit the PSAPPSRV.CFG in your favorite text editor.
  - Then, issue the **configure** command from the PSADMIN command line. This is the command that calls UBBGEN. In fact, you'll see the following after issuing this command:

```
C:\pt8apsrv\Appserv>psadmin -c configure -d 80manual

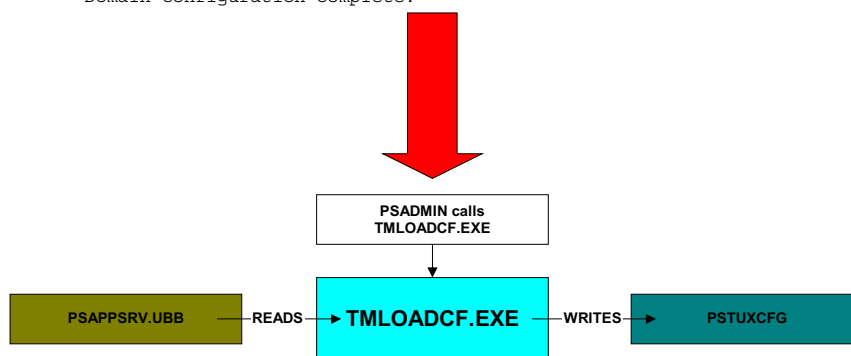
Loading UBBGEN configuration utility ...
```

---

## Loading a Configuration

After you configure a domain and PSADMIN creates the new configuration file, PSADMIN loads the new configuration settings into PSTUXCFG so that your domain can properly boot. This occurs automatically after you have completed all of the prompts for values in PSADMIN. For example, as shown in the following example, you will see the text Loading new configuration on the command line.

```
Setting Log Directory to the default... [PS_SERVDIR\LOGS]
Spawning enabled for server PSAPPSRV.
Configuration file successfully created.
CFG setting changes completed, loading configuration...
Domain configuration complete.
```



### Loading a New Configuration

To load the new configuration, PSADMIN makes a call to the BEA executable, TMLOADCF.EXE, which populates the PSTUXCFG file. TMLOADCF.EXE reads the newly entered values that appear in the PSAPPSRV.UBB file and writes them to the PSTXCFG file.

## Booting a Domain

When you select **Boot this domain** PSADMIN calls the Tuxedo executable called TMBOOT.EXE, which used the information that resides in both the PSAPPSRV.ENV and PSTUXCFG file to boot the appropriate domain.

The following example depicts this interaction.

```

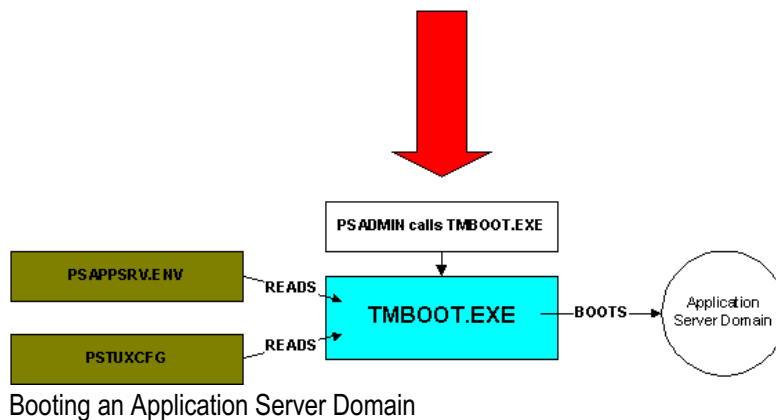
PeopleSoft Domain Administration

Domain Name: PSXXDEMO

1) Boot this domain
2) Domain shutdown menu
3) Domain status menu
4) Configure this domain
5) TUXEDO command line (tmadmin)
6) Edit configuration/log files menu
7) Messaging Server Administration menu
q) Quit

Command to execute (1-7, q) :1

```



## Stopping a Domain

When you select **Domain shutdown menu** and choose one of the shutdown options PSADMIN calls the Tuxedo executable called TMBOOT.EXE, which also uses the information that resides in both the PSAPPSRV.ENV and PSTUXCFG files to shutdown the appropriate domain.

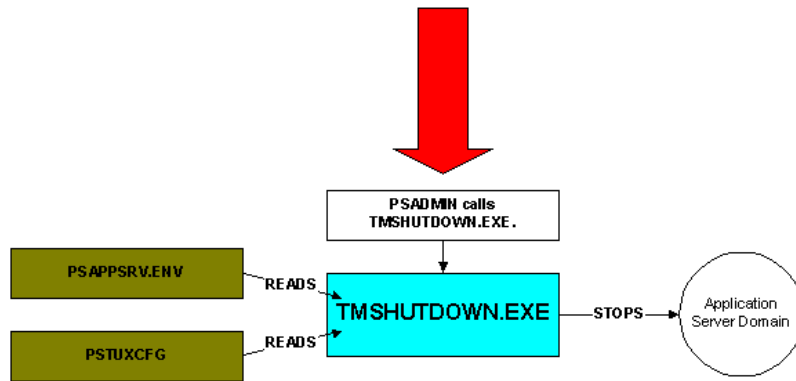
The following example depicts this interaction.

-----  
PeopleSoft Domain Administration  
-----

Domain Name: PSXXDEMO

- 1) Boot this domain
- 2) **Domain shutdown menu**
- 3) Domain status menu
- 4) Configure this domain
- 5) TUXEDO command line (tmadmin)
- 6) Edit configuration/log files menu
- 7) Messaging Server Administration menu
- q) Quit

Command to execute (1-7, q) :2



Stopping an Application Server Domain

## CHAPTER 3

# Application Server

The Application Server Administration menu offers by far the most extensive set of menu options. This is due to the fact that configuring an application server domain is more complicated than setting up a Process Scheduler Server for instance.

After reading this chapter, you will be familiar with the menus associated with configuring and administering an application server domain. The following chapter contains parameter-by-parameter documentation for all of the configuration sections you encounter while setting up an application server domain.

## Accessing the Application Server Options

To access the menu options available for configuring and administering your application server select *1* from the PeopleSoft Server Administration menu.

```

PeopleSoft Server Administration

```

- 1) Application Server
- 2) Process Scheduler
- 3) Web Components
- 4) Service Setup
  
- q) Quit

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

The **PeopleSoft Application Server Administration** menu appears.

```

PeopleSoft Application Server Administration

```

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

Command to execute (1-3, q) :

The menu options and parameters within the **Create a domain** and **Delete a domain** menus are straightforward, one-time tasks (per domain that is). On the other hand, the **Administer a domain** menu offers a multitude of very involved configuration, administration, and logging parameters.

## Administering a Domain

In order to administer a domain, you must have already created one to administer. After you have created a domain, you will need to specify environment-specific settings in order for the application server to function correctly with your system. The following sections describe all the menus and options within them that you will encounter as you administer and configure an application server domain.

To administer a domain

1. Select *1* from the PeopleSoft **Application Server Administration** menu.

```

PeopleSoft Application Server Administration

```

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

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

2. In the **Select domain number to administer** command line, enter the number that corresponds to the previously created domain you want to administer that appears in the

**Tuxedo domain list:**

Tuxedo domain list:

1) ps800dmo

Select domain number to administer: 1

The **PeopleSoft Domain Administration** menu appears.

3. Select the option that you want to perform from the **PeopleSoft Domain Administration** menu.

```

PeopleSoft Domain Administration

Domain Name: ps800dmo

1) Boot this domain
2) Domain shutdown menu
3) Domain status menu
4) Configure this domain
5) TUXEDO command line (tmadmin)
6) Edit configuration/log files menu
7) Messaging Server Administration menu

q) Quit
```

Command to execute (1-6, q) :

The following sections describe each option that appears in the **PeopleSoft Domain Administration** menu.

---

## Booting a Domain

The **Boot this Domain** option sets the following environment variable:

```
TUXCONFIG=PS_HOME/appserv/<domain-name>/pstuxcfg
```

And, then it boots the Tuxedo domain (the application server) using the **tmboot** command.

---

## Shutting Down a Domain

The **PeopleSoft Domain Shutdown Menu** offers two options: a *normal* shutdown and a *forced* shutdown.

```

PeopleSoft Domain Shutdown Menu

Domain Name: ps800dmo

1) Normal shutdown
2) Forced shutdown
q) Quit

Command to execute (1-2, q) [q]:

```

### Normal shutdown

Sets environment variable TUXCONFIG=PS\_HOME/appserv/<domain-name>/pstuxcfg, then shuts down the domain using the **tmshutdown** command. A normal shutdown is a quiesced shutdown that waits for users to complete their tasks and turns away new requests before terminating the BBL.

### Forced shutdown

Sets environment variable TUXCONFIG=PS\_HOME/appserv/<domain-name>/pstuxcfg, then shuts down the domain using the **tmshutdown -k TERM -c** command. A forced shutdown is a non-quiesced shutdown that *immediately* terminates the BBL process. Normally, you would only use the Forced shutdown when a BBL process is encountering errors and can not be shut down using a Normal shutdown.

---

## Checking Domain Status

The **PeopleSoft Domain Status Menu** enables you to view the status of the server, queues, or any clients connected through the domain.

```

PeopleSoft Domain Status Menu

```



-----

Domain Name: ps800dmo

- 1) Server status
- 2) Client status
- 3) Queue status
- q) Quit

Command to execute (1-2, q) [q]:

## Server status

By entering *1* for **Server status**, you invoke Tuxedo's tadmin **psr** subcommand (print screen), which displays the Tuxedo processes and server processes currently running.

| > Prog Name  | Queue Name  | Grp Name | ID  | RqDone | Load Done     | Current Service |
|--------------|-------------|----------|-----|--------|---------------|-----------------|
| -----        | -----       | -----    | --  | -----  | -----         | -----           |
| JSL.exe      | 00095.00200 | JSLGRP   | 200 | 0      | 0 ( IDLE )    |                 |
| PSSUBDSP.exe | 00098.00300 | PUBSUB   | 300 | 0      | 0 ( IDLE )    |                 |
| PSPUBDSP.exe | 00098.00200 | PUBSUB   | 200 | 0      | 0 ( IDLE )    |                 |
| PSBRKDSP.exe | 00098.00100 | PUBSUB   | 100 | 0      | 0 ( IDLE )    |                 |
| PSSAMSRV.exe | SAMQ        | APPSRV   | 100 | 0      | 0 ( IDLE )    |                 |
| BBL.exe      | 33969       | GSAWYER+ | 0   | 0      | 0 ( IDLE )    |                 |
| PSSUBHND.exe | SUBHQ_dfl't | PUBSUB   | 301 | 0      | 0 ( IDLE )    |                 |
| PSPUBHND.exe | PUBHQ_dfl't | PUBSUB   | 201 | 0      | 0 ( IDLE )    |                 |
| PSBRKHND.exe | BRKHQ_dfl't | PUBSUB   | 101 | 0      | 0 ( IDLE )    |                 |
| PSAPPSRV.exe | APPQ        | APPSRV   | 1   | 0      | 0 ( IDLE )    |                 |
| PSAPPSRV.exe | APPQ        | APPSRV   | 2   | 147    | 7350 ( IDLE ) |                 |
| PSAPPSRV.exe | APPQ        | APPSRV   | 3   | 4      | 200 ( IDLE )  |                 |
| WSL.exe      | 00001.00020 | BASE     | 20  | 0      | 0 ( IDLE )    |                 |
| JREPSVR.exe  | 00094.00250 | JREPGRP  | 250 | 0      | 0 ( IDLE )    |                 |

The number of items appearing in the Prog Name list depend on the number of server processes you have configured.

## Client status

By entering 2 for **Client status**, you invoke Tuxedo's **tmadmin pelt** subcommand (print client processes), which displays connected users.

```
> LMID User Name Client Name Time Status Bgn/Cmnt/Abrt

GSAWYER061199 NT WSH 0:10:06 IDLE 0/0/0
GSAWYER061199 NT JSH 0:10:04 IDLE 0/0/0
GSAWYER061199 NT JSH 0:10:03 IDLE 0/0/0
GSAWYER061199 NT JSH 0:10:03 IDLE 0/0/0
GSAWYER061199 NT JSH 0:10:02 IDLE 0/0/0
GSAWYER061199 NT JSH 0:10:02 IDLE 0/0/0
GSAWYER061199 PTDMO GSAWYER061199 0:05:45 IDLE/W 0/0/0
GSAWYER061199 PTDMO GSAWYER061199 0:06:25 IDLE/W 0/0/0
GSAWYER061199 NT tmadmin 0:00:00 IDLE 0/0/0
```

## Queue Status

Examining the status of the individual queues for each server process provides valuable tuning information. You check the queues using the Queue status option. Notice in the following example that the results of the Queue status option show the individual server processes, the associated queue, the number of server processes currently running, as well as the amount of queued requests.

```
Prog Name Queue Name # Serve Wk Queued # Queued Ave. Len Machine

BBL.exe 33835 1 - 0 - GSAWYER06+
PSMSGDSP.exe 00098.00100 1 - 0 - GSAWYER06+
JSL.exe 00095.00200 1 - 0 - GSAWYER06+
PSAPPSRV.exe APPQ 2 - 0 - GSAWYER06+
PSSAMSRV.exe SAMQ 1 - 0 - GSAWYER06+
WSL.exe 00001.00020 1 - 0 - GSAWYER06+
```

|              |             |   |   |   |              |
|--------------|-------------|---|---|---|--------------|
| PSMSGHND.exe | MBHQ        | 1 | - | 0 | - GSAWYER06+ |
| PSQCKSRV.exe | QCKQ        | 1 | - | 0 | - GSAWYER06+ |
| PSQRYSRV.exe | QRYQ        | 1 | - | 0 | - GSAWYER06+ |
| JREPSVR.exe  | 00094.00250 | 1 | - | 0 | - GSAWYER06+ |

The results alert you to any bottlenecks that may be occurring on your application server. With this information, you can make more informed performance decisions. For instance, if the bottlenecks appear to be persistent, it may indicate that you need to add more instances of a particular server process, such as PSAPPSRV for example. Or the results may indicate that you need to start either a PSQCKSRV or a PSQRYSRV.

---

## Configuring a Domain

The **Configure this domain** option sets the following environment variable:

```
TUXCONFIG=PS_HOME/appserv/<domain-name>/pstuxcfg
```

It also prompts users with a model configuration file to gather such parameters as port numbers, the number of various server processes desired, encryption enabling, and so forth. PSADMIN then invokes a PeopleSoft developed sub-program named UBBGEN, which takes the configuration parameters and builds the file, /PS\_HOME/appserv/<domain-name>/psappsrv.ubb, and executes the **tmloadcf -y psappsrv.ubb** command to generate the following binary file:

```
PS_HOME/appserv/<domain-name>/pstuxcfg
```

The following topics describe all of the parameters you will encounter while configuring an application server. We recommend that you either read this section before you fine tune the configuration of your application server or have it close by as you are doing so.

To configure a domain

1. Select option 4 from the PeopleSoft Domain Administration menu.

```

PeopleSoft Domain Administration

Domain Name: ps800dmo

1) Boot this domain
2) Domain shutdown menu
3) Domain status menu
4) Configure this domain
```

- 5) TUXEDO command line (tmadmin)
- 6) Edit configuration/log files menu
- 7) Messaging Server Administration menu
- q) Quit

Command to execute (1-6, q) : 4

If your domain is running you see the following prompt:

This option will shutdown the domain.

Do you want to continue? (y/n) [n] :y

You should see the following:

Trying to shutdown any servers, ignore errors...

Removing any existing configuration...

Generating new configuration...

Loading validation table...

2. When prompted with Do you want to change any config values, enter *y* for *Yes*.

Do you want to change any config values (y/n)? [n] : y

By entering *y*, you are prompted with a collection of configuration values in a number of sections, such as Startup and Domain Settings, but you will only encounter one configuration section at a time. For instance, the first configuration section is the Startup section. As shown in the following example, you see the configuration parameters contained in the section. If you need to change any of the values, just enter *y* on the **Do you want to change any values** line.

If you don't need to change any of the values, enter *n*. By doing so you create a new configuration file with the same values previously specified. You will want to enter *n*, or elect not to modify your PSADMIN parameters, in the following situations.

- You only have changed the location of TUXDIR.
- If you would rather edit the PSAPPSRV.CFG configuration file manually.
- You've installed a new Tuxedo patch.



**Note.** All of the parameters and configuration sections related to setting up an application server domain are documented in the following chapter.

---




---

For more information on each domain configuration parameter, see Domain Parameter Reference.

---



---

## The Edit configuration/log files Menu

From this menu you have the option to view the application server and Tuxedo log files. You can also manually edit the PSAPPSRV.CFG file if you do not want to use the PSADMIN interface.

```

PeopleSoft Edit Configuration/Log Files Menu

1) Edit psappsrv.cfg (current configuration file)
2) Edit APPSRV.LOG (current application server log file)
3) Edit TUXLOG (current Tuxedo log file)
4) Edit PSAPPSRV.tracesql (PSAPPSRV SQL trace file)
5) Edit PSSAMSRV.tracesql (PSSAMSRV SQL trace file)
q) Quit
```

Command to execute (1-7, q) [q]:

For PSADMIN to launch your text editor, such as Notepad or KEDIT, so that you can manually edit or view application server configuration and log files, you need to have your text editor specified in the Environment settings. For example, if you plan to use KEDIT, your editor environment setting should like the following:

```
set EDITOR=c:\apps\kedit\keditw32.exe
```

and if using Notepad:

```
set EDITOR=c:\Windows\Notepad.exe
```




---

You can view and edit a domain's PSAPPSRV.CFG file while the domain is up and running, but keep in mind that the changes you specify will not take effect until the next time you reconfigure the domain.

---

For the following options, you need to enter your Operator ID to view and edit the files.

- 4) Edit PSAPPSRV.tracesql (PSAPPSRV SQL trace file)
- 5) Edit PSSAMSRV.tracesql (PSSAMSRV SQL trace file)

For example

Command to execute (1-7, q) [q]: 5

Enter the operator ID : PTXYZ



PeopleSoft secures the SQL traces because in some instances the SQL traced may involve sensitive information.

## Edit PSAPPSRV.CFG

The PSAPPSRV.CFG file contains all of the configuration settings for an application server domain. The PSADMIN interface provides prompts so that you can edit and modify this file within a structured format. In many cases and perhaps due to personal preference, you may opt to edit the PSAPPSRV.CFG file manually. When editing this configuration file manually, you will see that it is similar to editing an INI file in that all of the parameters are grouped in sections.

```
[JOLT Listener]
;=====
; Settings for JOLT Listener
;=====
;Address Note: Can be either Machine Name or IP address.
;Address Note: %PS_MACH% will be replaced with THIS machine's name
Address=%PS_MACH%
Port=9000
Encryption=0
Min Handlers=1
Max Handlers=3
Max Clients per Handler=40
Client Cleanup Timeout=60
Init Timeout=5
Client Connection Mode=ANY
Jolt Compression Threshold=9999999

[JOLT Relay Adapter]
;=====
; Settings for JOLT Relay Adapter (JRAD)
;=====
;Listener Address Note: Can be either Machine Name or IP address.
;Listener Address Note: %PS_MACH% will be replaced with THIS machine's name
Listener Address=%PS_MACH%
Listener Port=9100

[Domain Settings]
;=====
; General settings for this Application Server.
```

PSAPPSRV.CFG in a text editor

## Edit APPSRV.LOG

Some of the things you might find in the APPSRV.LOG are a mismatch between the versions of PeopleTools on a Windows Workstation and application server or invalid passwords being used.

## Edit TUXLOG

The TUXLOG allows you to trace the Tuxedo component for troubleshooting information.

## Edit PSAPPSRV.tracesql

You can specifically trace the activity of the PSAPPSRV server process by setting the PSAPPSRV.tracesql.

## Edit PSSAMSRV.tracesql

You can specifically trace the activity of the PSSAMSRV server process by setting the PSSAMSRV.tracesql.

## Creating a Domain

The **Create a domain** option creates a subdirectory underneath /PS\_HOME/appserv using the domain name specified by the user and copies “model” files to that directory.

To create an application server domain

1. Select 2 from the PeopleSoft Application Server Administration menu.

```

PeopleSoft Application Server Administration

```

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

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

2. Enter the name of the domain that you want to create; the name must not exceed 8 characters.

Please enter name of domain to create :ps800dmo

### 3. Select a configuration template from the Configuration templates: list.

Configuration templates:

- 1) developer
- 2) large
- 3) medium
- 4) small

Select config template number:

The configuration templates are pre-configured sets of application server processes.



For more information on the configuration templates, see [Creating a Domain](#).

---

After you have selected the appropriate configuration template and pressed ENTER, you should see the following on the screen.

Creating domain...

Copying application server configuration files...

Copying Jolt repository file...

Domain created.



**Note.** If you are responsible for routinely creating many domains, you may want to consider either modifying the CFX files to reflect your environment or creating your own. You can manually edit any CFX file in the PS\_HOME\appserv directory with any text editor, such as Notepad. To create your own CFX files, just save the CFX file to a new name after modifying the template values. The next time PSADMIN prompts you for a configuration template to create a domain, your custom CFX file will appear in the configuration templates list.

---

## Deleting a Domain

The **Delete a domain** option shuts down the domain, if running, then deletes the domain's subdirectory.





---

For best results, before you delete a domain, make sure that it is not running.

---

To delete a domain

1. Select 3 from the PeopleSoft Application Server Administration menu.

```

PeopleSoft Application Server Administration

```

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

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

2. From the **Tuxedo domain list**: select the number that corresponds to the domain you want to delete

```
Tuxedo domain list:
```

- ```
1) ps800dmo
2) hr80ora
```

```
Select domain number to delete: 2
```

3. When prompted to continue, enter *y* and press ENTER.

```
This operation will delete the configuration files for this domain!
```

```
Warning: If this domain is still running,
```

```
you will not be able to shut it down cleanly!
```

```
Do you want to continue? (y/n) [n] :
```

You should see the following on the screen:

```
Attempting to delete domain...
```

```
Domain deleted.
```

## CHAPTER 4

# Domain Parameter Reference

This section focuses on all of the configuration options related to an application server domain. In the following topics you will find documentation pertaining to each of the configuration sections and the parameters they contain.

For the most part, the documentation reflects the order that the configuration sections appear in the PSADMIN interface or the PSAPPSRV.CFG configuration file. The exceptions to this order are the configuration options related to the PeopleSoft messaging architecture. Since the messaging architecture is somewhat separate from the basic application server configuration, the configuration sections for the messaging server processes appears in the Application Messaging chapter of this book.



**Note.** The application server dynamically scales server processes according to the volume of transaction requests. In PeopleSoft terminology, we call this "spawning" server processes. There is no explicit parameter you need to set in order to enable spawning. In the following configuration section descriptions, you will notice that some server processes allow you to specify a Min (minimum) and Max (maximum) number of server processes. To enable spawning, the Max value must exceed the Min value by at least one increment. As needed the application server spawns server processes up to the Max value. By setting the Max value greater than that of the Min value, you implicitly enable spawning.



For more information on configuring the server processes related to Application Messaging, see Messaging Server Administration.

## Startup

The first configuration section that you will see is the Startup section. This is where you set your database signon values.

Values for config section - Startup

DBName=

DBType=

OprId=

OprPswd=  
  
ConnectId=  
  
ConnectPswd=  
  
ServerName=

Do you want to change any values (y/n)? [n]:

The following topics explain what each parameter within the Startup section controls.

### **DBName**

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

### **DBType**

Enter the PeopleSoft database type, such as DB2, DB2ODBC, DB2UNIX, INFORMIX, MICROSOFT, ORACLE, or SYBASE. If you enter an invalid database type, PSADMIN will prompt you with a valid list. This value must be UPPERCASE.

### **OprID**

OprID refers to the PeopleSoft User ID that is authorized to start the application server. You use Maintain Security to add this property to a permission list that you apply to a user profile by way of an role. In order for the application server to boot, the appropriate user ID with the correct authorizations needs to be entered. Use UPPERCASE.

### **OprPswd**

Enter the password used by the specified User ID/OprID that will gain access to the database. The value you enter must be specified in UPPERCASE.. The reason for requiring that User ID and Password are specified in UPPERCASE is to simplify administration of the system.

### **Connect ID**

Required for all database platforms. This is a database level ID that is used by PeopleSoft to do the initial connection to the database. This username must have authority to select from PSLOCK, PSOPRDEFN, and PS.PSDBOWNER.

### **Connect Password**

The Connect ID's password. For instance, this would be the UNIX name's password (either upper or lower case).

### **ServerName**

Required for Sybase and Informix (ignored by any other RDBMS). This is the name of the server in which the PeopleSoft database is installed. This value is case sensitive.

## Database Options

The Database Options section allows you to specify certain environment variables that may improve the performance of your system. These options only apply to every RDBMS.

Values for config section - Database Options

SybasePacketSize=

UseLocalOracleDB=0

EnableDBMonitoring=0

Do you want to change any values (y/n)? [n]:

### SybasePackeSize

This option allows you to specify a TCP Packet Size. The minimum value is 512 and the maximum value is 65538. The default packet size will be 512. If you make any changes to the packet size, make sure that you make the corresponding changes to the Sybase server.



For more information on Sybase TCP packet sizes, refer to your Sybase reference manuals.

---

### UseLocalOracleDB

The UseLocalOracleDB parameter allows for a batch program to initiate a “local” connection to a PeopleSoft database running on the same machine. We recommend that this be used for all Process Scheduler (batch) and application server configurations that are local (on the same server) to the PeopleSoft Oracle instance. Our internal testing reveals that this type of connection allows batch processes to complete significantly quicker.



**Note.** This option only applies to Oracle customers.

---

### EnableDBMonitoring

This option allows you to view more information regarding the clients connected to a database server through application server. For instance, with this enabled, you can view the client machine name or User ID associated with a particular connection. Without this option enabled, all connections appear somewhat anonymously, as in PSFT or APPSERV.

Suppose a particular user keeps submitting a poorly crafted query that hampers the overall application server performance. Using this feature you can associate the query to particular machine and then take the “proper” steps to avoid any future problems.

To enable this option, enter ‘1’.



**Note.** The behavior/display that this option controls differs slightly per platform. Also, be aware that this feature is not supported on all platforms.

---

## Security

The Security section allows you to set an additional layer to the signon process.

```
Values for config section - Security
```

```
Validate Signon with Database=1
```

```
Do you want to change any values (y/n)? [n]:
```

The following topic describes how you would use this parameter.

### Validate Signon with Database

This option allows for an additional level of authorization checking performed at the database level. To enable this option, enter *1*, and to disable it enter *0*.

With this option *disabled* and a PeopleSoft user attempts to connect to an application server, the application server ensures that the user's PeopleSoft User ID and password exist on PSOPRDEFN. If it does not exist, the request to connect fails. This is PeopleTools-level authentication.

With the Validate Signon with Database option *enabled*, the application server first attempts to connect to the database using the User ID and password as part of the database connection string. If the authorization is successful it disconnects, and then the normal PeopleSoft Operator signon procedure is attempted.

This means that when you have this option enabled in order to successfully connect to the database the user must be defined on either the operating system or the RDBMS (depending on which RDBMS you're using) as well as within PeopleSoft.



**Note.** For DB2 for OS/390 (MVS) the User ID and password must be defined as MVS user logon IDs.

---



For more information on the PeopleSoft signon options, refer to Understanding PeopleSoft Signon.

---

## Workstation Listener

The Workstation Listener is the component to which Windows workstations send Tuxedo messages.

Values for config section - Workstation Listener

Address=%PS\_MACH%

Port=7000

Encryption=0

Min Handlers=1

Max Handlers=3

Max Clients per Handler=40

Client Cleanup Timeout=60

Init Timeout=5

Tuxedo Compression Threshold=5000

Do you want to change any values (y/n)? [n]:

The following sections explain the parameters within the Workstation Listener configuration section.

### Address

%PS\_MACH% resolves automatically to the machine name that PSADMIN obtains by using a system API call. You can also specify the machine's IP Address (dotted notation) or its resolvable name (DNS Name).

PeopleSoft suggests that you do not change this value except in the following rare cases. If you are configuring files to run an application server on another machine—you plan to copy PSAPPSRV.CFG and PSAPPSRV.UBB to a domain on another machine—you will need to overlay %PS\_MACH% with the other machine's name.

### Port

Enter the four digit port number that you will assign to the WSL. Port numbers are arbitrary numbers between 1000 and 16 K and must not already be in use by another service.

For example:

7000

## Encryption

Causes encryption (scrambling) of data messages between client workstations and the application server. You have the following options:

- 0=no encryption
- 128=128-bit encryption

## Min Handlers

Number of Workstation Handlers (WSH) started at boot time. Each WSH allows up to around 60 client connections. Numbers vary depending upon the resources of the server. In most cases, you will want to decrease this default as opposed to increasing it. The PeopleSoft default is 40.

## Max Handlers

Maximum number of WSHs that can be started for a domain. If Min Handlers = Max Handlers, this means that Tuxedo does *not* automatically spawn incremental WSHs.

## Max Clients per Handler

Specifies the maximum number of client workstation connections each WSH can manage.

## Client Cleanup Timeout

Specifies the amount of time, in minutes, that a client connection can remain idle (no work requested) before Tuxedo will terminate a client connection. Client disconnects are transparent to a client, and a user just needs to click the mouse to cause a reconnection.

## Init Timeout

This value, when multiplied by SCANUNIT (a UBB parameter value) specifies the amount of time, in seconds, Tuxedo will allow for a client connection request to bind to a WSH before terminating the connection attempt.

## Tuxedo Compression

Specifies the minimum length of a data message at which the application server initiates data compression. While compression results in favorable performance gains for transactions over a WAN, testing reveals that compression can degrade performance slightly over a LAN due to the compression/decompression overhead.

PeopleSoft recommends using the default threshold of 5000, which sets a balance between WAN and LAN environments. This means that only network request and response messages over 5000 bytes will be compressed, and those 5000 and under will be uncompressed. Customers supporting both WAN and LAN users may configure a hybrid environment by configuring two application servers: one to support WAN users (with compression set to 100) and another to support LAN users (with compression set to 100000, effectively turning compression off).



## JOLT Listener

After you've entered the values for the Workstation Listener section, the next configuration section you will see is the JOLT Listener section.

You'll need to set values for this section to enable PIA connections.

Values for config section - JOLT Listener

Address=%PS\_MACH%

Port=9000

Encryption=0

Min Handlers=1

Max Handlers=3

Max Clients per Handler=40

Client Cleanup Timeout=60

Init Timeout=5

Client Connection Mode=ANY

Jolt Compression Threshold=1000000

Do you want to change any values (y/n)? [n]:

### Address

See the equivalent parameter for Workstation Listener.

### Port

Enter the port number used for the JSL (Jolt Server Listener). This value can be any valid port number not already in use by another service. Note that the port number will not be used unless you answer *Yes* to the prompt that asks whether you want to start Jolt.

### Encryption

See the equivalent parameter for Workstation Listener.

### Min Handlers

Number of Jolt Server Handlers (JSH) started at boot time. Each JSH multiplexes up to 50 connections.

### Max Handlers

Specifies the maximum number of JSHs.



---

JSH Handlers spawn using successive port numbers starting at the port number set for the Jolt Server Listener in the PSAPPSRV.CFG file. Make sure that the additional ports are free before configuring spawning.

---

### **Max Clients per Handler**

Specifies the maximum number of client connections each JSH can manage.

### **Client Cleanup Timeout**

See equivalent parameter in the Workstation Listener section.

### **Init Timeout**

See equivalent parameter in the Workstation Listener section.

### **Client Connection Mode**

For this parameter you have the following options: RETAINED, RECONNECT, or ANY. This parameter controls the allowed connection modes from clients.

- RETAINED means the network connection is retained for the full duration of a session.
- RECONNECT means the client establishes and brings down a connection when an idle timeout is reached and reconnects for multiple requests within a session.
- ANY, the default, means the server allows client code to request either a RETAINED or RECONNECT type of connection for a session. Whereas, with the other two options the server dictates from which type of client it will accept a connection. This option translates to the -c Connection Mode parameter for the JSL section in the PSAPPSRV.UBB file.

### **Jolt Compression Threshold**

Jolt Compression can significantly improve performance. Jolt Compression allows messages transmitted through a Jolt connection to be compressed as they flow over the network. This has always been an option for the Windows Client, and now the equivalent technology is available for the Jolt requests. You are likely to see the most significant performance improvements over a WAN.

For compression, the configuration files contain a default compression threshold. This default value should give the best results for most situations. However, your application server administrator is welcome to adjust this value to suit your implementation.

The compression threshold indicates to the server how large a packet must be to require compressing. In other words, the value you set is the minimum number of bytes a single packet must be before the server will compress it.

Many of the XML messages being sent around your system are greater than 100,000 bytes. These messages contain HTML in compressed states so it's generally not required that these messages be compressed. Because of this, the PeopleSoft default is set to 1,000,000 bytes.

Be careful when adjusting compression settings. If you set the threshold too high, then no packets will be large enough to be compressed. If you set the threshold too low, you may greatly reduce network traffic, but be aware that the server will have an increased workload comprised of compressing numerous packets. Typically, PeopleSoft recommends decreasing the threshold according to the bandwidth of your workstation hardware as described in the following paragraphs.

If you are only handling LAN connections, you may want to disable compression by setting the threshold to 99999999. With the threshold set at such a value, only packets larger than 99,999,999 bytes will get compressed. Of course, such a large value effectively disables compression so that no packets get compressed. This means no extra work for the server compressing packets.

On the other hand, if you have mostly low bandwidth, as in 56Kb modem connections over a WAN, then you would most likely want to compress the packets as much as possible. When decreasing the compression threshold, keep in mind that the law of diminishing returns applies. Setting the threshold much below 1000 puts an increasing load on the server, and this can nullify any performance increases you may have gained from reduced network traffic.

### Additional Prompt

After you have finished *all* of the configuration sections PSADMIN prompts you with the following:

```
Do you want JOLT configured (y/n)? [n]:
```

If you have made changes to the JOLT Listener section, and you want JOLT configured to deploy applications to a browser, enter *y* for *Yes*. Essentially, if you are using PIA, you must have Jolt configured.

## JOLT Relay Adapter



Jolt Relay Adapter is primarily intended for use with the PeopleSoft Web Client, a product released with PeopleTools 7.0x and 7.5x. It is possible to use Jolt Relay Adapter with PIA, but performance may become an issue.

You may only need to configure this section if you are deploying the Web Client, *and* you plan to have your web server and application server reside separate machines. If your web server and application server reside on the same machine, then ignore this section.

```
Values for config section - JOLT Relay Adapter
```

```
Listener Address=%PS_MACH%
```

```
Listener Port=9100
```

```
Do you want to change any values (y/n)? [n]:
```

After you have finished *all* of the configuration sections PSADMIN prompts you with the following:

```
Do you want JRAD configured (y/n)? [n]:
```

If you have made changes to the JOLT Relay Adapter section, and you want JRAD configured so that you can deploy the Web Client from a web server running on a separate machine from the application server, enter *y* for *Yes*.

The following sections explain the settings for the JRAD offered through PSADMIN.



For more information on setting up Jolt Relay and Jolt Relay Adapter and when it's appropriate to use it, see Administration Considerations.

### Listener Address

Defaults to %PS\_MACH%. Specifies the machine on which the application server is running. See the equivalent parameter in the Workstation Listener section.

### Listener Port

This option is for advanced configurations requiring Jolt Internet Relay (JRLY). The Listener Port listens for JRLY requests and must match the JRLY “OUT” port setting in the JRLY configuration file of the sending machine. The port number, as in 9100, will not be used unless you answer “y” to the prompt asking if you want JRAD configured.

## Domain Settings

After the Jolt Relay Adapter section, comes the Domain Settings configuration section. This section allows you to specify general settings for the entire domain—not just a specific component of it.

```
Values for config section - Domain Settings
```

```
Domain ID=PT8
```

```
Add to PATH=C:\Apps\Db\Mssql7\Binn
```

```
Spawn Threshold=1,600:1,1
```

```
Restartable=Y
```

```
LogFence=3
```

```
Enable Debugging=0
```

```
Trace-Log File Character Set=ANSI
```

```
Do you want to change any values (y/n)? [n]:
```

## Domain ID

Enter the name of your application server domain. It does not need to match the name you specified when you created a domain. This name is important only in that the Tuxedo Web Monitor uses it to identify application server domains on each machine. It should not exceed eight characters. PeopleSoft suggests using the database name in lowercase.

## Add to PATH

The directory that contains your database connectivity software, as in `/apps/db/oracle/bin`, must be specified in the PATH. If the database connectivity directory is *not* already specified in the PATH, you can set it by specifying this parameter. The value will be added to the PATH.

On Windows NT, if you don't enter a value, it defaults to the current PATH.

On UNIX, if you don't enter a value, it defaults to the current directory—not the current PATH. So that it defaults to the current PATH, enter a `“.”` (a period without quotes).

## Spawn Threshold

Parameters supplied to BEA TUXEDO for control of process spawning using the `–p` command line option for all server processes. The default settings rarely need to be changed.

This allows the dynamic decay of spawned server processes as the transaction volume decreases. The value can be loosely translated to mean if in 600 seconds there is less than or equal to one job in the queue the decay process begins.



For more information, please refer to `servopts(s)` in the reference manual of the BEA TUXEDO online documentation.

---

## Restartable

For this parameter enter either a *y* or an *n*. A *y* indicates that Tuxedo can restart server processes (except the BBL process) if the server dies abnormally, as in a kill on UNIX or through Task Manager on Windows NT.

## LogFence

Sets a desired level of network tracing ranging from `-100` (suppressing) to `5` (all). The default is `3`. The following list reveals the choices available for LogFence.

- `-100`     - Suppress logging
- `-1`       - Protocol, memory errors
- `0`        - Status information
- `1`        - General errors
- `2`        - Warnings
- `3`        - Tracing Level 1 (default)

- 4            - Tracing Level 2
- 5            - Tracing Level 3

The Trace file will be generated in the following directory:

```
<PS_HOME>\appserv\<domain>\LOGS\psappsrv.log.
```

## Enable Debugging



This only applies to the Development Environment.

To enable debugging, enter *1* (one), and to disable debugging, enter *0* (zero).

By turning this feature on, you allow the PeopleCode debugger to run when connected to the PeopleSoft database using an application server. Interactive debugging of PeopleCode for internet and three-tier configurations requires that you have the following in place:

- An application server domain set up on your *local* machine.
- A COM connection between the application designer and the application server.



For more information on setting up PIA/three-tier debugging, see Debugging PIA Applications: Set up the PeopleCode Debugger.

## Trace-Log File Character Set

Specify the character set of the machine to which you typically write and read the traces and log files. If the character sets are not matched, the file is unreadable.

## Trace

After you have set the Domain Settings the Trace section appears. This section allows you to specify the tracing options that you can enable on the application server to track the SQL and PeopleCode of your domain(s).

Values for config section - Trace

```
TraceSql=0
```

```
TraceSqlMask=12319
```

```
TracePC=0
```

```
TracePCMask=4095
```

```
TracePPR=0
```

```
TracePPRMask=4095
```

```
TraceAE=0
```

```
Write crash dump to separate file=Y
```

```
Do you want to change any values (y/n)? [n]:
```

## TraceSQL

Sets logging level for SQL tracing for all clients. Traces are written to the following location:  
`<PS_HOME>/appserv/<domain>/LOGS/<DomainOPRID>_<svrname>.tracesql.`

If you enter 0 it disables tracing; use 7 to enable a modest tracing level for debugging. For other levels of tracing, set to a value equal to the sum of the desired options. For example, if you only want to trace SQL, `TraceSQL=1`; if SQL statements and Connect statements are desired, `TraceSQL` should be set to `1 + 2 + 4 = 7`. A setting of `TraceSQL = 7` is recommended for troubleshooting connection and other basic problems. Tracing can consume large amounts of disk space over time so be sure to reset `TraceSQL = 0` when you've finished trouble shooting.

## TraceSQLMask

Sets logging level ceiling for SQL tracing for individual clients. Traces are written to the following location:  
`<PS_HOME>/appserv/<domain>/LOGS/<ClientOPRID>_<svrname>.tracesql.` Clients must specify desired SQL tracing level using PeopleSoft Configuration Manager on the Trace tab. To prevent clients from turning on the application server trace, and consuming resources, the application server uses `TraceSQLMask` as an administrative control facility.

If a client transmits a request to trace SQL, the application server compares the value transmitted to `TraceSQLMask`. If the client value is less than or equal to `TraceSQLMask`, the application server enables the trace. However, if the client value is greater, application server will enable the trace up to the `TraceSQLMask` value. Trace files are written on the application server; no trace shows up on the client workstation.

Trace values are set in the application server configuration file `PSAPPSRV.CFG` file. Output files are written to directory `$PS_HOME/appserver/winx86/<domain>/logs.`

For `TraceSQL` and `TraceSQLMask` you have the following options:

- 1            - SQL statements
- 2            - SQL statement variables
- 4            - SQL connect, disconnect, commit and rollback
- 8            - Row Fetch (indicates that it occurred, not data)
- 16           - All other API calls except ssb
- 32           - Set Select Buffers (identifies the attributes of columns  
to be selected).

- 64           - Database API specific calls
- 128          - COBOL statement timings
- 256          - Sybase Bind information
- 512          - Sybase Fetch information
- 4096         - Manager information
- 8192         - Message Agent information

## TracePC

Sets a desired level for PeopleCode tracing for activity generated by all clients on a domain. Eligible values will be defined in the configuration file. TracePC values are displayed in the Configuration Manager on the **Trace** tab. You can find the results in the following location: <PS\_HOME>/appserv/<domain>/LOGS/<domain>.log.

## TracePCMask

This parameter controls which of the PeopleCode Trace options requested by client machines will be written to the trace file. The results of this trace are written to

<PS\_HOME>/appserv/<domain>/LOGS/<ClientMachine>.<domain>.log

For TracePC and TracePCMask you have the following options.

- 1            - Trace instructions
- 2            - List the program
- 4            - Show assignments to variables
- 8            - Show fetched values
- 16           - Show stack
- 64           - Trace start of programs
- 128          - Trace external function calls
- 256          - Trace internal function calls
- 512          - Show parameter values
- 1024         - Show function return value
- 2048         - Trace each statement in program

## TraceAE

To trace your Application Engine programs, you can activate the specific Application Engine traces using this parameter. For this parameter you have the following options:

- 1            - Trace STEP execution sequence to AET file



|      |                                                          |
|------|----------------------------------------------------------|
| 2    | - Trace Application SQL statements to AET file           |
| 4    | - not yet allocated                                      |
| 8    | - not yet allocated                                      |
| 16   | - not yet allocated                                      |
| 32   | - not yet allocated                                      |
| 64   | - not yet allocated                                      |
| 128  | - Timings Report to AET file                             |
| 256  | - Method/BuiltIn detail instead of summary in report     |
| 512  | - not yet allocated                                      |
| 1024 | - Timings Report to tables (ignored if Prcs. Instance=0) |
| 2048 | - DB optimizer trace to file                             |
| 4096 | - DB optimizer trace to tables                           |




---

For more information on all of the Application Engine trace options, see [Tracing Application Engine Programs](#).

---

## TracePPR and TracePPRMask

You use these options to trace the activity in the panel/page processor. Typically, these options are only used internally by PeopleSoft developers, however, you may need to view the results of this trace when troubleshooting. For the values of each trace, view the comments in the PSAPPSRV.CFG file.

## Write crash dump to separate file

In the event that the application server shuts down abnormally, or crashes, it's useful to view the log information related to the crash. However, information related to a crash, or crash dump, can be lengthy. To enable you to avoid large, difficult-to-read log files, PeopleSoft provides the option of having the system write crash information to a file other than the appserv.log file.

The advantages of this option are:

- You don't have to sort through all of the other trace information in the log file. Typically, when a crash occurs you only want to see information related to the crash.
- You can reduce the size of your application server log files by routing crash information to a separate file.

To enable this option, indicate *Y*, for Yes, at the **Write crash dump to separate file** parameter. For example,

```
Write crash dump to separate file=Y
```

The system writes the crash dump file to the following location:

```
PS_HOME\appserv\\logs
```

And the system names the crash dump file according to the following convention:

```
dump.<server_process_name>.<process_ID>
```

If you elect to write crash information to a separate file, when a crash occurs the system indicates in the appserv.log that a crash occurred and that it is redirecting the crash information to a particular file. The following example illustrates what appears in the appserv.log in the event of a crash.

```
(0) Unhandled exception occurred. Writing crash dump to dump.PSAPPSRV.213

(3) Switching to new log file b:\appserv\test\logs\dump.PSAPPSRV.213
```

To disable this option, enter *N*, for No, at the prompt.



If you do not enable this option, crash information appears in the appserv.log by default.

---

## Cache Settings

After you set your Trace settings, you will be prompted whether you want to enable caching on the application server.

```
Values for config section - Cache Settings
```

```
EnableServerCaching=1
```

```
Set ServerCacheMode=0
```

```
Do you want to change any values (y/n)— [n]:
```

The following sections explain each cache option

### EnableServerCaching

To enable application server disk caching the value must be set to *1* or *2*, and to disable application server caching the value must be set to *0*.

By entering *1*, the system caches only the most used classes of objects, and if you enter *2*, the system caches all object types regardless of the frequency of use. Which option you choose depends on internal testing at your site.

In most cases there is no reason to disable server caching.

## Set ServerCacheMode

---



This parameter is reserved for future use.

---

## Remote Call

You have one option to set for Remote Call through the PSADMIN: RCCBL Redirect.

```
Values for config section - RemoteCall
```

```
RCCBL Redirect=0
```

```
Do you want to change any values (y/n)? [n]:
```

Enter *0* to disable redirection and enter *1* to enable redirection. Redirection causes the server process to retain intermediate work files used to pass parameter values between the server process and a RemoteCall/COBOL program for debugging purposes. Redirect should always be *0* except for debugging. Work files are written to the /LOGS directory with “.in” and “.out” extensions.

---



For more information on Remote Call see PeopleCode PeopleCode Developer's Guide.

---

## PSAPPSRV

The PSAPPSRV server process performs the functional requests, such as building and loading panel groups. It also provides the in-memory-caching feature for PeopleTools objects on the application server. Each server process maintains its own cache.

```
Values for config section - PSAPPSRV
```

```
Min Instances=1
```

```
Max Instances=2
```

```
Service Timeout=300
```

```
Recycle Count=100000
```

```
Allowed Consec Service Failures=2
```

```
Max Fetch Size=5000
```

```
Do you want to change any values (y/n)? [n]:
```

The following topics describe each of the parameters within the PSAPPSRV configuration section.

### **Min Instances**

Specifies how many servers will be started at boot. Translates to the PSAPPSRV server's -m (min) parameter in the UBB file.

### **Max Instances**

Specifies the maximum number of servers that can be started. Translates to the PSAPPSRV server's -M (Max) parameter in the UBB file.

### **Service Timeout**

Specifies the number of seconds a PSAPPSRV will wait for a service request, such as MgrGetObj or PprLoad to complete, before timing out. Service Timeouts are recorded in the TUXLOG and APPSRV.LOG. In the event of a timeout, PSSAPSRV will terminate itself and Tuxedo will automatically restart this process.

### **Recycle Count**

Specifies the number of times each server will be executed before being terminated (intentionally) by PeopleSoft and then immediately restarted. Servers must be intermittently recycled to clear buffer areas. The time required to recycle a server is negligible—occurring in milliseconds. Recycle Count does not translate into a native Tuxedo parameter in the PSAPPSRV.UBB file. Instead the value is stored in memory and is managed by a PeopleSoft server.

### **Allowed Consec Service Failures**

This option allows for dynamic server process restarts for service failures. To enable this option enter a number greater than zero, and to disable this option enter 0. The default for this parameter is 2. The numerical value you enter is the number of consecutive service failures that will cause a recycle of the server process. This is a catchall error handling routine that allows PSAPPSRV, PSQCKSRV, and PSAMSRV to terminate itself if it receives multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively, but if they do it indicates that the server process needs to be recycled or cleansed. A “Retry” message will appear on the client machine when this occurs.

### **Max Fetch Size**

Default is 5000 (K). Specifies the maximum memory used by the server to store fetched rows for a transaction before sending the result set back to a client. If the memory limit is exceeded, the client receives the rows retrieved with a “Memory Buffer Exceeded” warning. PeopleSoft recommends keeping the default value. PSAPPSRV supports non-conversational transactions, so this parameter gives users a way to balance high-volume throughput with the needs of users working with large volumes of data. A value of 0 means unlimited memory will be used.



The memory is not pre-allocated but is acquired as needed for each transaction.

---

## PSSAMSRV

The PSSAMSRV server process communicates through the Tuxedo conversational mode. It performs transactional SQL requests (Updates).

Values for config section - PSSAMSRV

Instances=1

Service Timeout=300

Recycle Count=100000

Allowed Consec Service Failures=2

Max Fetch Size=32

Do you want to change any values (y/n)? [n]:

The following topics describe each of the parameters within the PSSAMSRV configuration section.

### Instances

Specifies how many servers will be started at boot. Translates to the PSSAMSRV server's -m (min) parameter in the UBB file.

### Service Timeout

Specifies the number of seconds the server processes will wait for a request before timing out. Meant to Stop Runaway processes like rcbl timeout.

### Recycle Count

Specifies the number of times each server will be executed before being terminated (intentionally) by Tuxedo. Tuxedo will immediately restart the server. Servers must be intermittently recycled to clear buffer areas. The time required to recycle a server is negligible, occurring in milliseconds. Recycle Count does not translate into a native Tuxedo parameter in the PSAPPSRV.UBB file, rather the value is stored in memory and is managed by a PeopleSoft server.

### Allowed Consec Service Failures

This option allows for dynamic server process restarts for service failures. To enable this option enter a number greater than zero, and to disable this option enter 0. The default for this parameter is 2. The numerical value you enter is the number of consecutive service failures that will cause a recycle of the server process. This is a catchall error handling routine that allows PSAPPSRV, PSQCKSRV, and PSSAMSRV to terminate itself if it receives multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively, but if they do it indicates that the server process needs to be recycled or cleansed.

A "Retry" message will appear on the client machine when this occurs.

## Ignore Undefined Subscription Messages

Ignore Undefined Subscription Messages=1 sets the system to ignore all incoming (subscription) messages not defined in the database. Ignore Undefined Subscription Messages=0 sets the system to send an exception reply to the publishing node for all incoming (subscription) messages not defined in the database.

## Max Fetch Size

Default 32 (K). Specifies the maximum memory used by server to store fetched rows for a transaction before sending results to client and refilling the memory buffer. When the memory limit is reached, the server sends rows to the client, but then resumes refilling the buffer and sending results to client until the query is complete. PeopleSoft recommends that users keep the default value.

PSSAMSRV supports conversational transactions, so this parameter gives users a way to tune performance by adjusting the number of network round-trips required for the “average” transaction. A value of 0 causes unlimited memory to be used, which means one round-trip no matter how large the result set. Note that the memory is not pre-allocated but is acquired as needed.

## PSQCKSRV

The PSQCKSRV is an optional server process designed to improve performance. Essentially, the PSQCKSRV, or quick server, is a copy of the PSAPPSRV. It performs “quick” requests such as non-transactional (read-only) SQL requests. The PSQCKSRV is designed to improve overall performance by allowing the PSAPPSRV process to direct a portion of its workload to PSQCKSRV.

Values for config section - PSQCKSRV

Min Instances=1

Max Instances=2

Service Timeout=300

Recycle Count=100000

Allowed Consec Service Failures=2

Max Fetch Size=5000

Do you want to change any values (y/n)? [n]:

The following topics describe each of the parameters within the PSQCKSRV configuration section.

**Min Instances**

Specifies how many servers will be started at boot. Translates to the PSQCKSRV server's -m (min) parameter in the UBB file.

**Max Instances**

Specifies the maximum number of servers that can be started. Translates to the PSQCKSRV server's -M (Max) parameter in the UBB file.

**Service Timeout**

Specifies the number of seconds a PSQCKSRV will wait for a request before timing out. Meant to Stop Runaway processes, like rcbl timeout. Applies to incremental PSQCKSRV servers dynamically started by the Max Instances parameter.

**Recycle Count**

Uses PSAPPSRV's specifications.

**Allowed Consec Service Failures**

This option allows for dynamic server process restarts for service failures. To enable this option enter a number greater than zero, and to disable this option enter 0. The default for this parameter is 2. The numerical value you enter is the number of consecutive service failures that will cause a recycle of the server process. This is a catchall error handling routine that allows PSAPPSRV, PSQCKSRV, and PSAMSRV to terminate itself if it receives multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively, but if they do it indicates that the server process needs to be recycled or cleansed.

A "Retry" message will appear on the client machine when this occurs.

**Max Fetch Size**

Uses PSAPPSRV's specifications.

## PSQRYSRV

PSQRYSRV is designed, specifically, to handle the SQL generated by PeopleSoft Query (PSQED.EXE). With PSQRYSRV configured, SQL intensive, complicated, user-defined queries get offloaded to a dedicated server process thus freeing PSAPPSRV and PSQCKSRV to handle the SQL requests for which they are more suited.

PSQCKSRV is also designed to handle SQLRequest services, however, if you have PSQRYSRV configured, it handles all SQLRequests initiated specifically by PSQuery (SQLQuery:SQLRequest).

Like the PSQCKSRV server process, PSQRYSRV is an optional server process. However, if you allow users to initiate queries from PeopleSoft Query, we recommend that you take advantage of this new server process.

Values for config section - PSQRYSRV

```
Min Instances=1

Max Instances=2

Service Timeout=300

Recycle Count=100000

Allowed Consec Service Failures=2

Max Fetch Size=5000

Use dirty-read on DB2/OS390=0
```

Do you want to change any values (y/n)? [n]:

The following topics describe each of the parameters within the PSQRYSRV configuration section.

### **Min Instances**

Specifies how many servers will be started at boot. Translates to the PSQRYSRV server's -m (min) parameter in the UBB file.

### **Max Instances**

Specifies the maximum number of servers that can be started. Translates to the PSQRYSRV server's -M (Max) parameter in the UBB file.

### **Service Timeout**

Specifies the number of seconds PSQRYSRV will wait for a request before timing out. Meant to Stop Runaway processes.

### **Recycle Count**

Specifies the number of times each server will be executed before being terminated (intentionally) by PeopleSoft and then immediately restarted. Servers must be intermittently recycled to clear buffer areas. The time required to recycle a server is negligible—occurring in milliseconds.

If the recycle count is set to zero, PSQRYSRV will never be recycled.

### **Allowed Consec Service Failures**

This option allows for dynamic server process restarts for service failures. To enable this option, enter a number greater than zero, and to disable this option enter 0. The default for this parameter is 2. The numerical value you enter is the number of consecutive service failures that will cause a recycle of the server process.

This is a catchall error handling routine that allows PSAPPSRV, PSQCKSRV, PSQRYSRV, and PSSAMSRV to terminate itself if it receives multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively, but if they do, it indicates that the



server process needs to be recycled, or cleansed. A "Retry" message will appear on the client machine when this occurs.

If this is set to zero, PSQRYSRV will never be recycled.

### **Max Fetch Size**

Specifies the maximum size (in KB) of a result set returned from a SELECT query. The default is 10000KB. Use 0 for no limit.

### **Use dirty-read on DB2/OS390**

This option applies to DB2 for OS/390 only.

## **Messaging Server Processes**

There are a variety of server processes devoted to application messaging. If you are not implementing the application messaging technology then you may skip through the delivered, default server processes. These server processes are:

- PSBRKDSP
- PSBRKHND
- PSPUBDSP
- PSPUBHND
- PSSUBDSP
- PSSUBHND

The configuration of the messaging server processes is discussed later in this document.



For more information on configuring the messaging server processes, see Messaging Server Administration.

---

## **SMTP Settings**

You have the ability to send electronic mail requests—issued with Workflow or PeopleCode—to the application server, and the application server will, in turn, pass the requests to the specified mail server (SMTPServer). By having the application server submit the email request you avoid having to install mail connectivity software on each client just as you avoid having to install database connectivity software on each client in a three-tier connection. To specify the appropriate SMTPServer and port to receive the email requests you will need to edit the SMTP Settings section, as shown below.

Values for config section - SMTP Settings

```
SMTPServer=

SMTPPort=25

SMTPServer1=

SMTPPort1=0

SMTPSender=PeopleSoft@peoplesoft.com

SMTPSourceMachine=

SMTPCharacterSet=UTF-8

SMTPEncodingDLL=

SMTPGuaranteed=0
```

Do you want to change any values (y/n)? [n]:

The following topics describe each of the parameters within the SMTP configuration section.

### **SMTPServer**

Enter the host name and IP Address of the mail server machine.

### **SMTPPort**

Enter the port number on the mail server machine.

### **SMTPServer1**

Enter the host name and IP Address of the failover mail server machine in case the other specified server is down.

### **SMTPPort1**

Enter the port number on the failover mail server machine

### **SMTPSender**

Enter the sender's Internet address. This must be a valid address such as user1@xyzcorp.com.

### **SMTPSourceMachine**

Sender's source machine name and internet address in the form of MACHINE.XYZCORP.COM. This value is required in some but not all environments

### **SMTPCharacter Set**

Here, specify the character set used on the sender's machine.

**SMTPEncodingDLL**

Specifies the name of a DLL used to translate the mail message from the sender's character set, as in latin1, sjis, big5, gb, ks-c-5601-1987, ks-c-5601-1992, to a desired 7-bit safe character set for transmission.

**SMTPGuaranteed**

Set this option to 1 if you want TriggerBusinessEvent email PeopleCode to be delivered through the messaging system. With this option on, the system periodically retries email sent with TriggerBusinessEvent until successful.

By enabling this feature you implement a mechanism to ensure that emails get routed to the appropriate place just in case SMTP mail fails for some reason, such as network timeouts, down mail servers, invalid parameters, and so on.

**Interface Driver**

You set the following parameter for configuring your Interface Driver.

Values for config section - Interface Driver

```
SCP_LOCALE="LOCALE=EN_US", "CHARSET=US-ASCII"
```

Do you want to change any values (y/n)? [n]:

The SCP LOCALE parameter applies to the interface driver for business interlinks. It defines the "RPS\_LOCALE" string which the driver sends to the Supply Chain Planning (SCP) server.

**PSTOOLS**

The following parameters are options that you may need to set in advanced configurations.

Values for config section - PSTOOLS

```
JavaVM Shared Library=
```

```
Add to CLASSPATH=
```

```
Proxy Host=
```

```
Proxy Port=
```

```
Suppress App Error Box=Y
```

```
Process exit grace period=5
```

```
DbFlags=1
```

```
Verity Dir=
```

Do you want to change any values (y/n)? [n]:

### JavaVM Shared Library

For PeopleCode and Java integration and to make sure the messaging servers can perform an HTTP post to the messaging gateway specify the local Java Virtual Machine location.

### Add to CLASSPATH

The CLASSPATH environment variable tells the Java Virtual Machine and other Java applications where to find the Java class libraries, including any user-defined class libraries. Because PeopleTools automatically generates CLASSPATH entries for core PeopleSoft delivered class libraries, use this field to specify any custom or additional class libraries that PeopleSoft needs to access.

### Proxy Host

If the HTTP destination, such as the Application Messaging Gateway or business interlink remote host, is "behind" a proxy server for security reasons, then you need to identify the proxy server.

Set the Proxy Host parameter to reflect the distinguished name of the proxy server, as in proxy.peoplesoft.com.

### Proxy Port

You also need to specify the port number on which the proxy server is "listening" for transmissions. For instance, set Proxy Port to 80, a typical default port number.

### Character Set (UNIX only)

This option specifies which character set to use for processing data on the application server. The default value for **Character Set** is *latin1*. This is the character set PeopleSoft supports for use with all Western European languages, including English. If the application server will *only* be used to process Western European data, then you should accept the default for this parameter.

Otherwise, select one of the valid character set choices from the following list:

- **latin1:** (default) Latin-1 - ISO 8859-P1 - Microsoft codepage 1252
- **sjis:** Japanese Shift-JIS - Microsoft codepage 932
- **big5:** Traditional Chinese - Microsoft codepage 950
- **gb:** Simplified Chinese - Microsoft codepage 936
- **ks-c-5601-1987:** Korean Wansung - Microsoft codepage 949
- **ks-c-5601-1992:** Korean Johab - Microsoft codepage 1361



---

The character set of the application server and the character set of any Windows workstations connecting to that application server must match.

---

### Suppress App Error Box (Windows NT Only)

To suppress an application error box or message from appearing after an application error occurs, enter Y for this parameter. If you want to view error dialogs/message boxes, enter N for this parameter.



---

If the system generates an error box for an application server process and this parameter is set to N, then Tuxedo can't restart the down process until you close the error box.

---

### Process exit grace period (PSREAPER.EXE)

Occasionally, when a server process is shutdown, recycled, or experiences a service timeout, for various reasons, the server process can enter a "hung" state. Application server processes in this state have a negative impact on performance. To avoid hung server processes, PeopleSoft provides the following parameter:

```
Process exit grace period=5
```

The grace period is an interval of time allotted to the server processes within a domain to complete a proper exit. Each server process has its own instance of PSREAPER. At the time of exit, the server process actually invokes a child process named PSREAPER.EXE. The PSREAPER program is designed to perform the following tasks:

- "Sleep" for the interval of time that you specify for the grace period.
- "Kill" the parent process by way of the TerminateProcess function (Tuxedo).

If the server process has not successfully performed a successful exit within the interval specified by the grace period, then PSREAPER shuts down the server process.

You want to set the grace period to a value greater than the typical amount of time your server processes require to perform an exit. The default that PeopleSoft ships is tuned to our internal environment and does not necessarily apply to your site. A value of 0 (zero) disables the reaper process.

The PSREAPER program applies to service timeouts, server recycles, and shutdowns.

### DbFlags

Using this parameter you can disable the %UpdateStats meta-SQL construct. To disable %UpdateStats enter 1. To enable %UpdateStats enter 0.

## Verity Dir

Verity is the search engine the PeopleSoft uses for the portal and PeopleBooks search. The Verity Dir needs to reflect the directory where the Verity tool set is installed. This directory should contain `_<platform>/bin/mkvdk` path.

```
%PSVERITYDIR%_nti40\bin\mkvdk
```

## Select Server Process Options (Final Prompts)

After you enter all of the previous parameter values for your application server, PSADMIN will prompt you for the following server process options. You can use these prompts to reduce the amount of server process that start when the domain boots. This, in turn, makes your configuration simpler while helping to conserve system resources.

For instance, if you enter *n* for any of the following prompts, the corresponding server process (or set of server processes) will not be configured for the domain. If you enter *n* to all the prompts, your domain will only contain the required server processes.

### Do you want the Publish/Subscribe servers configured (y/n)? [y]:

If you want the application messaging server processes configured and booted, enter *y*. If you are not implementing the application messaging technology, enter *n*.

### Move quick PSAPPSRV services into a second server (PSQCKSRV)?

Enter *n* if very few clients will access the domain and concurrency is not an issue. Enter *y* to enable the PSQCKSRV in situations where concurrency and optimal transaction throughput are desired.

### Move long-running SqlQuery service into a second server (PSQRYSRV)?

If you want all user generated queries initiated by PSQuery to be handled by a dedicated server process, enable this option. It will improve overall performance.

### Do you want JOLT configured?

JOLT listener is required to support the internet architecture. If you are not going to deploy internet architecture there is no need to configure JOLT.

### Do you want JRAD configured?

JRAD is used to support specific configurations. Accept the default unless you are attempting to configure JRAD for use with Jolt Internet Relay.



For more information on JRAD and Jolt Internet Relay, see Administration Considerations.

---

## CHAPTER 5

# Process Scheduler

You can configure and administer your Process Scheduler Server Agent with the PSADMIN utility. Even if you do not plan on running batch processes on the application server, you can use the PSADMIN utility to configure your Process Scheduler/Process Server Agent regardless of where it will ultimately run.

The following sections describe the menus and options within the PSADMIN utility related to the 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 your Process Scheduler Server. For example, the first menu item is

- 1) Start a Process Scheduler Server

However, logically, at some point previously, you would have already selected

- 4) Create a Process Scheduler Server Configuration

and

- 3) Configure a Process Scheduler Server.

Typically, after you have added and configured a Process Scheduler server, the majority of your administration tasks will involve the first two menu options:

- 1) Start a Process Scheduler Server
- 2) Stop a Process Scheduler Server

This section does not contain an all-inclusive reference to all of the parameters and configuration options that appear in the PSADMIN, Process Scheduler menus. For information on all of the parameters that apply to Process Scheduler, refer to the Process Scheduler documentation. After reading this section you will be familiar with the menus and with the procedures related to configuring a Process Scheduler Server Agent.



For more information on the actual procedural, step-by-step, instructions for installing and configuring the Process Scheduler for the first time, refer to the *Installation and Administration* book for your RDBMS. The following documentation is more reference oriented as opposed to procedural. Also, for more information on the parameters that you need to set while configuring Process Scheduler, refer to Process Scheduler.

---

To access the PeopleSoft Process Scheduler Administration menu

1. Select option 2 from the PeopleSoft Server Administration menu.

```

PeopleSoft Server Administration

```

- 1) Application Server
- 2) Process Scheduler
- 3) Web Components
- 4) Service Setup
- q) Quit

Command to execute (1-3, q):

2. Select the option from the **PeopleSoft Process Scheduler Administration** menu that corresponds to the action you need to perform.

**For Windows NT:**

```

PeopleSoft Process Scheduler Administration

```

- 1) Start a Process Scheduler Server
- 2) Stop a Process Scheduler Server
- 3) Configure a Process Scheduler Server
- 4) Create a Process Scheduler Server Configuration
- 5) Delete a Process Scheduler Server Configuration
- 6) Edit a Process Scheduler Configuration File
- q) Quit

Command to execute (1-6, q) :



**For UNIX:**

-----  
PeopleSoft Process Scheduler Administration  
-----

- 1) Start a Process Scheduler Server
- 2) Stop a Process Scheduler Server
- 3) Configure a Process Scheduler Server
- 4) Create a Process Scheduler Server Configuration
- 5) Delete a Process Scheduler Server Configuration
- 6) Show Status of a Process Scheduler Server
- 7) Kill a Process Scheduler Server
- q) Quit

Command to execute (1-7, q) :

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

## Starting a Process Scheduler Server

This describes the process for starting a Process Scheduler Server on the application server. For information on alternate methods of starting the Process Scheduler Server refer to the Process Scheduler or your Installation and Administration guide.

To start a Process Scheduler Server

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

Database list:

- 1) ps800dmo

Select item number to start: 1



If you are running Windows NT, and you have problems starting the server, PeopleSoft suggests that you modify the COBSW setting in the configuration file, setting it to -L1,+S5. That should allow you to see the error message.

## Stopping a Process Scheduler Server

This process describes the steps you need to complete in order to stop a Process Scheduler Server running on an application server using PSADMIN. If you need information on alternate methods for stopping a Process Scheduler Server refer to Process Scheduler or your *Installation and Administration* guide.



**Note.** You can also stop the server using the Process Monitor.

To stop a Process Scheduler Server

1. Select option 2 from the PeopleSoft Process Scheduler Administration menu.
2. If you want to stop the Process Scheduler server for a specific database, enter the number from the **Database list:** that corresponds to the appropriate database.

Database list:

1) ps800dmo

Select item number to stop: 1

### For Windows NT

You will see a COBOL output window momentarily appear on your screen. Soon after (15-20 seconds) your Process Scheduler Server will stop, and its output window will disappear from your Windows NT Task Bar.

The reason the server does not stop automatically lies in the fact that PSADMIN is designed to perform a "graceful" stop. Rather than immediately killing the server, the server refreshes, processes the request to stop, and then terminates. Depending upon your server sleep time, you may have to wait a few seconds. The default sleep time is 15 seconds. You can verify server status with the Process Monitor.

## Configuring a Process Scheduler Server

Configuring a Process Scheduler server is very 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 configuration file, and the PSADMIN provides an interface for and prompts you to edit the PSPRCS.CFG file.



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

---

For Windows NT:

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

```
%PS_HOME%\APPSERV\PRCS\<dbname>
```

For UNIX:

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

```
$PS_HOME/appserv/prcs/<dbname>
```

To configure a Process Scheduler Server (edit PSPRCS.CFG)

1. Select option 3 from the PeopleSoft Process Scheduler Administration menu.
2. From the **Database list**: select the number that corresponds to the server that you want to configure.

Database list:

```
1) ps800db
```

Select item number to configure: 1

3. Specify the appropriate values for your site in the following configuration section prompts.

The following sections describe each configuration section for the Process Scheduler and the options they offer.



For more information on all of the parameters and menu options available with the Process Scheduler configuration, refer to Process Scheduler Administration.

---

## Creating a Process Scheduler Server Configuration

This section describes the steps you need to complete in order to add a Process Scheduler server configuration on your application server. You must add or create a Process Scheduler before you can configure it.

To create a Process Scheduler server (configuration)

1. Select option 4 from the PeopleSoft Process Scheduler Administration menu.

```

PeopleSoft Process Scheduler Administration

1) Start a Process Scheduler Server
2) Stop a Process Scheduler Server
3) Configure a Process Scheduler Server
4) Create a Process Scheduler Server Configuration
5) Delete a Process Scheduler Server Configuration
6) Show Status of a Process Scheduler Server
7) Kill a Process Scheduler Server
q) Quit
```

Command to execute (1-7, q) : 4

2. Enter the name of the database that the Process Scheduler server will access.

```
Please enter name of Database that server will access :ps800db
```

3. Select the appropriate configuration template for the operating system on which your application server runs. For example, enter 1 if you're using Windows NT, 2 if you're using UNIX, or 3 if you're using VMS.

Process Scheduler Configuration templates:

1) nt

Select config template number: 1

You should see text on the screen similar to that which appears in the following example:

Creating Process Scheduler Server for Database ps800db...

Copying Process Scheduler Server configuration file(s)...

Process Scheduler Server configuration created.

When the add server process completes, you should see the **PeopleSoft Process Scheduler Administration** menu on the screen.

## Deleting a Process Scheduler Server

If you need to delete a Process Scheduler server process from the application sever, you can do so using PSADMIN.

To delete a Process Scheduler server (configuration)

1. Select option 5 from the PeopleSoft Process Scheduler Administration menu.

```

PeopleSoft Process Scheduler Administration

1) Start a Process Scheduler Server
2) Stop a Process Scheduler Server
3) Configure a Process Scheduler Server
4) Add a Process Scheduler Server Configuration
5) Delete a Process Scheduler Server Configuration
6) Show Status of a Process Scheduler Server
7) Kill a Process Scheduler Server
q) Quit

```

Command to execute (1-7, q) : 5

2. Select the number in the **Database list:** that corresponds to the database to which your server has access.

Database list:

```
1) ps800db
```

```
Select item number to delete: 1
```

3. PSADMIN prompts you to continue; if you want to delete the server, enter y.

```
Do you want to continue? (y/n) [n] :y
```

You should see the following text on the screen.

```
Attempting to delete directory...
```

```
Directory successfully deleted.
```

## Editing the Process Scheduler Configuration File

You can edit the Process Scheduler Server configuration file manually instead of using the prompts in the PSADMIN interface to specify environment variables if you want. This allows you to edit the configuration file in your favorite editor. You will need to set your EDITOR environment variable to point to the editor. For example:

```
set EDITOR=c:\apps\utils\kedit\keditw32.exe
```

or if you use Notepad:

```
set EDITOR=c:\Windows\Notepad.exe
```



When editing PSPRCS.CFG, make sure that there are no spaces between the equals sign and the entries. Also, make sure that there are no trailing spaces.

---

To manually edit the pspres.cfg

1. Select option 6) Edit a Process Scheduler Configuration File from the PeopleSoft Process Scheduler Administration menu.
2. Select the database associated with the file you want to edit.

Database list:

- 1) ptdmo
- 2) ptdmo80
- 3) ptdmotst

Select item number to configure:

3. Enter the variables for the parameters you need to specify.



The system invokes the text editor that you have set as the %editor% environment variable set on the particular machine, such as Notepad or KEDIT.

---

## Process Scheduler Options

You can elect to have the Process Scheduler Server run as a standalone component, or you can have the Process Scheduler server controlled by Tuxedo, which enables automatic restarts if the server goes down.



For more information on configuring Process Scheduler under Tuxedo see Process Scheduler.

---

## Process Scheduler Command Line Options

You can bypass the PSADMIN menus to start and stop your Process Scheduler server.

### Start the Process Scheduler

To start you process scheduler server from the command line, enter the following:

```
psadmin -p start -d <dbname>
```

### Stopping the Process Scheduler

To stop you process scheduler server from the command line, enter the following:

```
psadmin -p stop -d <dbname>
```





# Web Components



**Important!** You do not use the Web Components menus to configure the internet architecture. This set of PSADMIN menu options is intended mainly for upgrade support only. If you have implemented the Web Client for use with PeopleTools 7 or 7.5 then you may need to use these options at some time. Otherwise, you can safely ignore these options. Using the WEBGUI monitoring utility may be a useful administration tool in some circumstances.

The Web Components menu contains a collection of optional menu options. The options here are offered merely as a way to configure Tuxedo's web server that's shipped with your BEA CD-ROM, and to setup a minimal test environment for Web Client technology. The Tuxedo web server is not intended for use as your production web server. In the past PeopleSoft has used the Tuxedo web server for use as a demonstration web server only.

On the menu are the options to configure Tuxedo's WEBGUI monitoring utility. This is a BEA utility that can prove to be quite useful in monitoring the performance of your application server domains. However, keep in mind that since it is a BEA product, you'll need to refer to BEA documentation for information on its features.

Tuxedo's web server capability is not suitable and is not designed for a full-scale production environment. This section explains the PSADMIN menus related to the Tuxedo web server and the options and settings they offer. Typically, Tuxedo's web server is used only for deploying BEA's online documentation and for using the WEBGUI interface.



**Note.** Tuxedo's web server is not supported for the Application Messaging Gateway or PIA.



For more information on the procedures for installing and configuring the Tuxedo web server and the BEA documentation, refer to your *Installation and Administration* book. For more information on the supported web servers for the Application Messaging Gateway and the PIA, see the PeopleSoft Platforms database on Customer Connection and the *PeopleSoft Hardware and Software Requirements* guide.

To access the PeopleSoft Administrative Web Components Menu

1. Select option 3 from the PeopleSoft Server Administration menu.

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

- 1) Application Server
- 2) Process Scheduler
- 3) **Web Components**
- 4) Service Setup
- q) Quit

Command to execute (1-4, q):

2. From the **PeopleSoft Web Components Administration** menu, select the number that corresponds to the action you want to perform.

-----  
PeopleSoft Web Components Administration  
-----

- 1) Web Server Administration
- 2) Tuxedo Web Monitoring Facility Administration
- 3) PeopleSoft starter page Administration
- 4) Generate Web Components configuration files
- q) Quit

Command to execute (1-4, q):

The following sections provide descriptions of the options that appear in each menu.



You need to generate Web Components configuration files before starting Tuxedo Web Monitoring or the web server.

---

---

## Web Server Administration

-----  
PeopleSoft Web Server Administration  
-----

- 1) Start Tuxedo administrative web server
- 2) Edit pswebsrv.ini
- q) Quit

Command to execute (1-2, q) [q]:

### Start Tuxedo administrative web server

This options starts the tuxwsrv process (the Tuxedo web server).

### PSWEBSRV.INI

The PSWEBSRV.INI shows the virtual directory structure. You normally do not need to change any values in this file.

```
Tuxedo Web Server initialization file.

Created by PeopleSoft UBBGEN System installation program.

#

#Please note! The last entry of this file MUST be the entry for the (/)root
virtual directory!

#

#Below are the virtual directories that are required to run the BEA WebGUI
Administration Application.

#

#/cgi-bin, /doc and /java MUST point to the TUXEDO directories indicated below.

#

#/webmon should point to the webmon directory in the high level appserver
directory

#

#/appserv should point to the high level appserver directory
```

```

#

#/ MUST point to the directory where the Web Client files were transferred to.

(The javaclient subdirectory MUST be at the virtual location /javaclient
!!!!)

#

CGI /cgi-bin c:\tuxedo\udataobj\webgui\cgi-bin

HTML /java c:\tuxedo\udataobj\webgui\java

HTML /webmon c:\ptappsrv\appserv\webmon

HTML /appserv c:\ptappsrv\appserv

HTML /doc c:\tuxedo\doc

HTML / c:\ptappsrv\web

ubbgen substitution values:

#

[0]: {$TUXDIR} : c:\tuxedo

[1]: {FS} : \

[2]: {$PS_HOME} : c:\ptappsrv

ubbgen control values:

#

[0]: {UNIX} : FALSE

[1]: {WINDOWS} : TRUE

```

---

## Tuxedo Web Monitoring Facility Administration

The Web Monitor is not a required piece for the Web Client, but it is useful as a system management tool for monitoring application server domains through a browser.

-----

## PeopleSoft Tuxedo Web Monitoring Facility Administration

-----

- 1) Start Tuxedo Web Monitor wlisten process
- 2) Edit pswebgui.ini
- q) Quit

Command to execute (1-2, q) [q]:

### Start Tuxedo Web Monitor wlisten process

This option starts the wlisten process, which will display information on all domains specified in the PSWEBGUI.INI file.

### Edit pswebgui.ini

In this file you can change the port number to add additional domains.

```
Web GUI initialization file.

Created Thu Dec 04 13:02:07 1997 by TUXEDO System installation program.

#

TUXDIR=C:\TUXEDO

INIFILE=C:\ptappsrv\Appserv\pswebgui.ini

NADDR=//gsawyer090397:4003

CODEBASE=/java

DOCBASE=http://GSAWYER090397:80/doc

SNAPDIR=C:\ptappsrv\Appserv\webmon

SNAPBASE=/java/snapshot

#

In order to configure one or more domains as part of the Web GUI pull-down

menu, add lines to this file of the form DOMAIN=domainname;tuxconfig
```

---

## PeopleSoft Starter Page Administration

We supply a Starter Page so that you can test your installation and configuration and get a taste of the Web Client's look and feel.



**Note.** The Web Client is not offered in PeopleTools 8.0.

---

```

PeopleSoft Starter Page Administration

```

- 1) Launch PeopleSoft Starter Page
- 2) Edit psstarter.html
- q) Quit

Command to execute (1-2, q) [q]:

## Launching the Starter Web Page

PeopleSoft ships a starter web page to give you a sample of Tuxedo's web server capabilities and the PeopleSoft Web Client.

To launch the PeopleSoft Starter Web Page

3. Select option 3 from the PeopleSoft Administrative Web Components menu.

```

PeopleSoft Web Components Administration

```

- 1) Web Server Administration
- 2) Tuxedo Web Monitoring Facility Administration
- 3) PeopleSoft starter page Administration
- 4) Generate Web Components configuration files
- q) Quit

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

4. Then, from the **PeopleSoft Starter Page Administration** menu, select option 1.

```

PeopleSoft Starter Page Administration

```

- 1) Launch PeopleSoft Starter Page
- 2) Edit psstarter.html
- q) Quit

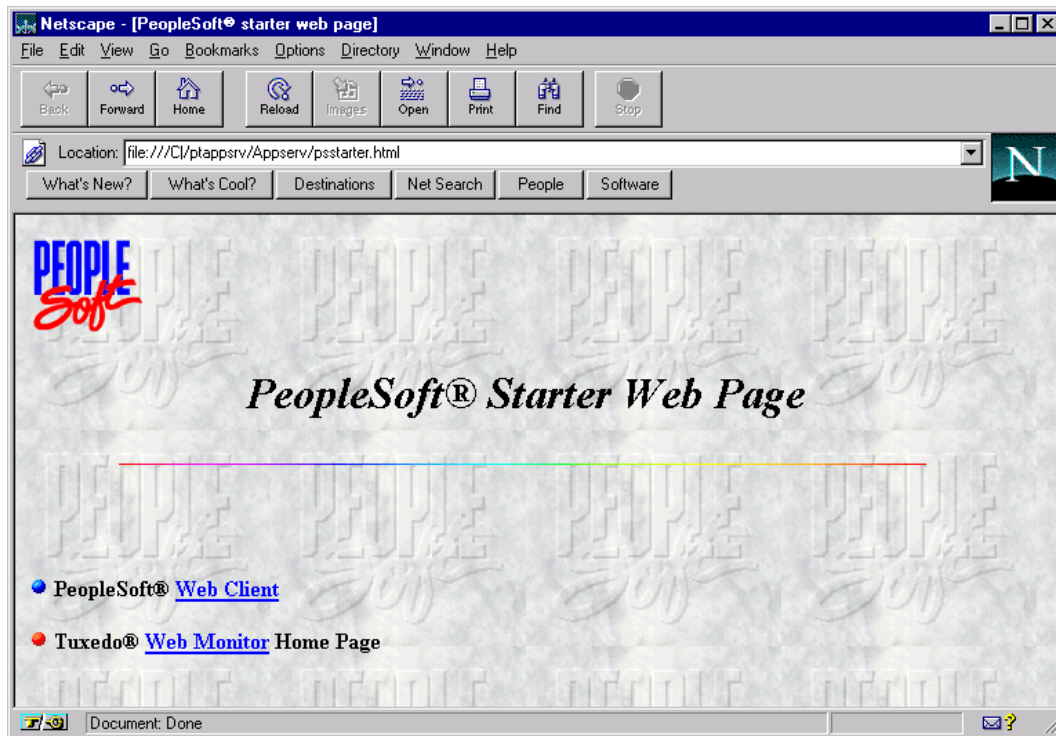
Command to execute (1-2, q) [q]:

Starting default browser with the PeopleSoft starter web page.

Please wait while the browser starts...

Sent this request to the system: start PSStarter.html

5. Select either PeopleSoft Web Client or the Tuxedo Web Monitor Home Page.



PeopleSoft Starter Web Page

## Edit psstarter.html

A user can copy the URL displayed into a browser to launch the Web Client.

```

<!--/* -->

<!--* Confidentiality Information: -->

<!--* -->

<!--* This module is the confidential and proprietary information of -->

<!--* PeopleSoft, Inc.; it is not to be copied, reproduced, or transmitted -->

<!--* in any form, by any means, in whole or in part, nor is it to be used -->

<!--* for any purpose other than that for which it is expressly provided -->

<!--* without the written permission of PeopleSoft. -->

<!--* -->

<!--* Copyright (c) 1988-1997 PeopleSoft, Inc. All Rights Reserved. -->

<!--* -->

<!--*/ -->

<HTML VERSION="2.0">

```



```
<HEAD>

<TITLE>

>PeopleSoft® starter web page</TITLE>

</HEAD>

<BODY>

 background="webmon/images/psbgkd.gif"

 bgcolor="#ffffff"

 text="#000000"

 link="blue"

 vlink="blue"

 alink="blue">

<CENTER>

<H1><i>PeopleSoft® Starter Web Page</i></H1>

</CENTER>

PeopleSoft® Web
Client

 Tuxedo® Web
Monitor Home Page

</BODY>

</HTML>
```

---

## Generate Web Components configuration files

This option acts as a prompt for editing the PSWEBSRV.INI file so that you can change the port number or the directory where the HTML resides.

To generate the web configuration files

6. Select option 4 from the PeopleSoft Administrative Web Components Menu.

```

PeopleSoft Web Components Administration

```

- ```
1) Web Server Administration  
2) Tuxedo Web Monitoring Facility Administration  
3) PeopleSoft starter page Administration  
4) Generate Web Components configuration files  
q) Quit
```

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

```
Generating web server configuration file...
```

7. If you want to change or view your web component configuration values, enter *y* when prompted to change configuration values. (These values are stored in the PSWEBSRV.CFG file.)

```
Do you want to change any config values (y/n)? [n]: y
```

8. View the parameters within the WebServer section, and enter *y* to change any of the values.

```
Values for config section - WebServer
```

```
ServerName=%PS_MACH%
```

```
Port=80
```

```
PSSite=SampleSite
```

Do you want to change any values (y/n)? [n]:y

Configuration file successfully created.

Generating PeopleSoft starter web page...

Configuration file successfully created.

9. When prompted to change any values associated with the administration web page, enter y if you wish to view or change these values.

Generating PeopleSoft application server administration web page...

Do you want to change any config values (y/n)? [n]: y

Values for config section - AdminWebPage

ServerName=PeopleSoft Application Server

10. After you have specified all the appropriate environment variables, press ENTER.

When the file generation process completes, the **PeopleSoft Administrative Web Components Menu** appears.

PeopleSoft Service Setup



Note. This chapter only applies to Windows NT servers. It involves setting up both the application server and Process Scheduler Server Agent as PeopleSoft Windows NT Services. PeopleSoft does not provide an equivalent feature for UNIX servers.

You can start application server domains and Process Scheduler Servers as Windows NT services. The PeopleSoft Service, if configured, automatically starts the application server or Process Scheduler when you boot the server machine. This means that administrators do not need to manually boot each application server or Process Scheduler Server after you reboot a Windows NT server.

After reading this section, you'll be familiar with the scope of the PeopleSoft Service and how to configure it.

Overview

A Windows NT service is a Microsoft-standard package that automatically starts and stops a process when you boot or shutdown the system. You can also start and stop Windows NT services manually through the Service Control Manager (SCM), which you can access through the Control Panel. A service uses a standard API so that it can interact with the Control Panel and log messages to the standard Event Log.

An important aspect for PeopleSoft is that the service we developed starts in an environment that is *separate* from any users logged on the system (or the machine). This means that administrators no longer need to log on to a machine, launch the Command Prompt, and enter the proper commands to start the server process. It's also important to note that if you use the PeopleSoft Service, an administrator's login session does not need to remain open while the Process Scheduler Server or the application server runs.

If you have multiple application server domains and Process Scheduler Servers on the same machine, you can start them all using the same Service Setup.



Note. The PeopleSoft Service supercedes the method provided in the Windows NT Resource Kit. PeopleSoft does not support using SRVANY.EXE or AT commands to start the Process Scheduler or the application server.

So, that's what the PeopleSoft Service does. It's also important to know what it's not designed to do. The following list lets you know what not to expect from the PeopleSoft Service.

- It doesn't change the behavior, performance or resource requirements of PeopleSoft applications.
- It doesn't create or configure servers or domains automatically. Your Process Scheduler Server and application server domain should be installed and configured *before* installing the service.
- It isn't intended to start *anything* other than the Process Scheduler Server or an application server domain.
- It doesn't make the Process Scheduler Server or an application server domain more fault-tolerant.
- The PeopleSoft Service *only* applies to Windows NT servers.

Configuring the PeopleSoft Service

The following procedure assumes that you have already installed and configured an application server domain and/or Process Scheduler Server Agent on the Windows NT server.

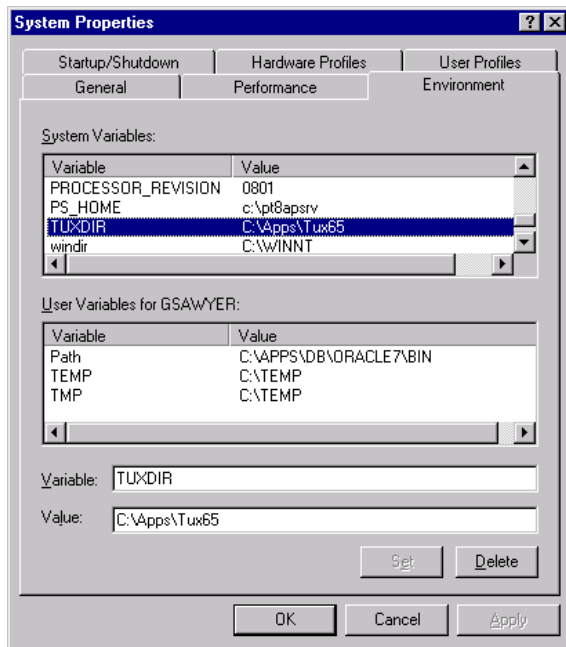
After completing this procedure, the specified application server domain(s) or Process Scheduler Server(s) start and shutdown automatically when the operating system recycles.

To set up the Windows NT Service for an Application Server or Process Scheduler Server

1. Open the System utility within the Control Panel, and set the following variables on the **Environment** tab.

| Variable | Value |
|-----------------|---|
| TEMP | Specify the location of the TEMP directory on the Windows NT server, as in C:\TEMP. |
| TUXDIR | Specify the location of the Tuxedo directory on the Windows NT server, as in C:\apps\tux65. |

These settings need to appear in the **System Variables** section, as shown in the following example.



System Properties Dialog

2. Run the PeopleSoft PSADMIN utility, and select option **4) Service Setup** from the **PeopleSoft Server Administration** menu.

```

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

```

- 1) Application Server
- 2) Process Scheduler
- 3) Web Components
- 4) Service Setup
- q) Quit

Command to execute (1-4, q): **4**

3. Select option 1) Configure a Service from the PeopleSoft Services Administration menu.

```

-----
PeopleSoft Services Administration
-----

```

- 1) Configure a Service
- 2) Install a Service
- 3) Delete a Service
- 4) Edit a Service Configuration File
- q) Quit

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

And, enter **y** to indicate that you want to change configuration values, as shown.

4. Do you want to change any config values (y/n)? [n]: y
5. Enter the name of the application server domain(s) and the Process Scheduler Databases that you wish to be included as part of the Windows NT Service.

For example

Values for config section - NT Services

Service Start Delay=60

Application Server Domains=**Domain1, Domain2**

Process Scheduler Databases=**PT80HRDB, PT80FSDB**

Do you want to change any values (y/n)? [n]:

To add multiple domains or databases, you delimit each value with a comma *and* a space as shown in the previous example.



Note. The NT Services section of the PSADMIN modifies the PSNTRV.CFG file located in the <PS_HOME>\appserv directory. You can edit this file manually by selecting the option **4) Edit a Service Configuration File** on the **PeopleSoft Services Administration** menu.

6. Select option 2) Install a Service from the PeopleSoft Services Administration menu.

 PeopleSoft Services Administration

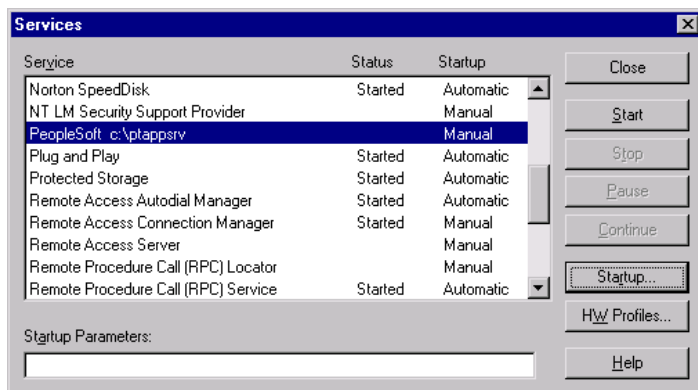
- 1) Configure a Service
- 2) Install a Service
- 3) Delete a Service
- 4) Edit a Service Configuration File
- q) Quit

Command to execute (1-4, q) : 2

7. Return to Control Panel, and launch the Services utility.
8. On the **Services** dialog, scroll to find the entry that adheres the following naming convention, and select it:

PeopleSoft <PS_HOME>

For example,

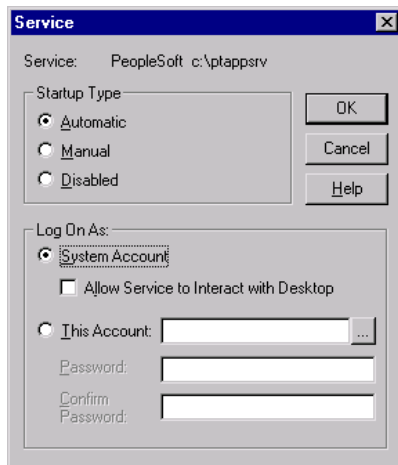


Services Dialog



Note. The default Startup mode is Manual.

9. Click Startup.
10. On the **Service** dialog in the **Startup Type** group, select **Automatic**, and in the **Log On As** group, select **System Account**.



Service Dialog

For Process Scheduler services, you need to select **This Account** otherwise problems while running Crystal Reports occur.



Note. The Log On As setting needs to reflect that which you set for your Tuxedo IPC Helper and Tlisten process. PeopleSoft recommends that when you install Tuxedo, you set these services to System Account. This only affects the application server since Process Scheduler is not integrated with Tuxedo.

When you've finished making the appropriate selections, click **OK** to dismiss the **Service** dialog.

11. On the **Services** dialog, make sure the PeopleSoft service is selected, and click **Start**.

Your application/Process Scheduler servers are now running and will launch automatically whenever you boot the server.

Monitoring the Executables

To test your Windows NT Service, you'll want to reboot your server, and then make sure that the appropriate server executables are running.

For the application server, use Windows NT Task Manager or the Server status option from the Domain status menu to see that the following executables are running:

- PSAPPSRV.EXE
- PSSAMSRV. EXE
- BBL. EXE
- WSL. EXE
- Also make sure any additional server processes you have configured, such as

PSQCKSRV.EXE, are also running.

For the Process Scheduler, use Windows NT Task Manager or the Process Monitor to make sure that PTPURCS.EXE is running. If you've customized the name of PTPURCS.EXE, look for your custom name instead.

PeopleSoft Services Administration Reference

There are three options related to your PeopleSoft Service setup. Each option you can specify either using PSADMIN or editing the PSNTSRV.CFG file manually.

```
Values for config section - NT Services
```

```
Service Start Delay=60
```

```
Application Server Domains=
```

```
Process Scheduler Databases=
```

```
Do you want to change any values (y/n)? [n]:
```

The following sections describe each parameter.

Service Start Delay

When an application server or Process Scheduler Server resides on the same machine as the database server, you should consider using the Service Start Delay setting. Using this feature, you can avoid the situation where the database server is in the process of booting (just like everything else on the machine) and is not ready to process requests at the time that the service attempts to boot the application server domain or Process Scheduler Server. In this scenario, without a delay set, the connection will fail.

You can configure a "Service Start Delay" parameter in the PSNTSRV configuration file that specifies a delay, in seconds, that elapses prior to a service attempting to start any application server domains or Process Scheduler Servers. This allows the RDBMS enough time to boot and become available to accept requests.

The default is 60 seconds.

Application Server Domains

Here specify the names of the domains that you want to automatically start 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 to which a Process Scheduler Server is associated. For each database you specify the associated Process Scheduler Server starts when you boot the Windows NT server.

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

Editing the PSNTRV.CFG File Manually

You can edit the file directly by selecting 4) Edit a Service Configuration File from the main menu. This opens PSNTRV.CFG in a text editor, where you can enter and save your changes.



Note. Make sure that your editor environment variable is set correctly.

A screenshot of a Notepad window titled 'psntrv.cfg - Notepad'. The window shows the contents of the PSNTRV.CFG file, which is a configuration file for the PS NT Service. The text in the file includes comments and configuration parameters. The window has a standard Windows interface with a menu bar (File, Edit, Search, Help) and a scroll bar on the right.

```
[NT Services]
;
; "Service Start Delay" -- Delay (in seconds) for starting the PS NT Service.
;
;
; Service Start Delay=60
;
; Add the names of the Application Server Domains that will be run as an NT Service.
; Example:
;   Application Server Domains=PTDMO, PTTST
; Application Server Domains=
;
; Add the names of the Process Scheduler Databases that will be run as an NT Service.
; Example:
;   Process Scheduler Databases=PTDMO, PTTST
; Process Scheduler Databases=
;
```

PSNTRV.CFG File

CHAPTER 8

Web Server Administration

The web server is a vital component within the PeopleSoft Internet Architecture. It is required for deploying PeopleSoft applications to the browser and for implementing PeopleSoft integration solutions.

This chapter contains information on the following topics:

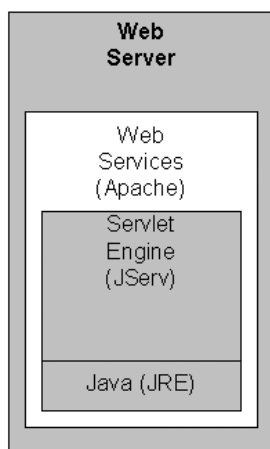
- How the different web server components interoperate.
- What the PeopleSoft servlets do and how they connect to the application server.
- Where you can find the configuration files on the web server.
- Setting timeout values.

Server Components

On the server that you have designated to be your web server, the following components are required for successful incorporation into the PeopleSoft Internet Architecture:

- **Web services software.** This is the major piece of software that manages the web server. There is a variety of web server software available. Apache is a common type of web server software.
- **Servlet Engine.** Because PeopleSoft servlets run on the web server, you need to install a servlet engine on the web server. A servlet engine is the environment in which servlets run. The servlet engine requires Java executables to run. A common servlet engine is JServ.
- **Java executables.** Java is a platform-independent programming language widely used for web-based programs. With Java you develop applets and servlets. Applets are small programs embedded in HTML pages that get download to the browser and invoked locally on the client. Servlets are Java programs that run on the web server, not the client. PeopleSoft only uses servlets. A common set of Java executables is the Java Runtime Environment (JRE).

The following example illustrates the logical architecture of these components on the web server.



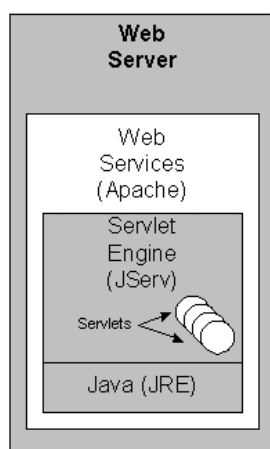
Web Server Components

The web services work within the web server environment, and the servlet engine runs on the Java executables within the web services.

PeopleSoft Servlets

There are four PeopleSoft-specific servlets that are installed directly to the web server. Depending on your installation, you may only choose to install a subset of the servlets. If you find that you want to implement a technology that requires a servlet, such as Application Messaging, that you did not install initially, you can always install any of the servlets as needed.

The PeopleSoft servlets run inside the servlet engine environment.



PeopleSoft Servlets

The following topics briefly describe the function of each PeopleSoft servlet. The following topics describe the PeopleSoft servlets that you install onto the web server.

Page Servlet

This servlet enables an end user to connect to a PeopleSoft application. The servlet relays requests to the application server, and it also formats the HTML for deployment in a browser. This servlet handles the building of the components on the application server and the presentation of the components in the browser.

For the initial connection the system provides a login HTML file that resides on the server. After a successful login, the application server generates the subsequent HTML pages to complete the transaction. The servlet relays the pages to the browser.

The Page Servlet requires the user to enter specific URL or click a specific link in order to signon to the system.

Portal Servlet

The Portal servlet handles all of the requests and formatting for the users accessing PeopleSoft through the PeopleSoft Portal. It also manages all aspects of the PeopleSoft Portal such as search, content management, and home page personalization.



For more information on the PeopleSoft Portal see Portal Technology.

Report Repository Servlet

The Report Repository servlet enables users to easily access and distribute the output of batch reports, such as Crystal and SQR, run through Process Scheduler over the Internet.

Application Messaging Gateway Servlet

The Application Messaging Gateway servlet transmits publish/subscribe messages between message nodes. The gateway handles PeopleSoft-to-PeopleSoft messages, PeopleSoft-to-third party messages, and third party-to-PeopleSoft messages.

Java Message Agent Client (JMAC)

PeopleSoft provides the Java Message Agent Client (JMAC) for backward compatibility. If you developed web pages for delivery using PeopleTools HTML Access, a previous web solution, or if you developed custom applications that used the JMAC API, you have the option of installing this component to your web server to support previous development. If you have not implemented JMAC/HTML Access in a previous PeopleTools version, you don't need the JMAC.



PeopleSoft does not recommend further or new development in the area of HTML Access.

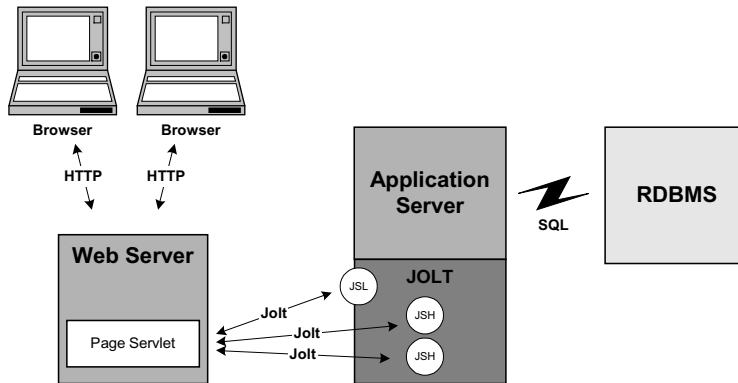
Transmitting Requests

When end users interact with a PeopleSoft application in the browser, they are looking at the HTML presentation produced by the servlet, not an actual HTML page. You can't find the "page" they are looking at anywhere on the web server.

PeopleSoft uses the servlets to enable the browser to communicate with an application server domain, forming a Java link between the browser and the application server (and therefore the database). Because Tuxedo uses Jolt to extend its capabilities to the Java realm, to establish the connection between the web server and the application server, the servlets need to be configured to send messages to a predefined Jolt port on the application server.

The following list shows the order of events that occur during a browser transaction:

- The PIA program establishes a connection to handle your browser's connection on the web server side.
- The Jolt Server Listener (JSL) at port 9000, by default, handles the initial connection to the application server. It hands off the communications responsibilities to an available handler (JSH).
- The JSH hands the request off to Tuxedo.
- Tuxedo analyzes the Jolt requests, and hands them off to the appropriate PeopleSoft server process queues.
- PSAPPSRV process does virtually all of the work when it comes to constructing web pages. Using Tuxedo resources, this PSAPPSRV retrieves object data from cache or the database and builds an HTML page, passing the completed work to the JSH.
- Assigned JSH passes the completed page to the Page Servlet.
- Page Servlet displays the page in the browser.



Browser connecting through the Page Servlet

When you install the servlets, you must supply the address of the JSL on the application server. For example, the default value is

```
<MachineName>:9000
```

If you've modified the default value of the Jolt port you need to enter the port that corresponds to the domain to which the servlet needs to send requests.

Multiple Servlets

Because you configure each servlet to communicate with a single JSL port address, the relationship between servlets and application server domains is one-to-one. However, you can install multiple instances of a servlet on a single web server, each servlet pointing to a different application server domain.

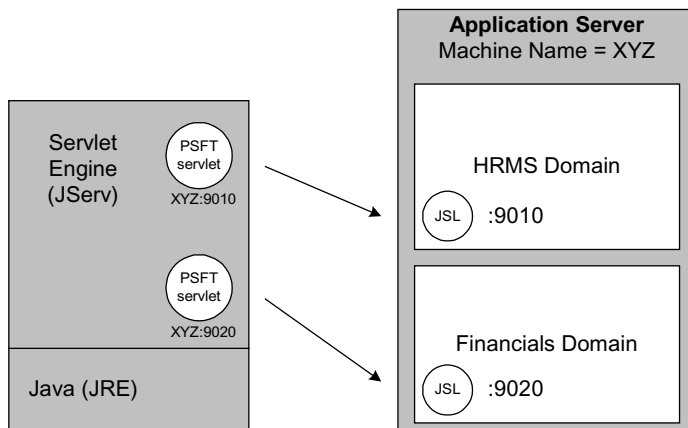
If the multiple domains happen to reside on the same application server machine, then it's just a matter of making sure that each servlet points to the appropriate Jolt port. The machine name value (assume it is XYZ) remains the same. For example,

Servlet 1

```
XYZ:9000
```

Servlet 2

```
XYZ:9010
```



Multiple Servlets and Multiple Domains



In the previous example the term PSFT servlet does not represent an actual servlet. It represents any of the PeopleSoft servlets. There is no PSFT servlet.

If the application server domains reside on separate application server machines, then you need to make sure that both the machine name value and Jolt port value are pointing to the appropriate targets. For example,

Servlet 1

XYZ:9000

Servlet 2

ABC:9010

You set the machine name and port values during the install of the servlets, and, if necessary, you can modify the configuration files

Configuration Files

Because there are numerous components involved in the delivering PeopleSoft applications to a browser, it follows that there are a variety of configuration options, parameters, and files. This section contains information regarding what most important configuration files are and where to find them.

PeopleSoft Configuration Files

The following files—delivered by PeopleSoft—are related to the general operations of the PIA architecture:

- **configuration.properties.** This is the main PIA configuration file. If you make modifications to the configuration files, you will find that you make most of your modifications to this file. It

contains connection settings, security settings, portal settings, just to name a few.

- **pstools.properties.** This configuration file contains a few select Tuxedo parameters and the locale_ settings for specifying global date and time values.
- **errors.properties.** This file contains the error numbers and messages that the system displays during the associated events. The errors.properties file is designed primarily for displaying error conditions and parameters. In most cases, you do not need to modify this file.
- **text.properties.** This file contains the text for pages, which might be indicating a general error condition. In most cases, you do not need to modify this file.

Once you run the internet setup program, these files are located on your web server under the following *default* directory:

Windows NT:

```
C:/Program Files/Apache Group/Apache/psftdocs/peoplesoft8
```

UNIX:

```
$ps_home/webserv/psftdocs/peoplesoft8
```

Non-PeopleSoft Configuration Files

There are also a few non-PeopleSoft configuration files with which you should be familiar. They are related to the Apache and JServ components.



These do not represent all of the Apache or JServ configuration files, just the ones that you want to be aware of for your PeopleSoft configuration.

Apache Configuration Files

The main configuration file for the Apache web services is the httpd.conf file. You can find this file in the following locations:

Windows NT

```
C:\Program Files\Apache Group\Apache\conf\httpd.conf
```

UNIX

```
$ps_home/webserv/htdocs
```

Refer to the Apache documentation and/or the comments in the httpd.conf file for information on this third party file.

JServ Configuration Files

JServ stores some of its configuration settings in the `zone.properties` file. You can find this file in the following locations:

Windows NT

`C:/Program Files/Apache Jserv 1.1/servlets`

UNIX

`$ps_home/webserv/conf/jserv`

Refer to the Apache documentation and/or the comments in the `zone.properties` for information on this third party file.

Modifying PeopleSoft Web Configuration Files

There is a collection of configuration files (properties files) that exist on the web server within a PIA implementation. This section identifies and provides information for all of the variables defined in the commonly edited PIA configuration files—`configuration.properties` and `pstools.properties`. This section also contains information on modifying some of the more typical configuration options, such as Jolt failover and system timeouts.

Many of the variables contain suitable default values, but some of them you may want to modify as you tune or customize your system. On the other hand, PeopleSoft does not recommend editing some of the variables; these variables are marked accordingly in the following sections.

Before you begin editing the configuration files, it's a good idea to have general understanding of which files are associated with which component, what the names of those files are, and where you locate them. The following sections identify the configuration files associated with PIA, the servlets, and the portal.



If you make changes to these files, keep in mind that whenever you run the internet setup program again, it overwrites the files on the web server with new copies. If you have any tuned variables you should either back them up to a remote location, or make note of your tuned values.

All keys in these files are case sensitive. Values may be too, for example *class names*.

configuration.properties variables

The following table contains the PIA configuration variables that are located in `configuration.properties` files. You will also find comments in the `configuration.properties` file for easy viewing while editing the file.



Variables marked with an asterisk (*) must be reviewed before attempting to use PeopleSoft 8. These variables enable the basic connection to occur between PIA and Tuxedo.

| Variable | Description |
|----------------------------------|--|
| General Settings | |
| *psserver="//machinename:port" | <p>The psserver parameter must point to your application server machine name or IP Address, including the Jolt port. PeopleSoft recommends using the server machine name. The PeopleTools version on the application server must match the version of the PeopleTools files on the web server. The syntax of the value is</p> <p><domain_name (or IP)>: Jolt port</p> <p>UNIX servers require a domain name. For example,</p> <p>NT:</p> <p>psserver=SERVER010499: 9000</p> <p>UNIX:</p> <p>psserver=server010400. peoplesoft. com: 9000</p> <p>To enable Jolt failover and load balancing, include multiple application server domains delimited by commas. For example,</p> <p>NT:</p> <p>psserver=SERVER1: 9000, SERVER2: 9010</p> <p>UNIX:</p> <p>SERVER1. peoplesoft. com: 9000, SERVER2. peoplesoft. com: 9010</p> |
| *pswebservername="//machinename" | <p>If UseCanonicalName is turned off in the httpd.conf file, leave pswebservername commented out. If UseCanonicalName is on, set this parameter.</p> <p>Enter the web server machine name (recommended) or IP address. The user URL must match this value.</p> <p>NT:</p> <p>pswebservername=WEBSERVER101898</p> <p>UNIX:</p> <p>pswebservername=WEBSERVER101898. peoplesoft. com</p> <p>If your web server is outside the firewall, you must provide the port number or the domain name.</p> |

| Variable | Description |
|---------------------------------------|--|
| *helpUrl= | This is where you add the location of your HTML PeopleBooks. When the user clicks the Help button, they will view the PeopleSoft documentation at the location you specify. |
| Debug and Trace Settings | |
| enableTrace=false | If set to true, during signon you can set trace parameters. |
| signontrace_page=signintrac
e.html | If you enable tracing, the URL that appears at signon is the signontrace_page. Here you set the trace parameters and then sign on to the system. |
| enableDebugDumpFile= | This parameter enables you to specify whether or not the system writes a dump file in the event that a Jolt exception error occurs. |
| testing=false | If set to true, this option alters the generated HTML to assist with testing and trouble shooting. For instance, it provides additional white space and comments in the HTML to aid readability. Also, it includes additional name attributes for reference from SQA robot scripts. |
| connectionInformation | If set to true, the database name, application server address, web server, and User ID information appears in the HTML generated for a "help" page. PeopleSoft provides a hotkey option (CTRL+J) to enable users and system administrators to view such system information for orientation and troubleshooting purposes. |
| debug_showlayout=false | If set to true, this option puts border and color attributes in a table layout for pages. This enables developers to see the position of Application Designer objects in HTML. |
| debug_inlinestylesheet=false | If set to true, this option inserts the page's stylesheet into the generated HTML for easy reference. |
| debug_inlinejavascript=false | If set to true, writes all the javascript functions used for processing into the generated HTML file. |
| debug_overlap=false | If set to true, includes comments in the generated html page that may help in diagnosing page layout problems, such as fields that may be overlapping other fields). |
| Cache Settings | |
| noCache=false | When this variable is set to true, the displayed page is not cached by the browser. Therefore, clicking the "Back" button from browser will result in "Warning: Page has Expired." from Internet Explorer or "Data Missing..." from Netscape. This variable should be used in Kiosk situations. |

| Variable | Description |
|--|---|
| imagedirphys=/cache/images
imagedirwebv=/cache/images | Image file cache directories.
Do not change these settings. |
| cssdirphys=/cache/CSS
cssdirwebv=/cache/CSS | Cascading style sheet cache directories.
Do not change these settings. |
| jsdirphys=/cache/js
jsdirphys=/cache/js | Java Script cache directories.
Do not change these settings. |
| Sign in Settings | |
| physicalpath= | If you are using an Apache alias to refer to PeopleSoft 8 files, you need to enter the actual location of the various load files required by PIA. By default this value is blank. Typically, you want to point to the standard installation directory.

NT:
Program Files/ApacheGroup/Apache/psftdocs/peoplesoft8

UNIX:
\$ps_home/webserv/psftdocs/peoplesoft8 |
| signon_page=signon.html | Redirects to the servlet for login process.
Do not change this setting. |
| signonError_page=signin.html | Page content that presents signin/login process. If you want to customize your signin page, it is recommended that you clone signin.html as a starting point.

Change with caution, though not recommended. |
| logout_page=signin.html | You may provide a custom logout page.
Change with caution, though not recommended. |
| Navigation Settings | |
| start_page=start.html | Indicates the page to which the system re-directs users after successful login. References the iScripts that build the PIA navigation.
Do not change this setting. |

| Variable | Description |
|----------------------------------|--|
| maxSavedState=5 | <p>Number of states supported by the browser Back button. Each trip to the server equates to one state.</p> <p>ICElementNum and ICStateNum work together for the datastructure on the web server to maintain state for the user: State[ICElementNum][ICStateNum]</p> <p>ICElementNum is system controlled; ICStateNum is user-configurable by this variable.</p> <p>Example: States for a user could look like:
 State[0][0]
 State[0][1]
 State[0][2]
 State[0][3]
 State[0][4]</p> <p>By the time a user goes to another page, you might end up with:
 State[0][7] . In this case, States[0][0,1,2] are no longer in memory. Trying to use the browser Back button to get there, gives the user a "Page Expired" message when they arrive at the web page that has the ICElementNum=2.</p> <p>Note: If you have applications that make numerous server trips, you may want to increase the maxSavedState value. Keep in mind that this does increase the virtual machine's memory requirements, so be prepared to allocate more memory accordingly.</p> |
| expirePage_ContentName=PT_EXPIRE | <p>Define the content name that is stored in the HTML catalog. It appears when a page has expired due to reaching the maxSaveState limit.</p> <p>Change with caution, though not recommended.</p> |
| exception_page=expire.html | <p>It is used in Java to handle exceptions such as java exception errors.</p> <p>Change with caution, though not recommended.</p> <p>The detailed errors are logged in:</p> <p>NT:
 C:/Program Files/Apache Group/Apache/logs/error.log</p> <p>UNIX:
 \$ps_home/webserv/logs/error.log</p> |
| enableNewWindow | <p>Controls whether or not the user can start a new window, using the New Window link. True enables the option, whereas False disables it.</p> |

| Variable | Description |
|-----------------------------------|---|
| breadcrumbsript | Defines the iScript used for breadcrumb navigation. It is not recommended that you change this parameter. |
| startpagescript | Defines the iScript used for starting a new window (if applicable). |
| Security Settings | |
| sessionTimeout=1200 | <p>Meta refresh tag in seconds. It should be less than or equal to the session.timeout for the servlet. For Apache/Jserv, it is defined in servlets/zone.properties file. The default is 1200 (20 minutes)</p> <p>Example usage:</p> <pre><meta HTTP-EQUIV='Refresh' CONTENT=' 1200; URL=http://SERVER060800/servlets/clientervlet/peoplesoft8/?cmd=expire' ></pre> |
| expire_page=expire.html | <p>The expire page is the HTML page containing text variables defined in text.properties. It appears when user inactivity exceeds the sessionTimeout limit.</p> <p>Do not change this setting.</p> |
| Cookiesrequired_page= | A page containing text variables defined in text.properties. It is displayed when the browser does not accept cookies. You should configure browsers to accept cookies. |
| SSLRequired=false | If this entire web site requires the SSL protocol, set this to true to enforce it. This prevents the users from using non SSL protocol to access any link within this web site or application. If only some pages require SSL access, leave this parameter set to false. |
| sslrequired_page=sslrequired.html | <p>This is the page that appears when the SSLRequired is set to true and the user is unable to proceed without SSL.</p> <p>Do not change this setting.</p> |
| byPassSignOn=false | If set to true, the system does not prompt the user to sign on during a direct link/page access. In this case, the system authenticates the user by defaultUSERID and defaultPWD. It is commonly used for informational web sites where sensitive data is not accessible. |
| defaultUSERID= | <p>Used to connect users when the byPassSignOn is set to true. For example,</p> <pre>defaultOPERID=PTDMO</pre> |
| defaultPWD= | <p>Used to connect users when the byPassSignOn is set to true. For example,</p> <pre>defaultTPWD=PTDMO</pre> |

| Variable | Description |
|------------------------------|--|
| defaultXMLLinkUSERID | User ID that is used to connect users for XML Link. |
| defaultXMLLinkPWD | Password used to connect users for XML Link. |
| disableSingleSignon= | Set to true or false. This parameter causes the server not to create a single signon cookie. |
| Portal Settings | |
| portalname=PORTAL | Name of the portal installed in the database. The default is PORTAL.

Do not change this setting. |
| AuthTokenDomain= | Specifies the domain for which the single-signon authentication token is valid. If you require a domain name here, you must also qualify the pswebservername= parameter with the same domain name.

For example, if the cookie is shared on web servers foo.peoplesoft.com and bar.peoplesoft.com, the property should be:

AuthTokenDomain=. peoplesoft.com |
| #PortalHTTPPort= | If you are using HTTPS, and your HTTP server is using a port other than 80, then uncomment the PortalHTTPPort setting and set PortalHTTPPort equal to the appropriate HTTP Port number. |
| #PortalHTTPSPort= | If you are using HTTPS, and your HTTPS server is using a port other than 443, then uncomment the PortalHTTPSPort setting and set PortalHTTPSPort equal to the appropriate HTTPS Port number. |
| customHeaders= | <i>Not currently implemented.</i> |
| Miscellaneous | |
| defaultContentType=text/html | This can be changed for a mobile application that uses HDML. |
| noProcessingWait=false | If set to false, this parameter enables Processing notification. |
| singleThreadNS= | If set to true, the requests from a Netscape browser are single-threaded to so that the browser handles the transaction with better performance. |
| ThreadDelay= | Specifies a delay, in milliseconds, for single-threaded requests. Used only when singleThreadNS is set to true. |
| defaultScheme= | Used to overwrite the scheme from request object "http" or "https". |
| defaultPort= | Used to overwrite the port from the request object. |

| Variable | Description |
|---------------------------------------|---|
| portalUseHttpForSameServer | Set this property to true when you wish to use the http protocol instead of https, for requests issued by the portal for content hosted on the same server as the portal servlet. You will need to set this property to true when using a hardware SSL accelerator. |
| Performance | |
| compressResponse=false | If set to true, this enables compression in the communication between the web server and the browser. Gzip and Compress are supported. |
| portalCompressResponse=false | If set to true, this enables compression in the communication between the web server and the browser. Gzip and Compress are supported. |
| portalAcceptCompressedEncodings=false | If set to true, enables the portal to accept compressed responses from the target components and template components it uses to assemble portal pages. If this setting is set to true and the compressResponse is set to true, PIA's responses to the portal are compressed. This setting is useful when PIA and the Portal are on separate web servers to reduce the size of the network transmission. |
| portalServletSessionCookieName= | The portal needs to know the name of the cookie used to store the session id. It normally automatically passes along any cookie that has the same value as the session id. If this is problematic for some reason, the cookie name can be specified here instead by setting the portalServletSessionCookieName to the name of the cookie storing the servlet session id. |



The noCache parameter should be set to true for public space machines, such as kiosks, where multiple users gain access to a computer. All users should be warned to log out. With noCache set to true subsequent users are not able to use the Back button to view the previous users' cached pages.

pstools.properties

The following table contains the variables and descriptions of the items that appear in the pstools.properties file.

| Variable | Description |
|------------------------|--------------------|
| Tuxedo Settings | |

| | |
|-------------------------------------|---|
| tuxedo_network_disconnect_timeout=0 | The amount of the time to wait and while disconnecting the Jolt connection. Entering 0 means no limit. |
| tuxedo_send_timeout=50 | The send timeout indicates the maximum number of seconds that the servlet allows for a request to be sent to the application server. This setting <i>does not</i> indicate a maximum amount of time for the service to complete; it only indicates the maximum amount of time to send the request to the application server. |
| tuxedo_receive_timeout=600 | The receive timeout indicates the maximum number of seconds that the servlet waits for a response from the application server. If you increase your application server service timeouts, such as the Service Timeout setting for PSAPPSRV, then increase the tuxedo_receive_timeout parameter to be greater than the Service Timeout values that appear in the PSAPPSRV.CFG configuration file on the application server. |
| Locale Settings | |
| locale_ | The locale_ settings relate to date and time values for globalization. Typically, you do not need to change these values. If you do, refer to the comments in the pstools.properties file. |

Setting Timeout Intervals

Because there are numerous components involved when deploying PeopleSoft applications to a browser transaction, setting timeout intervals involves setting configuration parameters at multiple levels. For instance, you need to set the timeout values at both the web server and the application server level.

The topics in this section describe the following information for setting the timeout values at each level, from the browser to the database:

- The configuration file storing the values.
- The parameter controlling timeout.
- The recommended value.

PeopleSoft provides recommended settings based on our internal testing. After you perform load testing at your site, you may need to adjust the recommended settings.



Note: The only value that needs to be changed from the default value, according to the PeopleSoft recommendations, is the Apache timeout.

Browser

You don't set a timeout value on the browser. The web server controls the timeout for the browser.

Apache

The timeout directive (HTTP GET, POST and ACK) for the Apache web services are stored in the httpd.conf file. The parameter is Timeout. The default value is 300. PeopleSoft recommends a value of 600.

```
Timeout=600
```



Values appear in seconds.

This setting is global for the entire Apache installation.

The Apache timeout generates the following error in the browser when a user exceeds the timeout limit.

```
"The page cannot be displayed"
```

This error appears even when the application server is processing a request.

JServ

The timeout values for the JServ component are stored in the zone.properties file.

The relevant parameters and recommended values are as follows:

```
init.timeout=10000  
destroy.timeout=10000  
session.timeout=1800000  
session.checkFrequency=30000
```



Values appear in milliseconds.

This value is global for JServ installation.

When the user exceeds the session.timeout, the following message appears on the browser:

```
"Your preferred language is not registered with the server's pstools.properties  
file..."
```

JRE

JRE enforces no timeouts.

Servlet

At the servlet level, the timeout values are stored in the configuration.properties file. The parameter is sessionTimeout, and PeopleSoft recommends a value of 1200.

```
sessionTimeout=1200
```



Values appear in seconds.

You should use this parameter to set an "idle session" timeout limit.

When the user exceeds the sessionTimeout value, the browser refreshes and displays the following error.

Either your connection has expired or your browser does not accept "cookies".

For increased security on this site, connections are expired after a delay of minutes. If you
would like to perform further transactions please
[Return to the Sign On Section](#)

Also at the servlet level, timeouts are stored in the pstools.properties file. The following parameters control timeout.

```
tuxedo_network_disconnect_timeout=0
```

```
tuxedo_send_timeout=50
```

```
tuxedo_receive_timeout=600
```



Values appear in seconds.

PeopleSoft recommends setting the tuxedo_receive_timeout to 600. At the very least it should be greater than the PSAPPSRV service timeouts.

If the user exceeds the tuxedo_receive_timeout value, the browser displays the following error:

```
"bea.jolt.ServiceException:
bea.jolt.JoltRemoteService(ICQuery) call()\nbea.jolt.SessionExceptio:

Connection recv error\nbea.jolt.JoltException: [1] NwHdlr.recv(): Timeout Error"
```

Jolt

The timeout values for Jolt appear in psapsrv.cfg, which you can edit directly or by using PSADMN. The following parameters appear in the Jolt Listener section.

```
Client Cleanup Timeout=60
```

```
Init Timeout=5
```



The Client Cleanup Timeout value is in minutes while the Init Timeout value is in seconds.

Application Server (TUXEDO service)

The timeout values for the application server appear in `psapsrv.cfg`, which you can edit directly or by using PSADMN. The timeout parameter appears in the PSAPPSRV section. PSAPPSRV is the main server process that handles the PIA transaction requests.

```
Service Timeout=300
```



Values appear in seconds.

This value should not be greater than the Apache timeout value.

Database

To set the timeout values at the database level, you do so in Permission Lists in Maintain Security. PeopleSoft recommends this to be set to "never timeout" for users logging in through PIA.

Setting Jolt Failover

To enable jolt failover and load balancing in the PeopleSoft Internet Architecture, you enter multiple application server domains in the `psserver` parameter in the `configuration.properties` file.

For example,

```
psserver=//SERVER1:9000, //SERVER2:9010, //SERVER3:9020
```

Linux Shell

You can use the Linux operating system for PIA web servers. If you use a Linux server, PeopleSoft requires that you use the ksh shell.



Important. Always check the PeopleSoft Platforms database on Customer Connection for information regarding the current support options. The appearance of the Linux reference in this document does not necessarily imply support for Linux and your particular PeopleTools version.

If version of Linux that you use is RedHat 6.2, ksh may not have been installed by default. To install it, mount the RedHat 6.2 CD-ROM and issue the following command to install ksh:

```
cd /mnt/cdrom; rpm -Uvh pdksh-5.2.14-2.i386.rpm
```

If you are not running RedHat 6.2, or if you prefer a more "UNIX-like" interface, the AT&T version of ksh is available at the following internet location:

<http://www.kornshell.com>

The build/install instructions for the AT&T version are available from the web site above. The AT&T ksh is similar to the ksh shell used on UNIX systems.

CHAPTER 9

Administration Considerations

This chapter offers a few additional topics that may be of interest to application server administrators.

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.

Overview

With single-signon and the portal, it may not be apparent to all end users just exactly what database or application 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 is having trouble accessing a particular application, the system administrator can instruct the user to provide the system information displayed in the help 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/5.0 |
| Operating System | WINNT |
| Tools Release | 8.12-D3 |
| Application Release | Core 8.12.00.000 |
| Service Pack | 0 |
| Page | ACTRSN_TBL_GBL |
| Component | ACTION_REASON_TBL |
| Menu | ADMINISTER_WORKFORCE_(GBL) |
| User ID | PTDMO |
| Database Name | TST812B |
| Database Type | MICROSFT |
| Application Server | //STNTDB03:8890 |

[continue](#)

System Information Help 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, as in Internet Explorer or Netscape. |
| Operating System | The operating system running 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 running on the application server, web server, or database server. |
| Tools Release | The version of PeopleTools that is currently installed at your site. For example, PeopleTools 8.12, 8.12.01, and so on. |
| Application Release | The version of PeopleSoft applications currently installed at your site. |
| Service Pack | Typically, updates to PeopleSoft applications arrive in the form of a Service Pack. This item shows the current Service Pack applied to your 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 currently user accessing PeopleSoft. |
| Database Name | The name of the database that the user is currently performing a transaction in. |

| <i>Item</i> | <i>Description</i> |
|--------------------|--|
| Database Type | The type of the current database, as in Microsoft, Oracle, DB2, and so on. |
| Application Server | The DNS name or IP Address and the JSL port number. |

Depending on your site's policy, you may not want the User ID, Database Name, Database Type, or Application Server information readily available. You use the Connectioninformation parameter in the configuration.properties file on the web server to determine what appears when a user or administrator presses CTRL+J.

If Connectioninformation is set to true, as in

```
Connectioninformation=true
```

then all information appears in the system help page. On the other hand, if you set this parameter to false the User ID, Database Name, Type, and Application Server name do not appear in the help page.



For more information on the configuration.properties file, see configuration.properties variables.

User Specifications

When an application server is booted, the User ID/Operator ID specified in the configuration file, PSAPPSRV.CFG must be authorized to start an application server. That authorization is provided by a setting in the STARTAPPSVR column in PSOPRDEFN—a value of 1 authorizes a user to start an application server. You grant this authorization through Maintain Security when creating permission lists. Just select the **Can start application server** checkbox.



For more information on the Allowed to start application server option, see Can Start Application Server?.

The authorization to start an application server does not (directly or indirectly) grant any authorizations or privileges beyond the ability to start application server. Each user who attempts to signon enters a unique User ID and Password. The application server uses these values to authenticate each user.

Setting Up Jolt Internet Relay

Jolt Internet Relay (JRLY) is a BEA product that is required for Web Client connections to an application server only in cases where the web server—containing PeopleSoft HTML and applets—is on a separate machine than the application server. If the web server is on the same

machine as the Application Server, JRLY is not required. PeopleSoft allows configurations where the web server and application server are on the same or separate machines.

Overview

Jolt Internet Relay consists of two components: Jolt Relay and Jolt Relay Adapter. It is important that you understand the difference between these two components.

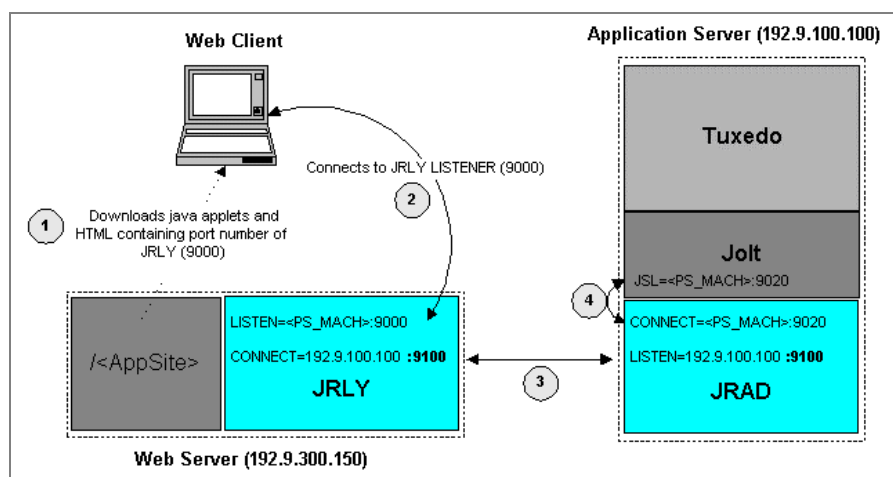
Jolt Relay (JRLY) consists of a stand-alone program and configuration file that runs on the same machine as the web server. JRLY receives Jolt messages from a PeopleSoft Web Client and routes those messages to the Jolt Relay Adapter on the application server. It receives the Jolt message through one port—the LISTEN port—and connects to the JRAD using another port—the CONNECT port. JRLY is sometimes referred to as a “Front-End Relay”.

Jolt Relay Adapter (JRAD) runs on the same machine as the application server. It is configured automatically on the application server domain as part of PeopleSoft's PSADMIN domain configuration procedure. JRAD listens for JRLY messages on its LISTENER port and transfers the message to the JSL or JSH. JRAD is sometimes referred to as a “Back-End Relay”.

The following example illustrates the relationship between the components and, most importantly, their respective port numbers. When you are configuring the Jolt Internet Relay system, it is very important to make sure that you specify the correct port numbers through which each component receives messages and to which port number they send messages. Any inconsistency will result in a failed connection.



Note. Using Jolt Internet Relay is intended for use with the Web Client (an internet solution from previous releases), however, you can use it with PIA too. Using Jolt Relay with PIA is *not* recommended for performance reasons. For use with PIA, you need to specify that the page servlet connect to the JRLY Listener port on the web server as opposed to specifying the JSL on the application server.



Jolt Internet Relay Port Numbers

In the example, assume that the web server and the application server reside on separate machines. The following list describes what takes place within each numbered step.

1. When the Web Client connects to a URL, PeopleSoft HTML and Java applets will be downloaded to the supported browser. Contained in the downloaded HTML is the port number used to connect to Jolt—in this case 9000.
2. After downloading the HTML, the Web Client reconnects to port 9000 on the web server machine. Port 9000 reflects the JRLY Listener. The JRLY Listener passes the Java message to the JRLY Connect process. For security reasons, the Web Client must always reconnect to the *same* machine from which it downloads the HTML. Keep in mind that the Web Client only reconnects to the same machine—not to the web server process on the machine.
3. The JRLY Connect process uses the machine IP Address and port number to connect to the Jolt Relay Adapter (JRAD) process on the application server machine.
4. Then JRAD passes the request on to the Jolt Station Listener, which initiates the transaction.

The return message to the Web Client follows the same path in reverse.

When you implement Jolt Internet Relay, the JRLY Listener must match the port number specified in the downloaded HTML. However, the Jolt Listener port on the application server may or *may not* match the port number specified in the HTML. For instance, in our example the JRLY Listener must be set to 9000 to match the port number specified in the HTML, but the JSL on the application server is set to 9020. The JSL could also be set to 9000 or any other valid port number. On the other hand, when the web server and application server are on the same machine and JRLY is not required, the JSL on the application server machine *must* match the port number specified in the downloaded HTML.



A firewall may separate (and probably does in most cases) the web server and the application server.

It's important to keep in mind that if you are planning to support Web Clients connecting through JRLY and Web Clients connecting *directly* to the JSL on the application server, the JRLY Listen port must equal the JSL port.

Summary:

- The JRLY Listener must match the port number specified in the HTML.
- JRLY Connect must match the JRAD Listener.
- JRAD Connect is set automatically by PeopleSoft to connect to the JSL.

Installing Jolt Internet Relay (JRLY)

This section contains the instructions for installing Jolt Internet Relay on UNIX and Windows NT. JRLY can connect to JRAD installed on any supported application server platform. For

information on the platforms on which JRLY is supported (by BEA), refer to BEA support documentation.

To install JRLY on UNIX

1. Log in as *root*, and create the UNIX group and the user-name of the individual who will be the owner of Jolt Relay.

Depending on your operating system, the utility you use to create user and group will be different. For example, HP-UX uses the *sam* utility, AIX uses the *smit* utility, and so on. For the exact utility you should use, refer to your operating system documentation.

2. Insert the CD-ROM into the CD-ROM drive, and mount the CD-ROM from the root login.
3. List the root directory on the CD-ROM.
4. Log in as the Tuxedo administrator.

You should no longer be logged on as *root*.

5. List the root directory on the CD-ROM, and change to the */jrelay* directory.
6. Execute the shell script, *install.sh*.

```
install.sh
```

7. You will be prompted for the following values for the installation of the Jolt Front-End product; enter the appropriate response for your site.

- You may install Jolt Front-End into any directory you choose.
- Select *1* to install BEA Jolt (it is your only choice).
- Select the menu option corresponding to your operating system and release. For example:

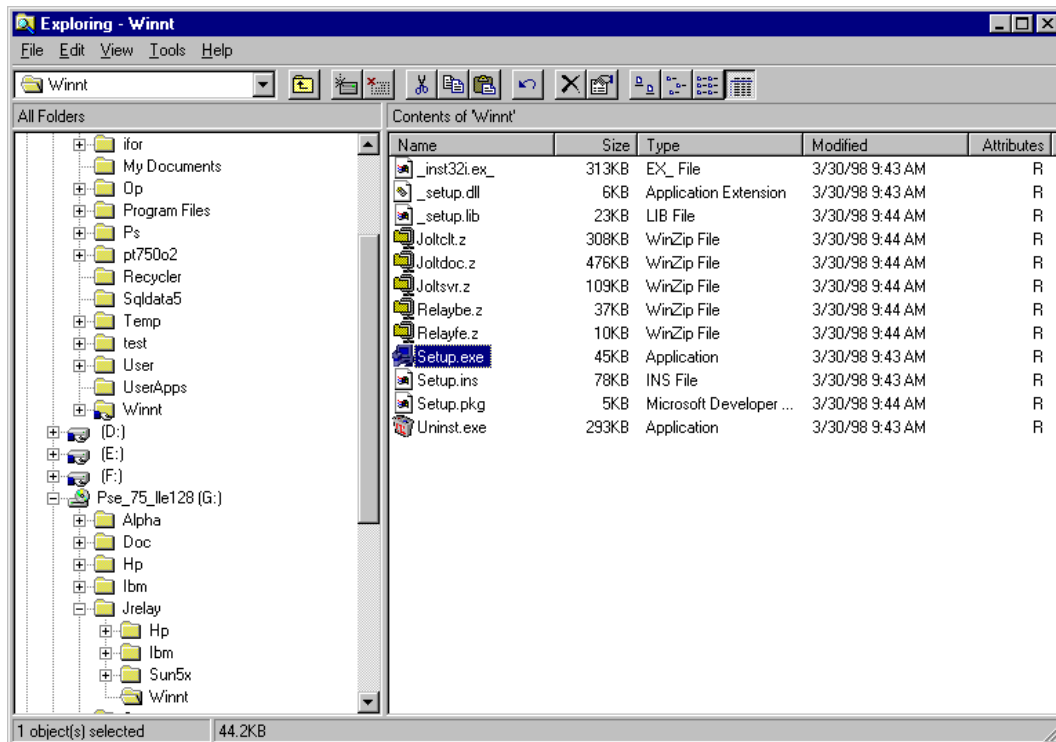
```
03= ibm/aix421/6.4
```

- Select *4* to install the Jolt Relay Front-End product.
8. Following the installation, examine the Jolt Relay installation directory.

Note that in *<joltrelay>/relay*, the following two files were installed: *jrly.config* and *jrly*. These are the only two files used in configuring and starting Jolt Relay and will be discussed in the next section.

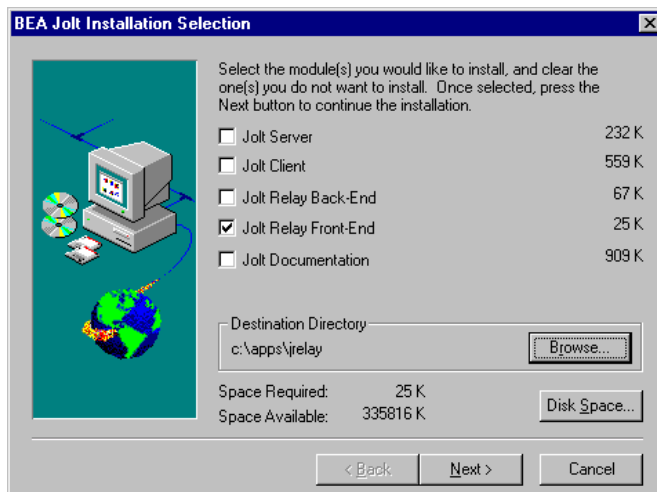
To install JRLY on Windows NT

1. Insert the CD-ROM into the CD-ROM drive.
2. Using Windows Explorer, navigate to the CD-ROM directory
3. Change directory to *Jrelay\winnt* and execute *setup.exe*



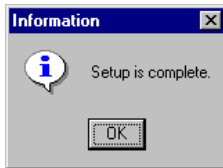
4. The installation program presents you with a series of dialogs, and, for the most part, they are intuitive, however make sure you make note of the following items:

- On the **BEA Jolt User Registration** dialog enter the appropriate information for your site.
- On the **Jolt Installation Selection** dialog, select the **Jolt Relay Front-End** check box and deselect all the other options. Also, use the **Browse** button to specify the desired installation directory.



BEA Jolt Installation Selection Dialog

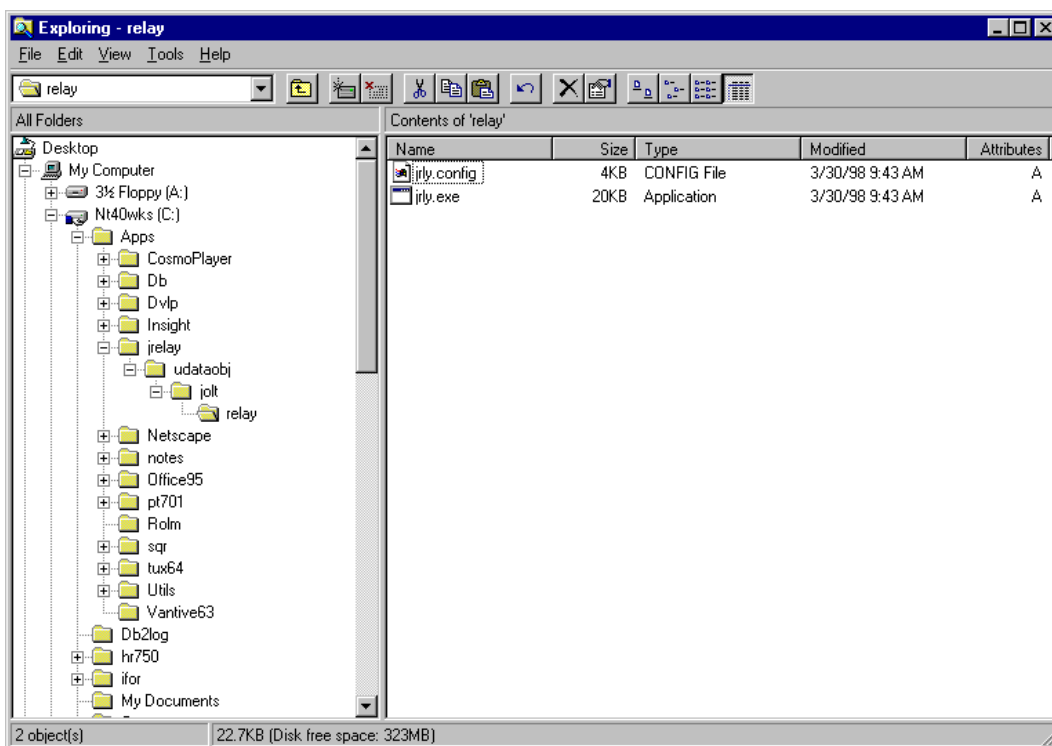
When the installation program completes you should see the following message:



Information Dialog

5. Using Windows Explorer, examine the Jolt Relay installation directory.

Note that in <joltrelay>\udataobj\jolt\relay, the following two files were installed: jrly.config and jrly.exe. These are the only two files used in configuring and starting Jolt Relay and are discussed in the following section.



Configuring Jolt Relay (Front-End) on UNIX or Windows NT

Configuring Jolt Relay is identical on UNIX and Windows NT.

To configure Jolt Relay

1. Go to <joltrelay>\relay and open JRLY.CONFIG using one of the following editors.
 - On UNIX edit the configuration file using VI or an equivalent editor.
 - On Windows NT, you must edit the configuration file using write.exe (WordPad). Notepad and other editors will format the file improperly.

2. Modify the parameters in the configuration file to reflect your site specifications.

Use the following table for guidance.

| Parameter | Description |
|----------------------------|--|
| LOGDIR=/tmp | LOGDIR is the directory where JRLY will create access and error log files. This directory must exist—the JRLY program will not start if it cannot find this directory. The path specified for LOGDIR should be an absolute path (starting from / on UNIX systems, starting from <DRIVE>: on Windows NT systems). The JRLY will accept relative path names but then the LOGDIR will be relative to the directory from which the JRLY program is started. |
| ACCESS_LOG=JRLY_access_log | ACCESS_LOG is the file name where JRLY records access information. This file will be created in \$LOGDIR. If the file already exists the most recent information will be appended to it. This can be any valid file name. Everything after the equal sign (=) to the end of the line is treated as the file name, so an entry of ACCESS_LOG="access log" would create a file named "access log" (including the double quotes). Leading and trailing blanks are ignored after the equals sign (=). So an entry of ACCESS_LOG=access_log would create a file named <i>access_log</i> (without the double quotes). If the JRLY program cannot create the ACCESS_LOG file or open it for appending, the program will exit. |
| ERROR_LOG=JRLY_error_log | ERROR_LOG is the file where JRLY records error information. This file follows all the rules that apply to the ACCESS_LOG parameter. JRLY_error_log will be created in /tmp. |
| LISTEN=sp-ibm02:9000 | <p>The LISTEN key word specifies the host and port on the current machine (i.e. the machine where you are installing Jolt Relay). JRLY will listen for client connections. The following formats are acceptable:</p> <pre> LI STEN=192. 9. 100. 100: 9000 LI STEN=//192. 9. 100. 100: 9000 LI STEN=sp-i bm02: 9000 LI STEN=//sp-i bm02: 9000 </pre> <p>Specify the port number in decimal; it must match the port number specified in the HTML. If a machine has multiple network interfaces, you should use the IP Address notation, since</p> |

CONNECT=192.9.100.100:9100

specifying the hostname could be ambiguous (OS dependent result). If the JRLY program cannot establish a network listening end-point at the host:port specified, it will print an error and exit. (The hostname specified for LISTEN must be the name of the host on which the program is running)

The CONNECT key word specifies the location of the Jolt Relay Adapter (JRAD) machine and process port on the application server machine to which the JRLY program connects. A JRLY only communicates with a single JRAD. The address specified in the JRLY connect parameter must match the JRAD listener address on the application server machine. (Check the PSAPPSRV.CFG file in <PS_HOME>/appserv/<domain> directory.) The JRAD does not have to be running when the JRLY is started. The JRLY will attempt to connect to the JRAD when it first starts, and if the JRAD is not available, the JRLY will try again whenever a new client connects to the JRLY. You can use the following format:

CONNECT=192. 9. 100. 100: 9100

CONNECT=//207. 135. 44. 91: 9105

CONNECT=sp-hp06: 9105

CONNECT=//sp-hp06: 9105

PeopleSoft has found that formats are operating system and environment dependent. If one fails to connect to the application server, try another format.

Configuring Jolt Relay Adapter (JRAD)

Jolt Relay's Connect port will connect to Jolt Relay Adapter's Listener Port specified on the application server machine. JRAD then routes the message to Jolt—either the JSL for initial connection from a web client or to the JSH for all subsequent connections from a web client. The return message will follow the same path in reverse.



Jolt Relay's Connect port must match JRAD's Listener port.

To configure JRAD

1. Launch PSADMIN, and navigate to the PeopleSoft Domain Administration menu and select

4) Configure this domain.

2. Follow the prompts until you reach the Jolt Relay Adapter section.

Values for config section - JOLT Relay Adapter

```
Listener Address=%PS_MACH%
```

```
Listener Port=9100
```

Do you want to change any values (y/n)? [n]:

3. Enter the appropriate port number for the Listener Port.



The Listener Port must match Jolt Relay Connect port.

4. Later in the series of PSADMIN prompts you will need to specify that you want JRAD configured; enter *y* to do so.

Do you want JRAD configured (y/n)? [n]:y

You must specify *YES* to this prompt if you plan to implement JRAD. This will start the JRAD Listener service each time you boot the application server. Even if you manually edit the PSAPPSRV.CFG file, you still need to launch PSADMIN, specify that you do not want to make any changes to the configuration parameters, and enter *y* at this prompt.

Starting Jolt Relay

To start Jolt Relay follow the appropriate procedure for your operating system.

To start Jolt Relay on Windows NT

1. Change directories to the Jolt Relay directory.

```
cd \apps\jrelay
```

2. Enter the following command:

```
JRLY.EXE -f JRLY.CONFIG
```

To start Jolt Relay on UNIX

1. Change directories to the Jolt Relay directory

```
cd /apps/jrelay
```

2. Enter the following command:

```
jrly -f jrly.config &
```



The & causes JRLY to run in the background.

Stopping Jolt Relay

To shutdown Jolt Relay on UNIX, use the UNIX kill -9 command.

To shutdown Jolt Relay on Windows NT, use Task Manager.

Jolt Relay Notes

The following list provides additional information that you will want to keep in mind as you configure your Jolt Internet Relay components.

- JRLY.EXE and its corresponding JRLY.CONFIG file must exist in the same directory. If you want to start multiple Jolt Relays on a machine, copy JRLY.EXE and JRLY.CONFIG into each subdirectory, modify the parameters in the JRLY.CONFIG file, and start Jolt Relay.
- You can start the JRLY process before or after you start JRAD. The JRLY will attempt 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.
- If you are installing Jolt Relay on UNIX and anticipate a large number of concurrent connected clients, we recommend increasing the file descriptors limit before running the JRLY executable.
- At runtime, if you get the following message:

```
[Fri Jun 6 20:25:11 1997] JRLY:accept():accept failed, err no: 23, strerror:  
File table overflow
```

We recommend that you increase the MaxUSERS kernel parameter and regenerate the kernel.

- If you are unable to connect, here are some common items that need double-checking:
 - Port numbers do not match. Print out the JRLY.CFG file and the PSAPPSRV.CFG file and compare the port numbers you have specified.
 - Make sure the application server is running.
 - Make sure JRLY is running.
 - Make sure the port numbers in the HTML are correct.

Make sure that JRAD is running on the application server and that in the PSADMIN interface you have opted to configure JRAD.

Development Environment

This section explains the client side of the PeopleSoft architecture. Specifically, we discuss the Windows Client and the PIA configuration. The first chapter of this book describes the high-level architecture that comprises your PeopleSoft system. In this chapter, we provide more detail regarding client connections.

Keep in mind that your *Installation and Administration* PeopleBook contains the information required to install and configure your Windows Clients and PIA configuration.

Windows Client

The Windows Client refers to the traditional PeopleSoft Windows Client that offers full-functionality and desktop integration with other Windows applications, such as Excel, Word, or your electronic mail software.

You can configure Windows Clients to use a two-tier connection to a database, a three-tier connection to an application server, or both. A Windows Client using a three-tier connection needs to meet the following requirements:

- Supported Windows version.
- TCP/IP connection.
- Access to the PeopleSoft client bin directory on the file server. PeopleSoft embeds Tuxedo client binaries in its client bin directory to eliminate the need for users to install Tuxedo client software on their machines.

The following sections provide additional considerations and tips to keep in mind when administering Windows Clients.



For more information on setting up workstations, see your *Installation and Administration* manual.

Specifying Application Server Names

Application server name(s) are defined in the client's Configuration Manager on the **Application Servers** tab within the **Profiles** section. The application server name is a user-defined name associated with the IP Address of the application server machine and the port number used by a particular application server domain. The application server name is not case sensitive and does not have to match the application server name defined in the application server machine's PSAPPSRV.CFG file.

Machine Host Name Versus IP Dotted Notation

You may specify a resolvable machine host name (DNS name) or IP dotted notation in the PeopleSoft Configuration Manager when setting up a client's application server name. Users can also enter the address directly on the signon dialog. Use the following format.

- 127.09.09.01:8000
- sp-ibm02:8000



No special characters are needed when specifying a machine host name.

How to Determine if Connected in Three-Tier Mode

You can determine if a client workstation is connected in two-tier or three-tier mode in the following ways.

Help About

In PeopleSoft, select **PeopleTools Help, About PeopleTools**. Under **Application Server**, if the user is connected in three-tier, it will show the application server name to which it is connected, as shown below:

Application Release

Core 8.0.00.00

Operator Id

VP1

Database Name

PS80DB01

Database Type

ORACLE

Application Server

APPSVR1



It will show N/A for **Application Server** if you are logged on in two-tier.

Hourglass

Clients connected in three-tier will always see an hourglass rather than the SQL lightning bolt while waiting for a transaction to complete.

Remote Call

Remote Call is an application server service that initiates a COBOL program to run a transaction.

If connected in three-tier, remote call will run on the application server. If connected in two-tier, remote call will run on the client workstation. So, if you want to run locally, you must run in two-tier mode.

Configuration parameters are in the PSAPPSRV.CFG file in your domain directory in the [RemoteCall] section:

- RCCBL Redirect=0
- Note that the NT Registry is not used to obtain parameter values

The RCCBL Redirect=0 value is the default and should always be disabled except for troubleshooting. For troubleshooting, set RCCBL Redirect=1, then look in the “LOGS” directory for files with “_in” and “_out” as part of the file names.

The NT Application Server automatically sets environment variable COBSW to +L1,+S5. The +L1 suppresses an “Application Completed” pop-up window following a successful COBOL job. The +S5 redirects COBOL output to files.



When running three-tier, nothing on the workstation client needs to be configured for RemoteCall.

Crystal and nVision in Three-Tier Configurations

Both Crystal and N-vision procedures can run in three-tier mode. This means that the client sends SQL calls to the application server, and the results are returned to the client in a message. Unlike two-tier, where Crystal and nVision execute on the client machine and execute SQL directly against a database server, in three-tier these programs run on the application server.

You can use Crystal in two-tier or three-tier mode, but Crystal will run on the client unless you schedule it to run with the Process Scheduler (PSNT).

Since Crystal is sending SQL to the application server, it does not really leverage the performance benefits of the three-tier configuration. The ODBC driver, required for the Crystal interface, does not need to reside on the workstation, which reduces some administration.

In some cases, Crystal performance may be somewhat improved over a WAN, because when sending messages over a WAN, Tuxedo utilizes packet compression and minimizes the amount of

round trips. In general, though, Crystal will perform about the same in a three-tier connection as it does in a two-tier connection.

Timeouts

The following table describes the various timeout facilities available to manage client transactions.

Client Connection

A Client Connection timeout is time the Tuxedo server will allow for the client workstation to connect to the application server, starting with the moment the client request is received by the WSL. Set using the “Init Timeout” parameter in PSADMIN’s configuration. If set to 6, this means that Tuxedo will allow a workstation client this value times the SCANUNIT time (defined in the PSAPPSRV.UBX file) to make a client connection to the application server. The application server must be reconfigured in order for new values to take effect. For example, if Init Timeout is 6, and SCANUNIT is 10, the Tuxedo server will allow 60 seconds for the client workstation to connect.

Wait in Queue

This is the time that a client request can remain in the queue waiting for a server process such as PSAPPSRV or PSQCKSRV to become available. Set using the BLOCKTIME * SCANUNIT parameter values in the PSAPPSRV.UBX file. The application server must be reconfigured in order for new values to take effect. For example, if BLOCKTIME is 6000 and SCANUNIT is 10, a client workstation request will remain queued for 60000 seconds.

Maximum In-Database Time

This value must be set using whatever facilities are available in the RDBMS.

Debugging PIA Applications: Set up the PeopleCode Debugger

When you develop applications for PIA, you need to debug in the same environment in which end users will access your application. This means that you need to have an application server configured and running on your local machine, and you also need to have the PeopleSoft Debug Broker Class (DbgBrkr Class) configured for three-tier/PIA debugging. The PeopleSoft software must also be installed on the local machine, and not on a network drive.

The following sections describe the details associated with each of these tasks.

Setting up the Debug Broker Class and Debug User Account

The following procedure involves the following fundamental tasks:

- Creating a user account for the Debug Broker process.

- Cleaning up previous COM registrations using Configuration Manager.
- Registering the COM components by launching Application Designer with a two-tier connection.
- Configuring the Distributed COM Configuration Properties for the Debug Broker class.



Note. If you are accessing the Application Designer in two-tier mode, the following instructions do not apply.

To set up the PIA/three-tier debugging environment

1. Create a User ID for the Debug Broker process to use.

Go to **Start, Programs, Administrative Tools (Common), User Manager.**

2. Add a new user.

Select **User, New User.**

3. Fill out the information for the new user.

- Enter *PSDBGBRKR* for **Username**.
- Enter *PSFT - Debug Broker* for **Full Name**.
- Leave the **Password** and **Confirm Password** field blank.
- Deselect **User Must Change Password at Next Logon**.
- Select **Password Never Expires**.

New User

Username:

Full Name:

Description:

Password:

Confirm Password:

☐ User Must Change Password at Next Logon

☐ User Cannot Change Password

☒ Password Never Expires

☐ Account Disabled

New User dialog box

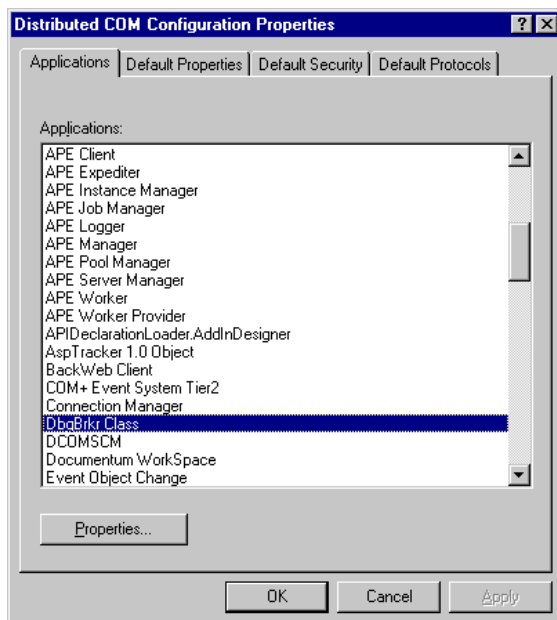
- Click **OK**, and exit User Manager.
- 4. Clean up the Distributed COM registration for the Debug Broker.
- Open the Configuration Manager.
- Go to the Client Setup tab.
- Click Unregister Com Components.
- Click **OK** to close the Configuration Manager.
- 5. Sign on to PeopleTools with a *two-tier* connection and open Application Designer and then sign off.

This registers the COM services for the debug broker.

6. Run the Distributed COM Configuration program.

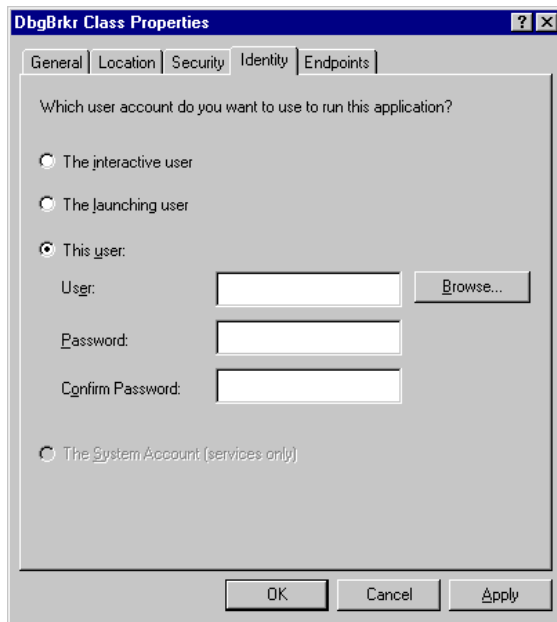
Go to **Start, Run**, and type in **DCOMCNFG**.

7. On the Distributed COM Configuration Properties dialog, select the **Applications** tab, find and select *DbgBrkr Class*, and click the **Properties**.



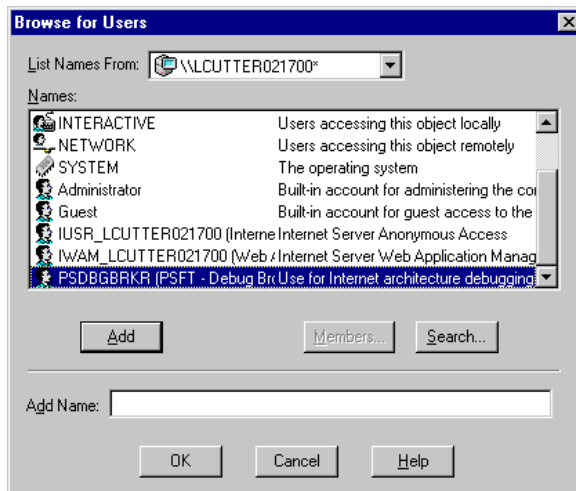
Distributed COM Configuration Properties dialog box

8. On the DbgBrkr Class Properties dialog, select the **Identity** tab, click **This user** and click **Browse**.



DbgBrkr Class Properties dialog box, Identity tab

9. From the **List Names From** dropdown list, select your machine name.



Browse for Users dialog box

10. Select user PSDBGBRKR, click **Add**, and then click **OK**.

This returns you to the DbgBrkr Class Properties, Identity tab.

11. Click **Apply** and **OK**.

The Debug Broker is now set up for the PeopleSoft Internet Architecture and three-tier debugging. You also need to configure the application server for three-tier debugging, which is covered in the following section.

Setting up the Application Server for Debugging

To set up your application server for PeopleCode debugging, make note of the following requirements for the application server:

- The application server must be running on the same computer as PeopleTools *and* the debugger. While Application Designer and the application server must be on the same computer, you can run the browser either from a different computer or the same computer.
- The application server domain requires at least two PSAPPSRV server processes.
- The application server must have Enable Debugging turned on in PSADMIN.



Windows Workstation You can set up a dual machine/debugger configuration. One computer can run the Windows client and a debugger for it, while another computer can run the application server and a debugger for it.

Configuration Templates

In order for the debugger to work, it has to run in parallel with the application it's debugging. Suppose your domain only has 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.

However, provided that you have two PSAPPSRV server processes running, one PSAPPSRV handles the debugger program while the other handles the application you're stepping through with the debugger. In this case, the two programs run in parallel and provide interactive debugging.

The delivered configuration templates have at least two PSAPPSRV processes. However, if you are using a custom template, make sure 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. For example,

```
Min Instances=2
```

It is also important to set the Service Timeout parameter for PSAPPSRV to zero. For example,

```
Service Timeout=0
```

Disabling service timeouts prevents the application server processes from timing out if you stop at a particular point in your program while debugging.



PeopleSoft recommends using the Developer configuration template because this template, by default, provides two PSAPPSRV server processes, has Enable Debugging activated, and has Service Timeout set to zero.

PSADMIN Settings

To debug your PeopleCode on the application server you need to enable debugging on the application server using PSADMIN.

The parameter that controls debugging is called Enable Debugging and you find it in the Domain Settings section of PSADMIN. To put the domain in debugging mode, set Enable Debugging to 1. Set it to 0 to disable debugging mode.



For more information on the Enable Debugging, see Domain Settings.

Keep in mind, that you also want to set the Min Instances of the PSAPPSRV to 2, and the Service Timeouts to zero.

When not debugging, the Enable Debugging should be turned off (set to 0). The debugging mode brings an unavoidable amount of overhead, and, due to the overhead, system performance is affected while you are debugging.



Keep in mind, that you should not be performing any debugging on a production domain. Debugging should *only* be performed on a designated testing domain.

UNIX Options

For the UNIX application server environment, you have the following options for debugging your PeopleCode.

- Because a COM connection between the Application Designer and the application server is required for PeopleCode debugging, interactive debugging is restricted to the Windows NT application server. If you are using UNIX, you can set up a temporary Windows NT application server to connect to your database for interactive debugging. Because all PeopleCode resides in your database and executes identically on UNIX and Windows NT, this solution provides ample debugging for the UNIX environment.
- Turn on the PSADMIN TracePC flag. This trace option records all the PeopleCode the system executes and writes it to the trace file.



For more information on the application server trace settings, see Trace.

Index

A

- application messaging gateway servlet 1-5
- application server 1-3, 1-6
 - server processes 1-10
- Application Server
 - administering a domain 3-1, 3-2
 - basic concepts 1-6
 - batch server environment 1-12
 - booting a domain 3-3
 - checking domain status 3-4
 - components 1-12
 - configuration files 2-19
 - configuration templates 2-9
 - configuring a domain 3-7
 - configuring web components 6-1
 - creating a domain 3-11
 - database connectivity software 1-12
 - deleting a domain 3-12
 - editing configuration files 3-9
 - executables 2-19
 - handlers 1-7
 - Jolt 1-5, 1-12
 - listeners 1-7
 - menus 2-6
 - Process Scheduler 1-13
 - processes 1-7
 - PSADMIN 1-6, 2-1, 3-1
 - quick configure 2-17
 - servers and services 1-11
 - shutting down a domain 3-4
 - Tuxedo 1-12
 - User IDs 9-3
 - viewing log files 3-9
- Application Server Domain 1-19
 - defined 1-19
- APPSRV.LOG
 - editing 3-11
- architecture
 - batch server support 1-13
 - logical 1-14
 - physical 1-14
- Architecture 1-1
 - application server components 1-12
 - browsers 1-4
 - database server 1-14
 - terminology 1-18
 - web server 1-4

B

- batch server 1-3
- browser 1-2

C

- Caching
 - application server 4-16
- Can start application server 9-3
- CD-ROM
 - ordering ii
- Command Line
 - Process Scheduler 5-9
 - PSADMIN 2-10
 - commands 2-10
 - creating domains 2-11
 - syntax 2-10
- components
 - server 1-4
- configuration files
 - web server 8-6
 - web server (non-PeopleSoft) 8-7
 - web server (PeopleSoft) 8-6
- Configuration Manager
 - specifying application server names 9-13
- Configuration Templates 2-9

D

- Database Options
 - PSADMIN 4-3
- database server 1-3
- Debugging
 - enabling (three-tier) 4-12
- Domain
 - parameter reference 4-1
 - parameters 4-1
 - settings 4-10
- domains 1-6
- Domains
 - booting 2-23
 - configuring 2-20, 3-7
 - creating 2-11
 - creating from command line 2-16
 - loading configuration 2-22

quick configure 2-17
stopping 2-23

F

failover
Jolt 8-19
Forced shutdown 3-4

H

Handlers 1-7
httpd.conf 8-7

I

interface driver 4-25
Internet Client Servlet 1-16
defined 1-20

J

java executables 8-1
Java Message Agent Client 8-3
JMAC 8-3
jolt
failover 8-19
Jolt
back end 9-10
compression threshold 4-8
defined 1-19
front end 9-8
Internet Relay
installing 9-5

setting up 9-3

starting 9-11

stopping 9-12

UNIX 9-6

Windows NT 9-6

JOLT Listener 4-7
Jolt Relay Adapter 9-10
JOLT Relay Adapter 4-9
Jolt Station Handler 1-9
Jolt Station Listener 1-9
JRAD 4-9, 9-10
configuring 9-10
JRLY 9-4
JSH 1-9
JSL 1-9

L

Listeners 1-7

M

Messaging Server 4-23
processes 4-23

N

Normal shutdown 3-4
NT Services
for PeopleSoft 7-1. *See also* Service Setup

P

page servlet 1-5, 8-3
PeopleBooks
CD-ROM, ordering ii
printed, ordering iii
PeopleCode Debugger
three-tier debugging 9-20
PeopleSoft Architecture 1-1
browser 1-2
components 1-1
PeopleSoft Internet Architecture 1-1
PeopleSoft Portal 1-17
PIA 1-1
batch server environment 1-12
deployment options 1-14
setting timeout intervals 8-16
transmitting requests 8-4
web server administration 8-1
portal servlet 1-5, 8-3
Process Scheduler
command line options 5-9
configuring 5-1
menus 2-7
on application server 1-13
PSADMIN menu 5-1
Process Scheduler Server
configuring 5-5
creating configuration 5-6
deleting 5-7
modifying 5-8
starting 5-3
command line 5-9

stopping 5-4
command line 5-9

PSADMIN
administering a domain 3-2
application server menu 3-1

- booting a domain 2-23
 - booting down a domain 3-3
 - cache settings 4-16
 - checking domain status 3-4
 - command line options 2-10
 - valid values 2-14
 - Config 6-1
 - configuring a domain 3-7
 - configuring an application server 2-1
 - creating a domain 3-11
 - database options 4-3
 - defined 1-20
 - deleting a domain 3-12
 - domain settings 4-10
 - editing configuration files 3-9
 - interface 2-2
 - Jolt Listener section 4-7
 - JOLT Relay Adapter section 4-9
 - loading configuration 2-22
 - menu structure 2-5
 - navigating 2-4
 - overview 2-1
 - Parameters 4-1
 - Process Scheduler menu 5-1
 - prompts 4-28
 - PSAPPSRV 4-17
 - PSQCKSRV 4-20
 - PSQRYSRV 4-21
 - PSSAMSRV 4-19
 - PSTOOLS 4-25
 - Reference 4-1
 - Remote Call section 4-17
 - Security 4-4
 - selecting options 2-3
 - shutting down a domain 3-4
 - SMTP settings 4-23
 - starting 2-1
 - Startup 4-1
 - stopping a domain 2-23
 - Trace section 4-12
 - using 2-2
 - viewing log files 3-9
 - workstation listener section 4-5
 - PSADMIN.EXE 2-19
 - PSAPPSRV 1-11, 1-20
 - configuring 4-17
 - PSAPPSRV.CFG 2-12, 2-19
 - editing 3-10
 - PSAPPSRV.ENV 2-20
 - PSAPPSRV.PSX 2-20
 - PSAPPSRV.UBB 2-19
 - PSAPPSRV.UBX 2-19
 - PSAPPSRV.VAL 2-20
 - psapsrv.cfg 8-7
 - PSNTRV.CFG 7-8
 - PSPRCS.CFG
 - editing manually 5-8
 - modifying 5-5
 - PSQCKSRV 1-11, 1-20
 - configuring 4-20
 - PSQRYSRV 1-11, 1-20, 4-21
 - PSSAMSRV 1-11, 1-20
 - configuring 4-19
 - psserver 8-19
 - psstarter.html 6-8
 - PSTOOLS 4-25
 - pstools.properties 8-7
 - PSTUXCFG 2-20
 - pswebgui.ini
 - editing 6-5
 - PSWEBSRV.INI 6-3, 6-10
- ## Q
- Quick Configure 2-17
- ## R
- Remote Call 4-17, 9-15
 - workstation 9-15
 - report repository servlet 1-5
- ## S
- SCP_LOCALE 4-25
 - Security
 - PSADMIN 4-4
 - Server Process 1-20
 - Server Processes 1-7
 - Service Setup 7-1
 - application server 7-1
 - configuration 7-2
 - menus 2-8
 - overview 7-1
 - Process Scheduler 7-1
 - PSNTRV.CFG 7-8
 - reference 7-7
 - Service Start Delay 7-4
 - servlet engine 8-1
 - servlets 1-5
 - application messaging gateway 8-3
 - gateway servlet 8-3
 - multiple servlets 8-5
 - page servlet 8-3
 - PeopleSoft 8-2
 - portal servlet 8-3
 - report repository servlet 8-3
 - Servlets
 - Jolt 1-5
 - SMTP
 - application server settings 4-23
 - settings 4-23

Spawn Threshold 4-11
 STARTAPPSVR 9-3
 Starter Web Page 6-6
 Startup 4-1

T

Three-Tier
 windows client 9-13
 timeout intervals
 Apache 8-17
 application server 8-19
 Jolt 8-18
 JRE 8-17
 JServ 8-17
 servlets 8-18
 setting on the web server 8-16
 TMBOOT.EXE 2-19
 TMLoadCF.EXE 2-19
 tmshutdown 3-4
 tmshutdown -k TERM -c 3-4
 TMSHUTDOWN.EXE 2-19
 TraceAE 4-14
 TracePC 4-14
 TracePCMask 4-14
 TracePPR 4-15
 TracePPRMask 4-15
 TraceSQL 4-13
 TraceSQLMask 4-13
 Tracing
 PSADMIN 4-12
 Tuxedo 1-12
 servers and services 1-11, 1-21
 web server 6-1
 TUXLOG
 editing 3-11

U

UBBGEN.EXE 2-19

V

Validate Signon with Database 4-4

W

Web Components
 menus 2-8
 web server 1-3
 administration 8-1
 components 8-1
 configuration files 8-6
 transmitting requests 8-4
 web services 8-1
 WEBGUI 6-1
 Windows Client 9-13
 Windows NT Services
 for PeopleSoft 7-1. *See also* Service Setup
 wlisten process 6-5
 Workstation
 considerations 9-13
 determining connection type 9-14
 Development Environment 9-13
 three-tier
 Crystal 9-15
 nVision 9-15
 timeouts 9-16
 Workstation Handler 1-9
 Workstation Listener 1-9, 4-5
 WSH 1-9
 WSL 1-9