PeopleSoft PeopleTools 8.63

System and Server Administration

G47467-01
July 2026

PeopleSoft PeopleTools 8.63 System and Server Administration,

G47467-01

Contents

5 Using PSADMIN Menus

7 Working with Oracle WebLogic

8 Using PeopleSoft Configuration Manager

16 PeopleSoft Timeout Settings

17 Ensuring Session Stickiness

18 Troubleshooting Server Issues

19 Replicating an Installed Environment

20 Working with PSADMUTIL

21 Implementing MCP Server for Chat UI

Preface

Understanding the PeopleSoft Online Help and PeopleBooks

The PeopleSoft Online Help is a website that enables you to view all help content for PeopleSoft applications and PeopleTools. The help provides standard navigation and full-text searching, as well as context-sensitive online help for PeopleSoft users.

Hosted PeopleSoft Online Help

You can access the hosted PeopleSoft Online Help on the Oracle Help Center. The hosted PeopleSoft Online Help is updated on a regular schedule, ensuring that you have access to the most current documentation. This reduces the need to view separate documentation posts for application maintenance on My Oracle Support. The hosted PeopleSoft Online Help is available in English only.

To configure the context-sensitive help for your PeopleSoft applications to use the Oracle Help Center, see Configuring Context-Sensitive Help Using the Hosted Online Help.

Locally Installed PeopleSoft Online Help

If you're setting up an on-premises PeopleSoft environment, and your organization has firewall restrictions that prevent you from using the hosted PeopleSoft Online Help, you can install the online help locally. Installable PeopleSoft Online Help is made available with selected PeopleSoft Update Images and with PeopleTools releases for on-premises installations, through the Oracle Software Delivery Cloud.

Your installation documentation includes a chapter with instructions for how to install the online help for your business environment. The documentation zip file may contain a README.txt file with additional installation instructions. See *PeopleSoft 9.2 Application Installation* for your database platform, "Installing PeopleSoft Online Help."

Installable PeopleSoft Online Help now includes PDF versions of each PeopleBook. For more information, see Understanding Locally Installed PeopleSoft Online Help.

Downloadable PeopleBook PDF Files

You can access downloadable PDF versions of the help content in the traditional PeopleBook format on the Oracle Help Center. The content in the PeopleBook PDFs is the same as the content in the PeopleSoft Online Help, but it has a different structure and it does not include the interactive navigation features that are available in the online help.

Common Help Documentation

Common help documentation contains information that applies to multiple applications. The three main types of common help are:

- Application Fundamentals
- Using PeopleSoft Applications
- Using and Managing the PeopleSoft Online Help

Most product families provide a set of application fundamentals help topics that discuss essential information about the setup and design of your system. This information applies to many or all applications in the PeopleSoft product family. Whether you are implementing a single application, some combination of applications within the product family, or the entire product family, you should be familiar with the contents of the appropriate application fundamentals help. They provide the starting points for fundamental implementation tasks.

In addition, the *PeopleTools: Applications User's Guide* introduces you to the various elements of the PeopleSoft Pure Internet Architecture. It also explains how to use the navigational hierarchy, components, and pages to perform basic functions as you navigate through the system. While your application or implementation may differ, the topics in this user's guide provide general information about using PeopleSoft applications.

Finally, *Using and Managing the PeopleSoft Online Help* (this guide) introduces you to the setup, features, and usage of the online help system.

Field and Control Definitions

PeopleSoft documentation includes definitions for most fields and controls that appear on application pages. These definitions describe how to use a field or control, where populated values come from, the effects of selecting certain values, and so on. If a field or control is not defined, then it either requires no additional explanation or is documented in a common elements section earlier in the documentation. For example, the Date field rarely requires additional explanation and may not be defined in the documentation for some pages.

Typographical Conventions

The following table describes the typographical conventions that are used in the online help.

Typographical Convention	Description
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 the **W** key.
. . . (ellipses)	Indicate that the preceding item or series can be repeated any number of times in PeopleCode syntax.
{ } (curly braces)	Indicate a choice between two options in PeopleCode syntax. Options are separated by a pipe (|).
[ ] (square brackets)	Indicate optional items in PeopleCode syntax.
& (ampersand)	When placed before a parameter in PeopleCode syntax, an ampersand indicates that the parameter is an already instantiated object. Ampersands also precede all PeopleCode variables.

Typographical Convention	Description
⇒	This continuation character has been inserted at the end of a line of code that has been wrapped at the page margin. The code should be viewed or entered as a single, continuous line of code without the continuation character.

ISO Country and Currency Codes

PeopleSoft Online Help topics use International Organization for Standardization (ISO) country and currency codes to identify country-specific information and monetary amounts.

ISO country codes may appear as country identifiers, and ISO currency codes may appear as currency identifiers in your PeopleSoft documentation. Reference to an ISO country code in your documentation does not imply that your application includes every ISO country code. The following example is a country-specific heading: "(FRA) Hiring an Employee."

The PeopleSoft Currency Code table (CURRENCY_CD_TBL) contains sample currency code data. The Currency Code table is based on ISO Standard 4217, "Codes for the representation of currencies," and also relies on ISO country codes in the Country table (COUNTRY_TBL). The navigation to the pages where you maintain currency code and country information depends on which PeopleSoft applications you are using. To access the pages for maintaining the Currency Code and Country tables, consult the online help for your applications for more information.

Region, Country, and Industry Identifiers

Information that applies only to a specific region or industry is preceded by a standard identifier in parentheses. This identifier typically appears at the beginning of a section heading, but it may also appear at the beginning of a note or other text.

An example of a region-specific heading is: "(Latin America) Setting Up Depreciation"

Region Identifiers

Regions are identified by the region name. The following are examples of region identifiers that may appear in PeopleSoft Online Help:

- Asia Pacific
- Europe
- Latin America
- North America

Country Identifiers

Countries are identified by a three-letter country code. The following are examples of country identifiers that may appear in the PeopleSoft Online Help:

- ARG (Argentina)
- AUS (Australia)
- CAN (Canada)
- NLD (The Netherlands)

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following are examples of industry identifiers that may appear in the PeopleSoft Online Help:

- USF (U.S. Federal)
- E&G (Education and Government)

Translations and Embedded Help

PeopleSoft 9.2 software applications include translated embedded help. With the 9.2 release, PeopleSoft aligns with the other Oracle applications by focusing our translation efforts on embedded help. We are not planning to translate our traditional online help and PeopleBooks documentation. Instead we offer very direct translated help at crucial spots within our application through our embedded help widgets. Additionally, we have a one-to-one mapping of application and help translations, meaning that the software and embedded help translation footprint is identical—something we were never able to accomplish in the past.

Using and Managing the PeopleSoft Online Help

Select About This Help in the left navigation panel on any page in the PeopleSoft Online Help to see information on the following topics:

- Using the PeopleSoft Online Help.
- Managing hosted Online Help.
- Managing locally installed PeopleSoft Online Help.

PeopleTools Related Links

PeopleSoft 8.63 Home Page

PeopleSoft Search and Insights Home Page

PeopleSoft Online Help

PeopleSoft Information Portal

PeopleSoft Spotlight Series

PeopleSoft Training and Certification | Oracle University

My Oracle Support

Oracle Help Center

Contact Us

Send your suggestions to psoft-infodev_us@oracle.com.

Please include the applications update image or PeopleTools release that you're using.

Follow Us

Icon	Link
	Watch PeopleSoft on YouTube
	Follow @PeopleSoft_Info on X.
	Read PeopleSoft Blogs
	Connect with PeopleSoft on LinkedIn

1

Getting Started with System and Server Administration

System and Server Administration Overview

This section provides an overview of the various components of system and server administration.

PSADMIN

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

- Tuxedo and Jolt.
- PeopleCode debugging.
- Caching.
- Analytic server framework.
- Transactional SQL requests.
- Performance enhancement.
- PeopleSoft Query.
- Integration Broker.
- email.
- Real time event notification.
- Performance Monitor.
- MultiChannel Framework.

You launch and run PSADMIN using a command line interface.

Related Topics

- Understanding PSADMIN
  PSADMIN simplifies the process of configuring and administering all of the server processes and features that are available for PeopleSoft servers. For example, you use PSADMIN to configure application server domains, Process Scheduler servers, and PIA domains. You can use PSADMIN interactively or through the command-line interface.
- Understanding PSADMIN Menus
  You use PSADMIN to manage all of the main PeopleSoft server types. After launching PS_HOME\appserv\psadmin.exe from the command line, the PeopleSoft Server Administration (PSADMIN) main menu appears.

Analytic Servers

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

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

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

Web Servers

Oracle supports Oracle WebLogic web servers for PeopleSoft implementations. Oracle WebLogic web servers provide the basic functionality to support PeopleSoft applications, including a console interface, secure sockets layer (SSL), and reverse proxy servers (RPS). This PeopleBook provides supplemental information about configuring and administering Oracle WebLogic web servers where it has particular relevance to PeopleSoft implementations.

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

Related Topics

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

PeopleSoft Configuration Manager

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

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

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

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

Related Topics

- Understanding PeopleSoft Configuration Manager
  PeopleSoft Configuration Manager simplifies Windows workstation administration by enabling you to adjust PeopleSoft registry settings from one central location. It contains a variety of controls that let you set up Windows workstations running development and administrative applications, such as Application Designer and Data Mover.

PeopleTools Utilities

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

- The System Information page.
- PeopleTools options.
- The message catalog.
- The spell check system dictionary.
- Translate values.

- Application server caching.
- Table management and sharing.
- Remote database connection.
- File attachments.
- Stored URLs.
- Trace Options.
- Update tracking.
- Platform-specific database features.
- Database auditing.
- International settings.
- Optimization utilities.
- PeopleSoft Ping.

Related Topics

- Understanding the PeopleTools Utilities
  As you work with the PeopleSoft system, you find that there are some administrative tasks that you only need to perform occasionally. These tasks include such things as maintaining error messages and setting DDL model defaults. The PeopleTools Utilities menu is where you find tools for accomplishing some of these more infrequent tasks.

Tracing and Debugging

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

Jolt Configuration Options

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

Timeout Settings

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

System and Server Administration Implementation

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

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

- Install your PeopleSoft application according to the installation guide for your database platform and application.

  PeopleSoft applications are installed with the PeopleSoft deployment packages and include PeopleTools, database, application server, web server (PIA), and Process Scheduler.

  See *PeopleSoft 9.2 Application Installation* for your database platform.
- Establish a user profile that gives you access to the tools, pages, and processes that you'll use.

  See the product documentation for *PeopleTools: Security Administration*.

Other Sources of Information

In addition to implementation considerations presented in this section, take advantage of all PeopleSoft sources of information, including the installation guides, new feature overviews (previously named release notes), technical briefs, and PeopleBooks.

2 Understanding PeopleSoft Internet Architecture

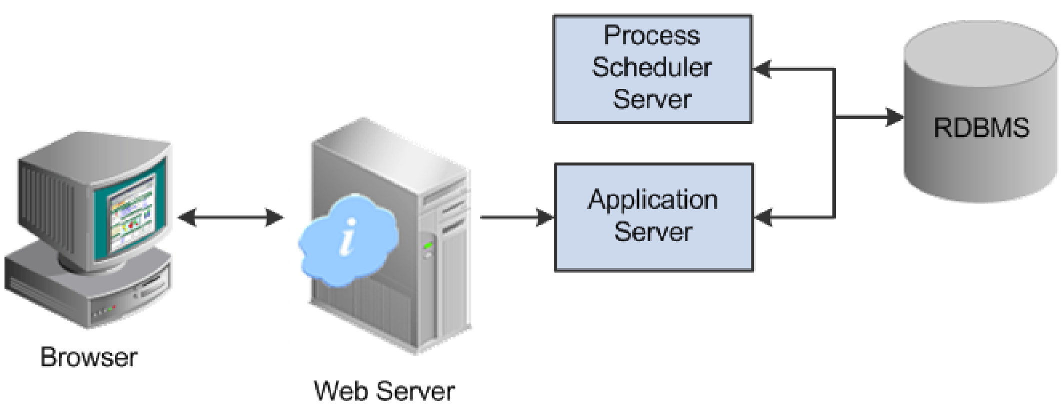
PeopleSoft Architecture Fundamentals

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

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

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

The browser sends requests to the web server, which transmits the request to the application server, which generates the SQL to run on the database.



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

- Integration Broker, our service oriented architecture (SOA).
- PeopleSoft Interaction Hub.
- Feeds Framework.
- Search Framework.

- Performance Monitor.

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

> **Note**
>
> Network connectivity between the PeopleSoft server elements support Internet Protocol versions 4 and 6 (IPv4 and IPv6).

Database Server

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

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

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

Application Servers

This section provides an overview of Application Servers and the components related to them.

Application Servers

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

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

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

Oracle Tuxedo and Oracle Jolt

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

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

Domains

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

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

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

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

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

PeopleSoft Server Processes

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

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

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

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

Note

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

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

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

The required server processes are:

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

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

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

Services

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

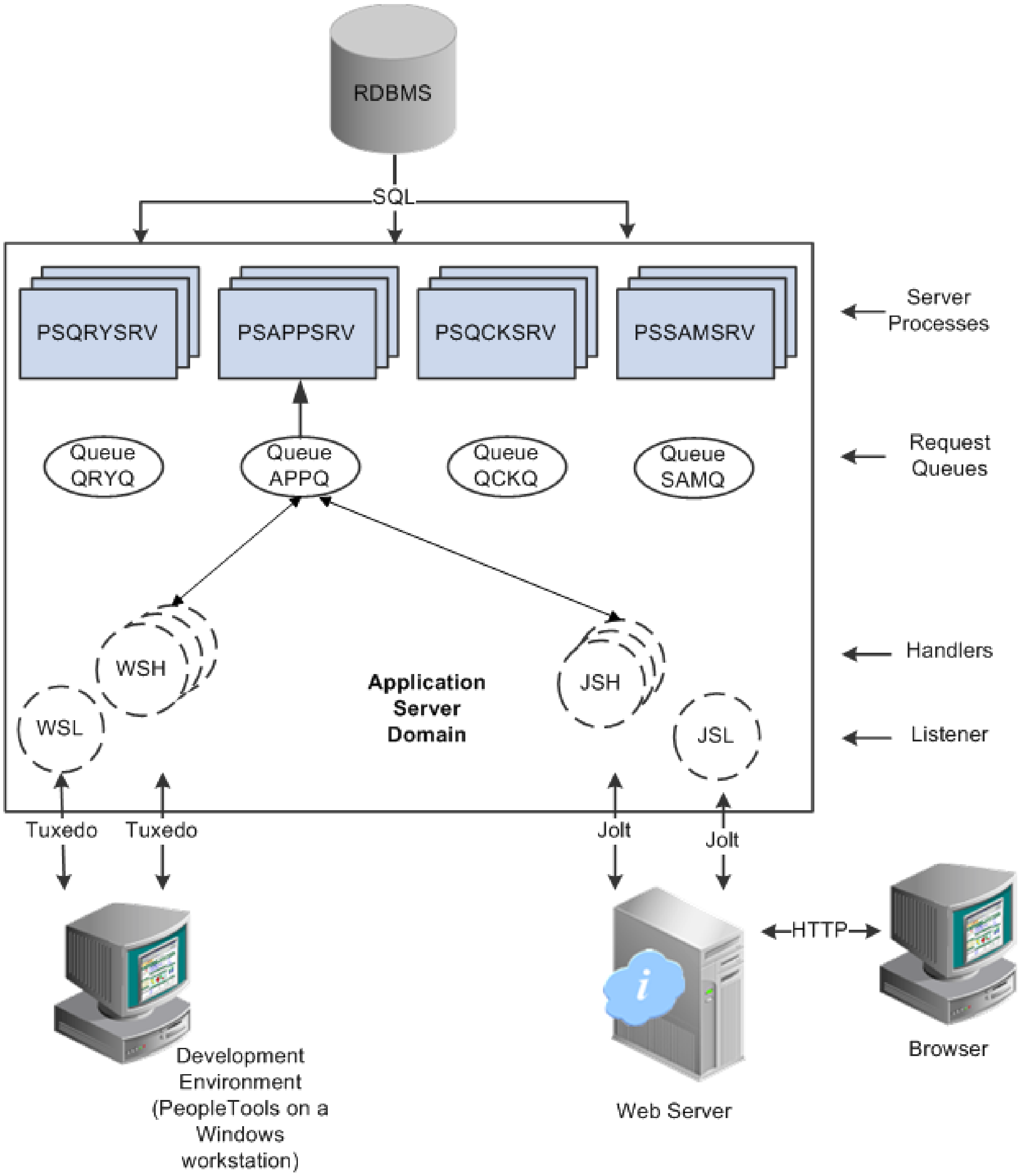
When a server process boots, it advertises to the system the predefined services it handles.

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

Listeners, Handlers, and Queues

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

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



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

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

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

Item	Description
Request queues	Each type of server process has a service request queue that it shares with other server processes of the same type. For example, PSAPPSRV processes use the queue, APPQ, and PSQCKSRV processes use the queue, QCKQ. The workstation handler and Jolt server handler insert requests into the appropriate queue, and then the individual server processes complete each request in the order that it arrives.
Server processes	The server processes act as the core of the application server domain. They maintain the SQL connection and make sure that each transaction request gets processed on the database and that the results are returned to the appropriate origin.

PeopleSoft Process Scheduler Server

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

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

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

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

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

Related Topics

- Process Scheduler

Web Server

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

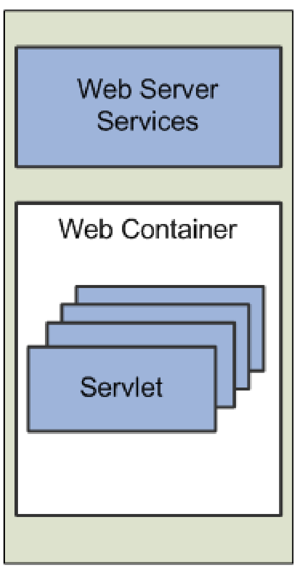
Web Server Software

Oracle supports Oracle WebLogic web server for use with PeopleSoft implementations.

The following software runs on the PeopleSoft web server:

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

The following diagram explains about Web server running the web server services and the web container that in turn contain PeopleSoft Servlet installations.



PeopleSoft Web Server

PeopleSoft Servlets

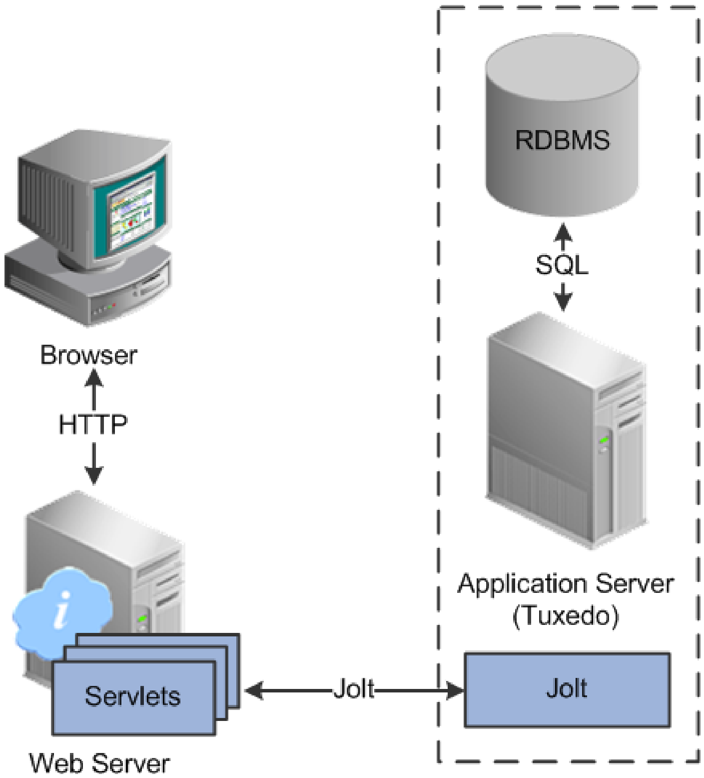
The following PeopleSoft servlets reside on the web server:

Servlet	Description
PORTAL	The portal servlet handles all of the requests and formatting for the users accessing PeopleSoft through PeopleSoft portal technologies. It manages various functionality, such as browser requests, search, content management, and homepage personalization.
PSIGW	The gateway servlet transmits service requests and responses between defined nodes as part of the PeopleSoft Service Oriented Architecture. The gateway handles PeopleSoft-to-PeopleSoft, PeopleSoft-to-third party, and third party-to-PeopleSoft services.
PSEMHUB	Supports the Environment Management Framework, used by various lifecycle management tools, such as Change Assistant, to collect and monitor system information.
Report Repository	This report repository servlet enables users to easily access and distribute the output of batch reports, such as SQR, run through PeopleSoft Process Scheduler. This servlet retrieves the report output in the report repository and serves it to the browser.
PSINTERLINKS	Used with the desupported product PeopleSoft Business Interlinks.

Oracle Jolt

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

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



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

Web Browser

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

The PeopleSoft system sends only the following to the browser:

- HTML
- JavaScript
- Cookies

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

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

- an in-memory cookie, never written to disk.

- encrypted to prevent snooping.
- check-summed to prevent tampering.

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

Browser Capabilities

Each PeopleSoft website has a browser capabilities file (browscap) located in:

`<PS_CFG_HOME>\webserv\<domain>\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs\<site>`

The system uses the browser capabilities file to form rules based on the values received from the user-agent string sent from client browser connections. The user-agent string identifies various characteristics of the browser and provides system details required for host web servers to tailor content sent to that browser and platform type, as needed. Upon receiving the values passed in the user-agent string, the system matches the values to the rules stored in the browser capabilities file to determine various characteristics, including browser support level, browser type, operating system, device type, and so on.

The file itself contains documentation for the syntax for the properties and rules. However, it is not a file that would be modified regularly outside of Oracle development. It should be modified only if needed or if instructed to do so by Oracle representatives. Keep track of changes, as any changes made to the file would be overwritten during upgrades and maintenance updates.

Rules for Browser Type and Version

Property	Values	Description
browserType	• CHROME • IE • FIREFOX • Safari	Browser type.
browserTypeClass	• chrome • ie • ie edge • firefox • safari	Browser type.
browserFluidCapable	Y or N	Indicates whether the browser will support PeopleSoft Fluid User Interface applications.
browserVersion	• 8.0 • 9.0 • 10.0 • 11.0 • 12.0	Indicates browser version. Used only for Microsoft Internet Explorer.

Property	Values	Description
ieTridentVersion	• 4.0 • 5.0 • 6.0 • 7.0 • 8.0	Indicates the version of the Trident layout engine used by the browser. Used only for Microsoft Internet Explorer.

Rule Examples:

```
if Chrome/
browserType=CHROME;
browserTypeClass=chrome;
browserFluidCapable=Y
```

Rules for Platform

Property	Values	Description
browserPlatform	• WIN • WINXP • WIN8 • WIN7 • WINVISTA • WIN2008 • WIN10 • MAC • MAC68K • MACPPC • MACIPAD	The operating system on which the browser runs.

Rule Examples:

```
if windows nt 10.0
browserPlatform=WIN10;

if (?=.*mac)(?=.*ipad)
browserPlatform=MACIPAD;
```

Rules for Form Factors

Property	Value	Description
browserFormFactors	• small: 519 • medium: 759 • large: 959 • extra large: 9999999 (no ceiling)	Platform form factor width ceiling—the maximum width in pixels.

Rules for Platform Feature Classes

Property	Value	Description
browserPlatformClass	• android • ios • mac • win	Operating system
browserDeviceTypeClass	• pc • phone • tablet	Indicates the type of device running the browser session, which determines the form factor for display.

Rule Examples:

```
if win
browserPlatformClass=win;
browserDeviceTypeClass=pc

if ipad
browserDeviceTypeClass=tablet

if (?=.*Mobile)(?=.*Touch)
browserDeviceTypeClass=phone
```

Dynamic Access Control on the Web Server

You can set up secure web folders to keep your content private and control which users have access.

See Portal Technology: Creating Access Control for Files on the Web Server

See PeopleCode API Reference: Authorize method: ContentAuthorizorBase class

Server Configuration Options

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

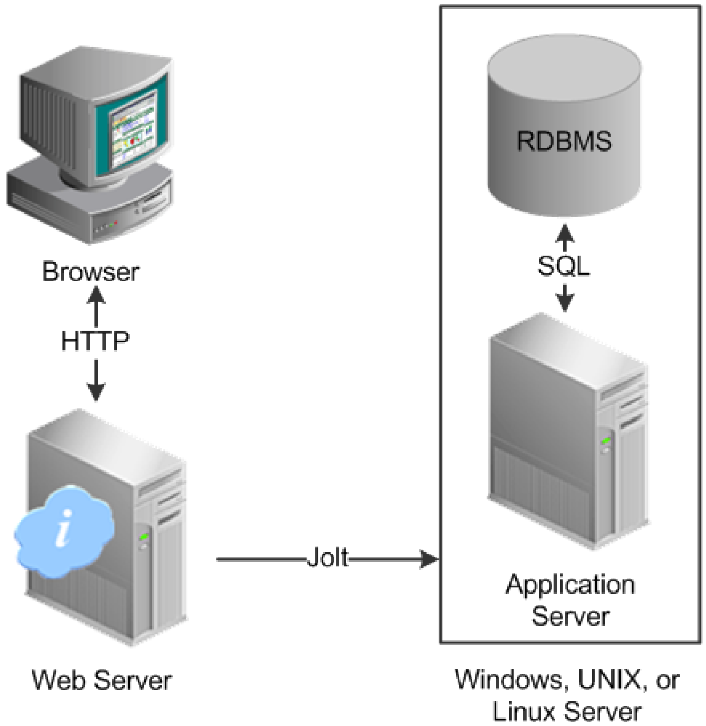
See *PeopleSoft 9.2 Application Installation* for your database platform.

Logically Separate Server Configuration

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

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

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



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

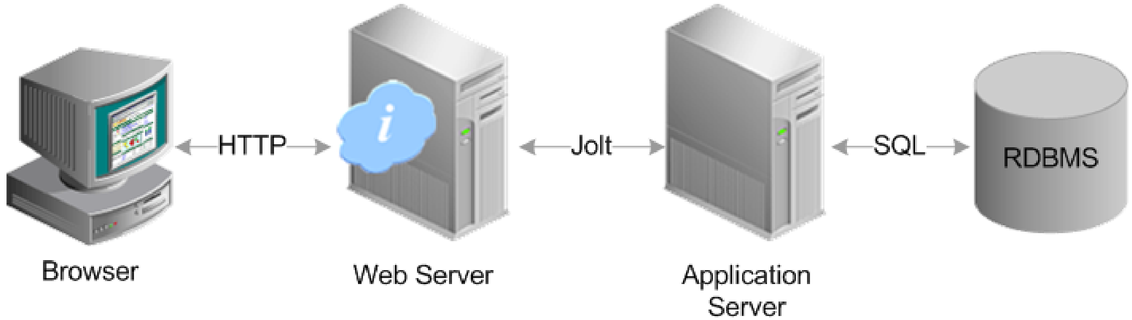
Although this diagram depicts a separate web server, the web server software could also reside on the same machine with both the application server and the database server. When installing multiple PeopleSoft architecture elements onto the same machine, the only requirement is that each element be supported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

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

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

Physically Separate Server Configuration

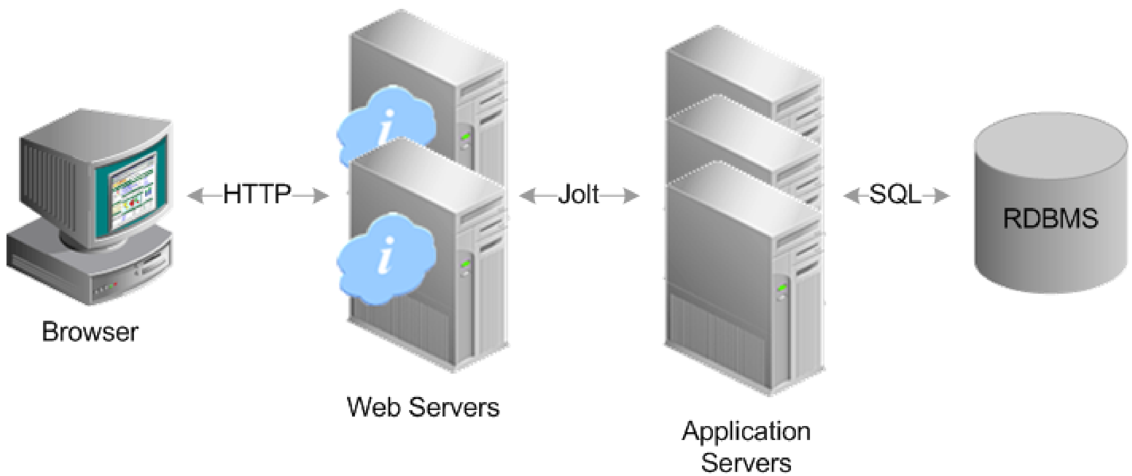
A physically separate server configuration means that the web server, application server, and database serer each reside on separate machines. The following diagram depicts a physically separate server configuration:



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

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

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



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

- scalability
- fail-over
- availability

Implementation Options

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

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

Technology	Description
Analytic Calculation Engine	Analytic Calculation Engine enables application developers to define both the calculation rules and the display of calculated data within PeopleSoft applications for the purposes of multidimensional reporting, data editing, and analysis. See the product documentation for *PeopleTools: Analytic Calculation Engine*.

3

Working with Server Domain Configurations

Understanding PS_HOME and PS_CFG_HOME

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

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

- PeopleSoft Application Server.
- PeopleSoft Process Scheduler Server.
- PeopleSoft Internet Architecture.

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

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

Location	File Type	Description	PeopleSoft Examples
PS_HOME	Binary	Compiled, non-modifiable executables and libraries.	PSADMIN.EXE PSAPPSRV.EXE PSAESRV.EXE PSAPPENG.DLL PSODBC.DLL
PS_CFG_HOME	ASCII	Text files associated with the configuration and administration of a domain that can be viewed, modified, or generated by the system.	PSAPPSRV.CFG PSAPPSRV.CFX PSTUXCFG APPSRV_<DATE>.LOG TUXLUG.<DATE>

> **Note**
>
> While some ASCII files are present in PS_HOME, these should be considered non-modifiable. For example, all of the files that are in PS_CFG_HOME when it is initially created, come from the associated PS_HOME.

The decoupling of these file types enables system administrators to:

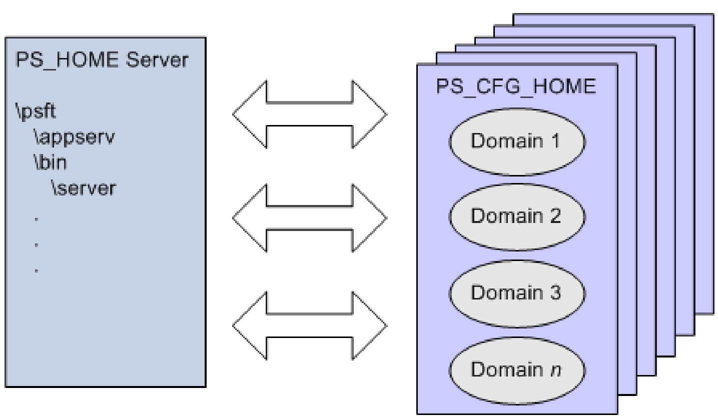
- Streamline and provide more flexible PeopleSoft server installations.
- Apply unique security restrictions to the binary file and configuration file locations.

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

When you launch PSADMIN, if a PS_CFG_HOME directory does not exist, PSADMIN creates the PS_CFG_HOME directory. PSADMIN creates PS_CFG_HOME when the first domain is created.

Implementing Flexible Server Installations

With the binary files separate from the domain configuration, you have the option of installing multiple domains on multiple separate servers all leveraging the binary files of a single PS_HOME. This diagram illustrates an example to implement multiple PS_CFG_HOME locations, containing multiple domains, all referencing the same PS_HOME installation on a remote server



Creating domains on separate servers enables you to:

- Install the PS_HOME binaries a single time.
- Incorporate additional server machines for additional domains per demand and performance requirements.

- Apply updates and upgrades to a single PS_HOME, reducing the time required for upgrading your system.

Applying Security Restrictions

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

See Understanding PS_HOME and PS_CFG_HOME Security.

Managing a Portable PS_HOME

After the initial installation of a single PS_HOME, that PS_HOME can be referenced, shared, and used as PS_HOME for any host server requiring the PS_HOME binaries for running PeopleSoft servers. Using a single, shared PS_HOME is the simplest, most efficient, and recommended approach.

However, the PS_HOME file system is portable, meaning the PS_HOME directory structure is self-contained and the content within it functions independent of a specific host or file path location. For example, if a system administrator elects to have a PS_HOME on each PeopleSoft server host, that PS_HOME file system can be moved or copied to additional locations on the network where it can be referenced by new PS_CFG_HOMEs, without requiring any additional setup program runs or manual configuration within the PS_HOME file system.

Using this portability, system administrators only need to run the PeopleTools installation program once to create the initial PS_HOME and then copy that PS_HOME to other required locations throughout the network. Likewise, PeopleTools maintenance can be applied only to one PS_HOME file system, which can then be copied to other locations.

> **Note**
>
> For efficient copying, PS_HOME can be compressed (zipped) to reduce network traffic.

For example, assume a PS_HOME has been installed on Host_A. A system administrator can copy that PS_HOME to a local directory on Host_B. The system administrator can launch PSADMIN, and begin configuring and booting application server domains, Process Scheduler domains, and PIA domains, without making any changes to the copied PS_HOME. When those domains are configured, the system automatically registers that PS_HOME location in PS_CFG_HOME/peopletools.properties

When the time comes to apply maintenance, the EMF agent crawling process discovers the PeopleSoft Home and database information from the domain configuration. Specifically, the PS_CFG_HOME/peopletools.properties file associates that PS_CFG_HOME with the appropriate PS_HOME installation (which is the PS_HOME from which the administrator launched PSADMIN to create the PS_CFG_HOME). The system stores the PS_HOME location in the `installlocation` key of the PS_CFG_HOME/peopletools.properties file. For example:

```
installlocation=C\:\\PT8.58
```

For UNIX, because the location of PS_HOME is relative, you need to make PS_HOME the current working directory prior to launching PSADMIN (PS_HOME/appserv/psadmin). For example, you need to make the current working directory PS_HOME prior to sourcing psconfig.sh, which does not contain a hard-coded path to PS_HOME. The psconfig.sh script derives the location of PS_HOME from the context in which it runs.

> **Note**
>
> The same physical PS_HOME may be mounted at different file system locations from different contexts (from different hosts and PS_CFG_HOMEs), which means the PS_HOME path appears differently for each of the mount contexts.

A single PS_HOME can be shared in situations where the system administrator decides new application server or Process Scheduler domains need to be created on a host, but the system administrator does not want to install PeopleTools again to create a local PS_HOME.

1. On Machine X, the system administrator installs PeopleTools to /u01/product/psft/pt/8.58.
2. On Machine Y, the system administrator creates a mount point to the remote PS_HOME that resides on Machine X. This mounted PS_HOME is at /mnt/psft/pt/8.57.
3. On Machine A a domain administrator logs in, opens a shell, changes directory to PS_HOME at /mnt/psft/pt/8.57 and sources psconfig.sh.
4. On Machine A a domain administrator starts PSADMIN and creates, configures and starts application server, Process Scheduler, or PIA domains.
5. The steps 2-4 are repeated on any other host where PeopleSoft servers are to be created.

If you intend to use the Environment Management Framework for applying updates to your system, you will need to set PS_AGENT_HOME. For more information on PS_AGENT_HOME, see Change Assistant and Update Manager: Configuring an Environment Management Agent.

Working with the Default PS_CFG_HOME

This section discusses how to locate and work with default PS_CFG_HOME.

Locating the Default PS_CFG_HOME

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

Operating System	Required Environment Variable
UNIX/Linux	HOME
Windows	USERPROFILE

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

Operating System	PS_CFG_HOME Location
UNIX/Linux	$HOME/psft/pt/<version>

Operating System	PS_CFG_HOME Location
Windows	%USERPROFILE%\psft\pt\<version>

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

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

```
/home/psftuser/peopletools/8.58
                     peopletools.properties
                     /appserv
                       /CRM
                       /HR
                       /PRCS
                         /CRM_PRCS
                         /HR_PCRS
                       /Search
                       /webserv
                       /ver_dom
```

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

```
C:\Documents and Settings\psftuser\psft\pt\8.58
                              peopletools.properties
                     \appserv
                        \CRM
                        \HR
                        \PRCS
                           \CRM_PRCS
                           \HR_PCRS
                        \Search
                        \Temp
                        \webserv
                        pswinsrv.cfg
```

Note

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

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

```
psadmin -defaultPS_CFG_HOME
```

Note

These commands are not case sensitive.

Using PSADMIN with the Default PS_CFG_HOME

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

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

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

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

Working with Alternate PS_CFG_HOME Locations

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

This section discusses how to specify and work with alternate PS_CFG_HOME locations.

Specifying Alternate PS_CFG_HOME Locations

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

For UNIX and Linux, you can:

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

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

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

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

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

Using the %V and %K Meta Variable

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

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

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

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

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

Use the %K variable if you wish to store PS_CFG_HOMEs by patch level. That is, while %V stores the files by version (8.50–00, 8.51–00), %K stores files down to the patch level. For example, using %K, a PS_CFG_HOME may look similar to:

```
C:\PT\installs
          \8.53
               \8.53-01
               \8.53-03
```

Configuring Domains in Alternate Locations of PS_CFG_HOME

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

Working with Remote PS_HOME Locations

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

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

Installing Necessary Components

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

Ensuring that PS_CFG_HOME is Set Appropriately

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

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

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

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

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

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

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

> **Note**
>
> You can have multiple PS_CFG_HOME locations on the same host server, but you will need to make sure the PS_CFG_HOME is set appropriately prior to starting PSADMIN, or by using the Switch Config Home PSADMIN option.

Managing Domains

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

- Use distinct PS_CFG_HOME locations for each PS_HOME. When a PS_CFG_HOME is created, PSADMIN brings content from the current PS_HOME into the PS_CFG_HOME, which effectively binds that PS_CFG_HOME to that PS_HOME. That is, all domains within a given PS_CFG_HOME should reference the same PS_HOME. Domains in the same PS_CFG_HOME that reference different PS_HOMEs can encounter unpredictable behavior, especially when applying updates to the system.

- Install as few PS_HOMEs as possible. Ideally, each PS_CFG_HOME requiring the same PeopleTools release level points to the same PS_HOME. This approach helps reduce the number of locations that require patches to be applied.
- After applying updates (patches) to the PS_HOME, recreate domains associated with that PS_HOME.
- A domain configured on one machine cannot be copied manually and run on another machine. Use the Replicate Config Home PSADMIN option to perform this task.
- The Windows service configuration file (PSWINSRV.CFG) will be invalid if it is shared between multiple PS_CFG_HOMEs on different network drives.
- As you maintain your system and upgrade domains, keep in mind that dormant domains will consume disk space. Make sure to keep track of where your retired domains reside and take action to remove them when no longer required. It is generally recommended to remove old log files and trace files for both efficiency and security purposes.

Working with PS_APP_HOME

This topic contains an overview and discusses how to manage environments where PS_APP_HOME has been implemented.

Understanding PS_APP_HOME

PS_APP_HOME refers to the installation location for the PeopleSoft application software. In previous PeopleSoft releases, you installed PeopleTools into a high-level directory referred to as PS_HOME, and then you installed your PeopleSoft application into the existing PS_HOME location such that PeopleTools and your PeopleSoft application coexisted within the same directory structure. In the current release, this configuration is not supported. PS_APP_HOME must not be the same as PS_HOME.

The installation performed by the PeopleSoft Deployment Packages (DPKs) supports only separate PS_APP_HOME and PS_HOME. For information on the default installation locations for PS_HOME and PS_APP_HOME set by the DPKs, see the PeopleSoft PeopleTools Deployment Packages installation guide on the PeopleSoft PeopleTools Patches Home Page, My Oracle Support, Doc ID 2062712.2.

The separate PS_APP_HOME configuration includes the following benefits:

- PeopleTools patches and updates can be applied knowing that no application content has been disturbed. Likewise, PS_APP_HOME application updates can be applied independently.
- Increased flexibility and modularity in that a single PS_HOME can be associated with multiple PS_APP_HOME locations.
- Disk space savings, due to shared PS_HOME amongst multiple PS_APP_HOME locations. Using the traditional method, each application installation requires its own PeopleTools installation.
- When troubleshooting it will be easier to isolate application and PeopleTools issues.

Managing Environments with PS_APP_HOME

As you configure your system, you need to be mindful of the PS_APP_HOME location. When you run PSADMIN, the PS_APP_HOME location is listed.

In many of the configuration settings where PS_APP_HOME might be involved, PeopleTools includes both PS_APP_HOME and PS_HOME references. For example,

```
RCCBL PRDBIN=%PS_APP_HOME%/cblbin:%PS_HOME%/cblbin
```

In environments where PS_APP_HOME has been implemented, system elements such as server domains have access to content supplied by both PS_HOME and PS_APP_HOME, but files added by the PeopleSoft application installation (those residing in PS_APP_HOME) that have the same name as those in the PeopleTools PS_HOME take preference.

The following table outlines key areas in the PeopleSoft system, where you need to consider the PS_APP_HOME location for your PeopleSoft applications. When you are installing, configuring, and maintaining your system, make sure to keep these items in mind and review the documentation associated with these areas.

PeopleSoft Infrastructure Element	PS_APP_HOME Consideration
Application Server	Remote Call COBOL can look for COBOL programs in both PS_APP_HOME or PS_HOME, so be sure to review those settings. See RCCBL PRDBIN.
Process Scheduler	The Process Scheduler domains reference PS_APP_HOME and PS_HOME locations for the following processes: • COBOL • SQR • nVision See Process Scheduler: Understanding the PeopleSoft Process Scheduler Configuration File.
COBOL	During your installation, you will need to perform different tasks when compiling COBOL. See the product documentation for *PeopleSoft 9.2 Application Installation*, "Installing and Compiling COBOL on UNIX" or "Installing and Compiling COBOL on Windows."
SQR	When specifying locations where the system should look for SQR reports, for example, in Process Scheduler or in Configuration Manager, make sure to consider reports that may exist in PS_APP_HOME.
Lifecycle Management	For applying updates and performing upgrades, PS_APP_HOME also needs to be considered for various tasks, including: • Crawling the environment with Environment Management Framework. • Creating change packages • Defining environments for upgrades. See Change Assistant and Update Manager.

PeopleSoft Infrastructure Element	PS_APP_HOME Consideration
Windows Development Environment	When setting up Windows Development workstation profiles using Configuration Manager, you set the PS_APP_HOME variable on the Process Scheduler tab, and you'll also consider your PS_APP_HOME location when specifying: • SQR report locations. • COBOL location. • nVision directory paths. • Data Mover directories. See Understanding PeopleSoft Configuration Manager.

Working with PS_CUST_HOME

This topic contains an overview and discusses how to manage environments where PS_CUST_HOME has been implemented.

Understanding PS_CUST_HOME

PS_CUST_HOME (PeopleSoft Customization Home) enables you to further clarify the ownership of files within your PeopleSoft system and separate them for more efficient lifecycle maintenance and system administration. While PS_APP_HOME enables you to separate PeopleSoft application files from PeopleTools files, PS_CUST_HOME enables you to identify your site's customized files and store them in a location separate from your PeopleTools and PeopleSoft application files. This creates a clear distinction between code and files delivered by Oracle (PeopleTools and PeopleSoft applications) and code produced or customized by individual customers.

> **Important**
>
> You can continue to store your custom and modified files within PS_HOME or PS_APP_HOME if desired. Using PS_CUST_HOME is an option that you can take advantage of if the additional flexibility provided by PS_CUST_HOME is required.

If your site does not create custom scripts, reports, or programs, then PS_CUST_HOME would not apply to your system. However, if your site does create custom scripts, reports, or programs, or if your site modifies delivered application files, then implementing PS_CUST_HOME should be considered.

Examples of customized files that might be stored in PS_CUST_HOME include:

- Data Mover scripts.
- DAT files.
- COBOL programs.
- SQR programs.
- Java class files.

With your custom files stored in the PS_CUST_HOME location, you can freely take PeopleTools or PeopleSoft application upgrades and maintenance updates, knowing that your customized files will not be affected. This approach also keeps PS_HOME and PS_APP_HOME consistent with the content provided by Oracle, being an exact replication of the files as they were installed.

In addition to lifecycle management benefits, PS_CUST_HOME also offers a significant advantage with sharing PS_HOME locations among multiple PeopleSoft application systems. If you are not using PS_CUST_HOME, customizations stored in PS_HOME or PS_APP_HOME that apply to only one application make it impossible to share that PS_HOME or PS_APP_HOME location with other PeopleSoft applications. With PS_CUST_HOME, all customizations can be consolidated into a single, isolated PS_CUST_HOME, and multiple runtime environments can utilize the resources within that PS_CUST_HOME. If separate environments require different customizations, they can share the same PS_HOME and PS_APP_HOME but point to separate PS_CUST_HOME locations.

An environment where PS_CUST_HOME is implemented may have the following environment variables set.

Environment Variable	Description
PS_HOME	Location where PeopleTools is installed. For example: `C:\psft`
PS_CFG_HOME	Location where the server domain configurations are installed. For example: `PS_CFG_HOME=C:\User\psftuser\psft`
PS_APP_HOME	Location where your PeopleSoft application files are installed. For example: `PS_APP_HOME=D:\HCM_920`
PS_CUST_HOME	Location where your customized PeopleSoft application files are installed. For example: `PS_CUST_HOME=D:\PSFT_CUSTOM`

Managing Environments with PS_CUST_HOME

Managing an environment where PS_CUST_HOME exists is similar to adding PS_APP_HOME to the list of environment variables, except with PS_APP_HOME you install the application files using the delivered installation program. As such, the application files are installed automatically into PS_APP_HOME using the standard directory structure based on PS_HOME. With PS_CUST_HOME, you need to create the directory structure manually, adding any required folder within the PS_CUST_HOME using the same structure and naming found in PS_HOME or PS_APP_HOME.

For example, if you have a custom SQR report that you want to store in PS_CUST_HOME, you need to create this directory structure in your PS_CUST_HOME and place your customized file within the \sqr directory:

D:\PS_CUST_HOME\sqr

Likewise, if you have custom SQL or Data Mover scripts, you need to create this directory structure in your PS_CUST_HOME and place your customized files within the \scripts directory:

D:\PS_CUST_HOME\scripts

If using PS_CUST_HOME, you need to make sure to set the PS_CUST_HOME environment variable prior to running PSADMIN. This can be done permanently in the environment variables interface on a Windows server, manually on the command line prior to running PSADMIN, or by using a .bat or shell script.

> **Note**
>
> If you do not set the PS_CUST_HOME environment variable explicitly, PeopleSoft runtime retrieves all runtime content from PS_HOME or PS_APP_HOME, if set.

When you configure your system, you need to be mindful of the PS_CUST_HOME location. In the PeopleTools configuration settings where the system would need to find files within PS_CUST_HOME, you need to include PS_CUST_HOME as a potential location for such files. For example,

```
RCCBL PRDBIN=%PS_CUST_HOME%/cblbin:%PS_APP_HOME%/cblbin:%PS_HOME%/cblbin
```

In environments where PS_CUST_HOME has been implemented, system elements, such as server domains have access to content supplied by PS_HOME, PS_APP_HOME, and PS_CUST_HOME, but files residing in PS_CUST_HOME take preference over those with the same name in either PS_HOME or PS_APP_HOME. If PS_CUST_HOME, PS_APP_HOME, and PS_HOME all exist within a PeopleSoft system, at runtime, the system checks for files in this order:

1. PS_CUST_HOME
2. PS_APP_HOME
3. PS_HOME

The following table outlines key areas in the PeopleSoft system, where you need to consider the PS_CUST_HOME variable if you added custom files to that location. When you are installing, configuring, and maintaining your system, make sure to keep these items in mind and review the documentation associated with these areas.

PeopleSoft Infrastructure Element	PS_CUST_HOME Consideration
Application Server	PS_CUST_HOME must be set explicitly before running PSADMIN, similar to setting PS_CFG_HOME manually. Remote Call COBOL can look for COBOL programs in PS_CUST_HOME, PS_APP_HOME, or PS_HOME, so be sure to review those settings. See RCCBL PRDBIN.
Process Scheduler	PS_CUST_HOME must be set explicitly before running PSADMIN, similar to setting PS_CFG_HOME manually. The Process Scheduler domains reference PS_CUST_HOME, PS_APP_HOME, and PS_HOME locations for the following processes: • COBOL • SQR • nVision See Process Scheduler: Understanding the PeopleSoft Process Scheduler Configuration File.
COBOL	During your installation, you will need to perform different tasks when compiling COBOL, which are described in detail in your PeopleTools installation guide. If you do have custom COBOL programs or files, you must set the PS_CUST_HOME environment variable prior to compiling the source code. If PS_CUST_HOME is set, the system compiles the source code in PS_HOME, PS_APP_HOME, and PS_CUST_HOME simultaneously, despite the source code residing in separate locations. See *PeopleSoft 9.2 Application Installation* for your database platform, "Installing and Compiling COBOL on UNIX" or "Installing and Compiling COBOL on Windows."
SQR	When specifying locations where the system should look for SQR reports, for example, in Process Scheduler or in Configuration Manager, make sure to consider reports that may exist in PS_CUST_HOME.
Lifecycle Management	For applying updates and performing upgrades, PS_CUST_HOME also needs to be considered for various tasks, including: • Crawling the environment with Environment Management Framework. • Creating change packages • Defining environments for upgrades. See Change Assistant and Update Manager. **Note:** By placing files within PS_CUST_HOME, you are taking complete responsibility for maintaining the source of that file and ensuring any changes are made to it for upgrade or maintenance purposes.

PeopleSoft Infrastructure Element	PS_CUST_HOME Consideration
Windows Development Environment	When setting up Windows Development workstation profiles using Configuration Manager, you set the PS_CUST_HOME variable on the Process Scheduler tab, and you'll also consider your PS_CUST_HOME location when specifying: • SQR report locations. • COBOL location. • nVision directory paths. • Data Mover directories. See Understanding PeopleSoft Configuration Manager.

4

Using the PSADMIN Utility

Understanding PSADMIN

PSADMIN simplifies the process of configuring and administering all of the server processes and features that are available for PeopleSoft servers. For example, you use PSADMIN to configure application server domains, Process Scheduler servers, and PIA domains. You can use PSADMIN interactively or through the command-line interface.

Topic	Reference
Use common parameters for setting up a domain.	Using the Quick-Configure Menu
Use PSADMIN interactively to manage your domains.	Understanding PSADMIN Menus
Use PSADMIN on the command line.	Using the PSADMIN Command-Line Interface
Edit application server configuration files.	Understanding Application Server Domain Parameters
Edit Process Scheduler configuration files.	Process Scheduler: Understanding the PeopleSoft Process Scheduler Configuration File

Accessing Network Drives in Microsoft Windows Server

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

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

Note

This section applies only to Tuxedo-managed servers (application server and Process Scheduler server). It does not apply to PIA domains.

Any PeopleSoft processes that reference mapped network drives by their drive letters in this environment must be able to find the drives, and must have appropriate permission to access them. In Windows Server, the operating system does not provide this access directly.

PeopleSoft uses Tuxedo's Oracle ProcMGR service and an associated environment variable, TM_TUXIPC_MAPDRIVER, to gain access to the network drives. You must configure these elements to provide the appropriate access before you start any PeopleSoft servers or other processes.

To configure access to mapped network drives:

1. Determine which shared network directories your PeopleSoft system will need to access with a drive and directory path.
2. In your Microsoft Windows system, make sure that the shared network directories are available, and grant the domain administrator privileges to access them.
3. Access the Services utility in Windows Control Panel.
4. Stop the Oracle ProcMGR service.
5. In the Oracle ProcMGR service properties, ensure that the “log on as” account is the account with domain administrator privileges for the required shared directories.
6. Define the TM_TUXIPC_MAPDRIVER environment variable to specify the drive mappings and paths of the required shared directories, with the following format:

   drive1:=*machine_name1**dirpath1*[;*drive2*:=*machine_name2**dirpath2*[...]]

   For example:

   ```
   U:=\\myMachine\e$;V:=\\myMachine\PSFT
   ```

7. Start the Oracle ProcMGR service.

   The service uses the value of TM_TUXIPC_MAPDRIVER to create the necessary drive mappings, and uses its own log on settings to provide your PeopleSoft system with access to those locations.

> **Note**
>
> Every time your PeopleSoft system needs to access a new mapped network location, you must repeat this procedure, including the new drive mapping along with the others.

Starting PSADMIN

This section assumes that you have already installed and configured the PeopleSoft application server.

See *PeopleTools Deployment Packages Installation* for the current release.

See *PeopleSoft 9.2 Application Installation* for your database platform.

> **Note**
>
> *PS_HOME* is the PeopleSoft PeopleTools installation directory.

To start the PSADMIN utility:

1. Launch the psadmin executable.

   PSADMIN is in PS_HOME\appserv.

2. Select the server that you want to configure, administer, or monitor from the PSADMIN menu.

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

  Config Home:  D:\PT_SERVER\8.60

  1) Application Server
  2) Process Scheduler
  3) Web (PIA) Server
  4) Switch Config Home
  5) Service Setup
  6) Replicate Config Home
  7) Refresh Config Home
  q) Quit
Command to execute (1-7, q):
```

Using PSADMIN

Using PSADMIN involves selecting the number of the menu item that reflects the action that you want to take, entering the correct number at the command line, and pressing **Enter.** However, in some cases, you may want to use the command-line options that PSADMIN offers.

See Using the PSADMIN Command-Line Interface.

Each PSADMIN menu has the same look and feel. To select a menu item, enter the corresponding number at the prompt and press **Enter.** To return to the previous menu enter *q* (quit) at the prompt.

> **Note**
>
> Because the numbers corresponding to the PSADMIN menu commands can change as minor releases and patches provide different features and capabilities to PSADMIN, this documentation generally refers to the names of the commands rather than their menu item numbers.

Using Configuration Templates

The initial values that you see in PSADMIN are derived from the configuration template that you select when you create your domain. The delivered templates provide a range of possible implementations. Each configuration template includes a number of server processes, such as PSAPPSRV, that is sufficient for its intended load.

These are the delivered templates:

Configuration Template	Default PSAPPSRV Server Processes	Example Usage
Small	2	Use for 1–50 users.
Medium	8	Use for 50–500 users.
Large	25	Use for 500–1000 users.
Developer	2	Use for development and demonstration environments only.

Note

The usage examples above are *sample* ranges. Each site must determine the optimal template and number of server processes to suit the types of transactions of a particular user base. The delivered templates provide reasonable starting points for your implementations. You can add additional server processes through PSADMIN or by editing the PSAPPSRV.CFG file. For optimum performance, the number of users per configuration template can vary significantly from application to application.

Note

Oracle does not support creating custom CFX or UBX templates nor modifying delivered CFX or UBX templates.

Using the Quick-Configure Menu

When you create a domain for the first time, PSADMIN presents you with the most commonly changed parameters on the Quick-Configure menu, so that you can get up and running quickly. After the initial setup, you may at any time select *Configure this domain* on the PeopleSoft Domain Administration menu to access the Quick-Configure menu.

```
----------------------------------------------
Quick-configure menu -- domain: HCMDEMO
----------------------------------------------
      Features                      Settings
     ==========                    ==========
  1) Pub/Sub Servers    : No    18) DBNAME      :[APPDOM]
  2) Quick Server       : No    19) DBTYPE      :[ORACLE]
  3) Query Servers      : No    20) UserId      :[PS]
  4) Jolt               : Yes   21) UserPswd    :[]
  5) Jolt Relay         : No    22) DomainID    :[TESTSERV]
  6) WSL                : No    23) AddToPATH   :[c:\psft\db\oracle-
server\19.3.0.0\bin]
  7) PC Debugger        : No    24) ConnectID   :[people]
  8) Event Notification: Yes    25) ConnectPswd :[]
  9) MCF Servers        : No    26) DomainConnectPswd:[]
 10) Perf Collator      : Yes   27) PHC Remote Admin Settings:[]
 11) Analytic Servers   : Yes   28) WSL Port    :[7000]
 12) Domains Gateway    : No    29) WSL SSL Port:[7010]
```

```
 13) Push Notifications: No    30) JSL Port    :[9000]
 14) MCP Server        : No    31) JSL SSL Port:[9010]
                               32) JRAD Port   :[9100]

      Actions
     =========
 15) Load config as shown
 16) Custom configuration
 17) Edit environment settings
  h) Help for this menu
  q) Return to previous menu

HINT: Enter 17 to edit DBNAME, then 15 to load

Enter selection (1-32, h, or q):
```

The Quick-Configure menu shows which features are currently set for the newly created domain. The menu contains the values that are most commonly changed when setting up a demonstration or test domain.

To change the value of a parameter under Features, just enter the number corresponding to the feature to 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.

> **Note**
>
> All of the values that you change remain in effect until you modify them again.

> **Note**
>
> If you select a Settings parameter, then press **Enter** without entering a new value at the prompt, the existing value of that parameter is deleted. If you then enter *q* to quit the Quick-Configure menu, your changes are discarded, and the original values remain. However, if you load the configuration as shown, the parameter is saved without a value.
>
> If the parameter is required, you'll see an error message indicating that the configuration could not be completed. The next time you access the Quick-Configure menu, PSADMIN reloads any empty required parameter with its original default value, just as it would appear if you were creating a new domain.

When you change the value of a password parameter, such as UserPswd or ConnectPswd, you are prompted to enter a password twice. If the password values do not match, an error message is displayed. The password values are masked.

To configure the rest of the parameters that are not presented on the Quick-Configure menu, select *Custom configuration* to view the full list.

The Quick-Configure menu is not intended to replace the series of configuration sections in the custom configuration interface. In most cases, your site requires the parameters and tuning options that are available only through the full custom configuration menu. For this reason, the

Quick-Configure menu is provided primarily for situations where you're setting up a demonstration domain for testing or for development purposes.

> **Note**
>
> When you use custom configuration, pressing **Enter** instead of entering a new value for a parameter does not delete the parameter's value. PSADMIN interprets **Enter** to mean that you want to retain the parameter's existing value. If you want to remove the value, you can enter one or more "white space" characters by pressing the space bar, or you can enter five underscore characters, and press **ENTER.**

To apply settings to the configuration files, you must select *Load config as shown.*

To add and specify environment variables at the domain level, select *Edit environment settings.*

For more information on editing domain environment variables, see Setting Domain-Level Environment Variables.

> **Note**
>
> For platforms that require the ServerName parameter to be set in order to boot, such as Microsoft SQL Server, you need to set that parameter through the custom configuration option in PSADMIN or by setting it manually in the configuration file.

Selecting Server Processes in the Quick-Configure Menu

Use the Quick Configure Menu to enable or disable server processes (listed under Features).

Pub/Sub Servers Option

If you want the messaging server processes to be configured and booted, enter *Yes*. If you are not implementing the Integration Broker technology for a domain, enter *No*.

> **Note**
>
> In addition to setting this option, in Integration Broker you must also activate the domain on which the pub/sub server resides before you can use the pub/sub system.
>
> See Integration Broker Administration.

Quick Server Option

Enter *No* if very few clients access the domain and concurrency is not an issue. Enter *Yes* to enable PSQCKSRV in situations where concurrency and optimal transaction throughput are needed.

Query Servers Option

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

Jolt Option

The Jolt listener is required to support the PeopleSoft Internet Architecture by enabling transmission between the web server and the application server.

Jolt Relay Option

See Using Jolt Internet Relay.

WSL

Configures the Workstation Listener for Development Environment (Windows) workstation connections.

PC Debugger Option

Enter *Yes* to debug PeopleCode programs with the current domain.

Event Notification Option

Select *Yes* to start the PSRENSRV servers.

See PSRENSRV Options.

MCF Servers Option

Select *Yes* to start the Multi Channel Framework servers.

See MultiChannel Framework.

Performance Collator Option

If the domain is servicing a Performance Monitor database, select *Yes* to start the PSPPMSRV servers.

See PSPPMSRV Options.

Analytic Servers Option

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

Domains Gateway Option

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

Push Notifications Option

Enables push notification framework for the application server domain.

See Fluid User Interface Developer’s Guide: Configuring the System for Push Notifications.

MCP Server Feature for the Chat UI

Enter *Yes* to enable the MCP option for use with Chat User Interface. Model Context Protocol (MCP) is an open-source standard that enables AI applications to connect seamlessly with external systems.

Enable the MCP Server feature for the Chat UI on only a single application server domain (similar to IB support configurations). The feature is only available for Linux application server domains.

> **Note**
>
> Use this feature only when working with the PeopleSoft-delivered Chat UI. If you want to implement a generic MCP deployment that exposes PeopleSoft APIs to external LLM agents, refer to Portal Technology: Configuring the MCP Stream Server.

See Portal Technology: Working with MCP Server and Implementing MCP Server for Chat UI

Setting Domain-Level Environment Variables

This section provides an overview of how to set domain-level environment variables and work with these settings.

Understanding Domain-Level Environment Settings

Environment settings can be set at the following levels:

- Operating System: At this level, the environment settings apply to all processes running at that time on the host, such as PS_CFG_HOME.
- Domain: At this level, you can specify settings that apply to only that domain. That is you can override any operating system environment settings, as well as any that come by default from the higher-level configuration file (.UBX).

> **Note**
>
> Setting a variable at the domain-level overrides the variable set in the parent environment. Variables like PS_HOME and PS_CFG_HOME cannot be set at the domain level.

The settings you modify or add for that domain will be saved in the *PS_ENVFILE section one of the following files, depending on the domain type:

- psappsrv.ubx
- psprcsrv.ubx
- pssrchsrv.ubx

When the configuration settings are loaded for that domain, then these settings will also be reflected in the .env file for that domain.

> **Note**
>
> It is not recommended to modify the .env file manually because it will be overwritten by .ubx values if the domain is ever re-configured (selecting the `Load config as shown` option on the Quick-configure menu). Environment settings should be set using the PeopleSoft Domain Environment Settings menu.

Working with Domain-Level Environment Settings

To set domain level environment settings, select *Edit Environment Settings* from the Quick-configure menu for an application server or Process Scheduler server.

The following example displays application server domain settings.

```
--------------------------------------
PeopleSoft Domain Environment Settings
--------------------------------------
     Domain Name: ORCLDM01

     TEMP                    :[{LOGDIR}{FS}tmp]
     TMP                     :[{LOGDIR}{FS]tmp]
     TM_BOOTTIMEOUT          :[120]
     TM_RESTARTSRVTIMEOUT    :[120]
     TM_BOOTPRESUMEDFAIL     :[Y]
     FLDTBLDIR32             :[{$TUXDIR}{FS}udataobj]
     FIELDTBLS32             :[jrep.f32,tpadm]
     ALOGPFX                 :[{LOGDIR}{FS}TUXACCESSLOG]
     COBPATH                 :[{$PS_APP_HOME}\CBLBIN%PS_COBOLTYPE%;
{$PS_HOME}\CBLBIN%PS_COBOLTYPE%]
   # IPC_EXIT_PROCESS        :[1]
     IPC_TERMINATE_PROCESS   :[1]
     PATH                    :[{$PATH};{$Domain Settings\Add to PATH}]
  1) Edit environment variable
  2) Add environment variable
  3) Remove environment variable
  4) Comment / uncomment environment variable
  5) Show resolved environment variables
  6) Save
  h) Help for this menu
  q) Return to previous menu
```

Field or Control	Description

Edit environment variable	Enables you to edit the values of the listed environment settings. After selecting this option, enter the row number of the environment setting you want to edit, and modify the value as needed.
	`Environment variable: TM_BOOTTIMEOUT`
	`Enter value [120]:150`
	For any settings that you modify, the system displays an asterisk next to it (*).

Add environment variable	Use this option to add new environment variables for the domain. After selecting this option, enter the name of the environment variable and its value when prompted.
	`Enter name of environment variable: MY_VARIABLE`
	`Enter value: D:\PSFT_HCM`
	For any settings that you add, the system displays an asterisk next to it (*).

Remove environment variable	Removes any of the currently listed environment settings. After selecting this option, enter the row number of the environment setting you want to remove.
Comment / uncomment environment variable	Use to comment or uncomment the environment setting. When you comment a setting, the system ignores that domain-level setting. Any default, or operating system-level settings would take effect. When you uncomment the setting, the system recognizes the domain-level setting value. After selecting this option, enter the row number of the environment setting you want to comment or uncomment. Environment variables that are commented (ignored by the system), have a # symbol next to it.

Field or Control	Description

Show resolved environment variables	Expands the meta variables for system settings, such as PS_HOME, TUXDIR, and so on, so that you can view the actual file path.
	For example, the following is unresolved:
	`:[{LOGDIR}{FS}TUXACCESSLOG]`
	And the following is resolved:
	`:[D:\PT_SERVER\8.59\appserv\FSDMO856\LOGS\TUXACCESSLOG]`

Save	Select to save the settings you've added or modified.
Help for this menu	Select to display online help information for this interface.

Setting Domain-Level Environment Variables

In most cases, you may not need to modify or add domain-level environment variables.

Some reasons to set a domain-level environment variable are:

- custom PeopleCode may require a domain-level environment variable.
- additional tracing may require additional Tuxedo environment variables, if specified in your Tuxedo documentation.
- an Oracle Support analyst may request a specific environment setting for troubleshooting issues.

> **Note**
>
> PS_HOME, PS_CFG_HOME, PS_APP_HOME and PS_CUST_HOME environment variables *should not* be set at the domain-level and can cause undesirable results if set at the domain level.

The following table describes the default environment settings for the Application Server that are exposed at the domain level:

Environment Setting	Description
TEMP	Path to the temp log access for that domain.
TMP	Path to the temp log access for that domain.
TM_BOOTTIMEOUT	Time allowed before Tuxedo performs a timeout for a domain boot.
TM_RESTARTSRVTIMEOUT	Time allowed before Tuxedo performs a timeout for a server process restart.

Environment Setting	Description
TM_BOOTPRESUMEDFAIL	TM_BOOTPRESUMEDFAIL is an indicator to the Tuxedo system that dictates how the domain should behave if a process fails to start before TM_BOOTTIMEOUT (above) has been reached. When set to *Y*, no further domain processes will be started and any existing domain processes will be terminated. If set to *N*, the boot process will proceed. By default, this is set to*Y* because domains cannot be relied upon to function correctly if any of the domain processes fail to start in a timely fashion.
FLDTBLDIR32	The directory in which field buffer descriptor files are located. This is an internal Tuxedo setting. Modification of this value could result in unpredictable behavior as Tuxedo will be unable to locate its buffer descriptor files.
FIELDTBLS32	The list of the buffer descriptor files that are required by the domain runtime. These files are located in the directory defined by the FLDTBLDIR32 variable. This is an internal Tuxedo setting. Modification of this value could result in unpredictable behavior as Tuxedo may allocate buffers incorrectly.
ALOGPFX	Path to the Tuxedo access log for that domain.
COBPATH	COBOL path for that domain.
IPC_EXIT_PROCESS	Indicates that the process should be shutdown in a graceful fashion, allowing cleanup functions to execute. This is analogous to normal shutdown and is achieved by using the MS C++ ExitProcess function. Typically this value or IPC_TERMINATE_PROCESS is set. These values should be considered to be mutually exclusive.
IPC_TERMINATE_PROCESS	IPC_TERMINATE_PROCESS is an indication that the process should exit immediately. This is analogous to a process kill or forced shutdown. The exit handlers do not have an opportunity to run. This would typically be used in a situation where the process has failed to respond to a normal shutdown instruction. Typically this value or IPC_EXIT_PROCESS is set. These values should be considered to be mutually exclusive.
PATH	Specifies a set of directories where executable programs are located for that domain.

The following table describes the default environment settings for the Process Scheduler Server that are exposed at the domain level:

Environment Setting	Description
COBPATH	COBOL path for that domain.
PATH	Specifies a set of directories where executable programs are located for that domain

Setting the PS_SERVDIR Environment Variable

To set PS_SERVDIR, start PSADMIN and select a domain.

When you select an Application Server domain, the PS_SERVDIR value is set to %PS_CFG_HOME%/appserv/<*Domain Name*>.

When you select a Process Scheduler domain, the PS_SERVDIR value is set to %PS_CFG_HOME%/appserv/prcs/<*Domain Name*>.

Note that you cannot change this value using the normal commands for setting environment variables. This setting applies only to server processes. Do not set PS_SERVDIR for client processes, such as Application Designer.

The PS_SERVDIR environment variable is referenced by several features, such as Application Engine, Integration Broker, and PeopleCode.

Related Topics

- Application Engine: Using the Command Line to Invoke Application Engine Programs
- Integration Broker Administration: Working With the SFTP Target Connector
- PeopleCode API Reference: Working With Relative Paths
- Process Scheduler: Log and Output Directory
- PS/nVision: Using Configuration Manager
- Query: Downloading Queries

Using the PSADMIN Command-Line Interface

This section provides an overview of the PSADMIN command-line interface and discusses how to work with commands.

Understanding the PSADMIN Command-Line Interface

In some cases, you might want to use the PSADMIN command-line interface rather than starting the PSADMIN interface and navigating to a particular menu. The command line offers a direct method of carrying out certain server administration tasks. It also enables you to include PSADMIN actions in scripts, and simplifies the task of creating numerous domains that use default server settings.

PSADMIN has several variations of its basic command-line syntax for miscellaneous activities and server administration, which are described in the following sections.

> **Note**
>
> Before you begin using the PSADMIN commands, you should become familiar with PSADMIN and the components that it controls.

> **Note**
>
> When using the PSADMIN command line and specifying paths containing spaces (such as directory paths on Windows), the paths must be enclosed within quotes.

> **Note**
>
> On Microsoft Windows, whenever you run a script to create or configure a domain (application server or Process Scheduler), a related rmiregistry MS-DOS window is started. You do not have to kill these command windows; these windows will be minimized.

Using the Miscellaneous Commands

Following are the available miscellaneous PSADMIN commands:

Command	Description
`psadmin -h`	Displays command help and syntax.
`psadmin -v`	Displays the PSADMIN version number, as in *Version 8.59*.
`psadmin -env`	Displays your current environment variables.
`psadmin -defaultPS_CFG_HOME`	Displays the current PS_CFG_HOME environment variable.
`psadmin -envsummary`	Displays a summary of PS_HOMEs and domains.

Command	Description
psadmin -replicate -ch <source PS_CFG_HOME> psadmin -replicate -ch <source PS_CFG_HOME> -r	Replicates a PS_CFG_HOME and its configured domains, including a PIA domain, to a new location, where they can be started without manual reconfiguration. The replicate originates from the source PS_CFG_HOME in the command line parameter and copies to the current PS_CFG_HOME that the user is working with as defined either by the value of the PS_CFG_HOME environment variable *or* the default location PSADMIN is using for the operating system. That is, you copy an existing PS_CFG_HOME into the current PS_CFG_HOME used by PSADMIN. If the current PS_CFG_HOME contains any existing domains, the replication process will abort and will display a message. You can overwrite the existing domains in the current PS_CFG_HOME using the –r option. **Note:** If the current PS_CFG_HOME contains any existing domains, ensure that the domains are shut down before you execute the replicate command. If the domains are not shut down, an error is displayed.
psadmin -refresh psadmin -refresh -keepbackup	Refreshes the domains in PS_CFG_HOME and applies the latest configuration settings to the domains. If any of the domains is started, the refresh process will first shut down the domain, refresh it, and then start the domain. A stopped domain will just be refreshed. Use the –keepbackup option to keep a back-up copy of a domain. The back-up copies of a domain are stored in PS_CFG_HOME with the following naming convention: <domain>_<date>_<time>.bak. For example, peoplesoft_081315_2158.bak. **Note:** Using this command does not update the integrationGateway.properties file on the web server. Save the integrationGateway.properties file so that you can replace it after using this command. See Integration Broker Administration: Backing Up the Integration Gateway Properties File.

Storing Parameter Key and its Value in a File

PeopleSoft allows you to store a parameter key and its value in a text file so that you need not enter passwords and configuration settings on the command line. System administrators must ensure that the file containing a parameter and its value is accessible to PSADMIN. When you enter a parameter and its value in a text file, you must follow the syntax that is applicable for the parameter value.

You can use the command line to read a file reference using the following syntax:

psadmin -c *command* -d *domain* -t *template* -@@ *file path and file name*

For example:

```
psadmin -c create -d NEWQ1 -t small -@@ D:\temp\key1.txt
```

The contents of the key1.txt file are as follows:

```
-s Q8588053/ORACLE/QEDMO/QEDMO/TESTSERV/D:\oracle\product\12.1.0\dbhome_1\BIN/
people/password/_____/password
```

> **Note**
>
> Use –@@ if you do not want the text file to be deleted after PSADMIN extracts the file’s contents. Use –@ if you want the text file to be deleted.

When you use –@ file references, consider the following:

- A PSADMIN command can contain multiple –@[@] file references that contain a parameter key-value pair, and you can use a mix of –@@ and –@ within a single command.

  For example:

  ```
  psadmin -c create -d psftdomain -t psfttemplate -@@ D:\temp\set_file.txt[-@
  D:\temp\pset_file.txt] [-@@ D:\temp\env_set.txt]
  ```

- A PSADMIN command can contain nested –@[@] file references.

  For example:

  ```
  psadmin -@@ D:\edw\temp\file1.txt
  ```

  where the contents of file1.txt are as follows:

  ```
  -c create -d NEWQ1 -t small -@@ D:\edw\temp\file2.txt
  ```

  and the contents of file2.txt are as follows:

  ```
  -s Q8588053/ORACLE/QEDMO/QEDMO/TESTSERV/D:\oracle\product\12.1.0\dbhome_1\BIN/
  people/password/_____/password
  ```

- When you have multiple –@ file references in a command, and if any of the –@ file reference fails, PSADMIN stops processing the command after an error is encountered.

  For example:

  ```
  psadmin -c -@@ D:\edw\temp\file3.txt
  ```

  Where the contents of file3.txt are as follows:

  ```
  create -@ D:\edw\temp\file4.txt -@ D:\edw\temp\file4.txt -@
  D:\edw\temp\file5.txt
  ```

  PSADMIN displays the following error:

  ```
  Can't open file D:\edw\temp\file4.txt. Please Check the correct
  file name.
  ```

  In this example, PSADMIN reads the contents of file3.txt, When it encounters an error, it deletes file4.txt and subsequently stops processing the command. PSADMIN does not delete file5.txt if the file exists in the location.

Using the Pure Internet Architecture Commands

The PeopleSoft Pure Internet Architecture (PIA) commands follow this basic syntax:

psadmin –w *command* –d *PIA_domain*

For example:

psadmin –w start –d peoplesoft

> **Note**
>
> The examples in this table include line feeds for readability.

Command	Example	Result of the Example
start	`psadmin -w start` `-d peoplesoft`	Starts the peoplesoft domain.
shutdown	`psadmin -w shutdown` `-d peoplesoft`	Stops the peoplesoft domain.
shutdown!	`psadmin -w shutdown!` `-d peoplesoft`	Performs a forced shutdown of the peoplesoft domain.
status	`psadmin -w status` `-d peoplesoft`	Displays the status of the domain (started, stopped, and so on).
remove	`psadmin -w remove` `-d peoplesoft`	Deletes the domain.

Command	Example	Result of the Example

import	`psadmin -w import <source PS_CFG_HOME>`
	
	`psadmin -w import <source PS_CFG_HOME> -d peoplesoft`
	
	`psadmin -w import <source PS_CFG_HOME> -d peoplesoft -n peoplesoft1`
		Imports the peoplesoft domain from a specified PS_CFG_HOME location, <source PS_CFG_HOME>.
		You may use the following options as required:
		• Use the –d option to import a specific domain if multiple domains exist in a source PS_CFG_HOME.
		• Use the –n option if you want to specify a name for the imported domain.
		If the –n option is not used, the database name is used as the default name for the imported domain.
		• Use the –r option to replace an existing domain folder in the current PS_CFG_HOME. If a domain folder exists and the –r option is not used, an error message is displayed. Use this option with all of the –import options.

Using the PeopleSoft Pure Internet Architecture Create Command

Use the following syntax to create a new PIA domain:

```
psadmin -w create -d PIA_domain -c c_set
```

The create command creates a PIA domain, using the specified configuration settings.

- *PIA_domain* — Enter the name of a PIA domain that you want to create, for example, peoplesoft.
- *c_set* (configuration settings) — Use the following order and format:

  In Microsoft Windows, the values must be separated by slashes:

  ```
  WebServer/WebServerRootDir/WebServerLoginId/WebServerLoginPwd/WebsiteName/
  AppserverHost/JSL Port/HTTP Port/HTTPS Port/AuthenticationTokenDomain/Web
  profile/WebProfUserId/WebProfUserPwd/IntegrGatewayId/IntegrGatewayPwd/
  AppsrvDomConnPwd/SSLKeystorePwd/RepDir
  ```

  In UNIX, the values must be separated by percent signs:

  ```
  WebServer%WebServerRootDir%WebServerLoginId%WebServerLoginPwd%WebsiteName%A
  ppserverHost%JSL Port%HTTP Port%HTTPS Port%AuthenticationTokenDomain%Web
  profile%WebProfUserId%WebProfUserPwd%IntegrGatewayId%IntegrGatewayPwd%Appsr
  vDomConnPwd%SSLKeystorePwd%RepDir
  ```

> **Note**
>
> To indicate an empty parameter, enter five underscore characters (_____).

See Creating a PIA Domain for descriptions of the configuration settings.

Specifying PIA Domain Configuration Settings

You can use the command line to set selected domain settings using the following syntax:

```
psadmin –w configure –d PIA_domain –c c_set –p p_set
```

For example:

```
psadmin –w configure –d peoplesoft –c 512/512/50/example.com –p 80/443
```

- *PIA_domain* — Enter the name of a PIA domain that you want to create, for example, peoplesoft.
- *c_set* (configuration settings) — Use the following order and format:

  minHeapSize/maxHeapSize/maxThreads/authTokenDomain
- *p_set* (port settings) — Use the following order and format:

  httpPort/httpsPort

See Configuring a PIA Domain for descriptions of the configuration settings.

Specifying PIA Site Configuration Settings

You can use the command line to set selected site settings using the following syntax:

```
psadmin –w configure –d PIA_domain –s site –c c_set
```

For example:

```
psadmin –w configure –d peoplesoft –s ps –c SERVER01:9000/PROD/Enabled/
D:\psreports/PS/password/password
```

- *PIA_domain* — Enter the name of a PIA domain that you want to create, for example, peoplesoft.
- *site* — Enter the web site name; for example, ps.
- *c_set* (configuration settings) — Use the following order and format:

  appSrvConnString/webProfile/pooling/reportPath/webProfUser/webProfPword/appSrvConnPword

The following table describes the configuration settings:

PIA configuration (c_set) settings	Description
appSrvConnString	Application server connect string, such as host:jolt port.
webProfile	The web profile the site will use (PROD, DEV, and so on).
pooling	Indicate whether Jolt pooling should be enabled. (Enabled/Disabled)
reportPath	The reports path, such as D:\psreports.
webProfUser	The user required for accessing the web profile.
webProfPword	The password for the web profile user.

PIA configuration (c_set) settings	Description
appSrvConnPword	If using a domain connection password, enter that value.

Using the Application Server Commands

For application server administration, PSADMIN has two syntax formats — one for creating new application server domains, and the other for administering existing domains.

Using the Application Server Create Command

Use the following syntax to create a new application server domain:

```
psadmin -c create -d appsrv_domain -t template -s s_set [-p p_set] [-cfg cfg_set
| –cfg_from_file cfg_file_path] [-env env_set]
```

The create command creates an application server configuration file with the specified domain name, using the specified configuration template.

Following is an example of the application server create command, including startup settings, ports settings, and a configuration file:

```
D:\PT\appserv>psadmin -c create -d HRDOM01 -t small -s PT858/ORACLE/QEDMO/
QEDMO
/TESTSRV2/"c:\oracle\product\12.1.0\client_1\bin"/people/password/_____/
ENCRYPT
-p 7200/9020/9200 -cfg_from_file d:\temp\app_cfg_file.txt
```

The other options are described below.

The *appsrv_domain* parameter must be the name of an application server domain that you want to create, for example, HCMDMO.

The *template* parameter must have one of the following values:

- *small*
- *medium*
- *large*
- *developer*

The *s_set* parameter is an optional string of startup values which provide initial configuration settings that you would otherwise specify on the PSADMIN application server Quick-Configure menu. You must enter the startup string as follows.

- In Microsoft Windows, the values must be separated by slashes (/):

```
DBNAME/DBTYPE/OPR_ID/OPR_PSWD/DOMAIN_ID/ADD_TO_PATH/DB_CNCT_ID/
DB_CNCT_PSWD/SERVER_NAME/DOM_CONN_PWD/SEC_PRINCIPAL_PASSWORD/
ENABLE_REMOTE_ADMIN/REMOTE_ADMIN_PORT/REMOTE_ADMIN_USRID/REMOTE_ADMIN_PSWD/
{ENCRYPT|NOENCRYPT}/REMOTE_ADMIN_SSL_ENABLED/REMOTE_ADMIN_SSL_KEYSTORE/
REMOTE_ADMIN_SSL_KEYSTORE_PWD/REMOTE_ADMIN_SSL_KEYSTORE_TYPE/
REMOTE_ADMIN_SSL_TRUSTSTORE/REMOTE_ADMIN_SSL_TRUSTSTORE_PWD/
REMOTE_ADMIN_SSL_TRUSTSTORE_TYPE
```

- In UNIX, the values must be separated by percent signs (%):

```
DBNAME%DBTYPE%OPR_ID%OPR_PSWD%DOMAIN_ID%ADD_TO_PATH%DB_CNCT_ID%DB_CNCT_PSWD
%SERVER_NAME%DOM_CONN_PWD%SEC_PRINCIPAL_PASSWORD%ENABLE_REMOTE_ADMIN%REMOTE
_ADMIN_PORT%REMOTE_ADMIN_USRID%REMOTE_ADMIN_PSWD%{ENCRYPT|NOENCRYPT}
%REMOTE_ADMIN_SSL_ENABLED%REMOTE_ADMIN_SSL_KEYSTORE%REMOTE_ADMIN_SSL_KEYSTO
RE_PWD%REMOTE_ADMIN_SSL_KEYSTORE_TYPE%REMOTE_ADMIN_SSL_TRUSTSTORE%REMOTE_AD
MIN_SSL_TRUSTSTORE_PWD%REMOTE_ADMIN_SSL_TRUSTSTORE_TYPE
```

Important

You must enter these values in the order shown. You can omit required values only by truncating the string from right to left. For example, you can specify *DBNAME/DBTYPE*, but you can't specify *DBNAME/DOMAIN_ID*.

These startup settings all have default values if you omit any of them. The default values are generally the values you provided when setting up your PeopleSoft environment, and are the same as they would initially appear on the PSADMIN application server Quick-Configure menu.

The following table describes the startup settings:

Startup (*s_set*) Settings	Description
DBNAME	Enter the name of the database to which the application server will connect. This is the same as the *DBName* parameter in the Startup section of the psappsrv.cfg file. **Note:** If you don't include the *s_set* parameter, the value of this setting is the same as the domain name that you specify in the command.
DBTYPE	Enter the database type. Valid values are ORACLE, MICROSFT, and DB2ODBC. This is the same as the *DBType* parameter in the Startup section of the psappsrv.cfg file. **Note:** Notice the spelling of MICROSFT. DB2ODBC is the database type for Db2 z/OS.
OPR_ID	Enter the user ID, such as QEDMO, PS, or VP1, for the domain to use to connect to the database. This is the same as the *UserId* parameter in the Startup section of the psappsrv.cfg file.
OPR_PSWD	Enter the user password that is associated with the specified user ID. This is the same as the *UserPswd* parameter in the Startup section of the psappsrv.cfg file.
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 because the Tuxedo Web Monitor uses it to identify application server domains on each machine. This is the same as the *Domain ID* parameter in the Domain Settings section of the psappsrv.cfg file.

Startup (*s_set*) Settings	Description
ADD_TO_PATH	(Optional) Enter the directory path that contains your connectivity software or database drivers. This is the same as the *Add to PATH* parameter in the Domain Settings section of the psappsrv.cfg file. **Note:** If this value contains spaces, it must be in double quotes (`" "`). For example:`"c:\Program Files"`. **Important:** If you want this setting to be blank, but you can't truncate the string to this point (you still need to specify a value for *CNCT_ID*), you can specify a value of "_____" (five underscores without the quotes) in this position. PSADMIN interprets this as a blank value.
CNCT_ID	Enter the connect ID, which is required for all platforms. This is the same as the *ConnectId* parameter in the Startup section of the psappsrv.cfg file. See Security Administration: Connect ID.
CNCT_PSWD	Enter the password that is associated with the connect ID. This is the same as the *ConnectPswd* parameter in the Startup section of the psappsrv.cfg file.
SERVER_NAME	(Optional) If your RDBMS requires that you specify the server name on which the database resides, enter the appropriate server name. This is the same as the *ServerName* parameter in the Startup section of the psappsrv.cfg file. **Important:** If you want this setting to be blank, but you can't truncate the string to this point (you still need to specify the ENCRYPT setting), you can specify a value of "_____" (five underscores without the quotes) in this position. PSADMIN interprets this as a blank value.
ENCRYPT | NOENCRYPT	Specify ENCRYPT to encrypt the values of the *UserPswd* and *ConnectPswd* parameters in the psappsrv.cfg file. If you specify NOENCRYPT, these values appear in clear text in the file. ENCRYPT is the default value.
SEC_PRINCIPAL_PASSWORD	Enter a password for use with Oracle wallet.

Startup (*s_set*) Settings	Description
ENABLE_REMOTE_ADMIN	Specify whether to enable embedded JMX agents for PeopleSoft Health Center. Enter *1* to enable remote administration, and *0* to disable remote administration. The default is enabled. When remote administration is enabled, you must also specify values for REMOTE_ADMIN_PORT, REMOTE_ADMIN_USRID, and REMOTE_ADMIN_PSWD. If you do not explicitly disable remote administration, and you do not provide values for REMOTE_ADMIN_PORT, REMOTE_ADMIN_USRID, and REMOTE_ADMIN_PSWD, the system will generate values for the user ID and password and place the encrypted values in the PSTOOLS section of the psappsrv.cfg file. See Performance Monitor: Enabling Monitoring of a Component or Domain
REMOTE_ADMIN_PORT	Enter a port number to use for remote administration of domain processes. All embedded JMX agents will use the port in the JMX service URL.
REMOTE_ADMIN_USRID	Enter a user name for remote administration connections.
REMOTE_ADMIN_PSWD	Enter a password for remote administration connections.
REMOTE_ADMIN_SSL_ENABLED	Specify whether to enable SSL for JMX agents. Enter *1* to enable SSL, and *0* to disable SSL. The default is disabled. When you enable SSL for JMX agents, you must also specify the location, password, and type for the keystore and truststore.
REMOTE_ADMIN_SSL_KEYSTORE	Specify the location of the keystore for SSL.
REMOTE_ADMIN_SSL_KEYSTORE_PWD	Specify the password for the SSL keystore. The password must be encrypted with PSCipher. See Security Administration: Using the PSCipher Utility
REMOTE_ADMIN_SSL_KEYSTORE_TYPE	Specify the keystore type, PKCS12 or JKS.
REMOTE_ADMIN_SSL_TRUSTSTORE	Specify the location of the truststore for SSL.
REMOTE_ADMIN_SSL_TRUSTSTORE_PWD	Specify the password for the SSL truststore. The password must be encrypted with PSCipher. See Security Administration: Using the PSCipher Utility
REMOTE_ADMIN_SSL_TRUSTSTORE_TYPE	Specify the truststore type, PKCS12 or JKS.

The *p_set* parameter is an optional string of port numbers that you would otherwise specify on the PSADMIN application server Quick-Configure menu. Typically, you include this parameter only if you have more than one domain on the same application server machine or if you need to provide a specific value due to your environment or testing needs. Otherwise, you should accept the defaults for easy configuration.

You must specify the port numbers as follows:

- In Microsoft Windows, the values must be separated by slashes.

  WSL_PORT/JSL_PORT/JRAD_PORT/WSL_SSL_PORT/JSL_SSL_PORT

- In UNIX, the values must be separated by percent signs.

  WSL_PORT%JSL_PORT%JRAD_PORT%WSL_SSL_PORT%JSL_SSL_PORT

Important

You must enter these values in the order shown. You can omit values only by truncating the string from right to left. For example, you can specify *WSL_PORT/JSL_PORT*, but you can't specify *WSL_PORT/JRAD_PORT*. These port numbers all have default values if you omit any of them. The default values are the values you provided when setting up your PeopleSoft environment, and are the same as they would initially appear on the PSADMIN application server Quick-Configure menu.

The following table describes the port settings:

Port (*p_set*) Setting	Description
WSL_PORT	Workstation listener port number. This is the same as the *Port* parameter in the Workstation Listener section of the psappsrv.cfg file. **Note:** Specify this value only if you intend for the domain to support Windows workstations connecting in the development environment.
WSL_SSL_PORT	SSL port for Workstation listener. This is the same as the *SSL PORT* parameter in the Workstation Listener section of the psappsrv.cfg file.
JSL_PORT	Jolt port number. This is the same as the *Port* parameter in the JOLT Listener section of the psappsrv.cfg file.
JSL_SSL_PORT	SSL port for Jolt. This is the same as the *SSL PORT* parameter in the JOLT Listener section of the psappsrv.cfg file.
JRAD_PORT	Jolt internet relay port number. This is the same as the *Listener Port* parameter in the JOLT Relay Adapter section of the psappsrv.cfg file. **Note:** Specify this value only if you intend for the domain to support browser deployment, *and* your web server resides on a separate machine from the application server.

Supply configuration settings in a string with the *cfg_set* parameter or reference a file with the *cfg_file_path* parameter.

The *cfg_set* parameter is an optional string of configuration settings from the psappsrv.cfg file. Use the number sign (#) to separate each setting, and surround the string with double quotes if any of the values include spaces. For example:

```
"[PSAPPSRV]/Min Instances=2#[PSAPPSRV]/Max Instances=8"
```

If you include a password that has a number sign (#), you must precede it with a slash (\). For example:

```
"[Security]/DomainConnectionPwd=password\##[Workstation Listener]/Port=7000"
```

Alternatively, you can supply configuration settings in a file. Specify the path and file name with the parameter *cfg_file_path*. Use this option rather than *cfg_set* if you are supplying passwords that contain special characters. Each line specifies a psappsrv.cfg file section, a parameter and value, with this format:

```
[<Section>]/<Param Name>=<Param Value>
[<Section>]/<Param Name>=<Param Value>
```

For example:

```
[Domain Settings]/Domain ID=APPDOM
[PSAPPSRV]/Min Instances=2
[PSAPPSRV]/Max Instances=8
[PSAPPSRV]/Max Fetch Size=15000
[Security]/DomainConnectionPwd=password
[JOLT Listener]/Port=9033
[JOLT Listener]/Address=0.0.0.0
[Workstation Listener]/Port=7000
```

The configuration parameters are described elsewhere in this document, organized by section name. See Understanding Application Server Domain Parameters.

The *env_set* specifies domain environment settings, having the following format:

```
ENVAR_NAME1=ENVVAR_VALUE1#ENVAR_NAME2=ENVVAR_VALUE2
```

For example:

```
TM_BOOTTIMEOUT=120#TM_RESTARTSRVTIMEOUT=120#TM_BOOTPRESUMEDFAIL=Y
```

When you launch the command, you see progress messages similar to the following:

```
Copying application server configuration files...
Copying [D:\PT\appserv\small.cfx] to [D:\PT_SERVER\8.59\appserv\HRDOM01\
psappsrv.cfg]
Stripping Annotations...
Copying [D:\PT\appserv\small.cfx] to [D:\PT_SERVER\8.59\appserv\HRDOM01\
psappsrv.cfx]
Copying Jolt repository file...
Domain created.
Performing load prechecks ...
```

```
Loading validation table...
  setting DBName=PT858DMO
  setting DBType=ORACLE
  setting UserId=QEDMO
  setting UserPswd=encrypted_password
  setting ConnectId=people
  setting ConnectPswd=encrypted_password
  setting ServerName=
  setting Port=7200
  setting Port=9020
  setting Listener Port=9200
  setting Domain ID=TESTSRV2
  setting Add to PATH=c:\oracle\product\12.1.0\client_1\bin
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]
Spawning disabled for server PSAPPSRV.
WARNING: PSSAMSRV is configured with Min instance set to 1. To avoid loss of
service, configure Min instance to atleast 2.
Configuration file successfully created.
CFG setting changes completed, loading configuration...
Domain configuration complete.
```

Using the Application Server Administration Commands

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

`psadmin -c command -d appsrv_domain`

The *appsrv_domain* parameter must be the name of an application server domain that you want to administer, for example, HCMDMO.

The valid values of the *command* parameter are as follows:

> **Note**
>
> The examples in this table include line feeds for readability.

Command	Example	Result of the Example
start	`psadmin -c start -d PSDMO`	Starts an application server domain named PSDMO.
parallelboot	`psadmin -c parallelboot -d PSDMO`	Starts an application server domain named PSDMO, using the parallel boot option.

Command	Example	Result of the Example
configure	`psadmin -c configure -d PSDMO [-cfg cfg_set] [-cfg_from_file cfg_file_path] [-u feature_set]`	Reloads the domain configuration for the PSDMO domain. See the following section Using the Application Server Configure Command.
pslist	`psadmin -c pslist -d PSDMO`	Displays the processes that have been booted for the PSDMO domain. This includes the system process ID for each process.
stop	`psadmin -c stop -d PSDMO`	Shuts down the PSDMO application server domain, by using a normal shutdown method. In a normal shutdown, the domain waits for users to complete their tasks and turns away new requests before terminating all of the processes in the domain.
kill	`psadmin -c kill -d PSDMO`	Shuts down the PSDMO application server domain by using a forced shutdown method. In a forced shutdown, the domain *immediately* terminates all of the processes in the domain.
sstatus	`psadmin -c sstatus -d PSDMO`	Displays the Tuxedo processes and PeopleSoft server processes that are currently running in the PSDMO application server domain.
cstatus	`psadmin -c cstatus -d PSDMO`	Displays the currently connected users in the PSDMO application server domain.
qstatus	`psadmin -c qstatus -d PSDMO`	Displays status information about the individual queues for each server process in the PSDMO application server domain.
preload	`psadmin -c preload -d PSDMO`	Preloads the server cache for the PSDMO domain.
cleanipc	`psadmin -c cleanipc -d PSDMO`	Cleans the IPC resources for the PSDMO domain.

Command	Example	Result of the Example
purge	`psadmin -c purge -d PSDMO -noarch` `psadmin -c purge -d PSDMO -arch` *`archive_directory`* `psadmin -c purge -d PSDMO -log` *`log_comments`*	Purges the cache for the PSDMO domain. You may use the following options as required: • Use the -noarch option if you do not want to archive the purged cache. • Use the -arch *archive_directory* option to quarantine the purged cache in the specified location. • Use -log *log_comments* to add the specified comments to the purge cache log entry.
import	`psadmin -c import PS_CFG_HOME -n NEWAPPDOM`	Imports the domain configuration from a specified PS_CFG_HOME location or a specified configuration file. See the following section Using the Application Server Import Commands for examples.
addmsgsrv	`psadmin -c addmsgsrv -d PSDMO -s MSGSRV -t type ch1,ch2,ch3`	Add a message server named MSGSRV (-s) to the domain PSDMO. Specify PUB, SUB, or BRK as the type (-t). The channels for the server are ch1, ch2, and ch3.
rmmsgsrv	`psadmin -c rmmsgsrv -d PSDMO -s MSGSRV`	Remove a message server named MSGSRV (-s) from the domain PSDMO.

See Integration Broker Administration: Understanding Messaging Server Administration

Using the Application Server Configure Commands

Use these commands to configure an existing application server domain.

```
psadmin -c configure -d appsrv_domain [-cfg cfg_set | -cfg_from_file
cfg_file_path] [-u feature_set]
```

Supply configuration settings in a string with *cfg_set*. or reference a file with *cfg_file_path*. See the discussion earlier in this section.

Use the -u *feature_set* option to enable or disable a set of features in the existing domain. To construct *feature_set*:

1. Access *PS_CFG_HOME*/appserv/*appsrv_domain*/psappsrv.ubx.
2. In the *FEATURES section, make a note of the {"-define-"} field for the feature you want to include.

   For example, the {"-define-"} field for Analytic Servers is {ANALYTICSRV}.

   ```
   *FEATURES
   {"-label-"},{"-define-"},{"-on-"},{"-servers-"},{"-Oracle Env Manager
   ```

```
Label-"}
...
{Analytic Servers},{ANALYTICSRV},{Yes},{PSANALYTICSRV},{Analytic Servers
(PSANALYTICSRV)}
```

3. Use the value from the {"-define-"} field in this format for *feature_set*.

   Use a slash (/) on Microsoft Windows to separate features.

   {FEATURE}=Yes|No/{FEATURE}=Yes|No...

   Use a percent sign (%) on UNIX to separate features.

   {FEATURE}=Yes|No%{FEATURE}=Yes|No...

For example, use this command to disable WSL and enable Analytic Servers in an application server domain.

On Microsoft Windows:

```
psadmin -c configure -d appsrv_domain -u "{WSL}=No/{ANALYTICSRV}=Yes"
```

On UNIX:

```
psadmin -c configure -d appsrv_domain -u "{WSL}=No%{ANALYTICSRV}=Yes"
```

Using the Application Server Import Commands

Use these commands to import an application server domain configuration:

- To import a domain from the location PS_CFG_HOME, use the following command.

  Use this command if there is a single domain in PS_CFG_HOME.

  `psadmin -c import ps_cfg_home_path`

- To import a domain when PS_CFG_HOME has more than one domain, use the following command.

  The imported domain has the same name as the original domain.

  `psadmin -c import ps_cfg_home_path –d appsrv_domain`

- To import a domain to a new name, from a multi-domain PS_CFG_HOME, use the following command:

  `psadmin -c import ps_cfg_home_path –d appsrv_domain –n new_domain_name`

- To import a domain to a new name, from a multi-domain PS_CFG_HOME.

  Specify the path to a template file and use the following command.

  `psadmin -c import ps_cfg_home_path –d appsrv_domain –n new_domain_name –t template_path`

- Use a configuration file to import a domain to a new name.

  Supply the path and name for psappsrv.cfg for config_file_path.

  `psadmin -c import config_file_path –n new_domain_name`

- Use a configuration file to import a domain to a new name.

  Specify the path and name for psappsrv.cfg for config_file_path, and specify the path to a template file.

```
psadmin -c import config_file_path –n new_domain_name –t template_path
```

Using the Process Scheduler Commands

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

Using the Process Scheduler Create Command

Use the following syntax to create a new Process Scheduler configuration:

```
psadmin -p create -d prcs_domain -t template -ps ps_set [-cfg cfg_set | -
cfg_from_file cfg_file_path] [-env env_set] [-s silent]
```

The PSADMIN create command creates a Process Scheduler configuration file for the specified database, using the specified configuration template.

The *template* parameter must be the name of a .cfx file located in *PS_HOME*\appserv\prcs, without the extension. This represents the operating system platform on which you're running PeopleSoft Process Scheduler. For example, to use the template file called windows.cfx on a Microsoft Windows machine, specify the value *windows*.

The *ps_set* parameter is an optional string of startup values which provide initial configuration settings that you would otherwise specify on the PSADMIN Process Scheduler Quick-Configure menu. You must enter the startup string as follows:

- In Microsoft Windows, the values must be separated by slashes.

  ```
  DBNAME/DBTYPE/PRCSSERVER/OPR_ID/OPR_PSWD/DB_CNCT_ID/DB_CNCT_PSWD/
  SERVER_NAME/LOGOUT_DIR/SQRBIN/ADD_TO_PATH/DBBIN/DOM_CONN_PWD/
  ENABLE_REMOTE_ADMIN/REMOTE_ADMIN_PORT/REMOTE_ADMIN_USRID/REMOTE_ADMIN_PSWD/
  {ENCRYPT|NOENCRYPT}/REMOTE_ADMIN_SSL_ENABLED/REMOTE_ADMIN_SSL_KEYSTORE/
  REMOTE_ADMIN_SSL_KEYSTORE_PWD/REMOTE_ADMIN_SSL_KEYSTORE_TYPE/
  REMOTE_ADMIN_SSL_TRUSTSTORE/REMOTE_ADMIN_SSL_TRUSTSTORE_PWD/
  REMOTE_ADMIN_SSL_TRUSTSTORE_TYPE
  ```

- In UNIX, the values must be separated by commas.

  ```
  DBNAME,DBTYPE,PRCSSERVER,OPR_ID,OPR_PSWD,DB_CNCT_ID,DB_CNCT_PSWD,SERVER_NAM
  E,LOGOUT_DIR,SQRBIN,ADD_TO_PATH,DOM_CONN_PWD,ENABLE_REMOTE_ADMIN,REMOTE_ADM
  IN_PORT,REMOTE_ADMIN_USRID,REMOTE_ADMIN_PSWD,{ENCRYPT|
  NOENCRYPT},REMOTE_ADMIN_SSL_ENABLED,REMOTE_ADMIN_SSL_KEYSTORE,REMOTE_ADMIN_
  SSL_KEYSTORE_PWD,REMOTE_ADMIN_SSL_KEYSTORE_TYPE,REMOTE_ADMIN_SSL_TRUSTSTORE
  ,REMOTE_ADMIN_SSL_TRUSTSTORE_PWD,REMOTE_ADMIN_SSL_TRUSTSTORE_TYPE
  ```

> **Note**
>
> The UNIX syntax does not include the DBBIN setting.

Important

You must enter these values in the order shown. You can omit required values only by truncating the string from right to left. For example, you can specify `DBNAME/DBTYPE`, but you can't specify `DBNAME/LOGOUT_DIR`.

These startup settings all have default values if you omit any of them. The default values are generally the values you provided when setting up your PeopleSoft environment, and are the same as they would initially appear on the PSADMIN Process Scheduler Quick-Configure menu.

Note

Because these PeopleSoft Process Scheduler settings are already documented in the *PeopleSoft Process Scheduler* product documentation, this section provides only a basic overview of the relationship between the settings on the command line and the equivalent settings on the PSADMIN Process Scheduler Quick-Configure menu.

The following table describes the startup settings:

Startup (*ps_set*) Setting	Description
DBNAME	This is the equivalent of the *DBName* parameter on the PSADMIN Process Scheduler Quick-Configure menu. **Note:** If you don't include the *ps_set* parameter, the value of this setting is the same as the database name that you specify in the command.
DBTYPE	This is the equivalent of the *DBType* parameter on the PSADMIN Process Scheduler Quick-Configure menu.
PRCSSERVER	This is the equivalent of the *PrcsServer* parameter on the PSADMIN Process Scheduler Quick-Configure menu. The server name must begin with a letter and include a maximum of 8 characters.
OPR_ID	This is the equivalent of the *UserId* parameter on the PSADMIN Process Scheduler Quick-Configure menu.
OPR_PSWD	Enter the user password that is associated with the specified user ID. This is the equivalent of the *UserPswd* parameter on the PSADMIN Process Scheduler Quick-Configure menu.
CNCT_ID	This is the equivalent of the *ConnectId* parameter on the PSADMIN Process Scheduler Quick-Configure menu.
CNCT_PSWD	This is the equivalent of the *ConnectPswd* parameter on the PSADMIN Process Scheduler Quick-Configure menu.

Startup (*ps_set*) Setting	Description
SERVER_NAME	(Optional) This is the equivalent of the *ServerName* parameter on the PSADMIN Process Scheduler Quick-Configure menu. **Important:** If you want this setting to be blank, but you can't truncate the string to this point (you still need to specify a value for *LOGOUT_DIR*), you can specify a value of "_____" (five underscores without the quotes) in this position. PSADMIN interprets this as a blank value.
LOGOUT_DIR	This is the equivalent of the *Log/Output Dir* parameter on the PSADMIN Process Scheduler Quick-Configure menu. **Note:** If this value contains spaces, it must be in double quotes (`" "`). For example:`"c:\psft app\log_output"`.
SQRBIN	This is the equivalent of the *SQRBIN* parameter on the PSADMIN Process Scheduler Quick-Configure menu. **Note:** If this value contains spaces, it must be in double quotes (`" "`). For example:`"C:\PeopleTools\bin\sqr\MSS\binw"`.
ADD_TO_PATH	(Optional) This is the equivalent of the *AddToPATH* parameter on the PSADMIN Process Scheduler Quick-Configure menu. **Note:** If this value contains spaces, it must be in double quotes (`" "`). For example:`"%WINDIR%\SYSTEM32;c:\Program Files"`.
DBBIN	(Windows only) This is the equivalent of the *DBBIN* parameter on the PSADMIN Process Scheduler Quick-Configure menu. **Note:** If this value contains spaces, it must be in double quotes (`" "`). For example:`"C:\my apps\db\MSSQL\Binn"`.
ENCRYPT | NOENCRYPT	Specify ENCRYPT to encrypt the values of the *UserPswd* and *ConnectPswd* parameters in the psprcs.cfg file. If you specify NOENCRYPT, these values appear in clear text in the file. ENCRYPT is the default value.

Startup (*ps_set*) Setting	Description
ENABLE_REMOTE_ADMIN	Specify whether to enable embedded JMX agents for PeopleSoft Health Center. Enter *1* to enable remote administration, and *0* to disable remote administration. The default is enabled. When remote administration is enabled, you must also specify values for REMOTE_ADMIN_PORT, REMOTE_ADMIN_USRID, and REMOTE_ADMIN_PSWD. If you do not explicitly disable remote administration, and you do not provide values for REMOTE_ADMIN_PORT, REMOTE_ADMIN_USRID, and REMOTE_ADMIN_PSWD the system will generate values for the user ID and password and place the encrypted values in the PSTOOLS section of the psprcs.cfg file. See Performance Monitor: Enabling Monitoring of a Component or Domain
REMOTE_ADMIN_PORT	Enter a port number to use for remote administration of domain processes. All embedded JMX agents will use the port in the JMX service URL.
REMOTE_ADMIN_USRID	Enter a user name for remote administration connections.
REMOTE_ADMIN_PSWD	Enter a password for remote administration connections.
REMOTE_ADMIN_SSL_ENABLED	Specify whether to enable SSL for JMX agents. Enter *1* to enable SSL, and *0* to disable SSL. The default is disabled. When you enable SSL for JMX agents, you must also specify the location, password, and type for the keystore and truststore.
REMOTE_ADMIN_SSL_KEYSTORE	Specify the location of the keystore for SSL.
REMOTE_ADMIN_SSL_KEYSTORE_PWD	Specify the password for the SSL keystore. The password must be encrypted with PSCipher. See Security Administration: Using the PSCipher Utility
REMOTE_ADMIN_SSL_KEYSTORE_TYPE	Specify the keystore type, PKCS12 or JKS.
REMOTE_ADMIN_SSL_TRUSTSTORE	Specify the location of the truststore for SSL.
REMOTE_ADMIN_SSL_TRUSTSTORE_PWD	Specify the password for the SSL truststore. The password must be encrypted with PSCipher. See Security Administration: Using the PSCipher Utility
REMOTE_ADMIN_SSL_TRUSTSTORE_TYPE	Specify the truststore type, PKCS12 or JKS

Supply configuration settings in a string with the *cfg_set* or reference a file with the *cfg_file_path*.

The *cfg_set* parameter is an optional string of configuration settings from the psprcs.cfg file. Use the number sign (#) to separate each setting, and surround the string with double quotes if any of the values include spaces. For example:

```
"[Trace]/TracePPM=1#[Trace]/TracePC=8#[Process Scheduler]/Max Reconnect
Attempt=20"
```

If you include a password that has a number sign (#), you must precede it with a backslash (\). For example:

```
"[Security]/DomainConnectionPwd=password\##[Trace]/TracePPM=1#[Trace]/
TracePC=8#[Process Scheduler]/Max Reconnect Attempt=20"
```

Alternatively, you can supply configuration settings in a file. Specify the path and file name with the parameter *cfg_file_path*. Use this option rather than *cfg_set* if you are supplying passwords that contain special characters. Each line specifies a psprcs.cfg file section, a parameter and value, with this format:

```
[<Section>]/<Param Name>=<Param Value>
[<Section>]/<Param Name>=<Param Value>
```

For example:

```
[Trace]/TracePPM=1
[Trace]/TracePC=8
[Process Scheduler]/Max Reconnect Attempt=20
```

The *env_set* specifies domain environment settings, having the following format:

```
ENVAR_NAME1=ENVVAR_VALUE1#ENVAR_NAME2=ENVVAR_VALUE2
```

For example:

```
COBPATH=C:\psft\cblpath
```

Following is an example of the Process Scheduler create command with start settings:

```
psadmin -p create -d PSHRDB1 -t nt -ps PSHRDB1/MICROSFT/PSNT/
PS/password/people/password/_____/"c:\psft app\log_output"/
c:\psfthr\bin\sqr\MSS\binw/
c:\WINNT\SYSTEM32//c:\apps\db\mssql\binn/ENCRYPT
```

Using the Process Scheduler Administration Commands

Use the following syntax to administer an existing Process Scheduler configuration:

```
psadmin -p command -d database
```

The *database* parameter must be the name of a database that's associated with a PeopleSoft Process Scheduler Server Agent, for example, PSHRDMO.

The valid values of the *command* parameter are as follows:

Command	Example	Result of the Example
list	`psadmin -p list`	Lists all domains.
start	`psadmin -p start -d psdmo`	Starts a Process Scheduler.
stop	`psadmin -p stop -d psdmo`	Stops a Process Scheduler.
configure	`psadmin -p configure -d psdmo [-cfg cfg_set] [-cfg_from_file cfg_file_path] [-u feature_set]`	Configures a Process Scheduler. See the following section Using the Process Scheduler Configure Commands.
status	`psadmin -p status -d psdmo`	Displays the status of a Process Scheduler.
sstatus	`psadmin -p sstatus -d psdmo`	Displays the status of a Process Scheduler server.
cstatus	`psadmin -p cstatus -d psdmo`	Displays the status of a Process Scheduler client.
qstatus	`psadmin -p qstatus -d psdmo`	Displays the status of a Process Scheduler queue.
cleanipc	`psadmin -p cleanipc -d psdmo`	Cleans the IPC resources for specified domain.
kill	`psadmin -p kill -d psdmo`	Kills the domain (similar to forced shutdown).
import	`psadmin -p import -d psdmo`	Imports a Process Scheduler domain configuration. See the section Using the Process Scheduler Import Command.

Using the Process Scheduler Configure Commands

Use these commands to configure an existing Process Scheduler domain.

```
psadmin -p configure -d prcs_domain [-cfg cfg_set | -cfg_from_file cfg_file_path]
[-u feature_set]
```

Supply configuration settings in a string with *cfg_set*. or reference a file with *cfg_file_path*. See the discussion earlier in this section.

Use the -u *feature_set* option to enable or disable a set of features in the existing domain. To construct *feature_set*:

1. Access *PS_CFG_HOME*/appserv/prcs/*prcs_domain*/psprcsrv.ubx.
2. In the *FEATURES section, make a note of the {"-define-"} field for the feature you want to include.

   For example, the {"-define-"} field for Real Time Indexing is {RTI}.

   ```
   *FEATURES
   {"-label-"},{"-define-"},{"-on-"},{"-servers-"},{"-Oracle Env Manager
   Label-"}
   ...
   {Real Time Indexing},{RTI},{Yes},{PSRTISRV},{Real Time Indexing (PSRTISRV)}
   ```

3. Use the value from the {"-define-"} field in this format for *feature_set*.

   Use a slash (/) on Microsoft Windows to separate features.

   {FEATURE}=Yes|No/{FEATURE}=Yes|No...

   Use a percent sign (%) on UNIX to separate features.

   {FEATURE}=Yes|No%{FEATURE}=Yes|No...

For example, use this command to disable real-time indexing (RTI) in a Process Scheduler domain.

```
psadmin -p configure -d prcs_domain -u "{RTI}=No"
```

Using the Process Scheduler Import Command

Use these commands to import a Process Scheduler domain configuration:

- To import a domain from the location PS_CFG_HOME, use the following command.

  Use this command if there is a single domain in PS_CFG_HOME.

  `psadmin -p import ps_cfg_home_path`
- To import a domain when PS_CFG_HOME has more than one Process Scheduler server, use the following command.

  The imported domain has the same name as the original domain.

  `psadmin -p import ps_cfg_home_path -d prcs_domain`
- To import a domain to a new name, from a PS_CFG_HOME with multiple Process Scheduler servers, use the following command:

  `psadmin -p import ps_cfg_home_path -d prcs_domain -n new_domain_name`
- Use a configuration file to import a domain to a new name.

  Supply the path and name for psprcs.cfg for config_file_path.

  `psadmin -p import config_file_path -n new_domain_name`
- If a domain folder exists in the destination PS_CFG_HOME, append `-r` to any of the preceding commands to replace the folder.

- If a domain folder exists in the destination PS_CFG_HOME, append `-keepfeatures` to any of the preceding commands to enable the same features as in the source domain.

Related Topics

- Using the Process Scheduler Menu
  PeopleSoft Process Scheduler is used to schedule and run batch processes against the database. Process Scheduler domains are managed by Tuxedo, just as application server domains are. You only need to configure PeopleSoft Process Scheduler on a server where you intend to run batch processes. The menu options are identical to those provided for administering an application server domain, except where the options are not applicable to a Process Scheduler domain.

Working with PSADMIN Exit Codes

The following table lists most PSADMIN command line options for the application server and Process Scheduler server and their return codes. This can provide insight into the return codes displayed when using PSADMIN from the command line or in scripts. In general, PSADMIN returns *0* for success and non-zero when an error occurs.

> **Note**
>
> *N/A* indicates the command is not applicable to that server.

PSADMIN command line return codes (Table 1 of 4)

Command	Application Server	Process Scheduler
`-v`	0	N/A
`-h`	0	N/A
`-env`	0	N/A
`-defaultPS_CFG_Home`	0	N/A
`-invalid`	-10	N/A
`-create -d <domain_name> -t <template>`	0	0

Command	Application Server	Process Scheduler
`-create -d <domain_name> -t <invalid_template>`	275	275
`create -d <already_existing_domain> -t <template>`	265	1
`create -d <domain_name> -t <template> -s <s_set>`	0	N/A
`create -d <domain_name> -t <invalid_template> -s <s_set>`	1	N/A
`create -d <domain_name> -t <template> -s <s_set> -p <p_set>`	0	N/A
`create -d database -t template -ps ps_set`	N/A	0
`create -d database -t template -silent`	N/A	0
`delete -d <domain_name>`	0	0
`delete -d <nonexistent_domain>`	235	235
`configure -d <domain_name>`	0	0
`configure -d <nonexistent_domain>`	235	235
`boot -d <domain_name>`	0	N/A
`boot -d <nonexistent_domain>`	235	N/A

Continuation of the table listing the PSADMIN command line return codes (Table 2 of 4)

Command	Application Server	Process Scheduler
parallelboot -d <domain_name>	-1	N/A
parallelboot -d <nonexistent_domain>	235	N/A
shutdown -d <domain_name>	0	N/A
shutdown -d <nonexistent_domain>	235	N/A
shutdown! -d <domain_name>	0	N/A
shutdown! -d <nonexistent_domain>	235	N/A
cleanipc -d <domain_name>	0	0
cleanipc -d <nonexistent_domain>	235	235
sstatus -d <domain_name>	0	0
cstatus -d <domain_name>	0	0
sstatus -d <domain_name>	0	0
sstatus -d <nonexistent_domain>	235	235
cstatus -d <nonexistent_domain>	235	235
qstatus -d <nonexistent_domain>	235	235

Command	Application Server	Process Scheduler
`status -d <domain>`	N/A	0
`status -d <nonexistent_domain>`	N/A	235
`start -d <domain_name>`	0	0
`start -d <nonexistent_domain>`	235	235

Continuation of the table listing the PSADMIN command line return codes (Table 3 of 4)

Command	Application Server	Process Scheduler
`stop -d <domain_name>`	0	0
`stop -d <nonexistent_domain>`	235	235
`kill -d <domain_name>`	0	N/A
`kill -d <nonexistent_domain>`	235	235
`import <path>`	0	0
`import <path> -d <domain_name>`	0	0
`import <path> -d <domain_name> -n <new_name>`	0	0
`import <path> -n <name>`	0	0

Command	Application Server	Process Scheduler
`import <path> -d <already_existing_domain>`	0	0
`import <path> -d <already_existing_domain> -r`	0	0
`import <path> -d <domain> -n <name> -t <template>`	0	N/A
`purge -d <domain_name> -arch <archive_directory>`	0	N/A
`purge -d <domain_name> -noarch`	0	N/A
`addmsgsrv -d <domain_name> -s <sname> -t <type> <ch1,ch2,...>`	0	N/A
`addmsgsrv -d <domain_name> -s <sname> -t <type>`	-1	N/A

Continuation of the table listing the PSADMIN command line return codes (Table 4 of 4)

Command	Application Server	Process Scheduler
`addmsgsrv -d <nonexistent_domain> -s <sname> -t <type> <ch1,ch2...>`	235	N/A
`rmmsgsrv -d <domain> -s <sname>`	0	N/A
`rmmsgsrv -d <nonexistent_domain> -s <sname>`	235	N/A
`rmmsgsrv -d <domain_name> -s <nonexistent_sname>`	25	N/A

Command	Application Server	Process Scheduler
`preload -d <domain_name>` PreloadFileCache not set in PSAPPSRV.CFG.	-1	N/A
`preload -d <domain_name>` PreloadFileCache set in PSAPPSRV.CFG but not in database.	1	N/A
`preload -d  <domain_name>` PreloadFileCache set in PSAPPSRV.CFG and in database.	0	N/A
`preload -d <nonexistent_domain>`	235	N/A
`stop -d <domain_name>` Domain is not started.	1	0
`pslist -d <domain_name>`	0	0
`pslist -d <domain_name>` Domain is not started.	0	0
`pslist -d <nonexistent_domain>`	235	235
`purge -d <domain_name> -noarch -log <log_comments>`	0	235
`purge -d <domain_name> -arch <archive_dir> -log <log_comments>`	0	235

Using PSADMIN Configuration Files

This section provides an overview of PSADMIN executables and configuration files.

Understanding PSADMIN Configuration Files

You can create, configure, and boot an application server domain from the PSADMIN interface or through its command-line options.

The configuration and data files on which the executables rely all reside in *PS_CFG_HOME*\appserv*domain_name*. Each domain has its own set of these files:

- PSAPPSRV.CFG

  This is the catch-all configuration file that contains the entire collection of configuration values for a given application server domain.
- PSAPPSRV.UBX

  This is the template or model file for the PSAPPSRV.UBB file. This is where the system stores environment settings after being edited through PSADMIN.
- PSAPPSRV.UBB

  This file stores and passes all of the domain values to the Tuxedo load configuration program (tmloadcf.exe).
- PSAPPSRV.PSX

  This is the template or model file specifically for the messaging server configuration sections.
- PSAPPSRV.ENV

  This contains environment information, such as the PS_HOME referenced by a domain.
- PSAPPSRV.VAL

  This contains the format specification for the configuration parameters and, for some parameters, a set of valid values that can assigned. This helps to prevent administrators from entering invalid values.
- PSTUXCFG

  This contains PeopleSoft and Tuxedo information regarding the location of executables, files, and command lines for server processes. This file is required to boot a domain.
- JREPOSITORY

  This file contains a list of the services handled by the application server on behalf of Jolt.

> **Note**
>
> Oracle does not support creating custom configuration files, such as CFX or UBX templates, nor modifying delivered templates. However, modifying PSAPPSRV.CFG and PSPRCS.CFG can be done through PSADMIN or manually through a text editor.

Configuring a Domain

Regardless of how you specify domain values, ultimately you must run PSADMIN to generate some necessary files that include your specific values. In the following example, PSADMIN invokes another PeopleSoft executable, UBBGEN, which reads the values and format in the psappsrv.cfg, psappsrv.val, and psappsrv.ubx files, and generates the psappsrv.ubb and psappsrv.env files.

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

If you have already entered values manually in the psappsrv.cfg file and enter n, UBBGEN reads those values and writes to the necessary files.

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

> **Note**
>
> In the previous example, 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 values to the new psappsrv.cfg file.

Here are the scenarios by which you can configure a domain:

- Start PSADMIN, and enter values at all of the prompts.

  This generates all of the necessary files automatically.
- Edit the psappsrv.cfg file.

  If you decide not to use PSADMIN you must complete the following tasks in order:
  - From the command line, create a domain based on a particular template.
  - Edit the psappsrv.cfg file in a text editor.
  - Issue the configure command from the PSADMIN command line. This is the command that calls UBBGEN. You see the following after issuing this command:

    ```
    n:\<ps_home>\appserv>psadmin -c configure -d HRDOM01


    Performing load prechecks ...
    Loading validation table...

    Loading new configuration...
    Domain configuration complete.
    ```

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 the domain can properly boot. This occurs automatically after you have completed all of the prompts for values in PSADMIN.

To load the new configuration, PSADMIN calls the Tuxedo 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 PSTUXCFG file.

Archiving Application Server Configuration Files

To track changes made to the psappsrv.cfg file and the history of the changes, a subdirectory, named "archive," stores various versions of the CFG file. You can find this subdirectory in the domain name directory, as in C:\PS_CFG_HOME\appserv\domain name\archive, where the current version of psappsrv.cfg resides.

When you boot the application server domain for the first time, PSADMIN places a copy of psappsrv.cfg in the archive directory. In subsequent boots, if PSADMIN detects a change in psappsrv.cfg based on the time stamp values, it replaces the current psappsrv.cfg with the latest version. The file name of the new version is then psappsrv_mmddyy_hhmm_ss.cfg, as displayed on the time stamp.

Booting a Domain

When you select Boot this domain, PSADMIN calls the Tuxedo executable, TMBOOT.EXE, which uses the information that resides in the PSAPPSRV.ENV and PSTUXCFG files to boot the appropriate domain.

> **Note**
>
> You can track the server processes that start by using PSADMIN, ps -ef command in UNIX, or Task Manager in Microsoft Windows.

Stopping a Domain

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

Following a successful domain shutdown, PSADMIN checks and stops orphaned processes in the domain. If PSADMIN identifies and stops any orphaned server processes, it displays a screen message at the end of the shutdown operation.

Monitoring a Domain

To detect any orphaned application server processes, a server process, PSWATCHSRV, monitors the application server domain. Every two minutes, PSWATCHSRV identifies and stops any hung or orphaned server processes. If any hung or orphaned processes are found, it writes a message to the application server log file. The PSWATCHSRV process is the first process to start when you boot up the domain and the last one to stop when you shut down the domain.

Domain ID Name

To identify orphaned application server processes, all server processes within a server's domain must be uniquely identified. Therefore, the system appends a unique number to the domain ID in the psappsrv.cfg file. If you refer to domain IDs in scripts or processes, you may need to change those to reflect the new naming convention.

The command line varies slightly depending on the application server process, but it looks like this:

```
PSAPPSRV -C dom=pt84_52692 ...
```

Configuring the Application Server to Handle Cache Files and Replay Files

When an application server instance crashes, cache files and replay files are generated automatically. Over time, the size of these files can consume a large amount of disk space if there are recurring crashes in a domain. To minimize the buildup of cache files and replay files, you can modify the psappsrv.cfg file based on the following rules:

- When a crash occurs, the system creates a directory in the domain's LOGS directory.
- The dump file is saved in a directory within the domain's LOGS directory.
- The DumpMemoryImageAtCrash setting in the Trace section of the psappsrv.cfg file saves the memory image of the failed process in Microsoft Windows.

  This functionality is only available for Windows. If the value of DumpMemoryImageAtCrash is MINI, a miniature memory image (with a size less than or equal to 64K) is generated. If the value is FULL, then a full memory image is created. Depending on how much memory is consumed by the application, this full memory image can be quite large. The location of the memory image is the same as the replay file.
- If DumpManagerObjectsAtCrash is set to Y, then the application server instance:
  1. Generates the replay file.
  2. Dumps the customized objects being used by the current service request into the special cache directory.

     The cache directory resides in the same location as the replay file.
  3. If the value of DumpMemoryImageAtCrash is NONE and the platform is set to MS Windows, a miniature memory image is created.
- The settings for DumpManagerObjectsAtCrash and DumpMemoryImageAtCrash are dynamic.

  That is, the application server doesn't need to be restarted for these settings to be effective.
- There is no separate setting for generating the replay file.

  This file is generated as mentioned previously.
- Regardless of the setting in DumpManagerObjectsAtCrash, a summary report of objects in each managed type for which at least one object is loaded in memory is written to the dump file or application log file.

  The summary report resembles the following example:

  ```
  PDM Definitions: Total=36   Customized=0   In-Use=10
  RDM Definitions: Total=53   Customized=52   In-Use=50
  MDM Definitions: Total=1   Customized=0   In-Use=0
  PCM Definitions: Total=199  Customized=0   In-Use=3
  PGM Definitions: Total=1   Customized=0   In-Use=1
  CRM Definitions: Total=67   Customized=0   In-Use=0
  SSM Definitions: Total=1   Customized=0   In-Use=1
  ```

```
CLM Definitions: Total=1   Customized=0   In-Use=0
UPM Definitions: Total=1   Customized=0   In-Use=0
```

Total indicates the total number of in-memory definitions being used by the current service. Customized indicates how many of those objects are customized, and In-Use indicates how many of those objects were being used at the time of the crash.

If DumpManagerObjectsAtCrash is set to Y, the summary for each managed object type follows the list of configured objects that are being dumped as part of the crash information gathering. If a configured object is in use, its name is prefixed with an asterisk.

A sample report for a managed object type follows:

```
RDM(PSOPTIONS/ENG)
*RDM(PSTREEDEFNLABLS/ENG)
RDM Definitions: Total=10   Customized=2   In-Use=1
```

Note

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

5

Using PSADMIN Menus

Understanding PSADMIN Menus

You use PSADMIN to manage all of the main PeopleSoft server types. After launching PS_HOME\appserv\psadmin.exe from the command line, the PeopleSoft Server Administration (PSADMIN) main menu appears.

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

  Config Home:  D:\PT_SERVER\8.57

  1) Application Server
  2) Process Scheduler
  3) Web (PIA) Server
  4) Switch Config Home
  5) Service Setup
  6) Replicate Config Home
  7) Refresh Config Home
  q) Quit
Command to execute (1-7, q):
```

Term	Definition
Config Home	Displays the PS_CFG_HOME the current PSADMIN session. If it is not the PS_CFG_HOME desired, use the *Switch Config Home* option to select the appropriate location.
Application Server	Select to manage your application server domains.
Process Scheduler	Select to manage your Process Scheduler domains.
Switch Config Home	Select to switch to an alternative PS_CFG_HOME location. Determine the current PS_CFG_HOME from the Config Home value displayed.
Service Setup	(Applies to Microsoft Windows only). Select to configure domains to start using Windows Services.
Replicate Config Home	Select to replicate configured PS_CFG_HOME locations so that you can share these pre-created, tuned, and configured server configurations on the local server or on remote servers.

Term	Definition
Refresh Config Home	Select to refresh all domains in the PS_CFG_HOME location so that the latest configuration settings are applied to the domains.

> **Note**
>
> As a configuration option, you can configure a domain to *spawn* server processes according to the volume of transaction requests. To enable spawning, enter a value for the Spawn Threshold configuration parameter. In the following configuration section descriptions, some servers enable you to specify a minimum and maximum number of server processes. To enable spawning for these server processes, the maximum value must exceed the minimum value by an increment of at least one. As needed, the domain spawns server processes up to the maximum value. As the volume of transactions decreases, the number of spawned server processes decreases, or decays, until the minimum value is reached. By default, spawning is *disabled*.

See Spawn Threshold.

> **Note**
>
> Generally, the documentation reflects the order in which the configuration sections appear in the PSADMIN interface or the PSAPPSRV.CFG file.

Using the Application Server Administration Menu

This section provides an overview of the executables in the Application Server Administration Menu.

Accessing the Application Server Options

To access the menu options for configuring and administering an application server, select *Application Server* from the PeopleSoft Server Administration (PSADMIN) menu.

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

  1) Administer a domain
  2) Create a domain
  3) Delete a domain
  4) Import domain configuration
  q) Quit

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

The **Administer a domain** menu offers numerous configuration, administration, and logging parameters that you may access frequently.

The menu options and parameters within the **Create a domain** and **Delete a domain** menus are one-time tasks (per domain).

The **Import domain configuration** menu enables you to import existing configurations.

Administering a Domain

To administer a domain, you must have already created a domain. After you have created a domain, specify environment-specific settings for the application server to function correctly with your system. The following sections describe all of the menus and menu options that you use to administer and configure an application server domain.

To administer a domain:

1. Select *Administer a domain* from the PeopleSoft Application Server Administration menu.
2. In the Select domain number to administer command line, enter the number that corresponds to the previously created domain that you want to administer that appears in the Tuxedo domain list.
3. Select the option that you want to perform from the PeopleSoft Domain Administration menu.

PSADMIN transparently sets several environment variables before invoking any Tuxedo administrative commands. You don't need to set these variables manually. These environment variables are:

- TUXCONFIG = *PS_CFG_HOME*/appserv/*domain_name*/PSTUXCFG
- APPDIR = *PS_CFG_HOME*/appserv/*domain_name*
- PATH = TUXDIR/bin; *PS_HOME*/bin/server/winx86; PATH
- APP_PW = Application Password (initialize)

Importing Domain Configurations

You import existing domain configurations by selecting the **Import domain configuration** option from the PeopleSoft Application Server Administration menu. From the PeopleSoft Import Application Server Configuration menu, you have the option to import a regular domain or an Integration Broker Primary Configuration.

```
--------------------------------------------
PeopleSoft Import Application Server Configuration
--------------------------------------------

  1) Import regular domain
  2) Import IB Primary Configuration
  q) Quit

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

Importing a Regular Application Server Domain

When importing a regular application server domain, you can import from a file or from a domain.

```
--------------------------------------------
PeopleSoft Import Application Server Configuration
--------------------------------------------

  1) Import from file
  2) Import from application domain
  q) Quit

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

To import configuration settings from a file:

1. From the PeopleSoft Import Application Server Configuration menu, select *Import from file*.
2. Enter the complete file path to the PSAPPSRV.CFG file containing the desired configuration settings.

   For example:

   c:\users\psftuser\psft\pt\8.59\appserv*domain_name*\PT859TST
3. Enter a name for the new domain.

To import configuration settings from an existing domain:

1. From the PeopleSoft Import Application Server Configuration menu, select *Import from application domain*.
2. Enter the path to the appropriate PS_CFG_HOME.

   For example:

   c:\users\psftuser\psft\pt\8.59\appserv
3. From the Tuxedo domain list, select the domain containing the desired configuration settings.
4. Enter a name for the new domain.

Importing an Integration Broker Primary Configuration

Use this option to import a Primary configuration, if you are implementing a Primary/Secondary integration server configuration. This implementation option is documented in detail within the Integration Broker Administration PeopleBook.

See Integration Broker Administration: Implementing Primary-Secondary Dispatchers

Booting a Domain

This boots the Tuxedo domain (the application server) by using the tmboot command. This command will start all of the server processes that have been configured for your domain.

```
-------------------------------
PeopleSoft Domain Boot Menu
-------------------------------
```

```
      Domain Name: DOC

   1) Boot (Serial Boot)
   2) Parallel Boot
   q) Quit

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

You have two booting options: a serial boot and a parallel boot.

Running a Serial Boot

A serial boot starts server processes in a sequential order, with one process beginning to start after the previous process has completely started.

Running a Parallel Boot

A parallel boot starts server processes at the same time, rather than having each process start sequentially. This option typically provides shorter boot durations.

Related Topics

- Reviewing Stdout and Stderr Files
  Oracle Tuxedo creates multiple stdout and stderr files when PeopleSoft domains are booted.

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: psdmo

 1) Normal shutdown
 2) Forced shutdown
 q) Quit

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

Performing a Normal Shutdown

A normal shutdown is a quiescent shutdown that waits for users to complete their tasks and turns away new requests before terminating all of the processes in the domain.

Performing a Forced Shutdown

A forced shutdown is a nonquiescent shutdown that *immediately* terminates all of the processes in the domain. Normally, you use the forced shutdown only when a Bulletin Board Liaison (BBL) process encounters errors and cannot be shut down by using a normal shutdown.

> **Note**
>
> The BBL is a primary Tuxedo process that controls the domain.

Checking the Domain Status

Use the PeopleSoft Domain Status menu to view the status of the server, queues, or clients connected to the domain.

```
----------------------------
PeopleSoft Domain Status Menu
----------------------------
   Domain Name: psdmo

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

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

Server Status

Select *Server status* to invoke the Tuxedo tmadmin `psr` subcommand (print server processes), which displays the Tuxedo processes and PeopleSoft server processes that are currently running. For example:

```
> Prog Name      Queue Name  Grp Name      ID RqDone Load Done Current Service
---------      ----------  --------      -- ------ --------- ---------------
BBL.exe        62645       RTDC796+       0  42801   2140050 (  IDLE )
PSAPPSRV.exe   APPQ        APPSRV         1    422     21100 (  IDLE )
PSMONITORSRV.e MONITOR     MONITOR        1      0         0 (  IDLE )
PSWATCHSRV.exe WATCH       WATCH          1      0         0 (  IDLE )
PSAPPSRV.exe   APPQ        APPSRV         2      4       200 (  IDLE )
JSL.exe        00095.00200 JSLGRP       200      0         0 (  IDLE )
PSMSGDSP.exe   00098.00100 PUBSUB       100      4       200 (  IDLE )
PSSAMSRV.exe   SAMQ        APPSRV       100      0         0 (  IDLE )
PSMSGHND.exe   MBHQ        PUBSUB       101      0         0 (  IDLE )
JREPSVR.exe    00094.00250 JREPGRP      250     29      1450 (  IDLE )
```

The number of items appearing depends on the number of server processes that you have configured.

Client Status

Select *Client status* to invoke the Tuxedo tmadmin `pclt` subcommand (printclient), which displays connected users. For example:

```
>      LMID          User Name       Client Name    Time     Status  Bgn/Cmmt/
Abrt
--------------- --------------- --------------- -------- ------- -------------
RTDC79623VMC    NT              JSH             **:**:** IDLE    0/0/0
RTDC79623VMC    NT              JSH             **:**:** IDLE    0/0/0
```

```
RTDC79623VMC    JPOOL_15        dhcp-adc-twvpn+  0:02:03 IDLE/W  0/0/0
RTDC79623VMC    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. Check the queues by using the Queue status option. In the following example, the results of the Queue status option show the individual server processes, the associated queue, the number of server processes currently running, and the number of requests waiting to be processed:

```
> Prog Name      Queue Name  # Serve Wk Queued  # Queued  Ave. Len    Machine
---------       ------------------- ---------  --------  --------    -------
PSAPPSRV.exe    APPQ             2          -         0         - RTDC79623+
JSL.exe         00095.00200      1          -         0         - RTDC79623+
JREPSVR.exe     00094.00250      1          -         0         - RTDC79623+
BBL.exe         62645            1          -         0         - RTDC79623+
PSMONITORSRV.e  MONITOR          1          -         0         - RTDC79623+
PSWATCHSRV.exe  WATCH            1          -         0         - RTDC79623+
PSSAMSRV.exe    SAMQ             1          -         0         - RTDC79623+
PSMSGHND.exe    MBHQ             1          -         0         - RTDC79623+
PSMSGDSP.exe    00098.00100      1          -         0         - RTDC79623+
```

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, for example, PSAPPSRV. Or the results may indicate that you need to start either a PSQCKSRV or a PSQRYSRV.

Purging the Domain Cache

A proven technique for resolving problem application server environments is to purge the application server domain cache located in *PS_CFG_HOME*\appserv*domain_name*\CACHE.

> **Important**
>
> You should purge the cache only after due consideration, and after consultation with PeopleSoft support or consultants.

Keep the following in mind:

- You cannot purge shared file cache.
- You can purge the cache regardless of whether the application server domain is running; there's no need to shut it down and reboot. However, the procedure is less disruptive and runs more quickly if the domain is shut down or its activity level is low.
- Purging the cache can take five minutes or more on a large or busy domain, depending on the domain configuration.
- If database cache is enabled, the database cache will be purged. If the database cache is purged, it affects all domains using that cache. The database cache is shared across all domains that use the same byte-order (little-endian or big-endian).

To purge the domain cache:

1. On the PeopleSoft Domain Administration menu, select Purge Cache.

   If the cache is currently empty, the purge operation is cancelled, and the PeopleSoft Domain Administration menu reappears.

   If the cache is not empty, the following prompt appears:

   ```
   Enter log comments about this purge, if any (maximum 256 characters):
   ```

2. Enter any information (up to 256 characters) that you want recorded explaining the circumstances of this cache purge operation, and press **Enter**. Your comments will be saved to a purge log file.

   The following prompt appears:

   ```
   Do you wish to archive the contents of the current cache? (y/n) [n] :
   ```

3. Enter *y* to archive the cache contents, or *n* to delete them permanently. The default response is *n*.

   If you enter *y*, the following prompt appears:

   ```
   Cache contents will be archived to
   PS_CFG_HOME\appserv\domain_name\Archive\CACHE_mmddyy_hhmm_ss.
   Hit Enter to continue or provide a new location:
   ```

   > **Note**
   >
   > At runtime, *PS_CFG_HOME* and *domain_name* are replaced with values appropriate to your system, and *mmddyy_hhmm_ss* represents the date and time of the cache purge operation.

4. (If you chose to archive the cache contents) Enter a different archive location if desired, and press **Enter**.

   If the location you enter is rejected, the following message appears, and you're prompted to continue:

   ```
   Failed to archive cache to location.
   ```

   > **Note**
   >
   > Continuing this procedure with an invalid location will purge the cache without archiving.

   The default location is a unique directory name. Keep in mind that the location you enter might have been rejected for the following reasons:

   - The directory can't be created due to an invalid drive or network mount.
   - The directory can't be created due to insufficient user privileges.
   - The directory has insufficient space for the cache files.

5. When prompted to continue, enter *y* to continue the purge operation, or *n* to cancel the operation and return to the PeopleSoft Domain Administration menu.

> **Note**
>
> Archiving the cache increases the time required to complete the purge, because the cache files must be copied to the archive location.

If the application server domain is running, you might see INFO: messages related to version, patch level, serial number and so on. These are normal and don't require any action. When the cache is successfully purged, the following message appears:

```
Purge Cache operation completed successfully.

You may notice that the cache directory is non-empty. Cache files have been
invalidated and will be refreshed from the database.
```

If the cache was archived, you'll also see the following:

```
You may also have noticed a number of Sharing Violation messages during the
Cache Purge.
These messages are no cause for alarm and are expected as part of the
cache archival.
```

If the application server domain is running, an entry is written to the application server log file to indicate that the cache has been purged.

The purge log file is saved (including any comments you entered in step 2) as *PS_CFG_HOME*\appserv*domain_name*\LOGS\PurgeCache_*mmddyy_hhmm_ss*.log.

> **Note**
>
> At runtime, *PS_CFG_HOME* and *domain_name* are replaced with values appropriate to your system, and *mmddyy_hhmm_ss* represents the date and time of the cache purge operation.

Following is an example of the purge log file contents:

```
Date:02/17/10 11:47
User Explanation: Processes appeared to take a long time to recycle.
Cache Contents archived to
C:\ptservers\appserv\QEDMO\Archive\CACHE_021710_1147_01
```

Configuring a Domain

This option prompts you with a model configuration file to gather such parameters as port numbers, the number of various server processes that are needed, encryption enabling, and so forth. PSADMIN then invokes a subprogram, UBBGEN, which takes the configuration parameters, builds the file *PS_CFG_HOME*/appserv/*domain_name*/psappsrv.ubb, and carries out the *tmloadcf - y psappsrv.ubb* command to generate the following file: *PS_CFG_HOME*/appserv/*domain_name*/PSTUXCFG.

The following topics describe all of the parameters that you encounter while configuring an application server. Either read this section before you fine tune the configuration of your application server or have it available while you are doing it.

To configure a domain:

1. Select *Configure this domain* from the PeopleSoft Domain Administration menu.

   Enter *n* (No), if you do not want to continue. This returns you to the previous menu. Otherwise, enter *y* (Yes).
2. When prompted to change configuration values, enter y.

   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 that were previously specified. Enter n, or elect not to modify the PSADMIN parameters, if:
   - You have changed only the location of TUXDIR.
   - You would rather edit the PSAPPSRV.CFG file manually.
   - You installed a new Tuxedo patch.

> **Note**
>
> If you edit the psappsrv.cfg file directly, it is recommended to reload your domain configuration. This is necessary because some settings in psappsrv.cfg are transferred to the PSTUXCFG file for the domain. This transfer of settings can only be achieved by running UBBGEN and tmloadcf, which the "Configure this domain" option performs.

Editing Configuration and Log Files

Use the Edit Configuration/Log Files menu 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.

To have PSADMIN start your text editor (such as Notepad or KEDIT) so that you can manually edit or view application server configuration and log files, you must specify the text editor in the environment settings. For example, to use KEDIT, the editor environment setting should look like this:

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

To use Notepad, it should look like this:

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

> **Note**
>
> You can view and edit a domain's PSAPPSRV.CFG file while the domain is running, but the changes that you specify do not take effect until the next time you reconfigure the domain.

For the following options, you must enter your operator ID to view and edit the files:

```
Edit PSAPPSRV.tracesql (PSAPPSRV SQL trace file)
Edit PSSAMSRV.tracesql (PSSAMSRV SQL trace file)
```

For example:

```
Command to execute (1-7, q) [q]: 5
Enter the operator ID : PTXYZ
```

> **Note**
>
> PeopleSoft secures the Structured Query Language (SQL) traces because, in some instances, the SQL that is traced may involve sensitive information.

> **Note**
>
> Server configuration files (.CFG) support the use of environment variables, such as %PS_HOME%, %TEMP%, and so on.

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 in a text editor. When editing this configuration file manually, note that it is similar to editing an INI file, because all of the parameters are grouped in sections.

Edit APPSRV.LOG

This log file contains PeopleTools specific logging information.

Edit TUXLOG

The TUXLOG file enables 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 option.

Edit PSSAMSRV.tracesql

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

Creating a Domain

The **Create a domain** option creates a subdirectory under *PS_CFG_HOME*/appserv using the domain name you specify as the directory name and copies the required domain files to that directory.

To create an application server domain:

1. Select *Create a domain* from the PeopleSoft Application Server Administration menu.
2. Enter the name of the domain that you want to create; the name must not exceed eight characters.
3. Select a configuration template from the Configuration template list.

   The configuration templates are preconfigured sets of application server processes.

Deleting a Domain

Use the Delete a domain option to shut down the domain, if running, and delete the domain's subdirectory.

> **Note**
>
> Before you delete a domain, make sure that it is not running.

To delete a domain:

1. Select *Delete a domain* from the PeopleSoft Application Server Administration menu.
2. From the Tuxedo domain list, select the number that corresponds to the domain that you want to delete.
3. When prompted to continue, enter *y* and press **Enter.**
4. If the application server domain is currently running, PSADMIN informs you of that, and you will need to determine if you want to continue.

   ```
   Domain processes are currently running.  This option will shutdown
   and delete the domain.

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

   If you enter *y*, PSADMIN performs a forced shutdown on the domain prior to deleting the domain.

   If you enter *n*, PSADMIN returns you to the PeopleSoft Application Server Administration menu, allowing you to shutdown the domain manually, if needed.

Configuring an Application Server Domain to Preload Cache

This section provides an overview and discusses how to:

- Create cache projects.
- Delete cache projects.
- Preloading File Cache.

See ServerCacheMode.

See Load Application Server Cache.

Understanding Preload Cache Projects

To improve performance, the application server uses a caching mechanism that keeps commonly used objects in memory, or in file form on the application server or in a database cache. Caching reduces the need for a complete database query each time a definition is accessed. As more definitions are accessed, more data becomes stored in the cache. However, if a page, for example, has not already been accessed, it does not exist in the current cache, and the user may experience a slower response time as the system requests the page from the database for the first time. To prevent this initial performance degradation, you can preload file, database, and memory cache with commonly used definitions inserted into a cache project.

Preloading cache involves creating a project containing commonly used definitions and then referring to these projects in the PSADMIN settings PreloadMemoryCache and PreloadCache. By default, PreloadMemoryCache and PreloadCache are commented out because the two parameters need to be set to a specific name of a project that you create. You can set the parameters to reference separate projects. You use the Select Preload Objects page to select frequently used definitions and build the preload project containing the objects selected.

The cache project is intended to be used for a new domain, where the cache is not yet built. Prior to providing the domain for production use, use the Preload Cache PSADMIN option to build the cache containing the objects specified in the project. The domain starts the PSAPPSRV process, builds the cache, and shuts down.

The memory cache project is intended to be used during a domain boot (serial or parallel) and during a server process recycle.

When you boot a domain in serial mode, a message warns the user of a possible delay in booting the domain and lists the project that is being loaded, but when you boot a domain in parallel mode, the message is not displayed.

When a system starts a new process to replace an old one, the new process loads memory cache based on the project specified by PreloadMemoryCache so that the new process will not have delays when processing the first few service requests. Because it is desirable to have new processes start as quickly as possible, there is a timeout (or limit) of 60 seconds for PreloadMemoryCache. That is, PreloadMemoryCache preloads as many definitions as possible before the timeout of 60 seconds. If a project does not load completely, an administrator can adjust the boot timeout setting.

Logs pertaining to preloading memory cache are logged in the application server log file. In the application server configuration file, the log fence setting must be set to 5 to view the preload memory cache logs.

> **Note**
>
> In general, it is not recommended to create large projects for the PreloadMemoryCache project. For all cache projects (used for file, database, and memory cache) the optimum selections for the projects will require tuning and testing at your site.

Creating Cache Projects

To create cache projects:

1. In a browser, select PeopleTools, Utilities, Administration, Preload Cache - App Server, Select Preload Objects.
2. Select the Add a New Value tab, and in the **Project Name** edit box, enter the name of the project that will contain the definitions you select, and click **Add**.

   > **Note**
   >
   > All project names used to contain definitions for preloaded cache *must* contain the "PLC_" prefix.

3. On the Select Preload Objects page, enter a Description, and select the Object Type to add.

   Object Type refers to the definitions you create with PeopleTools, such as components, menus, pages, Application Engine programs, and so on.
4. Use the Key fields, to refine the selection of the definition as needed.

   The keys will differ depending on the Object Type selected.

   > **Note**
   >
   > Project definitions support up to four keys for identification, so you can use up to four keys when selecting objects for the preload cache project.

5. Click **Save**.
6. Click Build Project Definition.
7. On the Process Scheduler Request page, click **OK**.

   This invokes an Application Engine program (PTCHPLC_PRJ) that creates the project definition in the database and populates it with the definitions you selected. The PTCHPLC_PRJ also automatically includes all related definitions in the project.

   > **Note**
   >
   > Cache projects can be created manually in Application Designer, without running the PTCHPLC_PRJ program. The project name must include "PLC_".

Deleting Cache Projects

To delete a preload file cache project:

1. Select PeopleTools, Utilities, Administration, Preload Cache - App Server, Delete Preload Project.
2. On the Find an Existing Value page, click the appropriate project name.
3. On the Delete Preload Project page confirm that you have selected the appropriate project and click **Delete the preload project.**

Preloading Cache Projects

To preload a cache project:

1. Shut down the application server domain.
2. Edit the PSAPPSRV.CFG configuration file for the appropriate domain.

   In the [Cache Settings] section, uncomment the `PreloadCache=` parameter, and enter the name of the pre-load project for this application server domain.

   For example:

   ```
   PreloadCache=PLC_PROJECTA
   ```

3. On the PeopleSoft Domain Administration menu in PSADMIN, select `9) Preload Cache.`

When preloading cache, keep the following in mind:

- Preload Cache does not work when shared caching is enabled.
- If database caching is enabled, Preload Cache loads the cache project into the database cache.

Preloading Memory Cache

To preload the memory cache:

Edit the PSAPPSRV.CFG configuration file for the appropriate domain.

In the [Cache Settings] section, uncomment the `PreloadMemoryCache=` parameter, and enter the name of the preload project that should be preloaded on this application server for memory cache.

For example:

```
PreloadMemoryCache=PLC_PROJECTB
```

Working with the Dynamic Cache Repair Utility

PeopleTools provides a dynamic cache repair utility that enables you to configure the system to monitor and adjust automatically a domain’s cache while it runs, requiring no intervention from system administrators, and without requiring the domain or a server process to be restarted.

For more information on enabling and using the cache repair utility, see EnableCacheRepair.

Related Topics

- Load Application Server Cache
  The Load Application Server Cache page enables you to run Application Engine programs that load the *shared* file cache or the database cache for application server domains.

Cleaning Domain IPC Resources

Use the Clean IPC Resources of this domain option to clear the interprocess communication (IPC) resources utilized by a domain. When a domain shuts down normally, the IPC resources it was using get released as part of the shut down process. However, if a domain terminates abnormally, in many cases the IPC resources are still assigned to the previous domain instance. This option enables you to clean any orphaned IPC resources assigned to a domain.

Using the Process Scheduler Menu

PeopleSoft Process Scheduler is used to schedule and run batch processes against the database. Process Scheduler domains are managed by Tuxedo, just as application server domains are. You only need to configure PeopleSoft Process Scheduler on a server where you intend to run batch processes. The menu options are identical to those provided for administering an application server domain, except where the options are not applicable to a Process Scheduler domain.

```
--------------------------------------------
PeopleSoft Process Scheduler Administration
--------------------------------------------
     Domain Name: QEDMO

  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) Clean IPC resources of this Domain
  q) Quit

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

The Process Scheduler PSADMIN options differ from the application server PSADMIN options for these items.

Function	Difference
Booting	The Process Scheduler domain boot does not provide the option to select a serial or parallel boot. All Process Scheduler domains start in serial mode.
Viewing//Editing log files and configuration files	The Process Scheduler Edit/View Configuration/Log Files Menu does not display options for viewing the PSAPPSRV and PSSAMSRV SQL trace files.

Function	Difference
Configuring Integration Broker (messaging) servers	For Process Scheduler domains there is no option to import a Primary Integration Broker Domain nor is there the option to configure messaging servers.
Caching	For Process Scheduler domains there are not options to purge or preload file cache.

All of the Process Scheduler server configuration information for a specific domain is contained in the PSPRCS.CFG file, and PSADMIN provides an interface for and prompts you to edit the PSPRCS.CFG file.

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

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

Related Topics

- Using the Application Server Administration Menu
  This section provides an overview of the executables in the Application Server Administration Menu.

Using the Web (PIA) Server Menu

This section provides an overview of the executables in the Web (PIA) Server Menu.

Understanding PeopleSoft Web Server Administration Using PSADMIN

Using PSADMIN you can complete the majority of the web server administrative tasks related to your PeopleSoft Pure Internet Architecture (PIA) installation. While you can perform these tasks outside PSADMIN, using PSADMIN provides a centralized interface from which you can install, delete, configure, tune, and manage your PeopleSoft web domains and PeopleSoft sites.

For example, you can install PeopleSoft Pure Internet Architecture domains and sites using the silent installation files found in *PS_HOME*/setup/PsMpPIAInstall, but you would need to run a separate BAT or SH file to start it, and navigate to log files manually to confirm it has started. Using PSADMIN you can install the domain and site, but also configure that domain, boot that domain, confirm it has started, and view log files all from the same interface. The centralized PSADMIN options provide a convenient alternative for system administrators.

> **Important**
>
> By default, the system installs your PeopleSoft Pure Internet Architecture installation within the current PS_CFG_HOME location on the web server host.

> **Note**
>
> PSADMIN only manages single-server PIA installations. If you intend to implement multi-server installations, those need to be created and managed using traditional methods, such as the PsMpPIAInstall program and the delivered scripts for starting and stopping domains.

Related Topics

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

Accessing the Web Server Options

To access the PSADMIN web server options, select *3) Web (PIA) Server* from the main PeopleSoft Server Administration menu.

```
----------------------------
PeopleSoft PIA Administration
----------------------------

  PIA Home:   D:\PT_SERVER\8.61

  1) Administer a domain
  2) Create a domain
  3) Delete a domain

  q) Quit

Command to execute:
```

Field or Control	Description
PIA Home	Displays the location of your PeopleSoft Pure Internet Architecture domains. The default location is PS_CFG_HOME. If your application server and web server run on the same machine, they will share the PS_CFG_HOME location, by default.
Create a domain	Select to create a PeopleSoft Pure Internet Architecture domain.
Delete a domain	Select to delete a PeopleSoft Pure Internet Architecture domain.

Creating a PIA Domain

You create a PIA domain by selecting *Create a domain* from the PeopleSoft PIA Administration menu.

```
--------------------------------------------
PeopleSoft PIA Administration - Create Domain
```

```
---------------------------------------------

  PIA Home: D:\PT_SERVER\8.61

  1) Create a domain
  2) Replicate a domain

  q) Quit

Command to execute:
```

Creating a Domain

Creating a PIA domain using PSADMIN provides an alternative to using the PIA silent installation.

> **Note**
>
> You cannot create a multi-server PIA installation using PSADMIN.

```
---------------------------------------------
PeopleSoft PIA Administration - Create Domain
---------------------------------------------

  PIA Home:   D:\PT_SERVER\8.61

  1)  Domain name                    :[peoplesoft]
  2)  Web server                     :[weblogic]
  3)  Web server root directory      :[c:\oracle]
  4)  Web server login id            :[system]
  5)  Web server password            :[]
  6)  Website name                   :[ps]
  7)  Application server             :[SERVER1]
  8)  JSL Port                       :[9000]
  9)  HTTP Port                      :[80]
  10) HTTPS Port                     :[443]
  11) Authentication Token Domain    :[]
  12) Web profile                    :[PROD]
  13) Web profile user id            :[PTWEBSERVER]
  14) Web profile password           :[]
  15) Integration Gateway Id         :[administrator]
  16) Integration Gateway Password   :[]
  17) AppServer Connection Password  :[]
  18) SSL Keystore Password:         :[*****]
  19) PeopleSoft reports directory   :[c:\psreports]

  c) Create the domain as shown
  q) Quit

Command to execute:
```

Enter the number of the menu option to edit the values.

Field or Control	Description
PIA Home	Ensure that the appropriate location is displayed. By default, PSADMIN installs the PIA domain into the PS_CFG_HOME environment setting.
Domain name	Enter the name for your domain, such as *PeopleSoft_HR*.
Web server	Accept weblogic as the web server type.
Web server root directory	Enter the location where you have installed your web server, such as *D:\oracle\WebLogic*.
Web server login id	Enter the administrative user ID for logging into the web server. This would be the same user ID you use to log in to the WebLogic Remote Console.
Web server password	Enter the password associated with the web server login ID.
Website name	Enter the name of the initial website, such as *PSHR*. **Note:** Using PSADMIN you can only add one site. To add additional sites, you need to use the PsMpPIAInstall program.
Application server	Enter the application server host machine, such as *SERVER01*.
JSL port	Enter the JSL port on which the appropriate application server domain listens for Jolt requests.
HTTP port	Enter the HTTP port for your web server installation
HTTPS port	Enter the HTTPS port for your web server installation.
Authentication Token Domain	Enter the domain used for your authentication tokens, if implemented, such as *.example.com.* See the product documentation for *PeopleSoft 9.2 Application Installation* for your database platform: "Setting Up the PeopleSoft Pure Internet Architecture in GUI Mode," Using Authentication Domains in the PeopleSoft Pure Internet Architecture Installation See Portal Technology: Understanding the Authentication Domain.
Web profile	Choose the name of the web profile you intend to use. A web profile is a named group of configuration property settings that the portal applies throughout your PeopleSoft system to control all portal-related behavior. The web profile name will be used to configure this web site. You can specify one of the delivered web profiles, PROD, DEV, TEST, or KIOSK. See Portal Technology: Configuring Web Profiles.
Web profile user ID	Enter the user ID associated with the web profile you've entered.

Field or Control	Description
Web profile password	Enter the password associated with the web profile user ID.
Integration Gateway Id	Enter the integration gateway ID. The default is administrator.
Integration Gateway Password	Enter the password for the integration gateway ID.
AppServer Connection Password	Set the domain connection password, which is required for connecting to the application server domain. This modifies the DomainConnectionPwd setting in the configuration.properties file.
SSL Keystore password	Enter the SSL Keystore password, which is required for web servers and Integration Broker gateways in SSL-enabled environments. The password you enter is used for the default keystore. If you need to reset the password, use the pskeymanager utility. See Implementing WebLogic SSL Keys and Certificates.
PeopleSoft reports directory	Specify the root directory for the Report Repository, where Process Scheduler writes report output. Make sure that the report repository directory is shared. The user that starts PIA must have write access to the Report Repository directory.

Replicating a Domain

Replicating a PIA domain enables you to copy a PIA domain configuration from another location into the current PS_CFG_HOME. This enables you to tune and configure a single PIA domain, and replicate that configuration in multiple locations. Or, you may simply need to move a configuration to a newer or different server.

> **Note**
>
> You can copy domains from one operating system to another, as in from Windows to Linux.

```
---------------------------------------------
PeopleSoft PIA Administration - Create Domain
---------------------------------------------

  PIA Home:   D:\PT_SERVER\8.61

  1) Source Domain home               :[D:\PT_SERVER\8.53]
  2) Source Domain name               :[PSHR1]
  3) Target Domain name               :[PSHR2]
  4) Target Web server                :[weblogic]
  5) Target Web server home           :[c:\oracle]
  6) Target Web server user id        :[system]
```

```
  7) Target Web server password         :[********]
  8) PeopleSoft reports directory       :[c:\psreports]

  r) Replicate the domain as shown
  q) Quit

Command to execute:
```

Field or Control	Description
Source Domain home	Enter the PS_CFG_HOME where the domain you want to copy is installed. **Note:** The source domain home can be on the local host or a remote host. The source domain can come from a mapped drive or mounted file system. The target domain will be created within the current PS_CFG_HOME.
Source Domain name	Enter the name of the domain to replicate.
Target Domain name	Enter the name for the new, replicated domain. **Note:** The name must be unique if you are replicating within the same PS_CFG_HOME.
Target Web server	Specify the web server type installed on both the target and source host.
Target Web server home	Enter the location where the web server software is installed.
Target Web server user ID	Enter the administrative user ID for the target web server (the user ID used to login to the WebLogic Remote Console).
Target Web server password	Enter the password associated with the administrative user ID on the target web server.
PeopleSoft reports directory	Enter the location of the Report Repository on the target web server.

Deleting a PIA Domain

You delete PIA domains by selecting *Delete a domain* from the PeopleSoft PIA Administration menu.

```
-------------------------------------------------------
PeopleSoft PIA Domain Administration - Choose a Domain
-------------------------------------------------------
  1) PT861042
  2) psftpia

  q) Quit

Command to execute:
```

To delete a PIA domain:

1. From the PeopleSoft PIA Administration menu, select *3) Delete a domain.*

2. From the PeopleSoft PIA Domain Administration - Choose a Domain menu, select the PIA domain to delete.
3. When prompted to continue, enter *y.*
4. View the screen messages to verify the system has deleted the domain successfully.

```
------------------------------------------------------
PeopleSoft PIA Domain Administration - Choose a Domain
------------------------------------------------------
  1) PT853802
  2) psftpia

  q) Quit

Command to execute: 1

This operation will delete the configuration files for the domain.
If the domain is currently running it will be stopped.
Do you want to continue? [Y/N]: y

Verifying domain status....................................
Removing the domain.................................
The domain has been removed.
```

Booting a PIA Domain

To boot a PIA domain:

1. From the PeopleSoft PIA Administration menu, select *1) Administer a domain.*
2. Choose the PIA domain from the domain list.
3. From the PeopleSoft PIA Domain Administration menu, select *1) Boot this domain.*

   This will boot the domain and verify its status.
4. View the screen messages to verify the system has booted the domain successfully.

```
------------------------------------
PeopleSoft PIA Domain Administration
------------------------------------

  PIA Home:      D:\PT_SERVER\8.61
  PIA Domain:    psftpia
  1) Boot this domain
  2) Shutdown this domain
  3) Get the status of this domain
  4) Configure this domain
  5) Edit configuration files
  6) View log files
  7) Administer a site
  8) Delete a site
  9) Windows Service Setup

  q) Quit

Command to execute: 1
```

```
Starting the domain................
Server state changed to STARTING................
Server state changed to STANDBY.
Server state changed to STARTING................
.
Server state changed to ADMIN.
Server state changed to RESUMING..
Server state changed to RUNNING.
Verifying domain status..............
The domain has started.
```

Note that the subsequent display of the PeopleSoft PIA Domain Administration menu indicates `Domain Status: started`.

```
------------------------------------
PeopleSoft PIA Domain Administration
------------------------------------

  PIA Home:      D:\PT_SERVER\8.61
  PIA Domain:    psftpia
  Domain Status: started

  1) Boot this domain
  2) Shutdown this domain
  3) Get the status of this domain
  4) Configure this domain
  5) Edit configuration files
  6) View log files
  7) Administer a site
  8) Delete a site
  9) Windows Service Setup

  q) Quit

Command to execute:
```

Shutting Down a PIA Domain

To shut down a PIA domain:

1. From the PeopleSoft PIA Administration menu, select *1) Administer a domain.*
2. Choose the PIA domain from the domain list.
3. From the PeopleSoft PIA Domain Administration menu, select *2) Shutdown this domain.*

   This will stop the domain and verify its status.
4. View the screen messages to verify the system has shut down the domain successfully.

```
------------------------------------
PeopleSoft PIA Domain Administration
------------------------------------

  PIA Home:      D:\PT_SERVER\8.61
  PIA Domain:    psftpia
```

```
  1) Boot this domain
  2) Shutdown this domain
  3) Get the status of this domain
  4) Configure this domain
  5) Edit configuration files
  6) View log files
  7) Administer a site
  8) Delete a site
  9) Windows Service Setup

  q) Quit

Command to execute: 2

Stopping the domain...........
Verifying domain status.......................

Command to execute: Server "PIA" was force shutdown successfully ...

Done
```

Checking PIA Domain Status

To verify status of a PIA domain:

1. From the PeopleSoft PIA Administration menu, select *1) Administer a domain.*
2. Choose the PIA domain from the domain list.
3. From the PeopleSoft PIA Domain Administration menu, select *3) Check the status of this domain.*
4. View the `Domain Status:` line at the top of the menu to verify the domain status.

For example,

```
Domain Status: started
```

or

```
Domain Status: stopped
```

Configuring a PIA Domain

To configure a PIA domain, select *4) Configure this domain* from the PeopleSoft PIA Domain Administration menu. You use the PeopleSoft PIA Administration - Quick Configure Domain menu to modify PIA domain settings.

```
------------------------------------------------------
PeopleSoft PIA Administration - Quick Configure Domain
------------------------------------------------------

  PIA Home:   D:\PT_SERVER\8.61
  PIA Domain: psftpia
  Domain Status: stopped
```

```
  1) HTTP Port         :[80]
  2) HTTP SSL Port     :[443]
  3) Min Heap Size     :[512m]
  4) Max Heap Size     :[512m]
  5) Max Threads       :[50]
  6) Auth Token Domain :[]

  s) Save
  q) Quit

Command to execute:
```

Field or Control	Description
HTTP Port	The HTTP port for your web server.
HTTPS Port	The HTTPS port for your web server.
Min Heap Size	The minimum JVM Heap size for the web server instance occupied by your PIA domain. **Note:** For Oracle WebLogic include the 'm' as in *512m*.
Max Heap Size	The maximum JVM Heap size for the web server instance occupied by your PIA domain. **Note:** For Oracle WebLogic include the 'm' as in *512m*.
Max Threads	Specify the maximum number of simultaneous transactions that the PIA domain web server instance can handle.
Auth Token Domain	Enter the Authentication Token Domain. See Portal Technology: Understanding the Authentication Domain.

After making configuration changes and saving, PSADMIN responds differently, displaying different prompts, depending on the state of your domain, and the configuration options you've modified.

Configuration Option	Description
Standard	Can be modified while the server is running. A reboot is required for the change to take effect.
Dynamic	Can be modified while the server is running. Takes effect immediately with no reboot required.
Volatile	Cannot be modified while the server is running. The server must be stopped before the change can be saved.

Editing PIA Configuration Files

To view and edit PIA configuration files, select *5) Edit configuration files* from the PeopleSoft PIA Domain Administration menu.

> **Note**
>
> These files are provided for informational purposes. In most cases, you will not be required to modify these files unless asked to do so by a Global Customer Support representative.

> **Note**
>
> To open the file editor and view or edit the PIA configuration files on a non-Windows environment, such as UNIX or Linux, you need to have an X Window System configured and running.

```
-----------------------------------------------------------
PeopleSoft PIA Domain Administration - Choose a Config File
-----------------------------------------------------------
  1) setEnv
  2) config.xml
  3) portal/weblogic.xml
  5) logging.properties

  q) Quit

Command to execute:
```

Field or Control	Description
setEnv	Opens the setEnv file in your text editor. This file contains environment properties that the system initializes when you start the web server.
config.xml	Opens the config.xml file for the PIA domain web server instance. The config.xml file contains numerous web server settings.
portal/<web server type>.xml	Opens the <web server type>.xml file associated with the PeopleSoft Internet Architecture portal deployment.
logging.properties	Opens the logging.properties file for adjusting global and handler-specific logging levels.

Viewing Log Files

To view PIA domain log files, select *6) View log files* from the PeopleSoft PIA Domain Administration menu.

```
--------------------------------------------------------
PeopleSoft PIA Domain Administration - Choose a Log File
--------------------------------------------------------

  1) PIA_access.log     0
  2) PIA_stderr.log     0
```

```
  3) PIA_stdout.log     7951
  4) PIA_weblogic.log   384999
  5) psftpia.log      57793

  q) Quit

Command to execute:
```

Field or Control	Description
PIA_access.log	Opens the PIA_access.log in your text editor.
PIA_stderr.log	Opens the PIA_stderr.log in your text editor.
PIA_stdout.log	Opens the PIA_stdout.log in your text editor.
PIA_weblogic.log	Opens the PIA_weblogic.log in your text editor.
<domain name>.log	Opens the log for the current domain in your text editor.

Administering a PIA Site

To administer a PIA site:

1. From the PeopleSoft PIA Administration menu, select *7) Administer a site.*
2. Choose the PIA site from the site list.

```
----------------------------------
PeopleSoft PIA Site Administration
----------------------------------

  PIA Home:   D:\PT_SERVER\8.61
  PIA Domain: psftpia: stopped
  PIA Site:   PT86104

  1) Configure this site
  2) Edit configuration file

  q) Quit

Command to execute:
```

Configuring a PIA Site

To configure a PIA site, select *1) Configure this site* from the PeopleSoft PIA Site Administration menu.

```
----------------------------------------------------
PeopleSoft PIA Administration - Quick Configure Site
----------------------------------------------------

  PIA Home:   D:\PT_SERVER\8.61
  PIA Domain: psftpia
  Domain Status: stopped
  PIA Site:   PT86104
```

```
  1) Application Server Connect String :[SERVER1:9010]
  2) Jolt Pooling                      :[Enabled]
  3) AppServer Password                :[]
  4) Dynamic Config Reload Setting     :[0]
  5) Web Profile                       :[PROD]
  6) Web User Id                       :[PTWEBSERVER]
  7) Web Password                      :[]
  8) Report Repository File Path       :[d:/psreports]
  9) OBIEE Analytics Server URL        :[]
  s) Save
  q) Quit

Command to execute:
```

Field or Control	Description
Application Server Connect String	Specifies the application server(s) from which this web site will accept connections if Jolt failover and load balancing are implemented. This modifies the `psserver` setting in the configuration.properties file. See Configuring Jolt Failover and Load Balancing.
Jolt Pooling	Select to toggle Jolt Pooling [Enabled | Disabled]. This modifies the `joltpooling` setting in the configuration.properties file. See Configuring Jolt Session Pooling.
AppServer Password	Set the domain connection password, which is required for connecting to the application server domain. This modifies the `DomainConnectionPwd` setting in the configuration.properties file. See Configuring Domain Connection Password.
Dynamic Config Reload Setting	This setting enables the psserver string (Application Server Connect String) to be reloaded dynamically. That is, the domain does not need to be restarted for the changed values to take effect. Set Dynamic Config Reload Setting to 1 enables the feature, while setting it to 0 disable the feature. Default value is 1. This modifies the `DynamicConfigReload` setting in the configuration.properties file. **Note:** This setting applies only to the Application Server Connect String. See Configuring Dynamic Reloading of the Application Server Connect String and Web Profile Attributes.
Web Profile	Specify the web profile used to store the configuration settings for this web site. This modifies the `WebProfile` setting in the configuration.properties file.

Field or Control	Description
Web User ID	The user ID that the Portal uses to access the web profile. This modifies the `WebUserId` setting in the configuration.properties file.
Web Password	The user password that the Portal uses to access the web profile. This modifies the `WebPassword` setting in the configuration.properties file.
Report Repository File Path	The location of the Report Repository. **Note:** The system uses this setting value only if a Report Repository location is not specified in the current web profile. This modifies the `ReportRepositoryPath` setting in the configuration.properties file.
OBIEE Analytics Server URL	Enter the location of the OBIEE analytics server, if implemented.
Save	After you have made your changes, select *S) Save* to save your changes to the PIA site.

Editing PIA Configuration Files

To edit the configuration.properties file directly in your text editor, select *1) Edit configuration file* from the PeopleSoft PIA Site Administration menu.

Deleting a PIA Site

To delete a PIA site:

1. From the PeopleSoft PIA Domain Administration menu, select *8) Delete a site.*
2. Choose the PIA site from the site list.

> **Note**
>
> The domain must have at least two sites for the delete operation to succeed. If only one site exists in the domain, the system will not allow you to delete the only remaining site. A domain must contain at least one site.

3. When prompted to make sure you want to delete the site, enter *y*.

```
----------------------------------------------------
PeopleSoft PIA Domain Administration - Choose a Site
----------------------------------------------------
  1) PSHR
  2) PT86104

  q) Quit

Command to execute: 1

Are you sure you want to remove the site [Y/N]: y
```

```
Removing the site.................
The site has been removed.
```

Managing Windows Services for PIA Domains

This section discusses:

- Installing a Windows Service for a PIA Domain.
- Uninstalling a Windows Service for a PIA Domain.

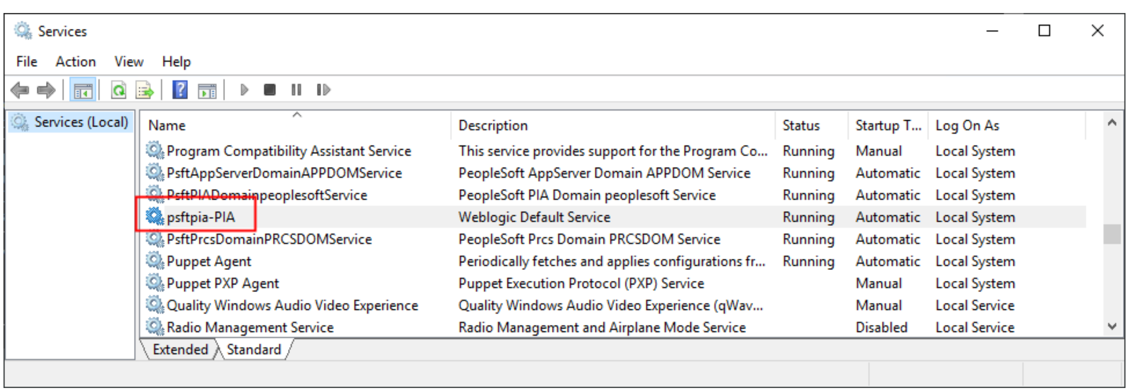
Installing a Windows Service for a PIA Domain

To install a Windows Service for starting a PIA domain, select *9) Windows Service Setup* from the PeopleSoft PIA Domain Administration.

To install the Windows Service, select *1) Install Service* from the Windows Service Setup menu.

```
---------------------
Windows Service Setup
---------------------

  PIA Home:   D:\PT_SERVER\8.61
  PIA Domain: psftpia: stopped

  1) Install Service
  2) Uninstall Service

  q) Quit

Command to execute: 1

Defining NT service for WebLogic admin server PIA
psftpia-PIA installed.
```

This example illustrates the Windows Services dialog box showing the installed service for PIA domain.



Uninstalling a Windows Service for a PIA Domain

To uninstall a Windows Service for starting a PIA domain, select *9) Windows Service Setup* from the PeopleSoft PIA Domain Administration.

To uninstall the Windows Service, select *2) Uninstall Service* from the Windows Service Setup menu.

```
---------------------
Windows Service Setup
---------------------

  PIA Home:   D:\PT_SERVER\8.61
  PIA Domain: psftpia: stopped

  1) Install Service
  2) Uninstall Service

  q) Quit

Command to execute: 2

psftpia-PIA stopped.
psftpia-PIA removed.
```

Replicating Configuration Home (PS_CFG_HOME)

The application server, Process Scheduler, and web (PIA) server domains exist within PS_CFG_HOME directory structure. You replicate these server elements by using the Replicate Config Home option in PSADMIN. The Replicate Config Home option enables you to clone a PS_CFG_HOME to a second location using PSADMIN internal functions to ensure that the environment is correct and cloned domains are readily bootable. The destination PS_CFG_HOME can reside on the same or a separate host. PSADMIN accounts for and modifies any host-specific settings and omits files that do not require cloning, such as log files.

This feature enables you to create a PS_CFG_HOME containing all the required domains that is reliable, tuned, and configured correctly for your site and deploy it to multiple locations across the enterprise. Replicating PS_CFG_HOME can be useful in a variety of scenarios, such as creating:

- backup environments.
- testing environments.
- scaling environments.
- failover environments.

> **Note**
>
> When replicating a source PS_CFG_HOME, PSADMIN overwrites the destination PS_CFG_HOME and any domains currently existing within it. If the destination PS_CFG_HOME contains domains, PSADMIN alerts you before proceeding.

> **Note**
>
> PS_CFG_HOME/data is also copied from the source PS_CFG_HOME to the destination PS_CFG_HOME.

> **Note**
>
> File cache is also replicated in the destination PS_CFG_HOME. Consider that replicating a PS_CFG_HOME with 30 domains with file cache will take longer to copy over a network to a remote host than replicating a PS_CFG_HOME with two domains and no cache.

> **Important**
>
> Using the PSADMIN replication feature is not designed to support replication from one operating system to another. For example, you *cannot* configure a domain on UNIX and then replicate that domain on a Windows server. You can replicate between environments on the same operating system platform only.

To replicate a PS_CFG_HOME:

1. Set PS_CFG_HOME to reflect the *destination* PS_CFG_HOME location.
2. Launch PSADMIN.
3. Select 6) Replicate Config Home from the PeopleSoft Server Administration menu.

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

     1) Application Server
     2) Process Scheduler
     3) Web (PIA) Server
     4) Switch Config Home
     5) Service Setup
     6) Replicate Config Home
     7) Refresh Config Home
     q) Quit

   Command to execute (1-7, q): 6
   ```

4. When prompted, enter the location of the destination PS_CFG_HOME and press ENTER.
5. When the process completes, verify the contents of the destination PS_CFG_HOME.
6. Use PSADMIN to make any adjustments to the replicated domain environment information as needed.

   For example, consider the following items for the new host:

   - database connectivity software location.

- port availability
- cache location
- file system settings
- (if replicating to the same machine) JSL and WSL port conflicts

Refreshing Configuration Home (PS_CFG_HOME)

Use the Refresh Config Home feature to refresh the domains in PS_CFG_HOME and to apply the latest configuration settings to the domains.

To refresh configuration home:

1. Launch PSADMIN.
2. Select 7) Refresh Config Home from the PeopleSoft Server Administration menu.

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

  1) Application Server
  2) Process Scheduler
  3) Web (PIA) Server
  4) Switch Config Home
  5) Service Setup
  6) Replicate Config Home
  7) Refresh Config Home
  q) Quit

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

3. You will be prompted whether to back up the current configuration.

   If you select yes, a back up of a domain is created and stored in PS_CFG_HOME with the following naming convention.

   <domain>_<date>_<time>.bak

   For example, peoplesoft_081315_2158.bak.

   If any of the domains is started, the refresh process will first shut down the domain, refresh it, and then start the domain. A stopped domain will just be refreshed.
4. When the refresh process completes successfully, boot the domains you need.

Switching Configuration Homes (PS_CFG_HOME)

By default, when you install application servers, Process Scheduler servers, and PIA domains, the system installs them in the current PS_CFG_HOME location. If you do not modify or override the PS_CFG_HOME location, and you install all of these server types on the same host, all of these server types will exist and be managed by PSADMIN in the same PS_CFG_HOME location. With all of the server types installed in the same PS_CFG_HOME, you would not need to use the Switch Config Home feature. This is the recommended approach.

PSADMIN supports multiple PS_CFG_HOME locations, but you need to make sure to set the PS_CFG_HOME variable correctly before installing domains, and use the Switch Config Home option to specify the appropriate PS_CFG_HOME when performing administrative and configuration tasks.

> **Note**
>
> You can set the appropriate PS_CFG_HOME outside of PSADMIN manually, or from a BAT or shell script, or you can set it using the Switch Config Home feature.

To switch configuration homes:

1. Navigate to the main PSADMIN PeopleSoft Server Administration menu.
2. Select option *5) Switch Config Home.*

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

  Config Home:  D:\HR_SERVER\8.57

  1) Application Server
  2) Process Scheduler
  3) Web (PIA) Server
  4) Switch Config Home
  5) Service Setup
  6) Replicate Config Home
  7) Refresh Config Home
  q) Quit
Command to execute (1-7, q): 5
```

3. At the prompt, enter the complete path to the desired PS_CFG_HOME location, and press ENTER.

> **Note**
>
> If the directory structure does not currently exist, PSADMIN creates it.

```
Please provide new Config Home location or q to quit: D:\PT_SERVER\8.57
```

4. Verify that the correct Config Home value appears on the PeopleSoft Server Administration menu.

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

  Config Home:  D:\PT_SERVER\8.57

  1) Application Server
  2) Process Scheduler
  3) Web (PIA) Server
  4) Switch Config Home
```

```
  5) Service Setup
  6) Replicate Config Home
  7) Refresh Config Home
  q) Quit
Command to execute (1-7, q):
```

> **Note**
>
> For PIA domains, you can set the PIA_HOME environment variable prior to starting PSADMIN. If this variable is set, PSADMIN recognizes it, and will install your PIA domains in the PIA_HOME location. You can switch to this location using the Switch Config Home feature.

Setting Up the PeopleSoft Windows Service to Start an Application Server Domain

This section discusses how to set up the PeopleSoft Windows Service to start an Application Server Domain.

> **Note**
>
> This section applies only to Microsoft Windows servers. It involves setting application server and Process Scheduler server domains to start as PeopleSoft Windows services.

Understanding Microsoft Windows Services

A Microsoft Windows service is a Microsoft-standard package that automatically starts and stops a process when you boot or shut down a Windows system. You can also start and stop Microsoft Windows services manually through the Administrative Tools - Services utility, which you can access through the Control Panel. A service uses a standard application programming interface (API) so that it can interact with the Control Panel and log messages to the standard event log.

For PeopleSoft domains, the service starts in an environment that is separate from any users who are signed in to the system (or to the server machine). If using the Windows service, administrators do not need to log on to a machine, start PSADMIN, and enter the proper commands to start a domain. In addition, if you use the PeopleSoft service, an administrator's logon session does not need to remain open while the domain runs.

If you have multiple application server domains or Process Scheduler servers on the same machine, you can start them all by using the same service setup.

> **Note**
>
> The PeopleSoft service supersedes the method that is provided in the Microsoft Windows resource kit. Do not use SRVANY.EXE or AT commands to start PeopleSoft domains.

Configuring the PeopleSoft Service

The following procedure assumes that you have already installed and configured a domain on the Microsoft Windows server.

After completing this procedure, the specified PeopleSoft domains start and shut down automatically when the operating system recycles.

To set up the Microsoft Windows service:

1. Open the System utility within the Control Panel, and make sure the following variables are set on the Environment tab in the System Variables section:

Variable	Value
TEMP	Specify the location of the TEMP directory on the Microsoft Windows server, as in C:\TEMP.

2. In PSADMIN utility, select *Service Setup* from the main menu.
3. Select *Configure a Service* from the PeopleSoft Services Administration menu.
4. Enter *y* to indicate that you want to change configuration values.
5. Enter the names of the application server domains or the Process Scheduler server domains that you want to include as part of the Microsoft Windows service.

   To add multiple values, separate each value with a comma and a space.

   For example,

   ```
   Application Server Domains=HRDOM1, HRDOM2, HRDOM3
   ```

6. Select *Install a Service* from the PeopleSoft Services Administration menu.

   > **Note**
   >
   > All of the domains and databases that you specified are part of a single Windows service called PeopleSoft *PS_CFG_HOME*, where *PS_CFG_HOME* is the location of the domain installations.

7. Return to the Windows Control Panel, and start the Administrative Tools, Services utility.
8. In the Services utility, scroll to find the entry that adheres to the PeopleSoft *PS_CFG_HOME* naming convention and access its properties.
9. On the General tab of the service properties, select a startup type of *Automatic*.

> **Note**
>
> The default startup mode is *Manual*.

10. On the Log On tab, the **Log On As** setting must match the**Log On As** setting that's defined for the Oracle ProcMGR service, which was created when you installed Tuxedo.

    Both services should either be configured to *Log On As Local System Account*, or to *Log On As This Account* (referring to the same account).

11. On the General tab of the service properties, click Start.

> **Note**
>
> If using a secure, remote PS_HOME, the user ID that starts the Oracle ProcMGR service and the PeopleSoft service must be same domain user ID and have sufficient privileges to at least read-execute the remote resources. Also, the TM_IPC_MAPDRIVER environment variable needs to be set appropriately. The PeopleSoft Windows Service must run under the same user id as the ProcMGR service because the PeopleSoft Windows service executes the PSADMIN command line options to start and stop domains.

See Understanding PS_HOME and PS_CFG_HOME.

Service Start Failure

It's possible that one or more of the domains or databases that are configured as part of the PeopleSoft service will fail to start, for reasons unrelated to the service.

The service is marked as started even if only one of its assigned domains starts. A message is written to the Windows event log for each domain, indicating whether it has started or not. If you experience problems with any domain or database, check the event log to see if it started successfully.

If all of the assigned domains and databases fail to start, the service is marked as stopped, and the following message is written to the event log:

```
Unable to start any of the domains configured for service service_name.
```

Testing the Service

To test the Microsoft Windows service, reboot the server, and make sure that the appropriate server executables are running.

For example, for the application server, use the Microsoft Windows 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 that any additional server processes that you have configured, such as PSQCKSRV.EXE, are running.

Editing the Service Configuration File

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

The following sections describe each parameter.

Service Start Delay

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

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

The default is 60 seconds.

Application Server Domains

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

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

Process Scheduler Databases

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

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

6

Setting Application Server Domain Parameters

Understanding Application Server Domain Parameters

These sections describe the application server domain parameters. You can use PSADMIN to review and change these parameters, or edit the psappsrv.cfg file directly.

See Editing Configuration and Log Files

Startup Options

Set database sign-in values in the Startup section, so that the domain can connect to the database.

> **Note**
>
> When specifying passwords, such as UserPswd, Connect Password, and StandbyUserPswd you will be prompted with: `Do you want to encrypt this password`? Selecting *y* encrypts the password so that it does not appear in clear text in the configuration file (PSAPPSRV.CFG) or PSADMIN.

DBName

Enter the PeopleSoft database name, such as FSDMO or HRDMO. This parameter is case sensitive.

DBType

Enter the PeopleSoft database type, such as Db2 for z/OS, Microsoft SQL Server, or ORACLE. If you enter an invalid database type, PSADMIN prompts you with a valid list.

UserID

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

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

Related Topics

- Security Administration: Setting General Permissions

UserPswd

Enter the password to be used by the specified user ID that will gain access to the database.

The password:

- should contain between 1 and 32 characters.
- should not contain any spaces.

Connect ID

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

Maximum length of an ID (user name) is 30 characters.

Connect Password

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

Ensure that the password meets the length and complexity requirements for your database platform.

ServerName

Optional for Microsoft SQL Server. Enter the name of the server on which the PeopleSoft database is installed. This value is case sensitive.

StandbyDBName

(Used only for Oracle databases with Oracle Active Data Guard implemented). Specify the standby database name.

StandbyDBType

(Used only for Oracle databases with Oracle Active Data Guard implemented). Specify the standby database type, such as *ORACLE*.

StandbyUserId

(Used only for Oracle databases with Oracle Active Data Guard implemented). Specify the user ID required to connect to the standby database.

StandbyUserPswd

(Used only for Oracle databases with Oracle Active Data Guard implemented). Specify the user password required to connect to the standby database.

InMemoryDBName and InMemoryDBType

These parameters are reserved for Oracle internal use only.

Database Options

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

UseLocalOracleDB

(Applies only to Oracle databases.). Use this option to enable processes to initiate a local connection to a PeopleSoft database that is running on the same machine. You should use this option for all PeopleSoft Process Scheduler (batch) and application server configurations that are local (on the same server) to the PeopleSoft Oracle instance. This type of connection enables processing to complete significantly quicker.

Enter 1 to enable this option, and enter 0 to disable it.

> **Note**
>
> Using the local Oracle connection disables the Query Kill function.

Related Topics

- Data Management: PeopleSoft Servers and the Oracle Connection String

EnableDBMonitoring

Required for database-level auditing. How this works varies slightly, depending on the database platform. Use this option to view more information regarding the clients that are connected to a database server through the application server. For instance, with this enabled, you can view the client machine name or user ID that is associated with a particular connection. Without this option enabled, all connections appear somewhat anonymously, as in PSFT or APPSERV.

The default value is *1* (enabled). Enter *0* to disable it.

EnableAEMonitoring

See Data Management: Monitoring PeopleSoft Database Connections.

PSDB Maximum Cursors

Enables you to configure the maximum number of cursors opened at one time. In general, the maximum number of cursors that can be open at one time within PeopleTools is 64,000. However, some database platforms place greater restrictions on the number of cursors that can be open at one time. By default, if no value is provided, the system allows a maximum of 1024 cursors.

To increase or decrease the default value, assign your custom value to this parameter. The value you set should be determined on multiple factors including:

- Any predefined limits imposed by your database platform.
- Internal testing of your PeopleSoft application(s).

Always refer to your database documentation for the latest recommendations related to maximum cursors settings. The following information can be used as a guideline.

Database	Maximum Cursors Limit
Oracle	No predefined limit.
IBM Db2 z/OS	Limited only by available storage.
Microsoft SQL Server	No predefined limit.

> **Note**
>
> When running Application Engine programs or Data Mover scripts you may see the following message in your log files:
>
> `Initializing PSDB maximum cursor = 1024`
>
> This is an informational message, not an error.

DisableGTTStats

Enables you to disable or enable the execution of the %UpdateStats meta-SQL function on an Oracle GTT (Global Temporary table).

The default value is *0*, that is, the %UpdateStats function is executed on an Oracle GTT.

Enter *1* to disable the execution of the %UpdateStats function when the target table is an Oracle GTT.

Related Topics

- Application Engine: Understanding Global Temporary Tables
- Data Management: Updating Statistics

Security Options

Use the Security section to set an additional layer to the sign-in process.

Validate Signon With Database

Use this option to set an additional level of authorization-checking to be performed at the database level. Enter 1 to enable this option, and enter 0 to disable it.

With this option disabled, if 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 this option enabled, the application server first attempts to connect to the database by 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 sign-in procedure occurs.

With this option enabled, to connect successfully to the database, the user must be defined on either the operating system or the database and within PeopleSoft.

> **Note**
>
> For Db2 for z/OS, the user ID and password must be defined as z/OS user logon IDs.

DomainConnectionPwd

The domain connection password adds an extra layer of security between the application server domain and any connections made to it. This password enables you to further prevent unauthorized clients from establishing connections to an application server domain. It is recommended to use PSADMIN to update this value. The value can be up to 30 characters.

All domains, PeopleSoft Pure Internet Architecture, and three-tier workstations used for a particular database, must use the same domain connection password.

> **Note**
>
> It is not required to add a domain connection password. It is an additional security layer for use if desired. If you add or change this value in the domain configuration, you *must* also update any PeopleSoft Internet Architecture sites and three-tier Windows workstations to reflect the new password expected by the application server domain.

For the PeopleSoft Internet Architecture configuration, you enter the password in the configuration.properties file using the same setting, DomainConnectionPwd.

For a three-tier Windows workstation connection, you enter the password in the Configuration Manager profile using the Domain Connection Password field on the Database/Application Server tab of the Edit Profile dialog box.

Related Topics

- Configuring Domain Connection Password
  The DomainConnectionPwd setting in the [Security] section of the application server domain, sets the domain connection password for all PeopleSoft Internet Architecture Jolt connections to that application server domain. It is not a required password setting, but *if set*, for the Jolt session to connect successfully to the application server domain, the DomainConnectionPwd property value in the configuration.properties file needs to match the DomainConnectionPwd parameter set for the application server domain.

Enable Login Audit

See Security Administration: Tracking User Sign-In Attempts for description of this parameter.

AddTo SID

Users can use the Actions menu on any fluid page to add an ad hoc tile to a selected fluid homepage, the NavBar, or the My Favorites section of the NavBar. When users select one of the “Add to” (or pin) actions, the request is accompanied by a default security check to prevent an unauthorized user from pinning or adding a page or web site to a homepage, NavBar, favorites or dashboard.

You can disable this security check by adding the AddTo SID setting to the psappsrv.cfg file. If you do not add the setting to psappsrv.cfg, the default security check will remain in effect.

To disable the security check:

1. Open the psappsrv.cfg file for editing and locate the Security section.
2. Add this line to the end of the section:

   ```
   AddTo SID Test=N
   ```

   > **Note**
   >
   > You must enter the text exactly as shown. The only allowable parameters are “N” and “n”. For example, entries such as `AddTo SID Test=` or `AddTo SID Test=False` are invalid and will *not* disable the security check.

   This example shows the Security section after editing:

   ```
   [Security]
   ;=========================================================================
   ; Security settings
   ;=========================================================================
   Validate Signon with Database=0
   DomainConnectionPwd=xxxxxxxxxxxxxx
   ; Enable/Disable Login Audit
   ; Y - Enable Login Audit (Default)
   ; N - Disable Login Audit
   Enable Login Audit=Y
   AddTo SID Test=N
   ```

3. Restart the application server.

   > **Note**
   >
   > This is not a dynamic flag. You must restart the application server for a new setting to take effect.

Related Topics

- Applications User’s Guide: Using the Actions Menu

Inter-Domain Event Settings

Use the Inter-Domain Events section to set the Process Scheduler Credentials and the Application Server Port options.

For a description of these options, see Fluid User Interface Developer's Guide: Configuring the System for Push Notifications

Oracle Wallet Options

Oracle Wallet is a container that stores authentication and signing credentials. Use the Oracle Wallet options to specify the location of an Oracle wallet, the wallet name, and the password.

See Configuring SSL for Workstation Listener and Jolt Listener

Workstation Listener Options

The workstation listener is the component to which PeopleSoft development environments running on Windows send Tuxedo messages. Development environments run Application Designer, for example.

By default, the workstation listener is disabled.

Address

%PS_MACH% resolves automatically to the machine name that PSADMIN obtains by using a system application programming interface (API) call. You can also specify the machine's Internet Protocol (IP) address (dotted notation) or its resolvable name (domain name server [DNS] name).

You should not change this value except in the following rare cases. If you are configuring files to run an application server on another machine (that is, you plan to copy PSAPPSRV.CFG and PSAPPSRV.UBB to a domain on another machine), you must overlay %PS_MACH% with the other machine's name.

> **Note**
>
> If you enter a literal IP address or machine name in place of the %PS_MACH% system variable, PSADMIN automatically adds '//' to the beginning of the value during the configure process. For example, if you enter *192.0.2.117* in place of %PS_MACH%, after configuring the domain, the value appears as *//192.0.2.117* in both PSADMIN and the psappsrv.cfg file.

Port

Enter the 4-digit port number to assign to the WSL. Port numbers are arbitrary numbers between 1000 and 64,000 and must not be in use by another service. The default values is 7000.

SSL PORT

Enter a 4-digit SSL port number to assign to WSL. The default values are:

- 7010 for a small application server domain template
- 7012 for medium application server domain template
- 7025 for large application server domain template

Min Encryption

Use this option to specify the minimum encryption level for encrypting data messages between client workstations and the application server. Choose one of the following values, which indicate the encryption key length (in bits):

- *0* — No encryption.

> **Important**
>
> This is the default value.

- *40* — 40-bit encryption.
- *128* — 128-bit encryption.

> **Note**
>
> Because this is a dynamic parameter, you must modify it by selecting Custom Configuration on the Quick-Configure menu, and reboot the application server domain for it to take effect.

Max Encryption

Use this option to specify the maximum encryption level allowed for encrypting data messages between client workstations and the application server. Choose one of the following values, which indicate the encryption key length (in bits):

- *0* — No encryption.
- *40* — 40–bit encryption.
- *128* — 128–bit encryption.
- *256* — 256–bit encryption.

> **Important**
>
> The default value is 256.

Min Handlers

Enter the number of workstation handlers (WSHs) to be started at boot time. The minimum number of handlers required depends on the domain. A smaller domain, handling less connections, may only need one WSH to start at boot time. A larger domain, handling more connections, may require ten WSHs to start at boot time.

Here are the default Min Handler values for the delivered application server templates:

- Small –- 1
- Medium — 5
- Large — 10
- Developer — 1

Max Handlers

Enter the maximum number of WSHs that can be started for a domain. As connections increase, the system spawns more WSHs (up to the Max Handler value) to keep up with system demand. If the Min Handlers value equals the Max Handlers value, Tuxedo does not spawn incremental WSHs. Make sure the additional ports are free.

If you configure more than one application server domain, to avoid port conflicts, you must allow for spawned ports, up to the Max Handler value. For example, for a small template, if the first port is 7000, to derive the port number for the second domain, skip 3 and set the port for the second domain to 7004.

Here are the default Max Handler values for the delivered application server templates:

- Small –- 3
- Medium — 12
- Large — 25
- Developer — 2

See Listeners, Handlers, and Queues.

Max Clients per Handler

Enter the maximum number of client workstation connections that each WSH can manage. Each WSH can handle approximately 60 client connections. Numbers vary depending upon the resources of the server. In most cases, you should decrease the default as opposed to increasing it. The default is *40*.

Client Cleanup Timeout

Enter the amount of time, in minutes, that a client connection can remain idle (no work requested) before Tuxedo terminates the client connection. Client disconnects are transparent to a client, and a user just clicks the mouse to cause a reconnection. The default value for this setting is *60* minutes.

Init Timeout

This value, when multiplied by SCANUNIT (a UBB parameter value that is defined in the PSAPPSRV.UBB file) specifies the amount of time, in seconds, that Tuxedo allows for a client connection request to bind to a WSH before terminating the connection attempt.

Tuxedo Compression

Enter the minimum length of a data message for which the application server initiates data compression. While compression results in favorable performance gains for transactions over a wide area network (WAN), testing reveals that compression can degrade performance slightly over a local area network (LAN) due to the compression and decompression overhead.

You should use 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 are compressed, and those 5000 and under are uncompressed. If you support both WAN and LAN users, you can 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 Options

Use this section to configure PeopleSoft Internet Architecture connections. The Jolt listener enables Tuxedo to exchange messages with the web server.

Address

See the equivalent parameter for the workstation listener.

Port

Enter the port number that is used for the Jolt server listener (JSL). This value can be any port number that is not already in use by another service on the machine that runs the application server domain. By default, the JSL port is enabled.

SSL PORT

Enter a 4-digit SSL port number to assign to JSL. The default values are:

- 9010 for a small application server domain template
- 9012 for a medium application server domain template
- 9025 for a large application server domain template

Min Encryption

Use this option to specify the minimum level of encryption required for encrypting data messages between client workstations and the web server. Choose one of the following values, which indicate the encryption key length (in bits):

- *0* — No encryption.

> **Important**
>
> This is the default value. Incoming Jolt requests from the web server (portal, PIA, and Integration Broker) are not encrypted.

- *40* — 40-bit encryption.
- *128* — 128-bit encryption.
- *256* — 256–bit encryption.

Max Encryption

Use this option to specify the maximum level of encryption allowed for encrypting data messages between client workstations and the web server. Choose one of the following values, which indicate the encryption key length (in bits):

- *0* — No encryption.
- *40* — 40–bit encryption.
- *128* — 128–bit encryption.
- *256* — 256–bit encryption.

> **Important**
>
> The default value is 256.

Min Handlers

Enter the number of Jolt server handlers (JSH) to be started at boot time. Each JSH multiplexes up to 50 connections.

Here are the default Min Handler values for the delivered application server templates:

- Small — 5
- Medium — 9
- Large — 15
- Developer — 1

Max Handlers

Enter the maximum number of JSHs.

JSHs spawn by using successive port numbers starting at the port number for the JSL in the PSAPPSRV.CFG file. Make sure that the additional ports are free before configuring spawning.

If you configure more than one application server domain, to avoid port conflicts, you must allow for spawned ports, up to the Max Handler value. For example, for a small template, if the first port is 9033, to derive the port number for the second domain, skip 7 and set the port for the second domain to 9041.

Here are the default Max Handler values for the delivered application server templates.

- Small — 7
- Medium — 12
- Large — 25
- Developer — 2

See Listeners, Handlers, and Queues.

Max Clients per Handler

Enter the maximum number of client connections that each JSH can manage.

Client Cleanup Timeout

Enter the amount of time, in minutes, that a client connection can remain idle (no work requested) before Tuxedo terminates the client connection. Client disconnects are transparent to a client, and a user just clicks the mouse to cause a reconnection. The default value for this setting is *10* minutes.

Init Timeout

See the equivalent parameter for the workstation listener.

Client Connection Mode

Enter one of these options to control the allowed connection modes from clients:

- RETAINED: The network connection is retained for the full duration of a session.
- RECONNECT: The client establishes and brings down a connection when an idle timeout is reached and reconnects for multiple requests within a session. The reconnection is transparent to the user.
- ANY: (Default) 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 accepts connections. 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 enables messages that are transmitted through a Jolt connection to be compressed as they flow over the network. 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 provide the best results for most situations. However, your application server administrator can 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 that you set is the minimum number of bytes that a single packet must be before the server compresses it.

Many of the XML messages being sent around the 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 from compressing numerous packets. Typically, you should decrease the threshold according to the bandwidth of the workstation hardware as described in the following paragraphs.

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

Alternatively, if you have mostly low bandwidth, as in 56-kilobyte (KB) modem connections over a WAN, then you 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 that you may have gained from reduced network traffic.

Additional Prompt

After you finish all of the configuration sections, PSADMIN prompts you to configure Jolt (which is enabled by default).

If you are using the PeopleSoft Internet Architecture, you must have Jolt enabled for browser access.

Jolt Relay Adapter Options

The Jolt relay adapter is disabled by default. Unless you have a specific need for JRAD, you should skip this section.

Related Topics

- Using Jolt Internet Relay
  This section provides an overview of the Jolt Internet Relay architecture and how to use it.

Listener Address

The default is %PS_MACH%. Enter the machine on which the application server is running. See the equivalent parameter for the workstation listener.

Listener Port

This option is for advanced configurations requiring the 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, is not used unless you enter y at the prompt that asks if you want to configure JRAD.

Domain Settings

Use this section to specify general settings for the entire domain—not just for a specific component of the domain or server process.

Domain ID

Enter the name of the application server domain. It does not need to match the name that you specified when you created the domain. This name is important only because the Tuxedo Web Monitor and PeopleSoft Watch Server (PSWATCHSRV) use it to identify application server domains and the processes associated with each machine. It should not exceed 8 characters. Generally, you should use the database name in lowercase.

Add to PATH

Enter the directory that contains your 64–bit database connectivity software, as in /apps/db/oracle/bin. If the database connectivity directory is not already specified in the path, you can set it by specifying this parameter. The value is added to the path.

On UNIX, if you don't enter a value, it uses the current directory—not the current path. To have it set by default to the current path, enter a period (.).

> **Note**
>
> On Windows, entries that contain a space must be surrounded by quotes. If you don't enter a value, it uses the current path.

Spawn Threshold

Enter a parameter that's supplied to Tuxedo for control of process spawning by using the -p command-line option for all server processes. The default setting (*1,600:1,1*) rarely needs to be changed.

This setting enables the dynamic decay of spawned server processes as the transaction volume decreases. The value can be loosely translated to mean that if, in 600 seconds, there is less than or equal to one job in the queue, the decay process begins. This is described in more detail in the timeout settings topic of this product documentation.

New server processes will be spawned according to the rule defined here. By default, if there is one outstanding request in the queue for one second or more, an additional process is spawned. Additional processes will be spawned all the way up to the Max Instances defined for that server type. If Max Instances and Min Instances are identical, this setting has no effect.

> **Note**
>
> This parameter applies only if, for PSAPPSRV, the value of *Max Instances* is greater than that of *Min Instances*. By default, spawning is disabled.
>
> For more information, also see servopts(s) in the Oracle Tuxedo reference manual.

Spawning results in resource-intensive demands at process initialization because it takes place mainly when the system is starting to undergo increased load and increased queues. In this scenario, you may want to manually scale the size of a domain by starting additional domain processes as desired. In order to manually scale the size of a domain, you need to disable spawning by *not* entering a value for the Spawn Threshold configuration parameter, which disables the dynamic process spawning for the domain. As a result, you can set the minimum and maximum instances of PSAPPSRV to different values, and you can manually start additional domain processes as desired.

Related Topics

- Application Server Timeouts
  All configurable settings for the application server require modification in PSADMIN:

Log Directory

The log directory contains log files the system generates for a domain, such as Tuxedo logs (TUXLOG) and APPSRV logs.

The default log directory for a domain is %PS_SERVDIR\logs.

To specify a custom log directory:

1. Uncomment the Log Directory setting in the domain's PSAPPSRV.CFG file.
2. Enter the desired log directory location either directly into the PSAPPSRV.CFG file or through PSADMIN.
3. Reconfigure the domain using PSADMIN for the new setting to take effect.

When entering custom log directory locations, keep the following length restrictions in mind.

Domain Type	Log Directory Character Length Limit
Application Server	60
Process Scheduler Server	70

Restartable

Enter *y* to have Tuxedo restart server processes (except the BBL process) if the server dies abnormally, as in a 'kill' on UNIX or through the Task Manager on Microsoft Windows. Otherwise, enter *n*.

Allow Dynamic Changes

Often, administrators must set a trace or performance parameter while the domain is up and running. If you enable this option, then you don't need to reboot the domain for the modified parameter value to take effect.

Enter *y* or *n* to enable or disable dynamic changes. When disabled, you must reboot (or cycle the processes) for changes to take effect.

When enabled, the server checks an internal time stamp for a particular service request to see if any values have changed for the parameters for which dynamic changes are valid. If values have changed, the system uses the modified parameter value.

You should enable this option in your test and development domains. For production environments, you should enable dynamic changes selectively.

These parameters allow dynamic changes:

- Recycle Count.
- Consecutive service failures.

- Trace SQL and Trace SQL Mask.
- Trace PC and Trace PC Mask.
- Trace PPR and Trace PPR Mask.
- Log Fence.
- Enable DB Monitoring.
- Enable Debugging.
- Dump Memory Image at Crash.
- Dump Managed Objects at Crash.
- Log Error Report.
- Mail Error Report.
- SMTP Settings (all except SMTPGuaranteed, SMTPTrace, and SMTPSendTime).
- Analytic Instance Idle Timeout.
- Analytic Per Server Log.

> **Note**
>
> The parameters that allow dynamic changes are also identified through comments in the PSAPPSRV.CFG file. Look for the phrase “Dynamic changes allowed for X,” where *X* is the parameter name. This option does not apply to configuration parameters that Tuxedo relies on, such as the number of processes, whether restart is enabled, the port numbers, the amount of handlers, and so on.

LogFence

Enter a level of network tracing, ranging from –100 (suppressing) to 5 (all). The default is 3.

The trace file is generated in *PS_CFG_HOME*\appserv*domain*\LOGS\psappsrv.log.

LogSizeLimit

Enter a whole number for the log size threshold in MBytes. Logs are rotated when the threshold is exceeded. The default value of LogSizeLimit=0 means that there is no threshold configured.

The log size based rotation works in conjunction with date-based rotation. When this setting is enabled, PeopleTools server processes will switch to creating a newer log file when the log file size threshold is breached or when the date changes; whichever occurs earlier.

All PeopleTools Tuxedo Server logs will have a sequence number at the end, such as APPSRV_1110.LOG.1, APPSRV_1110.LOG.2, APPSRV_1110.LOG.3. The current log file can be identified by the current date and the highest sequence number.

Log files with sequence number up to 1000 can be created by default. Once this limit is reached, servers will continue to append to the same log file. This behavior can be changed by adding a new environment variable to the domain—PS_LOG_ROTATION_MAX. This environment variable denotes the maximum rotations that can take place per log file. Its value should be between 1 and 2000.

LogFieldSeparator

Separate individual fields of data within the domain log file(s). If this setting is not explicitly defined, the default value is a single SPACE character. This setting must be uncommented in the configuration file to be enabled.

These meta values can be used as the field separator:

- *TAB*
- *SPACE*

While the actual key strokes can be used to specify a separator, the escape sequence equivalent of them is not supported. For example \t is not recognized as a tab character. For purpose of clarity, avoid key strokes that are not visible to the naked eye, instead use their meta value equivalents.

AppLogFence

This setting is not available through the PSADMIN interface, but can be entered directly into the PSAPPSRV.CFG file.

You can use this parameter conditionally to determine whether you want to do specific logging from your application. You can implement this parameter from PeopleCode using the %AppLogFence system variable. Complete documentation for this option is in the PeopleCode Developer's Guide.

See PeopleCode Developer’s Guide: Using Application Logging.

Trace-Log File Character Set

Enter the character set (default is UTF-8) of the machine to which you typically write and read the traces and log files. If the character sets are not matched between the file and the machine, the file is unreadable. Supported character sets are:

- ANSI
- UTF-8
- UTF8
- UTF8BOM
- UNICODE

> **Note**
>
> The UNICODE character set can be used, but it will be desupported.

Related Topics

- Global Technology: Understanding Character Sets

ProcessRestartMemoryLimit

Enter a memory threshold for recyclable server processes running within the domain. When the server processes exceed the limit, the system restarts, or recycles, them. This setting applies to all recyclable server processes within the domain. If set to *0,* this feature is disabled and the system recycles server processes according to the recycle count setting for that recyclable server process.

To enable the restart memory limit, enter a value, in megabytes, as the memory threshold. For example, if you set the restart memory limit for the domain to *500,* any recyclable server process running within the domain restarts after it has consumed 500 megabytes of memory and after the current service request has been processed.

Recyclable server processes include:

- PSAPPSRV
- PSANALYTICSRV
- PSSAMSRV
- PSQCKSRV
- PSQRYSRV
- Integration Broker server processes

The restart memory limit can be set at the individual server process level as well. That is, you can set the restart memory limit for the entire domain using the setting in the [Domain Settings] section of the configuration file, or you can set it for a specific server process to override the domain-level limit for that server process type.

When the system recycles a server process, the system writes in the log file a statement similar to the following:

```
PSAPPSRV.4356 (9) [2014-05-12T09:02:44.273 QEDMO@<machine-info>.example.com]
(0) Recycling server due to reaching the configured memory threshold of 150
(cfg section [Domain Settings]).
PSAPPSRV.4356 (9) [2014-05-12T09:02:44.273 QEDMO@<machine-info>.example.com]
(0) In the last request the memory increased up to 193.
PSAPPSRV.4356 (9) [2014-05-12T09:02:44.277 QEDMO@<machine-info>.example.com]
(0) Recycling server after 96 services
PSAPPSRV.5016 (0) [2014-05-12T09:02:45.112](0) PeopleTools Release 8.54-901-
R1 (Windows) starting. Tuxedo server is APPSRV(99)/1
```

> **Note**
>
> If you implement this feature, you need to test and monitor the appropriate memory threshold for your site's performance requirements and resource usage to reach the optimal setting.

The system implements the ProcessRestartMemoryLimit recycling behavior according to these rules:

- If ProcessRestartMemoryLimit is disabled for the domain (set to *0*), and the ProcessRestartMemoryLimit parameter is disabled for each individual server process configuration section, the Recycle Count setting controls when a server process recycles (restarts).

- If ProcessRestartMemoryLimit is enabled for the domain, all recyclable server processes running in the domain are subject to that memory limit, unless the ProcessRestartMemoryLimit parameter is set to a different value in that server type's configuration section. For example, assume at the domain level, ProcessRestartMemoryLimit is set to *500,* but the ProcessRestartMemoryLimit in the PSQRYSRV section is set to *600.* In this case, all recyclable server processes running in the domain will be recycled when a server process consumes 500 megabytes off memory, except for PSQRYSRV, which will not recycle until it reaches the 600 megabyte limit.
- The ProcessRestartMemoryLimit cannot be set too low so as to prevent the domain from establishing the minimum memory requirements for booting and performing basic operations. If the memory limit is set too low, the domain will not boot, and the system writes an error message into the log file similar to the following:

```
PSAPPSRV.4272 (0) [2014-05-12T09:18:25.535](0) Taken configuration
variable ProcessRestartMemoryLimit from the section [Domain Settings] 20
Mb is not big enough. PSAPPSRV initially needs at least 40 Mb.
```

For the [Publish&Subscribe] (Integration Broker) server process group, there is an additional layer of priority given to the ProcessRestartMemoryLimit parameter, because you can set the ProcessRestartMemoryLimit parameter at the group level as well. For example:

- If ProcessRestartMemoryLimit is set within the [Publish&Subscribe] configuration section, that setting overrides any ProcessRestartMemoryLimit setting at the domain level. The ProcessRestartMemoryLimit value set within the [Publish&Subscribe] section applies to all integration server processes (handlers and dispatchers) running in that group within the domain.
- If ProcessRestartMemoryLimit is set within an individual server process configuration section, such as [PSPUBHND_dflt] (settings for publication contract handler), that setting overrides any ProcessRestartMemoryLimit setting at the [Publish&Subscribe] section level or the domain level.

PeopleCode Debugger Options

Use this section to enable and configure the PeopleCode debugging environment. Configuring PeopleCode debugging is discussed in detail in another section of this PeopleBook.

Related Topics

- Setting Up the PeopleCode Debugger
  This section discusses how to set up and use the PeopleCode Debugger.

Trace Options

This section enables you to specify the tracing options that you can enable on the application server to track the Structured Query Language (SQL) and PeopleCode of the domains. You can also set all of the trace parameters from the PeopleSoft sign-in page. Just beneath the Sign In button, click the link that opens the trace flags page. This enables you to set the trace options and then sign in to the system.

> **Note**
>
> With many of the following trace options, you need to view the comments in the PSAPPSRV.CFG file to understand what to enter to return the trace information you require.

TraceSQL

Enter the logging level for SQL tracing for all clients. Traces are written to *PS_CFG_HOME*/appserv/*domain*/LOGS/*domain_user_IDservername*.tracesql.

Enter 0 to disable tracing; enter 7 to enable a modest tracing level for debugging. For other levels of tracing, set this option to a value that equals the sum of the needed options. For example, to trace only SQL, enter 1; to trace SQL statements and connect statements enter 7 (1+ 2 + 4 = 7). A setting of 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 this option to 0 when you finish troubleshooting.

> **Important**
>
> The trace file stores elapsed times for SQL events to a precision of one microsecond (six decimal places). However, due to limitations of the operating system, Windows precision is actually in milliseconds (three decimal places), so the last three digits in a Windows trace will always be zero. Elapsed times in UNIX are accurate to one microsecond.

TraceSQLMask

Enter the logging level ceiling for SQL tracing for individual clients. Traces are written to *PS_CFG_HOME*/appserv/*domain*/LOGS/*client_user_IDservername*.tracesql.

For Windows clients, you specify the necessary SQL tracing level by using the 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 that is transmitted to the TraceSQLMask value. If the client value is less than or equal to the TraceSQLMask value, the application server enables the trace. However, if the client value is greater, the application server enables the trace up to the TraceSQLMask value. Trace files are written on the application server; no trace shows up on the client workstation.

TracePC

Enter a level for PeopleCode tracing for activity that is generated by all clients on a domain. Eligible values are defined in the configuration file. TracePC values are displayed in the PeopleSoft Configuration Manager on the Trace tab. You can find the results in *PS_CFG_HOME*/appserv/*domain*/LOGS/*domain*.log.

> **Important**
>
> The trace file stores elapsed times for PeopleCode events to a precision of one microsecond (six decimal places). However, due to limitations of the operating system, Windows precision is actually in milliseconds (three decimal places), so the last three digits in a Windows trace will always be zero. Elapsed times in UNIX are accurate to one microsecond.

TracePCMask

Enter which PeopleCode trace options that are requested by client machines will be written to the trace file. You can find the results in *PS_CFG_HOME*/appserv/*domain*/LOGS/*client_machine*.*domain*.log.

TracePPR and TracePPRMask

Use these options to trace the activity in the page processor. Typically, these options are used internally only by PeopleSoft developers; however, you may need to view the results of this trace when troubleshooting.

Tracing-related display processing is useful for seeing when and if related displays are being updated and if they are updating successfully. Some sample output in the log file from setting this flag includes:

```
Starting Related Display processing
Related Display processing - PPR_RELDSPLVALID not set
Related Display processing - All Rows
  Starting Related Display processing for - PSACLMENU_VW2.MENUNAME
  Related Display processing for - PSACLMENU_VW2.MENUNAME - completed
successfully
Finished Related Display processing
```

By using the keylist generation tracing in addition to the related display tracing, you can determine why the related displays have the wrong value. It shows where the keys are coming from. The following is a sample of keylist generation tracing:

```
 Starting Keylist generation
       Keylist generation - FIELDVALUE is a key
       FIELDVALUE is low key
       Low key value was supplied =
       Key FIELDVALUE =
       Keylist generation - FIELDNAME is a key
       Keylist generation - Finding value for USRXLATTABLE_VW.FIELDNAME
         Not Found in key buffer
           Seaching for field FIELDNAME in component buffers
           Scanning level 1
             Scanning record DERIVED_USROPTN for field FIELDNAME
             Field FIELDNAME found in record DERIVED_USROPTN
         Found in component buffers, value = PT_TIME_FORMAT
       Key FIELDNAME = PT_TIME_FORMAT
       Keylist generation - USEROPTN is a key
       Keylist generation - Finding value for USRXLATTABLE_VW.USEROPTN
```

```
         Not Found in key buffer
           Seaching for field USEROPTN in component buffers
           Scanning level 1
             Scanning record DERIVED_USROPTN for field USEROPTN
             Scanning record PSUSROPTLIST_VW for field USEROPTN
             Field USEROPTN found in record PSUSROPTLIST_VW
         Found in component buffers, value = TFRMT
       Key USEROPTN = TFRMT
     Keylist Generation complete
     FIELDNAME = PT_TIME_FORMAT
     FIELDVALUE =
     USEROPTN = TFRMT
```

In this example, you can see how the system builds the keylist by first searching in the current record (key buffer), then searching the buffers in the current level, and then searching up a level, and so on. It also indicates exactly what record the key value is taken from. This is useful on complex components where there are often several instances of a particular field; a common problem is that the value is derived from an unexpected location.

Combining the keylist tracing and the related display tracing provides a good view of the system behavior. For example:

```
Starting Related Display processing


Related Display processing - All Rows
  Starting Related Display processing for - PSACLMENU_VW2.MENUNAME
    Starting Keylist generation
      Keylist generation - MENUNAME is a key
      MENUNAME is low key
      Low key value was supplied = APPLICATION_ENGINE
      Key MENUNAME = APPLICATION_ENGINE
    Keylist Generation complete
     MENUNAME = APPLICATION_ENGINE
  Related Display processing for - PSACLMENU_VW2.MENUNAME - completed
successfully
```

Each related display goes through the keylist generation process, and you can see exactly what key values are used to populate the related displays and where those key values came from.

The work record flag is a performance feature. If every field in a level-0 record has a value from the keylist and is display-only, then it is marked as a work record because the values can't be changed. After it is marked as a work record, that affects how the record behaves. For example, PeopleCode for fields in the record but not in the component don't run, data isn't saved, and so on. By enabling this tracing option, you can see which records are flagged as work records. The output looks like this:

```
   Work flag cleared for record PSCLASSDEFN_SRC
   Work flag cleared for record PSCLASSDEFN_SRC
   Work flag cleared for record PSCLASSDEFN
   Work flag cleared for record PSPRCSPRFL
   Work flag cleared for record SCRTY_QUERY
   Work flag set for record PSCLASSDEFN
```

```
Work flag set for record PSPRCSPRFL
Work flag set for record SCRTY_QUERY
```

Because the flag is turned on and off at various points, the last setting (set or cleared) is the most important. In the previous trace, PSCLASSDEFN, which is marked as a work record, is cleared and then set again.

TracePIA and TracePIAMask

Use these options for tracing PeopleSoft page (PIA page) generation.

TraceAE

Use this parameter to activate specific Application Engine traces for tracing Application Engine programs.

TraceAnalytic and Trace AnalyticMask

The bits enable logging for analytic servers beyond the standard LogFence setting.

TracePPM

The Performance Monitor agent is a thread that reports performance metrics for each instrumented server if monitoring is enabled for the database. Select *1* to enable and *0* to disable.

See Performance Monitor.

DumpMemoryImageAtCrash

This parameter determines whether or not a memory image of the failing process is created when a crash occurs. By default, the value is 'NONE' which means that a memory image will not be written during a crash. This value can be set to 'MINI' or 'FULL'. MINI means that a shorter memory image is written. This may be a better option if you are leaving this option turned on permanently. FULL may be appropriate when you are debugging a known issue. Typically, this option is used internally only by PeopleTools.

DumpManagerObjectsAtCrash

This parameter assists PeopleSoft in debugging any crash problems at your site by providing insight into the customized object definitions to reproduce the crash on another database.

Log Error Report, Mail Error Report

If you enter *y* (enabled) and runtime errors are detected (nonfatal error conditions), the system writes a message and information regarding the runtime error to the current log file. If you assign the MailErrorReport parameter an email address, an individual, such as a system administrator, can be alerted whenever the system writes an error to the log. If MailErrorReport is enabled but LogErrorReport is set to *n*, the system still sends a message for application server crashes, which always cause data to be written to the log. The following is an example of setting this parameter to send notifications to an email address: MailErrorReport=user.name@example.com.

AETFileSize, TraceAECombineOutput, and TraceAEEnableSection

For descriptions of these parameters, see Process Scheduler: Trace Section.

Cache Settings

Use this section to specify how to handle caching at your site. Enabling caching on the application server improves performance, but you need to review the available cache options and determine which option best suits your site.

> **Note**
>
> The majority of the cache settings need to be uncommented in the PSAPPSRV.CFG file.

EnableServerCaching

With EnableServerCaching, you specify what objects the system stores in cache on the application server. To enable application server disk caching the value must be set to 1 or 2.

If you enter 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 select depends on internal testing at your site.

To disable application server caching, set this value to 0. In most cases there is no reason to disable server caching. Doing so significantly degrades performance, because it requires the application server to retrieve an object from the database each time the system needs it.

CacheBaseDir

This setting is the location where cache files will be written and stored for this domain.

> **Note**
>
> You must preload your shared cache before you enable shared caching on the application server.
>
> Application Engine processes are independent from application server domains, directories, and configuration files. Therefore, Application Engine processes do not share cache with application server domain processes.

ServerCacheMode

Caching enables the system to store definitions locally on the application server, eliminating unnecessary trips to the database server. If caching were disabled, the system would need to retrieve definitions from the database with each request, every time. This adds significant overhead to each transaction and affects system response times.

If server caching is enabled on the application server, which is usually the recommended approach, there are two modes of caching from which to choose.

Cache Mode	Description
Non-shared cache	By default, non-shared cache mode is enabled (ServerCacheMode set to *0*). With the non-shared cache mode, each server process that starts within a domain maintains its own separate cache file. For example, assume you had two PSAPPSRV processes configured in a domain. In non-shared cache mode, there is one cache directory per PSAPPSRV server process, which each individual PSAPPSRV process uses separately. In this case, you can find the cache files in *PS_CFG_HOME*\appserv*domain*\cache\PSAPPSRV_1 *and* \PSAPPSRV_2. While the cache directories will grow over time to include the most used definitions, you have the option to *preload* the non-shared cache directories with the most used system definitions. See Configuring an Application Server Domain to Preload Cache.
Shared cache	To set shared caching for the domain, enter 1 at the Set ServerCacheMode prompt. With this option enabled, you can find the cache files in *PS_CFG_HOME*\appserv*domain*\cache\share. The system assumes that a preloaded cache exists in the share directory. The preloaded cache contains most instances of the managed object types that are cached to file. When you boot the application server, if shared cache files are enabled but no cache files exist in the expected location, the system reverts to unshared caching. To preload shared cache, you run delivered Application Engine programs that build your shared cache. See Load Application Server Cache.

MaxCacheMemory

PeopleTools stores metadata in a memory cache on the server machine to increase system performance. However, in rare situations if the memory cache becomes too large, it can reduce the amount of memory available to other processes running on the server, which can degrade general performance. Use the MaxCacheMemory setting in certain situations where it is required to specify the maximum allowable size of the memory cache.

> **Note**
>
> Typically, this setting only needs to be implemented in rare circumstances, or when instructed to do so by GCS support staff. In most situations, it is recommended that sufficient memory be installed on the server machine to ensure optimal performance.

The MaxCacheMemory setting enables you to establish a size limit in megabytes, that the memory cache should not exceed. Every time the system retrieves an object from the database, such as a page, it updates the object's LastUsedDate value. When your system reaches the memory cache threshold controlled by the MaxCacheMemory setting, the system "prunes" the oldest objects in the memory cache until the desired memory cache size is reached.

> **Note**
>
> If a memory cache limit has been set, the system prunes the memory cache to keep it roughly 10% below the specified limit.

The default value of this setting is *0* (disabled), which disables memory cache pruning altogether, allowing for an unlimited memory cache. That is, when MaxCacheMemory is set to *0,* the system continues to store a memory cache, but the system will not attempt to prune or reduce the size of the memory cache.

The default setting (disabled) might not be optimal for your application. You can adjust this setting to achieve the best trade-off between speed and available memory based on your system requirements. If you are in a situation where you need to limit the size of the memory cache, enter that limit in megabytes. Make sure to test your memory cache limit thoroughly in realistic scenarios, reflecting your production environment.

EnableDBCache

If enabled, the metadata (cache) is accessed from database, rather than the file system.

To implement database caching, uncomment the parameter and enter *Y* to enable database caching, or *N* to disable database caching. After changing this parameter, the domain does not need to be reconfigured.

The database cache is shared by all domains that enable this option.

When database caching is enabled, these settings are ignored:

- EnableServerCaching
- ServerCacheMode
- CacheBaseDir

You can load the database cache using the Load Application Server Cache utility or the preload cache utility.

> **Note**
>
> Database caching is also available for Process Scheduler domains.

Related Topics

- Load Application Server Cache
  The Load Application Server Cache page enables you to run Application Engine programs that load the *shared* file cache or the database cache for application server domains.
- Configuring an Application Server Domain to Preload Cache
  This section provides an overview and discusses how to:

PreLoadCache and PreLoadMemoryCache

If you have created a cache project, specify the project name. Before booting a domain, you have the option to preload the cache, and this option creates the cache based on the load project you specify.

If database caching is enabled, the cache is preloaded into the database.

Related Topics

- Configuring an Application Server Domain to Preload Cache
  This section provides an overview and discusses how to:

EnableCacheRepair

Set to *Y* to enable the cache repair utility, which can be configured using these settings:

- AdjustMaxCacheMem
- ExcessivePruningThreshold
- CacheMissThreshold
- MaxRepairCollection

If set to *N,* the cache repair utility does not run, and the system ignores these settings.

The cache repair settings enable you to configure the system to monitor and adjust automatically a domain's cache while it runs, requiring no intervention from system administrators, and without requiring the domain or a server process to be restarted.

> **Note**
>
> When cache repairs occur, the system writes a message to the log file. The messages are preceded with the text "Cache Repair:" so they can be identified easily. The system begins recording these messages when the log fence is set to *4.*

AdjustMaxCacheMem

AdjustMaxCacheMem only applies if the MaxCacheMemory setting has been enabled for the domain. MaxCacheMemory is used to remove (also referred to as pruning) old objects from the memory cache so that it does not become too large and consume more system resources than expected.

Set the number of megabytes by which the system can increase the MaxCacheMemory setting if the memory cache size has reached the MaxCacheMemory setting. So, rather than pruning more objects from the memory cache, this setting increases the memory cache limit incrementally. For example, if MaxCacheMemory is set to 50 and AdjustMaxCacheMem is set to 5, the system increases MaxCacheMemory to 55 megabytes when the 50 megabyte limit is reached.

> **Note**
>
> When the system increases the MaxCacheMemory value, the increase will not be reflected in the configuration file. When the application server domain recycles, it assumes the original MaxCacheMemory value set in the configuration file. You can review the log files to determine what the system automatically increased the MaxCacheMemory to prior to the recycle, and set the static value in the configuration file accordingly.

ExcessivePruningThreshold

Expressed as a percentage and represents the percentage of objects fetched from the database or persistent cache during the service request (and therefore added to memory cache) that have also been pruned from the memory cache at the end of the service request. If more than 80% of the objects fetched during a service request are pruned at the end of the service request, then the system is over-pruning.

Once the threshold has been reached, the system adjusts the MaxCacheMemory limit by the AdjustMaxCacheMem value so that the system is not constantly fetching from the persistent cache.

When this occurs, a message appears in the logs similar to:

```
Cache Repair: Warning excessive pruning has been detected. 83 percent or more
of cached items have been removed from memory cache. MaxCacheMemory may be
set too low.
```

If AdjustMaxCacheMem is not set to *0,* then the system also logs the following message:

```
Cache Repair: Adjusted internal MaxCacheMemory setting from 50MB to 55MB.
```

CacheMissThreshold

Threshold for the number of times an object is found to be invalid within the memory and persistent cache. Once this threshold is reached, the system deletes the object from both memory and persistent cache, forcing the object to be refreshed from the database.

The system tracks the types of cache separately. If the object is invalid only in memory cache, then it's only removed from there. But if the object is invalid in the persistent cache (file or database cache), then it's removed from memory *and* the persistent cache.

The log file message is similar to:

```
Cache Repair: PDM(VAT_OPTIONS_SBP/ENG) Found invalidated object in memory
cache- Removed from memory
Cache Repair: PDM(VAT_OPTIONS_SBP/ENG) Found invalidated object in file
cache- Removed from memory and cache
Cache Repair: PDM(VAT_OPTIONS_SBP/ENG) Found invalidated object in db cache-
Removed from memory and cache
```

MaxRepairCollection

To keep track of various cache metrics, the system stores some caching information for automatic analysis to determine if corrective measures need to be taken. The default limitation for this storage is 5 megabytes. This data is not used for historical reporting or analysis, and it is used only by the system while a server process is running. Once the limit is reached, the system purges all cache tracking information.

The log file message is similar to:

```
Cache Repair: maxed out memory at 5MB, purging cache repair data
```

Remote Call Options

Use the Remote Call options to enable executing a COBOL program remotely from within a PeopleSoft application.

RCCBL Redirect

Enter *0* to disable redirection and *1* to enable redirection. Redirection causes the server process to retain intermediate work files that are used to pass parameter values between the server process and a Remote Call/COBOL program for debugging purposes, and it also generates Remote Call trace information in the COBOL output files.

If set to *0,* the system generates no COBOL output files. If set to *1* (enabled), the system generates the following files, where *<pid>* is the Process ID:

- rmtcall_in.<pid>
- rmtcall_out.<pid>
- rmtcall_cblmsg.<pid>
- Remote Call Tracing entries for COBOL output files.

> **Note**
>
> RCCBL Redirect should be enabled only when debugging because writing Remote Call trace entries to the COBOL output files can add to system overhead and overall performance.

Remote Call Trace Information

The information captured for Remote Call tracing follows this general structure:

Trace Field	Description
Remote Call trace identifier (RCCT)	Remote Call COBOL Trace (RCCT) helps you to identify the Remote Call tracing in the COBOL output files.
Remote Call Component	Enables you to identify the type of Remote Call being traced: • PSPCM • PTPECOBL • PTPNETRT • PTPSQLRT • Application COBOL

Trace information	The trace record can be different depending on what is being traced:
	• In some cases there are parameter/value pairs written in the record, such as:
	`RCCT: PSPCM: pSamParms->szDBName QE850GA`
	`RCCT: PTPNETRT: DATA-FIELD-REG   BEGIN ->a<-`
	`END`
	• In other cases there will be trace information that reports the program flow:
	`RCCT: PTPNETRT: RemoteCall PSCOBNET function`
	`CBL_GET_OS_INFO`

Remote Call Output Files

The location and naming of the COBOL output files differ depending on whether the Remote Call is invoked on the application server or the Process Scheduler server (by the PSAESRV server process).

Location	Description
Application Server	If the Remote Call is invoked on the application server, the COBOL output file(s) are written to: *PS_CFG_HOME/appserv/<APP_SERVER_DOMAIN>/LOGS* The output file with the Remote Call trace entries follows this naming convention: *<COBOL_PGM>_<OPERID>_<TIMESTAMP>.out*
Process Scheduler Server	If Remote Call is invoked on the Process Scheduler server (PSAESRV), the COBOL output file will be written to: *PS_CFG_HOME/appserv/prcs/<PROCESS_SCHEDULER_DOMAIN>/ log_output/CLB _<Program Name>_<Process_Instance>* The output file with the Remote Call trace entries follows this naming convention: *<COBOL_PGM>_<TIMESTAMP>.out* For example, if a program named PTPTSTAE runs it generates the following: *PS_CFG_HOME/appserv/prcs/HCMPRCS/log_output/ CBL_PTPTSTAE_94/PTPECOBL_1121155222.out* **Note:** PTPECOBL is the initial target COBOL program invoked by the Remote Call function because PTPECOBL handles the parameter passing from the PeopleCode program to the target application Remote Call COBOL program.

RCCBL PRDBIN

Use this parameter to specify where RemoteCall can find the COBOL executables. By default, RemoteCall looks in a predefined location. This might not be the location where you've installed them on your system:

- In UNIX, RemoteCall looks in $PS_HOME/cblbin.
- In Windows, RemoteCall looks in %PS_HOME%\cblbin%PS_COBOLTYPE%. The %PS_COBOLTYPE% variable contains a single letter that indicates the character encoding for the database platform. It's automatically set to one of the following values when the application server starts:
  - *U* — Unicode.
  - *A* — Non-Unicode.
  - *E* — EBCDIC.

To override this default behavior, set RCCBL PRDBIN to the absolute path of your COBOL executables, for example:

- In Windows: `RCCBL PRDBIN=c:\pscobol\MYDOMAIN\cblbin`
- In UNIX: `RCCBL PRDBIN=/app/psoft/MYDOMAIN/cblbin`

> **Note**
>
> This parameter doesn't appear in the PSADMIN custom configuration interface if it's not already set. You must define it by editing the application server configuration file directly. On the PeopleSoft Domain Administration menu, select *Edit configuration/log files menu*, then select *Edit psappsrv.cfg (current configuration file)* to open psappsrv.cfg in a text editor. Define the RCCBL PRDBIN parameter in the RemoteCall section of the file.

PSAPPSRV Options

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.

Min Instances

Enter the minimum number of application server instances that start when you boot the domain. There's always at least this number of instances running. This translates to the PSAPPSRV server's -m (min) parameter in the UBB file.

Max Instances

Enter the maximum number of server instances that can be started. This translates to the PSAPPSRV server's -M (Max) parameter in the UBB file.

Service Timeout

Enter the number of seconds that a PSAPPSRV waits 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, PSAPPSRV is terminated and a replacement process is started by Tuxedo.

Recycle Count

Enter the number of service requests that each server has carried out before being terminated (intentionally) and then immediately restarting. Servers must be intermittently recycled to clear buffer areas. The time that is required to recycle a server is negligible, occurring in milliseconds. The 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.

> **Note**
>
> The default value for the small, medium, and large templates is 10,000.

> **Note**
>
> Serial Recycle or Dynamic Recycle parameters can delay the recycle even if the number of the requests is greater than or equal to the Recycle Count value.

ProcessRestartMemoryLimit

This parameter can be set at the individual server process level or the domain level.

See ProcessRestartMemoryLimit.

Allowed Consec Service Failures

Enter a number greater than 0 to enable dynamic server processes to restart for service failures. To disable this option, enter 0. The default is 2. The value that 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 enables PSAPPSRV, PSQCKSRV, and PSAMSRV to terminate themselves if they receive multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively, but if they do, the server process must be recycled or cleansed. A retry message appears on the client browser when this occurs.

Max Fetch Size

The default is 5000 (K). Enter the maximum memory that is 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. You should use the default value. PSAPPSRV supports nonconversational transactions, so this parameter provides 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 is used. The memory is not preallocated—it is acquired as needed for each transaction.

Auto Select Prompts

Enter 1 (the default) to enable automatic prompting on lookup pages. When the user selects the prompt lookup button, the application server automatically returns all values for that field, up to 300 rows. If necessary, the user can refine the search further by entering partial data in the Search By field.

Enter 0 to require the user to enter a partial value before the automatic prompt list appears.

AutoLoad JVM

You can add this parameter, if needed, manually to the [PSAPPSRV] section of the PSAPPSRV.CFG file. AutoLoad JVM controls whether the JVM gets loaded automatically when the domain boots.

By default, domain behavior reflects a setting of AutoLoad JVM=0 (not enabled).

If you have configured EnablePPM Agent=1, then JVM will be loaded at the domain boot time. If you have configured EnablePPM Agent=0, then to load JVM when the domain boots, you need add AutoLoad JVM=1 to the [PSAPPSRV] section of the PSAPPSRV.CFG file.

Serial Recycle

When serial recycling is enabled for a server process, the system recycles server processes of that type within a domain on a serial basis–one after another–to allow processing to continue uninterrupted. By default, the domain behavior reflects Serial Recycle=Y (enabled).

When Serial Recycle=Y for PSAPPSRV, for example, then only one PSAPPSRV process will recycle during the recycle time of 60 seconds when the Recycle Count limit is reached. Then the next PSAPPSRV will recycle.

When serial recycling is not enabled, all the server processes of that type recycle simultaneously when the Recycle Count limit is reached, which can cause throughput to pause.

Serial recycling applies to these server process types: PSAPPSRV, PSQCKSRV, PSQRYSRV, and the Integration Broker server processes.

To disable serial recycling, manually add the Serial Recycle parameter, and assign the value *N*. For example:

```
Serial Recycle=N
```

The recycle time is 60 seconds.

Tuxedo Queue Size

The Integration Broker messaging system uses the Tuxedo queue size indicated in the application server domain section of PSADMIN to determine when the Tuxedo queue size has reached its maximum.

For more details on this setting, see Integration Broker Administration: Setting the Oracle Tuxedo Queue Size.

PSANALYTICSRV Options

PSANALYTICSRV relates to the server processes that are associated with the *analytic server framework*. Most of the parameters for PSANALYTICSRV need to be reviewed in the PSAPPSRV.CFG file and uncommented as needed.

Related Topics

- MultiChannel Framework

Min Instances

Enter the minimum number of analytic server instances that start when you boot the application server domain. There's always at least this number of instances running. The default value of this parameter is *3*.

Max Instances

Enter the maximum number of analytic server instances that can result from dynamically spawning new processes. The default value of this parameter is *3*.

Analytic Instance Idle Timeout

Enter the number of minutes of inactivity before the analytic instance times out and is unloaded.

This value takes effect only if the PeopleCode AnalyticInstance class Load method specifies a value of *-1* for its IdleTimeOut parameter when loading an analytic instance. This includes Load PeopleCode that's launched from an analytic grid, which enables you to avoid having to explicitly specify a timeout.

The default value of this parameter is *0* (no timeout limit) for domains that are configured with a developer template, and *30* minutes for other domains.

ProcessRestartMemoryLimit

You can set this parameter at the individual server process level, or the domain level.

See ProcessRestartMemoryLimit.

PSSAMSRV Options

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

Min Instances

Enter how many servers are started at boot time. This translates to the PSSAMSRV server's -m (min) parameter in the UBB file.

Max Instances

Enter the maximum number of servers that can be started. This translates to the PSSAMSRV server's -M (Max) parameter in the UBB file.

Service Timeout

Enter the number of seconds that the server processes waits for a request before timing out. This stops runaway processes, like an rccbl timeout.

Recycle Count

Enter the number of service requests that each server carries out before being terminated (intentionally). Tuxedo immediately restarts the server. Servers must be intermittently recycled to clear buffer areas. The time that is required to recycle a server is negligible, occurring in milliseconds. The 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.

ProcessRestartMemoryLimit

This parameter can be set at the individual server process level or the domain level.

See ProcessRestartMemoryLimit.

Allowed Consec Service Failures

Enter a number greater than zero to enable dynamic server process restarts for service failures. To disable this option, enter 0. The default is 2. The value that you enter is the number of consecutive service failures that cause a recycle of the server process. This is a catchall error handling routine that enables PSAPPSRV, PSQCKSRV, and PSSAMSRV to terminate themselves if they receive multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively, but if they do, the server process must be recycled or cleansed. A retry message appears on the client browser when this occurs.

Max Fetch Size

The default is 32 (K). Enter the maximum memory that is used by the server to store fetched rows for a transaction before sending results to the 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 the client until the query is complete. You should leave the default value unchanged.

PSSAMSRV supports conversational transactions, so this parameter enables users to tune performance by adjusting the number of network round-trips that are 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. The memory is not preallocated, but is acquired as needed.

PSQCKSRV Options

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

Min Instances

Enter how many servers are started at boot time. This translates to the PSQCKSRV server's –m (min) parameter in the UBB file.

Max Instances

Enter the maximum number of servers that can be started. This translates to the PSQCKSRV server's –M (Max) parameter in the UBB file.

Service Timeout

Enter the number of seconds that a PSQCKSRV waits for a request before timing out. This stops runaway processes, like an rccbl timeout. This applies to incremental PSQCKSRV servers that are dynamically started by the Max Instances parameter.

Recycle Count

Use the PSAPPSRV specifications.

ProcessRestartMemoryLimit

This parameter can be set at the individual server process level or at the domain level.

See ProcessRestartMemoryLimit.

Allowed Consec Service Failures

Enter a number greater than zero to enable dynamic server process restarts for service failures. To disable this option, enter 0. The default is 2. The value that 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 enables PSAPPSRV, PSQCKSRV, and PSAMSRV to terminate themselves if they receive multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively, but if they do, the server process must be recycled or cleansed. A retry message appears on the client browser when this occurs.

Max Fetch Size

Use the PSAPPSRV specifications.

Serial Recycle

Use the PSAPPSRV specifications.

PSQRYSRV Options

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

> **Note**
>
> When running PS/nVision reports from a three-tier, Windows client connection, the system also routes the SQL generated by both matrix (ledger-based) and tabular (PS Query-based) reports through PSQRYSRV if it is enabled.

PSQCKSRV also processes SQLRequest services; however, if PSQRYSRV is configured, it processes all SQLRequests that are initiated specifically by PSQuery (SQLQuery:SQLRequest) or PS/nVision.

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

Min Instances

Enter how many servers are started at boot time. This translates to the PSQRYSRV server's –m (min) parameter in the UBB file.

Max Instances

Enter the maximum number of servers that can be started. This translates to the PSQRYSRV server's –M (Max) parameter in the UBB file.

Service Timeout

Enter the number of seconds that PSQRYSRV waits for a request before timing out. This stops runaway processes.

Recycle Count

Enter the number of service requests that each server carries out before being terminated (intentionally) by Tuxedo and then immediately restarted. Servers must be intermittently recycled to clear buffer areas. The time that is required to recycle a server is negligible, occurring in milliseconds.

If the recycle count is set to 0, PSQRYSRV is never recycled.

ProcessRestartMemoryLimit

This parameter can be set at the individual process level or at the domain level.

See ProcessRestartMemoryLimit.

Allowed Consec Service Failures

Enter a number greater than 0 to enable dynamic server process restarts for service failures. To disable this option, enter 0. The default is 2. The value that 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 enables PSAPPSRV, PSQCKSRV, PSQRYSRV, and PSSAMSRV to terminate themselves if they receive multiple, consecutive, fatal error messages from service routines. Such errors should not occur consecutively, but if they do, the server process must be recycled or cleansed. A retry message appears on the client browser when this occurs.

Max Fetch Size

Enter the maximum size (in KB) of a result set that is returned from a SELECT query. Use 0 for no limit.

Use Dirty-Read

(Applies only to DB2 systems.) Enter 1 to enable the application server to read uncommitted data from a table. Enter 0 to disable dirty reads.

This parameter can be used for general reporting and PeopleSoft Query. You can run dirty read queries through the application server, the Process Scheduler, and in a two-tier connection. The Use Dirty-Read setting in the application server configuration controls the behavior of PSAPPSRV, PSQCKSRV, and PSQRYSRV.

> **Note**
>
> Dirty reads are not recommended if you are reading data and doing subsequent processing based on the disposition of the data at the time that it is read. Between the time the data is read by a subsequent process and the time the unit of work is completed by the first process, any activity affecting the table data at the time a subsequent process read could be rolled back, invalidating the accuracy of the data that a subsequent process read.

Related Topics

- Query
- Process Scheduler: Setting Parameters for the Application Engine Server

Serial Recycle

Use the PSAPPSRV specifications.

Integration Broker Server Processes

A variety of server processes are devoted to Integration Broker processing. These servers are configured in these sections:

- Publish&Subscribe
- PSBRKDSP
- PSBRKHND
- PSSUBDSP
- PSSUBHND
- PSPUBDSP

- PSPUBHND

These server processes act as dispatchers and handlers of the messages being distributed by the Integration Broker.

Related Topics

- Integration Broker Administration

SMTP Settings

You can send electronic mail requests, issued with workflow or PeopleCode, to the application server, which passes the requests to the specified mail server (SMTPServer). To specify the appropriate SMTP server and port to receive the email requests, you must edit the SMTP Settings section.

When set in the PSAPPSRV.CFG file, these SMTP settings are not dynamic: SMTPTrace and SMTPSendTime. They require a domain reboot to take full effect.

> **Note**
>
> You can also control most of these settings using the PeopleCode SMTPSession class, which temporarily overrides them without changing the values in PSAPPSRV.CFG. You use this class primarily when you want to send multiple emails in a single session of the SMTP server, instead of having to change the permanent SMTP settings for every email. In the event that your application PeopleCode does not specify its own SMTP settings, the system uses the settings in the PSAPPSRV.CFG file.

> **Note**
>
> Most of the SMTP settings documented here also apply to Process Scheduler configurations, and can be updated in the PSPRCS.CFG file. See Process Scheduler: SMTP Section.

See PeopleCode API Reference: SMTPSession Class.

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@example.com.

SMTPBlackberryReplyTo

Enter the internet address that you want to be the reply to address for Blackberry email responses. This must be a valid address such as user1@example.com.

See Workflow Technology: Setting Properties for BlackBerry Email Responses

SMTPSourceMachine

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

SMTPCharacterSet

Enter the character set that is used on the sender's machine.

SMTPEncodingDLL

Enter the name of a dynamic-link library (DLL) that is used to translate the mail message from the sender's character set (such as latin1, sjis, big5, gb, ks-c-5601-1987, or ks-c-5601-1992) to a 7-bit safe character set for transmission.

SMTPGuaranteed

Set this parameter to *1* if you want TriggerBusinessEvent email PeopleCode to be delivered through the Integration Broker system, which provides some additional administration capabilities for ensuring delivery of the message. By enabling this feature you implement a mechanism to ensure that emails get routed to the appropriate place in case SMTP mail fails for reasons such as network timeouts, downed mail servers, invalid parameters, and so on.

This setting enables the system to track email messages sent using Integration Broker queues. When sending an email with this option enabled, the system performs an asynchronous local-to-local publish, and for the subscription the system calls MCFOutboundMail.send to email the message.

See Workflow Technology: Setting Properties for BlackBerry Email Responses and PeopleCode Language Reference: TriggerBusinessEvent function.

SMTPTrace

Use this parameter to control the tracing of email details. The system writes the log information to SMTP<DD_MM>.log in %PS_SERVDIR%/LOGS, by default, or the custom value set for Log Directory. For example:

PS_CFG_HOME\APPSERV\domain\LOGS\SMTP6_27.log

LogFence should be set to *5*.

See LogFence

- Enter *0* to disable SMTP tracing.
- Enter *1* to enable the tracing of email details to the log file.

  With this option the log file includes extensive details in the log file.
- Enter *2* to enable the tracing of summary information to the log file.

  With this option the log file includes summary information for each email in this format:

  ```
  <Date_Time> Email sent with return code [<Return Code>] From: [<Sender>]
  To:[<Recipient>] Sender:[<Sender>] CC:[<CC Field>] BCC:[<BBC Field>]
  Subject:[<subject_text>]
  ```

  For example:

  ```
  Wed Apr 29 18:48:46 CDT 2020 Email sent with return code [NOT DELIVERED]
  From:[test@example.com] To:[test9@example.com] Sender:[test@example.com]
  CC:[null] BCC:[null] Subject:[This is a test]
  ```

  For a list of the Return Codes, see PeopleCode API Reference: Send method: SMTPSession class

SMTPSendTime

Enter *1* to have messages contain a send time that is populated by the application server. Enter *0* to leave the send time blank and have it populated by the receiving gateway (depending on the gateway).

SMTPUserName

Enter the user name to log in to the SMTP server. This applies only when authentication is enabled on the SMTP server.

SMTPUserPassword

Enter the password for the user specified by *SMTPUserName* to access the SMTP server. This applies only when authentication is enabled on the SMTP server.

Use PSCipher to encrypt the SMTPUserPassword value.

See Security Administration: Using the PSCipher Utility.

SMTPUserName1

Enter the user name to log in to the backup SMTP server. This applies only when authentication is enabled on the backup SMTP server.

SMTPUserPassword1

Enter the password for the user specified by *SMTPUserName1* to access the backup SMTP server. This applies only when authentication is enabled on the backup SMTP server.

Use PSCipher to encrypt the SMTPUserPassword1 value.

See Security Administration: Using the PSCipher Utility.

SMTPTimeToWaitForResult

Enter the time in milliseconds for the mail system to wait for the result of sending each email. If the time is set to *0*, the system doesn't wait, and the returned result will be always be *%ObEmail_SentButResultUnknown (= -1)*. If you set this parameter to *-1*, the system will wait for the completion of the send process. The default value of this setting is 10000 (ten seconds).

SMTPUseSSL

This parameter indicates whether SSL connections are enabled. Enter *Y* for yes or *N* for no. The default value is *N*.

SMTPTLSEnable

To allow PeopleTools to issue the STARTTLS or STLS command to the mail server to secure an insecure connection, add this parameter manually to the PSAPPSRV.CFG file and PSPRCS.CFG file in the SMTP settings section, and set it to *Y* (yes).

Use this to specify whether STARTTLS can be enabled (if supported by the server). If STARTTLS is enabled and configured correctly on the server it will send mail through SSL by issuing a STARTTLS or STLS command. If SSL is not configured correctly or if the STARTTLS command fails, the mail will not be sent. If STARTTLS or STLS is not supported on the server, it will send mail in the normal means, without using SSL.

SMTPTLSRequired

To allow PeopleTools to issue the STARTTLS or STLS command to the mail server to secure an insecure connection, and require that the mail server support the command, add this parameter manually to the PSAPPSRV.CFG file and PSPRCS.CFG file in the SMTP settings section, and set it to *Y* (yes).

Use this to specify whether using STARTTLS or STLS is required. If this property is set to *Y,* mail will be sent only if STARTTLS or STLS is supported on the server. If the server doesn't support these commands, or the command fails, the connect method will fail.

SMTPSSLPort

If using SSL, specify the SSL port on the SMTP server. The default is 465.

SMTPClientCertAlias

If the SMTP server is configured for client authentication, enter the alias name of the client certificate.

SMTPSSLProtocol

Use this parameter to specify the SSL protocol to connect to the mail server.

The allowed values are TLSv1.2 or TLSv1.3. The default is TLSv1.3.

SMTPUseSSL1

This parameter applies to the backup SMTP server.

Use it to indicate whether SSL connections are enabled. Enter *Y* for yes or *N* for no. The default value is *N*.

SMTPTLSEnable1

To allow PeopleTools to issue the STARTTLS or STLS command to the backup mail server to secure an insecure connection, add this parameter manually to the PSAPPSRV.CFG file and PSPRCS.CFG file in the SMTP settings section, and set it to *Y* (yes).

Use this to specify whether STARTTLS can be enabled (if supported by the backup server). If STARTTLS is enabled and configured correctly on the server it will send mail through SSL by issuing a STARTTLS or STLS command. If SSL is not configured correctly or if the STARTTLS command fails, the mail will not be sent. If STARTTLS or STLS is not supported on the server, it will send mail in the normal means, without using SSL.

SMTPTLSRequired1

To allow PeopleTools to issue the STARTTLS or STLS command to the backup mail server to secure an insecure connection, and require that the mail server support the command, add this parameter manually to the PSAPPSRV.CFG file and PSPRCS.CFG file in the SMTP settings section, and set it to *Y* (yes).

Use this to specify whether using STARTTLS or STLS is required on the backup server. If this property is set to *Y*, mail will be sent only if STARTTLS or STLS is supported on the server. If the server doesn't support these commands, or the command fails, the connect method will fail.

SMTPSSLPort1

This parameter applies to the backup SMTP server.

If using SSL, specify the SSL port on the SMTP server. The default is 465.

SMTPClientCertAlias1

This parameter applies to the backup SMTP server.

If the SMTP server is configured for client authentication, enter the alias name of the client certificate.

SMTPSSLProtocol1

Use this parameter to specify the SSL protocol to connect to the backup SMTP server.

The allowed values are TLSv1.2 or TLSv1.3. The default is TLSv1.3.

SMTPDNSTimeoutRetries

This parameter enables you to configure the number of times the IsDomainNameValid method of the MCFMailUtil class retries to verify that the domain of an email address submitted to the method is valid. If you need to override the system default, manually add this parameter to the [SMTP Settings] section of the PSAPPSRV.CFG file.

Related Topics

- PeopleCode API Reference: IsDomainNameValid method: MCFMailUtil class

SMTPUseOAuth

Use this to specify whether to use OAuth2 to connect to the primary SMTP server for outbound email.

To use OAuth 2.0 to connect to the primary SMTP server, set this parameter to *Y*. Set it to *N* if you do not want to use OAuth2.

To use OAuth2 for outbound email, you must complete the tasks to register the OAuth2 application through Microsoft Azure and configure the service operation in PeopleSoft Pure Internet Architecture (PIA).

See MultiChannel Framework: Setting Up MCF Email Using Azure.

SMTPOAuthRedirectURI

Enter the redirect URI for your OAuth2 application.

The SMTPOAuthRedirectURI is listed in the Redirect URI field on the Services Application page (Create OAuth2 Service apps) for the OAuth2 application for outbound email in PeopleSoft Pure Internet Architecture (PIA).

See MultiChannel Framework: Setting Up MCF Email Using Azure.

SMTPOAuthApplicationName

Enter the name for the OAuth2 application.

The SMTPOAuthApplicationName is listed as OAuth Name on the Services Application page (Create OAuth2 Service apps) for the OAuth 2 application for outbound email in PeopleSoft Pure Internet Architecture (PIA).

See MultiChannel Framework: Setting Up MCF Email Using Azure.

SMTPOAuthExternalProxyHost

Use this to specify the external proxy host for OAuth2.

If a proxy is needed to contact microsoft.com, then enter the host name for the proxy here. This is optional for setting up OAuth2 for outbound email.

SMTPOAuthExternalProxyPort

Use this to specify the port for the external proxy.

If a proxy is needed to contact microsoft.com for the OAuth2 application, enter an available port number for the proxy here. This is optional for setting up OAuth2 for outbound email.

SMTPInternalProxyHost

Use this to specify the internal proxy host for OAuth2.

If a proxy is needed to contact your Integration Broker Gateway, then enter the host name for the internal proxy here. This is optional for setting up OAuth2 for outbound email.

SMTPInternalProxyPort

Use this to specify the port for the internal proxy for OAuth2.

If a proxy is needed to contact your Integration Broker Gateway, then enter an available port number for the proxy here. This is optional for setting up OAuth2 for outbound email.

SMTPOAuthPIAUser

Use this to specify a user for OAuth2.

Enter the name of a user authorized to use the MultiChannel Framework OAuth2 Service Application. The user who sends the emails must have the PeopleTools MCF Email role.

See Security Administration: Setting Roles.

See MultiChannel Framework: Setting Up MCF Email Using Azure.

SMTPOAuthPIAPassword

Use this to specify the password for the SMTPOAuthPIAUser.

Enter an encrypted password for the user that is specified in the SMTPOAuthPIAUser property.

See Integration Broker Administration: Encrypting Passwords.

SMTPUseOAuth1

Use this to specify whether to use OAuth2 to connect to the backup SMTP server.

To use OAuth2 to connect to the backup SMTP server, set this parameter to *Y*. Set it to *N* if you do not want to use OAuth2 for a backup server.

SMTPOAuthRedirectURI1

Enter the redirect URI for the OAuth2 application on the backup SMTP server.

The SMTPRedirectURI is listed in the Redirect URI field on the Services Application page (Create OAuth2 Service apps) for the OAuth2 application for outbound email in PeopleSoft Pure Internet Architecture (PIA).

See MultiChannel Framework: Setting Up MCF Email Using Azure.

SMTPOAuthApplicationName1

Enter the name for the OAuth2 application for the backup SMTP server.

The SMTPOAuthApplicationName is listed as OAuth Name on the Services Application page (Create OAuth2 Service apps) for the OAuth2 application for outbound email in PeopleSoft Pure Internet Architecture (PIA).

See MultiChannel Framework: Setting Up MCF Email Using Azure.

SMTPOAuthExternalProxyHost1

Use this to specify the external proxy host for the backup SMTP server.

If a proxy is needed to contact microsoft.com, then enter the host name for the proxy here. This is optional for setting up OAuth2.

SMTPOAuthExternalProxyPort1

Use this to specify the port for the external proxy host for the backup SMTP server.

If a proxy is needed to contact microsoft.com for the OAuth2 application, enter an available port number for the proxy here. This is optional for setting up OAuth2.

SMTPInternalProxyHost1

Use this to specify the internal proxy host for the backup SMTP server.

If a proxy is needed to contact your Integration Broker Gateway, then enter the host name for the internal proxy here. This is optional for setting up OAuth2.

SMTPInternalProxyPort1

Use this to specify the port for the internal proxy for the backup SMTP server.

If a proxy is needed to contact your Integration Broker Gateway, then enter an available port number for the proxy here. This is optional for setting up OAuth2.

SMTPOAuthPIAUser1

Use this to specify a user for OAuth2 on the backup SMTP server.

Enter the name of a user authorized to use the MultiChannel Framework OAuth2 Service Application. The user who sends the emails must have the PeopleTools MCF Email role.

See Security Administration: Setting Roles.

See MultiChannel Framework: Setting Up MCF Email Using Azure.

SMTPOAuthPIAPassword1

Use this to specify the password for the SMTPOAuthPIAUser on the backup SMTP server.

Enter an encrypted password for the user that is specified in the SMTPOAuthPIAUser property.

See Integration Broker Administration: Encrypting Passwords.

SMTPSMIMEEncryption

Use this to specify whether to encrypt outbound emails using the Secure/Multipurpose Internet Mail Extensions (S/MIME) format.

Set to N (no) to disable encrypted outbound emails, or Y (yes) to turn on S/MIME encryption. The default is N.

See MultiChannel Framework: Configuring Signed and Encrypted Emails for additional configuration required for encrypted outbound emails.

You can override the setting using PeopleCode. See PeopleCode API Reference: SMIMEEncryption property: MCFEmail class.

SMTPSMIMEEncryption1

Use this to specify whether to encrypt outbound emails using the Secure/Multipurpose Internet Mail Extensions (S/MIME) format on the backup SMTP server.

Set to N (no) to disable encrypted outbound emails, or Y (yes) to turn on S/MIME encryption. The default is N.

See MultiChannel Framework: Configuring Signed and Encrypted Emails for additional configuration required for encrypted outbound emails.

You can override the setting using PeopleCode. See PeopleCode API Reference: SMIMEEncryption property: MCFEmail class.

SMTPSMIMESign

Use this to specify whether to send outbound emails with digital signatures using the Secure/Multipurpose Internet Mail Extensions (S/MIME) format.

Set to N (no) to disable digital signatures for outbound emails, or Y (yes) to turn them on. The default is N.

See MultiChannel Framework: Configuring Signed and Encrypted Emails for additional configuration required for outbound emails with digital signatures.

You can override the setting using PeopleCode. See PeopleCode API Reference: SMIMESign property: MCFEmail class.

SMTPSMIMESign1

Use this to specify whether to send outbound emails with digital signatures using the Secure/Multipurpose Internet Mail Extensions (S/MIME) format on the backup SMTP server.

Set to N (no) to disable digital signatures for outbound emails, or Y (yes) to turn them on. The default is N.

See MultiChannel Framework: Configuring Signed and Encrypted Emails for additional configuration required for outbound emails with digital signatures.

You can override the setting using PeopleCode. See PeopleCode API Reference: SMIMESign property: MCFEmail class.

SMTPSMIMEHandling

Use this to specify whether to send outbound email if there are encryption and digital signing errors.

Set to 0 (zero) to ignore S/MIME errors and send email in plain text if it can not be signed or encrypted. The errors are recorded in the SMTP log.

Set to 1 (one) to not send email if S/MIME errors occur. The default is 1.

See MultiChannel Framework: Configuring Signed and Encrypted Emails for additional configuration required for outbound emails with digital signatures.

SMTPSMIMEHandling1

Use this to specify whether to send outbound email if there are encryption and digital signing errors on the backup SMTP server.

Set to 0 (zero) to ignore S/MIME errors and send email in plain text if it can not be signed or encrypted. The errors are recorded in the SMTP log.

Set to 1 (one) to not send email if S/MIME errors occur. The default is 1.

See MultiChannel Framework: Configuring Signed and Encrypted Emails for additional configuration required for outbound emails with digital signatures.

Implementing NT LAN Manager

PeopleTools supports NT LAN Manager (NTLM) as a primary authentication mechanism. NTLM is a suite of authentication and session security protocols used in various Microsoft network protocol implementations. To use NTML with PeopleTools, you need to add these parameters manually to the PSAPPSRV.CFG file and PSPRCS.CFG file in the SMTP settings section.

Parameter	Description
SMTPNTLMEnable	Specify whether NTLM is enabled. Add *true* or *false*.
SMTPNTLMDomain	Specify the NTLM domain.
SMTPNTLMEnable1	Specify whether NTLM is enabled for the backup server.
SMTPNTLMDomain1	Specify the NTLM domain for the backup server.

Implementing Simple Authentication and Security Layer

PeopleTools supports Simple Authentication and Security Layer (SASL) as a primary authentication mechanism. SASL is a framework for authentication and data security in many internet protocols. While there are several SASL mechanisms, PeopleTools supports PLAIN SASL only. To use SASL with PeopleTools, you need to add these parameters manually to the PSAPPSRV.CFG file and PSPRCS.CFG file in the SMTP settings section.

Parameter	Description
SMTPSASLEnable	Specify whether SASL is enabled.
SMTPSASLMechanism	Specify the SASL mechanism.

Parameter	Description
SMTPSASLAuthorizationid	Specify the authorization ID
SMTPSASLRealm	Specify the realm.
SMTPSASLEnable1	Specify whether SASL is enabled for the backup server.
SMTPSASLMechanism1	Specify the SASL mechanism for the backup server.
SMTPSASLAuthorizationid1	Specify the authorization ID for the backup server.
SMTPSASLRealm1	Specify the realm for the backup server.

Implementing Transport Layer Security (TLS)

PeopleTools supports TLS for SMTP security. To use TLS with PeopleTools, see the preceding definitions for these parameters:

- SMTPTLSEnable
- SMTPTLSRequired
- SMTPTLSEnable1
- SMTPTLSRequired1

SMTP Further Considerations

Keep in mind the following considerations:

- PeopleSoft mail integration is on the application server only.

  PeopleSoft software does not support VIM/MAPI, because this option is client-side-only integration, and PeopleSoft Internet Architecture applications run on the server side.
- The application server communicates directly with the SMTP server through telnet by using standard SMTP commands with Multipurpose Internet Mail Extensions (MIME) 1.0 messages.
- PeopleSoft software currently supports UTF-8 encoding of the email messages by default, and you can encode email messages in other ways.

  With server-side integration, you do not have to certify any specific email client application. You can use any application to read email.
- PeopleSoft recommends using the Multichannel Framework mail classes for all email sent from a PeopleSoft application.

> **Note**
>
> For more information on certificates for SSL communication, see How To Configure SMTP SSL on Application Servers and Process Schedulers on My Oracle Support (Doc ID 1612188.1). For more information on setting up client certificate authentication, see How to Setup SMTP Client Certificate Authentication in PeopleSoft (Doc ID 2876697.1).

Related Topics

- PeopleCode API Reference: Understanding PeopleSoft MultiChannel Framework Mail Classes

Interface Driver Options

Set the following parameter for configuring the interface driver for business interlinks.

> **Note**
>
> PeopleSoft Business Interlinks is a desupported product and these options currently exist only for upgrade compatibility and transition.

SCP_LOCALE

Enter the RPS_LOCALE string, which the driver sends to the Supply Chain Planning (SCP) server.

PSTOOLS Options

You may need to set the following parameters in advanced configurations.

Enable Remote Administration

Use the Remote Administration parameters to enable embedded Java Management Extensions (JMX) agents for use with PeopleSoft Health Center (PHC). By default the value is set to enabled. With this enabled, a Server Runtime JMX agent will be started in all managed servers, and the Domain Runtime JMX agent will be started in PSMONITORSRV.

To improve security, you can configure SSL for JMX agents. Create a keystore and truststore for the JMX agents and enter the information here.

See Performance Monitor: Enabling SSL for JMX Agents

- Enable Remote Administration is enabled by default.

  Change the value to *1* to enable, and to *0* to disable. When you enable Remote Administration, you must also specify the port, user ID and password.

  > **Note**
  >
  > When configuring an Application Server domain in psadmin, if you do not disable the setting and you do not supply any information, you see a message such as “Enable PS JMX is not 0, but some of PS JMX parameters (userId, userPwd) are empty. PHC will not be able to work.”

- Remote Administration Port

  Specify the port number to use for remote administration of domain processes. All embedded JMX agents will use the port in the JMX service URL.

- Remote Administration UserID

Specify the user name for remote administration connections between JMX agents.

- Remote Administration Password

  Specify the password for remote administration connections. You do not have to enter an encrypted password. When you use psadmin to edit a psappsrv.cfg or psprcs.cfg file, you can enter the password in clear text, and psadmin will encrypt it when a domain is created and configured.

- Remote Administration SSL Enabled

  Change the value to *1* to enable SSL for JMX agents, and to *0* to disable. The default is *0*, disabled. When you enable SSL for JMX agents, you must also specify the location, password, and type for the keystore and truststore.

- Remote Administration SSL Keystore

  Specify the location of the keystore for SSL.

- Remote Administration SSL Keystore Password

  Specify the password for the SSL keystore. The password must be encrypted with PSCipher.

  See Security Administration: Using the PSCipher Utility

- Remote Administration SSL Keystore Type

  Specify the keystore type, PKCS12 or JKS.

- Remote Administration SSL Truststore

  Specify the location of the truststore for SSL.

- Remote Administration SSL Truststore Password

  Specify the password for the SSL truststore. The password must be encrypted with PSCipher.

  See Security Administration: Using the PSCipher Utility

- Remote Administration SSL Truststore Type

  Specify the truststore type, PKCS12 or JKS.

Related Topics

- Performance Monitor: Configuring PeopleSoft Health Center
- Performance Monitor: Enabling Monitoring of a Component or Domain

EnablePPM Agent

Controls whether the Performance Monitor agent runs and collects performance monitor data. Enter *1* to enable the Performance Monitor agent, and *0* to disable the Performance Monitor agent. This setting overrides the value for this parameter that is set in the database.

Related Topics

- Performance Monitor

Add to CLASSPATH

The Add To CLASSPATH environment variable parameter enables the Java Virtual Machine (JVM) and other Java applications to locate the Java class libraries, including any user-defined class libraries. Because PeopleTools automatically generates CLASSPATH entries for core, delivered class libraries, use this variable to specify additional class libraries that the PeopleSoft software needs to access. To use this parameter, you need to uncomment it in the PSAPPSRV.CFG file.

The PeopleCode API Reference provides details on where you can place custom and third-party Java classes.

See PeopleCode API Reference: System Setup for Java Classes.

JavaVM Options

Specify additional options to be passed to the JVM loaded by the domain's server processes. Separate the options with spaces, for example:

```
-Xrs -Xmx256m -Xms256m
```

If the domain will run as a Windows service, you must specify at least the default option, -Xrs.

> **Note**
>
> If you are using the AIX operating system, these options may be required: `JavaVM Options=-Xrs -Djava.awt.headless=true -Xcheck:jni`.

The JavaVM Options parameter specified in the [PSTOOLS] section specifies global JavaVM options used by every server process in a domain. To override this global value for a particular server process, you can apply custom JavaVM options to individual server processes by adding the JavaVM Options parameter manually to the configuration section for that server process.

JavaVM Options can appear multiple times in a single PSAPPSRV.CFG or PSPRCS.CFG file. While the JavaVM Options value in the [PSTOOLS] section applies to all server processes governed by a particular configuration file, the system only uses the JavaVM Options value in the [PSTOOLS] section for server processes that do not have the JavaVM Options parameter added to its configuration settings section.

For example, if the JavaVM Options parameter has been added to the [PSAPPSRV] section of the PSAPPSRV.CFG file and has been assigned a value, then that value will be used when loading the JVM as a thread of that PSAPPSRV process. If the JavaVM Options parameter does not appear, or has no value, in the [PSAPPSRV] section, then the system uses the value specified in the [PSTOOLS] section when loading the JVM as a thread of the PSAPPSRV process. This applies to any server process: PSAPPSRV, PSQRYSRV, PSAESRV, and so on.

See your JRE documentation for valid JVM options.

> **Note**
>
> If the server runs behind a proxy server, and Java code will be invoked, you must set the -D Java options to reflect your proxy connection settings. For example, Java code invoked by any Tuxedo-managed server, such as the application server or Process Scheduler server, requires the following Java options: -Dhttp.proxyHost=<value>, -Dhttp.nonProxyHosts=<value>, and -Dhttp.proxyPort=<value>.

Proxy Host

If the HTTP destination, such as the gateway host, is behind a proxy server for security reasons, enter the distinguished name of the proxy server, as in proxy_server.example.com.

> **Note**
>
> The proxy settings configured in the PSAPPSRV.CFG file do not apply to the embedded JVM. For Java code invoked on a PeopleSoft server (application server, Process Scheduler server, and so on) that requires a connection to the proxy, set the Java proxy settings using the Java VM Options setting in the PSAPPSRV.CFG file or PSPRCS.CFG file.

Proxy Port

Enter the port number on which the proxy server is listening for transmissions. For instance, 80 is a typical default port number.

Non Proxy Hosts

Enter a list of the hosts that should be connected to directly, not through a proxy server. Separate the host names with a pipe symbol (|). You can use an asterisk (*) as a wildcard character to specify a pattern of similar host names.

For example, `localhost|*.example.com` indicates that the local host and all hosts with names ending in .example.com will be accessed directly.

> **Note**
>
> The length of this string cannot exceed 1024 characters.

Display

Remove the comment character to set the DISPLAY environment variable on headless UNIX boxes running applications with chart graphics. Xvfb must also be started and configured to use this value.

For example, Display=localhost:10.0.

Character Set

Enter the character set to be used as the system locale. This setting is used to convert text to and from UNICODE when using interfaces that do not support UNICODE. Such interfaces include file names, text file contents, and other operating system calls that require non-UNICODE text. This setting controls how files used by the PeopleTools file attachment feature are named.

The default values are:

Operating System	Value
UNIX	*latin1*
z/OS	*ccsid1047*
Windows	*CP_ACP*

Note

By default, the parameter is commented out. With no value specified, invalid values, or blank value (" "), the system assumes the above default values as the character set.

To override the default, remove the comment character for the Character Set parameter, and select a character set from the following list corresponding to the languages that this application server will process. For example, to set the character set for Simplified Chinese:

```
Character Set=gb2312
```

Note

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

The *utf8* option is valid only when the locale character set is UTF-8.

Related Topics

- Global Technology: Character Sets Across the Tiers of the PeopleSoft Architecture

Suppress App Error Box (Microsoft Windows Only)

Enter *y* to suppress an application error box or message from appearing after an application error occurs. Enter *n* to view error dialogs and message boxes.

Note

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

DbFlags

The following values are valid for the DbFlags parameter:

Value	Description
0	Enable the %UpdateStats meta-SQL construct.

Value	Description
1	Disable the %UpdateStats meta-SQL construct.
4	Disable a secondary database connection (used with the GetNextNumberWithGapsCommit PeopleCode function). This prevents the creation of a secondary database connection, bundling all SQL into a single unit of work. Without the additional database connection, the database row lock is held for a longer time, reducing concurrency in a multiple-user environment. **Important:** Analytic instance processing requires a secondary database connection, so if you're using analytic servers, ensure that this value is not set. In addition, the CallAppEngine PeopleCode function requires a secondary connection when used with the allowcommit parameter set to *True*. Therefore, ensure that this value is not set when using CallAppEngine with allowcommit set to *True.*
8	Disables persistent secondary database connections (used by either the GetNextNumberWithGapsCommit PeopleCode function or the CallAppEngine PeopleCode function when the allowcommit parameter is set to True). When this value is specified, any secondary database connection created by GetNextNumberWithGapsCommit or by CallAppEngine is immediately closed after the commit completes execution. This keeps the number of database connections to a minimum, but requires each call to create a new secondary database connection on demand. **Important:** The performance impact of creating each new secondary database connection is significant, especially in high volume user production environments. Therefore, don't use this setting without carefully considering its effect.

DbFlags uses a bit mask so that you can specify one or more of these values. You set this parameter to the total of the values that you want to apply. For example, to disable %UpdateStats and ignore the Truncate command, set DbFlags to *3* (setting bits one and two).

The default value is *0*.

See Application Engine: %UpdateStats meta-SQL element.

Suppress SQL Error

This option is not available through the PSADMIN interface, but it exists in the PSTOOLS section of the PSAPPSRV.CFG file for small, medium, and large configurations.

For security purposes, this option has a default value of *1* to prevent SQL error details from being displayed to users. Any SQL errors that occur don't display details, but refer users to consult the system log. The details that were in the SQL message are written to the log file. This helps to prevent SQL injection vulnerabilities.

If you want SQL error details to be visible to users, set this property as follows:

```
Suppress SQL Error=0
```

> **Note**
>
> In developer configurations, the Suppress SQL Error option doesn't exist in PSAPPSRV.CFG, and the system assumes a value of *0*.

Related Topics

- PeopleCode Developer's Guide: Preventing SQL Injection

Usage Monitor Specific Option

Controls whether Usage Monitoring is enabled for the domain. Enter *1* to enable usage monitoring for this domain. Enter *0* to disable usage monitoring for this domain even when Usage Monitoring is enabled system-wide.

The default value is `UMOption=0`, meaning monitoring is not enabled for the domain.

For information on using Usage Monitor, see Usage Monitor: Enabling Usage Monitor.

To enable or disable Usage Monitoring for the entire PeopleSoft system, see Security Administration: Understanding General Options.

This table describes how the server domain configuration setting and the System Personalizations setting work together to determine whether Usage Monitoring is enabled for a domain.

Configuration Option UMOption	System Personalizations PTUSEMON Option	Usage Monitoring Feature
1	Enabled	ON
0	Enabled	OFF
1	Disabled	OFF
0	Disabled	OFF

Integration Broker Options

The following parameters apply to Integration Broker. All of the options for Integration Broker are discussed in detail in the Integration Broker PeopleBooks.

Related Topics

- Integration Broker Administration

Min Message Size for Compression

Use this parameter to configure the threshold of a message before the system compresses the message.

Thread Pool Size

Set the thread pool size used by the SyncRequest PeopleCode event. The Minimum value is 1 and maximum allowable is 20.

Disable Physical Document Cache

Use this parameter to disable cache for physical representations (json, xml).

Related Topics

- Integration Broker Administration: Specifying the Disable Physical Document Cache Parameter

PSRENSRV Options

PSRENSRV is a modified web server designed for real time event notification. The primary purpose of PSRENSRV is to publish events to the browser.

See MultiChannel Framework: Understanding REN Servers.

log_severity_level

This is the log severity level for the PSRENSRV process. Settings are Error, Warning, Notice or Debug. Default is Warning.

io_buffer_size

This is the TCP buffer size when serving content. This should not exceed a value of *65536*.

default_http_port

This is the REN server's HTTP port. The default value is *7180*.

default_https_port

This is the REN server's HTTPS port. The default value is *7143*.

default_auth_token

The fully qualified domain name of the server (network domain, not the PeopleSoft application server domain). This value should match the value of the web server's authentication token domain.

PSPPMSRV Options

PSPPMSRV servers subscribe to performance metrics published by the web service at the PPMI URL (specified in the Performance Monitor administration pages) and insert them into the database. If you select *Y* when you are asked whether you want Performance Collators configured, then the number of PSPPMSRVs specified in Min Instances=1 will be started. Min and Max instances should be set to the same value, as new PSPPMSRV servers *are not* spawned on demand.

Related Topics

- Performance Monitor

Min Instances

The number of servers started at boot time. This translates to the PSPPMSRV server's –m (min) parameter in the UBB file.

Max Instances

The maximum number of servers that can be started. This translates to the PSPPMSRV server's –M (max) parameter in the UBB file.

LDAP Settings

See Security Administration: Configuring LDAP Connection Parameters.

7

Working with Oracle WebLogic

Understanding WebLogic

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

- Single server.

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

- Multi server.

  This domain configuration contains seven unique server definitions and a WebLogic cluster, and the PeopleSoft application is split across multiple server instances on a *single* machine. This configuration is intended for the production environment.

- Distributed managed server.

  This option is an extension of the “Multi server” selection and installs the necessary files to boot managed servers spread across *multiple* machines. This option requires a “Multi server” installation to be performed to some other location that contains the configuration for this managed server.

> **Note**
>
> The topics related to WebLogic in this documentation discuss PeopleSoft-specific subjects. Always refer to your Oracle WebLogic documentation for all general server administration information.

Using WebLogic Remote Console

Use the Oracle WebLogic Remote Console to administer and monitor the WebLogic server processes.

> **Note**
>
> The WebLogic Remote Console replaces the WebLoigc Server Administration Console that was used in previous releases.

The WebLogic Remote Console can be installed as a desktop application (recommended) or a hosted web application. This section gives a brief introduction to using the WebLogic Remote Console for a PeopleSoft environment. Use these resources for complete information:

- Oracle WebLogic Remote Console download location in GitHub, https://github.com/oracle/weblogic-remote-console/releases
- Oracle WebLogic Remote Console documentation in GitHub, https://oracle.github.io/weblogic-remote-console/
- Oracle WebLogic documentation, https://docs.oracle.com/en/middleware/fusion-middleware/weblogic-remote-console/administer/index.html

Accessing the WebLogic Remote Console

To access the WebLogic Remote Console:

1. Follow the installation instructions in the Oracle WebLogic documentation.
2. Access the WebLogic Remote Console in one of these ways:
   - If you are using the desktop application, add the provider with the admin server user name, password, and this URL: https://*<server>:<port>*.
   - If you are using the hosted web application, enter this URL in your browser address field:

     http://*<server>:<port>*/rconsole

     For example,

     http://server08.example.com:8000/rconsole
   - If it is a multi server installation the Remote Console from the Admin Server is always running on port 9999.

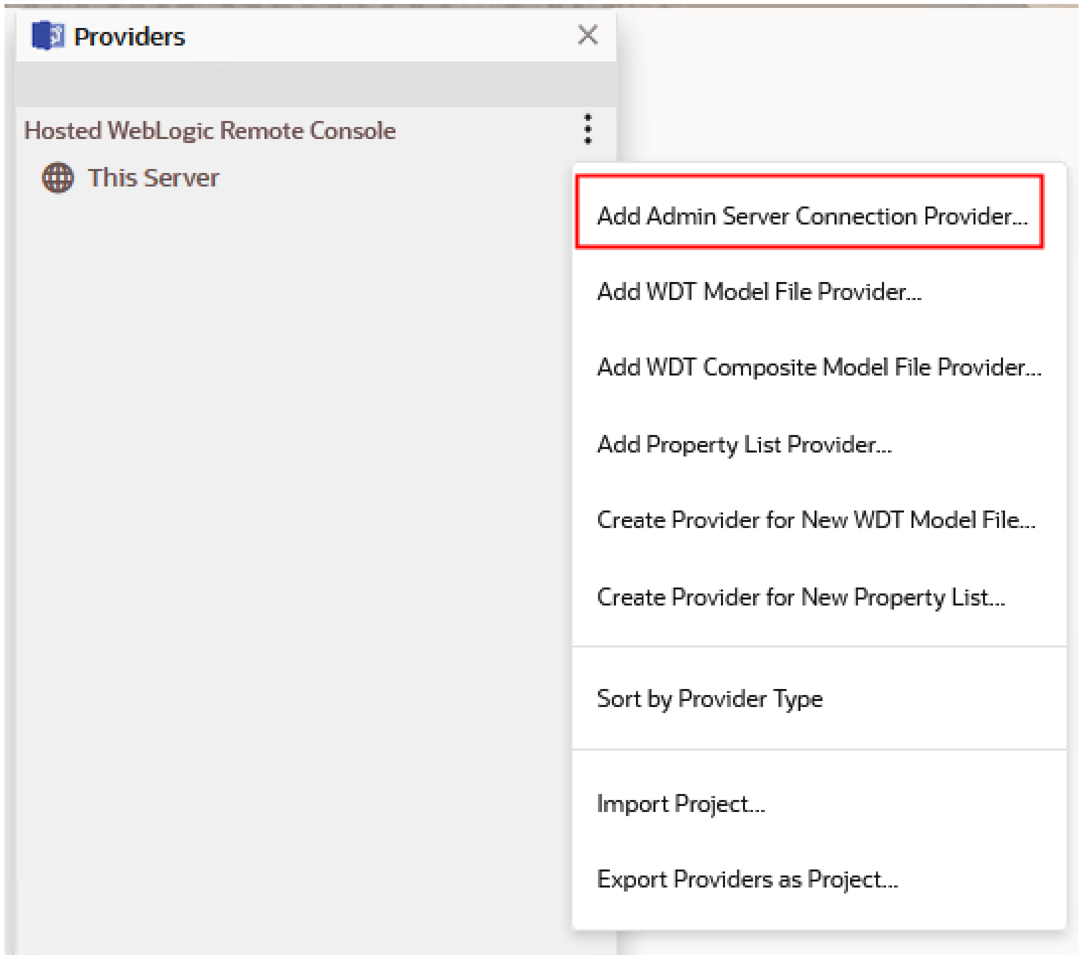
     You must set up a multi server installation manually. If you use the Remote Console hosted web in a multi server installation, the URL is: http://*<server>*:9999/rconsole.

     If you use the desktop application, enter http://*<server>*:9999 as the URL for the provider.

   The default ports for the web server (PIA) set by the DPK setup script are 8000 for HTTP and 8443 for HTTPS. The ports are listed in the psft_configuration.yaml file. See *PeopleSoft PeopleTools <current release> Deployment Packages Installation*.
3. On the sign in page, enter the name for the WebLogic server administrator and password that you supplied during the PeopleSoft environment installation.

   If you accepted the default, the administrator name is system.
4. Under Providers, select Add Admin Server Connection Provider.

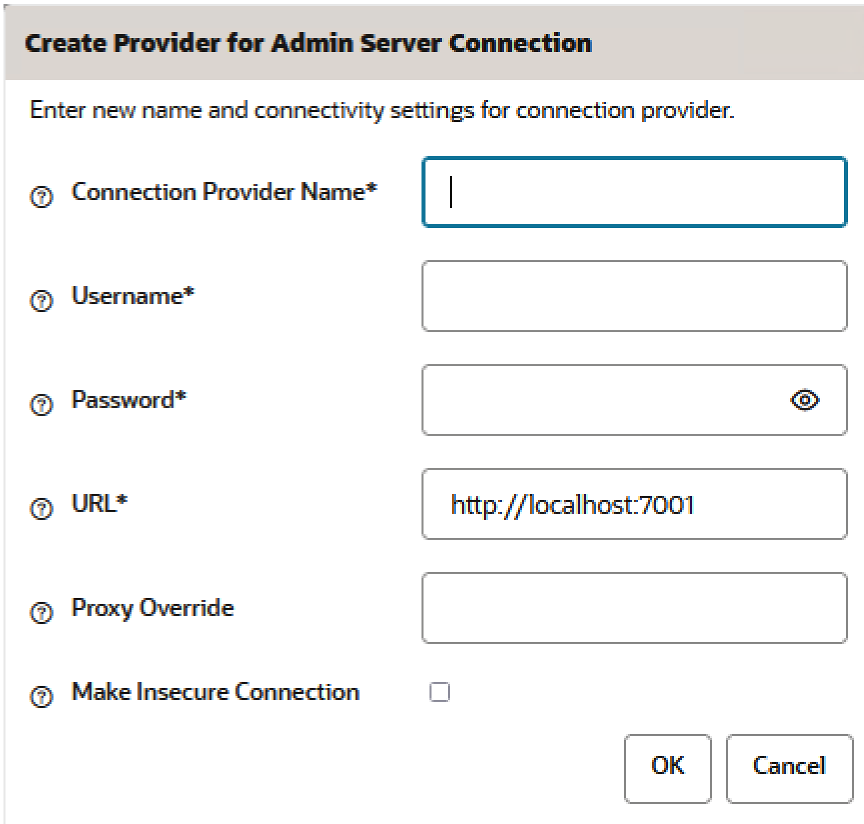
   This example illustrates the Providers menu.



5. Supply the required connection information.
   - Connection Provider Name — Enter the host name for the web server.
   - Username — Enter the WebLogic administrator name. The default is system.
   - Password — Enter the password for the WebLogic administrator.
   - URL — Enter the URL to access PIA, http://*<server>*:*<port>*.
   - Proxy Override (optional)
   - Make Insecure Connection

     Do not select this option for SSL environments.

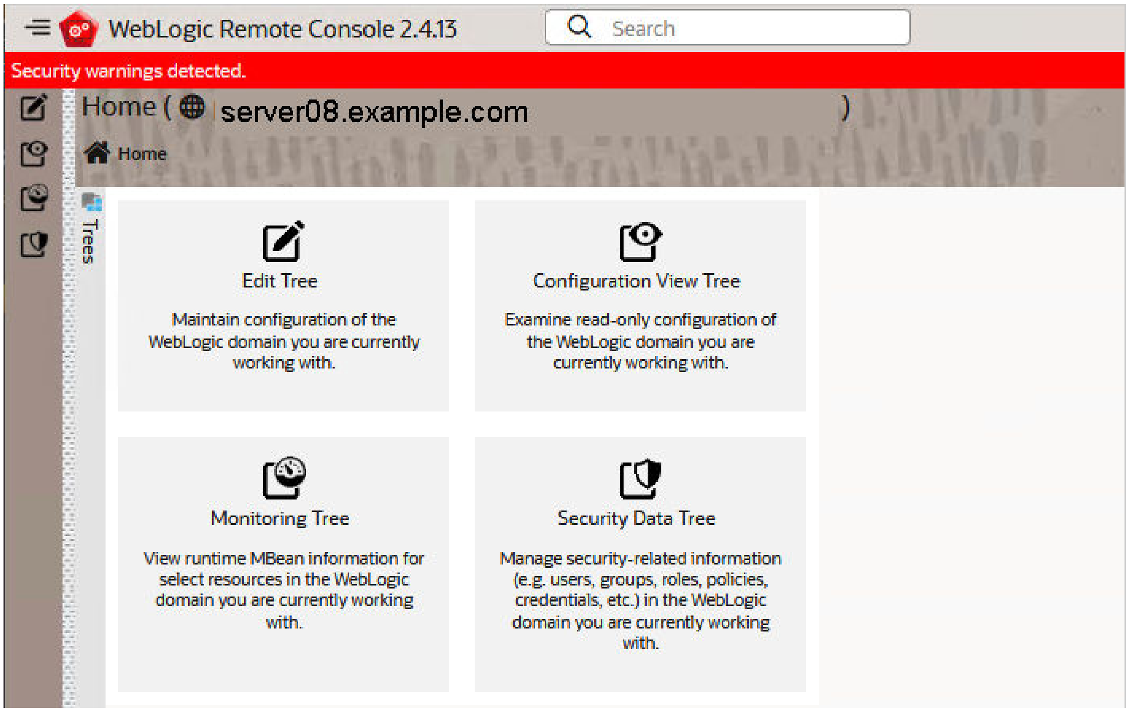
This example illustrates the Create Provider for Admin Server Connection window.



Reviewing the WebLogic Remote Console

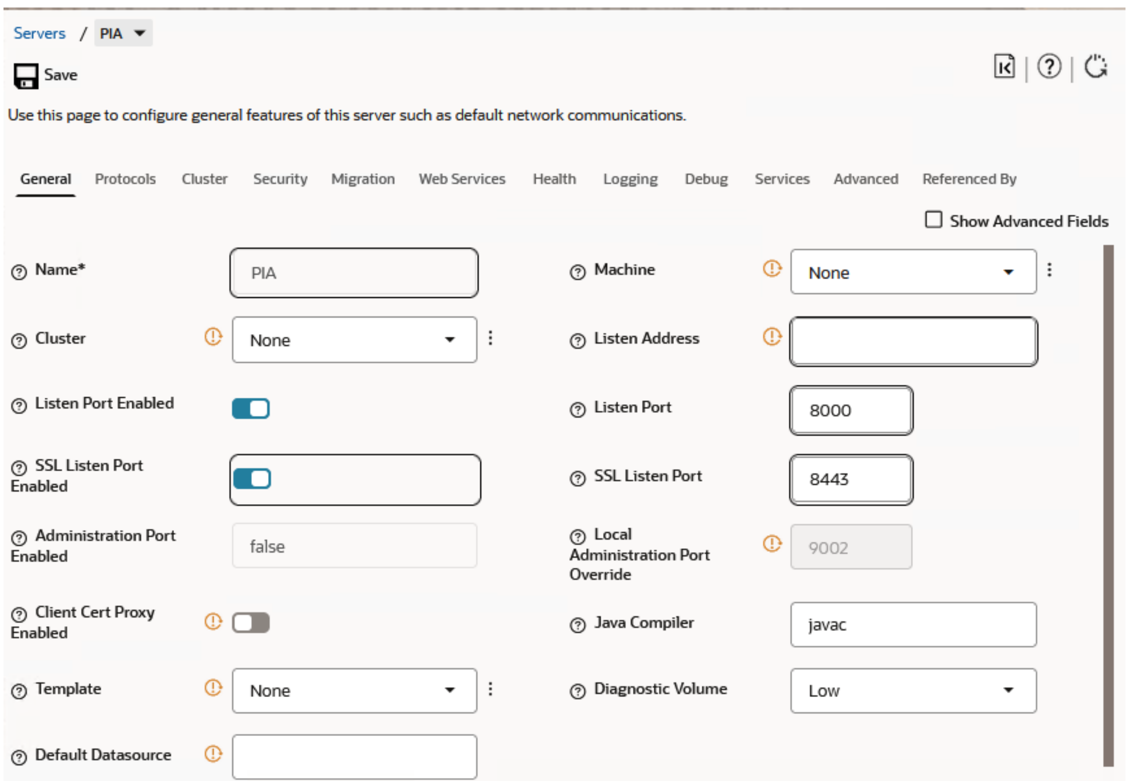
The home page for the WebLogic Remote Console includes four tiles –- Edit Tree, Configuration View Tree, Monitoring Tree, and Security Data Tree. You can access each area by clicking the tile or the icons in the left panel.

This example illustrates the WebLogic Remote Console home page.



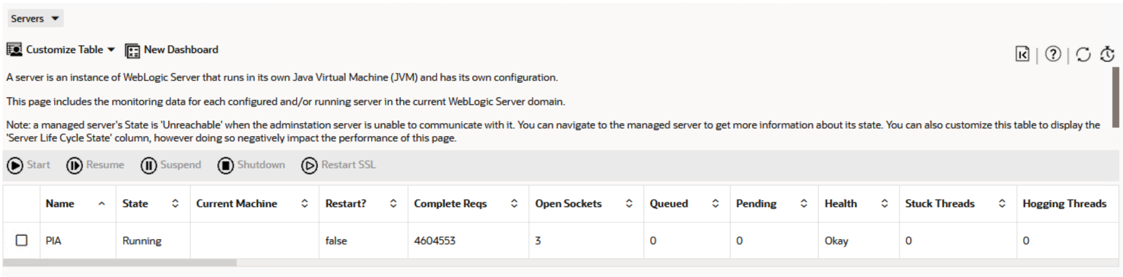
Select **Edit Tree**, and then **Environment**, and then **Servers**, and then **PIA** to view and modify the current configuration for PIA.

This example illustrates the PIA information accessed in WebLogic Remote Console.



Select **Monitoring Tree**, and then **Environment**, and then **Servers** to view the running status of the web server (PIA).

This example shows a portion of the monitoring page for PIA.There are several more columns on the page that are not shown.



Starting WebLogic

This section discusses how to start WebLogic in various operating systems.

Related Topics

- Administering a WebLogic Server Life Cycle
  This section provides an overview of the WebLogic server life cycle and discusses how to work with server processes.

Starting WebLogic Using PSADMIN

For single-server domains on any supported operating systems, use the PSADMIN Web (PIA) Server menus to start the PIA domain.

See Using the Web (PIA) Server Menu.

Starting WebLogic on Microsoft Windows

On Microsoft Windows, you can start WebLogic using:

- command prompt.
- Windows service.

Using the Command Prompt

Running WebLogic as a foreground process is beneficial if you need to monitor WebLogic in real time. To run WebLogic as a foreground process, enter one of the following commands at the command prompt in *PIA_HOME*\webserv*domain*\bin.

Configuration Type	Command
Single server	`startPIA.cmd`
Multi server - domain admin server	`startWebLogicAdmin.cmd`

Configuration Type	Command
Multi server - managed server	`startManagedWebLogic.cmd` *`server`* For example, `startManagedWebLogic.cmd PIA`

Using the Windows Service

Benefits of running WebLogic as a Windows service include:

- WebLogic can automatically start when the Windows server boots.
- You can start and stop the service from a remote Windows machine.

To install the service, open the command prompt, and enter the appropriate command from your *PIA_HOME*\webserv*domain*\bin directory:

Configuration Type	Command
Single server	`installNTservicePIA.cmd`
Multi server	`InstallNTservice.cmd` *`weblogic_server_instance_name`* For example: `installNTservice.cmd PIA`

To start WebLogic as a Windows service, you can:

- Start the service by using the Services utility in the Windows Control Panel.

  The DPK setup script creates a service with the default name PsftPIADomainpeoplesoftService.
- Start the service from a command prompt by entering the following command:

  ```
  NET START PsftPIADomainpeoplesoftService
  ```

> **Note**
>
> If WebLogic fails to start as a service, try starting it as a foreground process to make sure there are no general issues related to the startup.

To uninstall the service, enter the following command:

```
uninstallNTservicePIA.cmd
```

Starting WebLogic on UNIX

To start PeopleSoft on UNIX run the appropriate script in the WebLogic domain directory that the PeopleSoft install created, as in PIA_HOME/webserv/peoplesoft/bin.

Configuration Type	Command
Single server	`startPIA.sh`
Multi server - domain admin server	`startWebLogicAdmin.sh`
Multi server - managed server	`startManagedWebLogic.sh` *server* For example, `startManagedWebLogic.sh PIA`

When you run the above scripts, the server runs as background process.

Stopping WebLogic

This section discusses how to stop WebLogic.

Related Topics

- Administering a WebLogic Server Life Cycle
  This section provides an overview of the WebLogic server life cycle and discusses how to work with server processes.

Stopping WebLogic Using PSADMIN

For single-server domains on any supported operating systems, use the PSADMIN Web (PIA) Server menus to stop the PIA domain.

See Using the Web (PIA) Server Menu.

Stopping WebLogic Using the WebLogic Remote Console

To stop the web server (PIA):

1. Access the WebLogic Remote Console.

   See Using WebLogic Remote Console.
2. Select **Monitoring Tree**, and then **Environment**, and then **Servers**.
3. Select the row for PIA and click the Shutdown icon.

Stopping WebLogic Using the Command Line

To stop WebLogic using the command line, enter the appropriate command.

Configuration	Windows	UNIX
Single server	`stopPIA.cmd`	`stopPIA.sh`
Multi server (WebLogic Admin server)	`stopWebLogic.cmd`	`stopWebLogic.sh`
Multi server (Managed WebLogic server)	`stopWebLogic.cmd <ManagedServerName>`	`StopWebLogic.sh <ManagedServer Name>`

If WebLogic is running as a Windows service you can also stop the service in Windows Control Panel or using the NET STOP command.

Working With Reverse Proxy Servers For PeopleSoft Implementations

PeopleSoft applications support the use of reverse proxy servers (RPS) with WebLogic. When using an RPS, the RPS provides the URL to which the browsers connect, but another web server handles the actual transaction processing. Implementing the use of RPS servers provides an additional protective layer between the content server and the internet.

A reverse proxy is a server that appears to outside clients to be the content server. It relays requests from outside the firewall to servers behind the firewall, and delivers retrieved content back to the client. A firewall rule allows access only to the proxy server, so that the content servers are protected. The proxy server changes URLs listed in the headers of any messages generated by the content servers, so that external clients are given no information about the servers behind the firewall. No configuration of clients is necessary with a reverse proxy (the client makes requests for content in the name-space of the reverse proxy). The reverse proxy decides where to send the requests, and returns the content as if it was the origin server.

See Web Server Proxy Plug-ins for configuring supported reverse proxy servers.

Setting The HTTP Session Timeout

PeopleSoft Internet Architecture does not use session timeout controls set on the web server.

The HTTP session timeout controls are accessible on the Security page of the web profile interface. At runtime, the session timeouts set in the current web profile override any HTTP session timeouts set on the web server.

See Portal Technology: Configuring Portal Security.

Setting Authentication Failure Timeout

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

For example,

```
SET JAVA_OPTIONS_WIN32=-server -Xms256m -Xmx256m -Dpsft_failtimeout=60
-XX:MaxPermSize=128m -Xcomp
```

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

Enabling or Disabling HTTP Keep Alive

This section describes how to change HTTP Keep-Alive settings for a single server configuration of PIA. When in production, a multi server configuration would be used to perform these steps to your managed server instance of PIA, PIA1, PIA2, and so on.

Keep-Alive, or more accurately termed "Persistent Connections" is a default feature of HTTP. Keep-Alive allows for the client (generally a web browser) and the web server to maintain open connections between requests for specified period of time. That time period is generally less then 60 seconds. The benefit of a persistent connection is that the client and the server do not need to perform the overhead of opening a new connection with every single request.

Enabling Keep-Alive is generally recommended, but in some situations it may introduce problems. Sporadic "The Page cannot be displayed" errors, for example, can be the result of a problem with Keep-Alive. In situations where Keep-Alive issues are suspected, disabling the web server Keep-Alive and performing tests will help to determine if the problem is indeed related to connection persistence.

To enable or disable Keep-Alive:

1. Start the PIA server.

   See Starting WebLogic.

   See Stopping WebLogic.
2. Sign in to the WebLogic Remote Console.
3. Click the **Edit Tree** tile, and then select **Environment**, and then **Servers** from the left navigation panel.
4. Select the PIA server (or your custom named PeopleSoft server).
5. On the PIA page, select the Protocols tab, then the HTTP tab.
6. Change the Keep-Alive settings.
   - To disable Keep-Alive: Deselect **Enable Keepalives** and click **Save**. With Keep-Alive disabled, **Duration** and **HTTPS Duration** are not used.
   - To enable Keep-Alive: Select **Enable Keepalives** and update Duration and HTTPS Duration values as needed for your system.
7. Click **Save**.

   Click the Shopping Cart icon to check for pending changes and commit them if necessary.
8. Restart WebLogic Server.

Changing WebLogic User Passwords

The WebLogic domain built by the PIA install includes the following WebLogic IDs:

- system
- operator
- monitor

It is *highly* recommended to change the passwords for all of these user IDs on any production servers. You set the system password for the web server (PIA) domain during the DPK setup script installation.

To change the passwords:

1. Start the PIA server.

   See Starting WebLogic.

   See Stopping WebLogic.
2. Sign in to the WebLogic Remote Console.

3. Click the Security Data Tree tile, and then select **Realms**, and then **myrealm**, and then **Authentication Providers**, and then **Default Authenticator** from the left navigation panel.
4. Select **Users**, then select *system* from the list of users.
5. Select the Password tab, and enter a new password for the WebLogic server user.
6. Click **Save**.
7. With the *system* user ID, modify the boot.properties file with the new values:
   a. Open <*PIA_HOME*>\webserv\<*domainname*>\servers\PIA\security\boot.properties.
   b. Remove the previous, encrypted system value, and enter the new value in clear text.

      For example,

      ```
      ## Generated by Configuration Wizard on Mon Jun 25 18:46:08 MDT 2018
      password=password
      username=***Encrypted user name***
      ```
   c. Save the file.

      > **Note**
      >
      > Rebooting the server should encrypt the clear text value you entered.

8. After you have changed the operator password in the WebLogic Remote Console, you must modify wlop.properties file with the encrypted value of the new operator password.

   If you are on a Microsoft Windows operating system:
   a. Enter the following command:

      ```
      cd %DOMAIN_HOME%
      ```

      > **Note**
      >
      > *%DOMAIN_HOME%* is the PIA WebLogic domain root.

   b. Enter the following command to start the **wlst** shell:

      ```
      %WEBLOGIC_HOME%\wlserver\common\bin\wlst.cmd
      ```

      > **Note**
      >
      > *%WEBLOGIC_HOME%* is the root of WebLogic installation.

   c. Enter the following command after the **wlst** shell starts:

      ```
      print encrypt('password');
      ```

> **Note**
>
> In this example *password* is the new operator password that you have set in the previous steps.

d. The encrypted value is echoed in the shell. Copy this encrypted value.

e. Open the file wlop.properties from `<%DOMAIN_HOME%>\piaconfig\properties`.

f. Paste the new encrypted string (from step 'd') as the value of the key "password".

g. Save the modified file.

If you are on Linux operating system:

a. Enter the following command:

```
cd $DOMAIN_HOME
```

> **Note**
>
> *$DOMAIN_HOME* is the PIA WebLogic domain root.

b. Enter the following command to start the **wlst** shell :

```
$WEBLOGIC_HOME/wlserver/common/bin/wlst.sh
```

> **Note**
>
> *$WEBLOGIC_HOME* is the root of WebLogic installation.

c. Enter the following command after the **wlst** shell starts:

```
print encrypt('password');
```

> **Note**
>
> In this example *password* is the new operator password that you have set in the previous steps.

d. The encrypted value is echoed in the shell. Copy this encrypted value.

e. Open the file wlop.properties from `$DOMAIN_HOME/piaconfig/properties`.

f. Paste the new encrypted string (from step 'd') as the value of the key "password".

g. Save the modified file.

Implementing WebLogic SSL Keys and Certificates

This section provides an overview of Secure Sockets Layer (SSL) encryption with WebLogic.

Understanding SSL Encryption with WebLogic

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

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

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

When setting up SSL, you need to:

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

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

Obtaining Encryption Keys

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

The following procedure outlines how to obtain new encryption keys.

To obtain new encryption keys:

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

   PIA_HOME\webserv*domain_name*\piabin

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

   ```
   pskeymanager -create
   ```

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

3. Follow the prompts and enter the requested information to create a new private key and a CSR for your web server.
   - Pskeymanager uses the keystore in *PIA_HOME*\webserv*domain_name*\piaconfig\keystore\pskey.
   - Pskeymanager prompts you for an alias for the new keys, for example, *ServerABC*. This is the name you'll use to refer to the keys in the future.
   - Pskeymanager prompts you for a Subject Alternate Name (SAN). Enter one or more domain name server (DNS), IP address, EMAIL, URI, or an arbitrary object identifier (OID) in the format type:value.
   - Pskeymanager prompts you for distinguished name fields. Enter the appropriate values for your organization.
   - Pskeymanager prompts you for information about the CSR expiration date, key size, key algorithms, and the private key password. All of these fields have default values.
   - Pskeymanager makes a backup copy of an existing pskey file.

   Pskeymanager creates the private key inside the keystore, and creates the CSR as a file called ServerABC_certreq.txt in the current directory. You use the CSR to obtain your signed public key certificate and a root certificate from a CA.
4. Decide which trusted CA you want to sign your web server's public key, and follow their instructions to obtain the certificate.

   You can use any CA that's compatible with PKCS12 or Sun's Java JKS standard.
5. Follow the instructions in the product documentation to install the certificate.

   See Security Administration: Installing Web Server-Based Digital Certificates.

> **Note**
>
> Make backup copies of the server certificate and the root certificate before proceeding.

Preparing Keys and Certificates for the Keystore

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

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

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

Converting DER Files to PEM Format

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

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

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

   PIA_HOME\webserv*domain_name*

   Where *domain_name* is the name of an installed PeopleSoft Pure Internet Architecture domain.
2. Enter the following command:

   ```
   setenv.cmd
   ```

   This sets the appropriate environment for Java commands.
3. For each DER-formatted key or certificate file, enter the following command:

   ```
   java utils.der2pem filename.der
   ```

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

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

To modify the private key file header and footer:

1. Open the PEM-formatted private key file in a text editor.
2. Change the following line:

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

   To this:

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

3. Change the following line:

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

   To this:

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

4. Save and close the private key file.

Establishing the Server Certificate Chain of Trust

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

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

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

```
...
...
Dosdxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx==
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
DMICxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx
...
...
```

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

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

Password Protecting the Private Key

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

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

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

   WL_HOME\server\native\win\32

   Where *WL_HOME* is the root directory where you installed WebLogic.
2. Enter the following command:

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

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

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

The following message appears:

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

3. Press **Enter**.

   The following message appears:

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

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

   The following message appears:

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

5. Enter the password again to confirm it.

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

Importing Keys and Certificates Into the Keystore

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

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

Utility	When to Use
pskeymanager	If you created the private key using the pskeymanager utility on a WebLogic platform, it's already installed in the keystore. You need only use pskeymanager to import your server certificate, which should contain your web server's signed public key, your trusted CA's root certificate, and any public keys necessary to establish a chain of trust between them.
ImportPrivateKey	If the private key was previously configured as an external file on an earlier WebLogic platform, you must import it into the WebLogic keystore along with the server certificate, using the *ImportPrivateKey* utility. The private key should be password-protected.

Using pskeymanager to Import the Server Certificate

To import the server certificate into the WebLogic keystore:

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

   PIA_HOME\webserv*domain_name*\piabin

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

2. Enter the following command:

```
pskeymanager -import
```

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

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

   Keep the following in mind:

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

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

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

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

   PIA_HOME\webserv*domain_name*\bin

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

2. Enter the following command:

```
setEnv.cmd
```

   This sets the appropriate environment for Java commands.

3. Enter the following command:

```
java utils.ImportPrivateKey keystore\pskey
store_passprivatekey_aliasprivatekey_passservercert_fileprivatekey_file
```

   The parameters for this command are as follows:

Field or Control	Description
store_pass	Specify the password for the WebLogic pskey keystore.
privatekey_alias	Specify an alias for the private key. This is the name by which the key will be accessible inside the keystore.

Field or Control	Description
privatekey_pass	Specify the password for the private key.
servercert_file	Specify the path and name of the server certificate file that includes the issuing CA's root certificate.
privatekey_file	Specify the path and name of the private key file.

Configuring WebLogic SSL Encryption Keys

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

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

To configure WebLogic SSL encryption keys for the PIA server:

1. With the PIA server running, sign in to the WebLogic Remote Console.
2. Click the Edit Tree tile, and select **Environment**, and then **Servers**, and then **PIA**, and then **Security**, and then **Keystore** from the left navigation panel.
3. Select *Custom Identity and Custom Trust* from the Keystores drop-down list.
4. Update the fields on the Configure Keystore Properties page as follows:

Field	Value
Custom Identity Key Store File Name	*piaconfig/keystore/pskey* This should be the relative path and name of the keystore into which you imported your SSL keys.
Custom Identity Key Store Type	*JKS* or *PKCS12* These are the allowed values.
Custom Identity Key Store Passphrase	*password*
Custom Trust Key Store File Name	*piaconfig/keystore/pskey* This should be the relative path and name of the keystore into which you imported your SSL keys.
Custom Trust Key Store Type	*JKS* or *PKCS12* These are the allowed values.
Custom Trust Key Store Passphrase	*password*

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

Field	Vale
Server Private Key Alias	Specify a unique identifier, such as the web server's machine name. This should be the alias that you specified for this server's private key.

Field	Vale
Server Private Key Pass Phrase	Enter a password.

7. Click Save.

   Click the Shopping Cart icon to check for pending changes and commit them if necessary.
8. Restart the WebLogic PIA server.

Enabling TLS-Only on WebLogic

This section discusses the configurations for enabling TLS-Only on WebLogic.

Configuring TLS-Only on WebLogic Server

The weblogic.security.SSL.protocolVersion command-line argument lets you specify which protocol is used for SSL connections.

To enable TLS-only on WebLogic Server:

1. Open <*PIA_HOME*>/webserv/<*domain_name*>/bin/setEnv.cmd (Windows) or setEnv.sh (UNIX) in edit mode.
2. Append the following option to the JAVA_OPTIONS variable:

   ```
   -Dweblogic.security.SSL.protocolVersion=TLS1
   ```

   For example, on Windows in the setEnv.cmd, the updated JAVA_OPTIONS_WIN would appear as:

   ```
   SET JAVA_OPTIONS_WIN=-jrockit -XnoOpt -XXnoJITInline -Xms512m -Xmx512m
   -Dtoplink.xml.platform=oracle.toplink.platform.xml.jaxp.JAXPPlatform
   -Dweblogic.security.SSL.protocolVersion=TLS1
   ```

3. Save the change.
4. Restart PeopleSoft Internet Architecture.

For more information about Supporting TLS1 protocol, refer to the Oracle WebLogic documentation for specifying the version of SSL to be used.

See Specifying the SSL Protocol Version..

Configuring Reverse Proxy Servers for TLS

It is strongly recommended that you access the vendor's documentation of the web server you are using for a reverse proxy server and use their instructions for setting up TLS.

Working With WebLogic Session Cookies

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

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

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

To ensure valid WebLogic session cookie names:

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

   You can find it in <*PIA_HOME*>\webserv\<*domain_name*>\applications\peoplesoft\PORTAL\WEB-INF.
3. Check the value of the session parameter called CookieName.

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

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

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

   ```
   <session-param>
     <param-name>CookieName</param-name>
     <param-value>c57-28-208-21-80-WebLogicSession</param-value>
   </session-param>
   ```
4. Save and close the file.
5. Restart your WebLogic server.

Adjusting the JVM Heap Size

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

PIA_HOME\webserv\peoplesoft\bin

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

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

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

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

```
JAVA_OPTIONS_AIX="-Xms128m -Xmx256m -Dtoplink.xml.platform
=oracle.toplink.platform.xml.jaxp.JAXPPlatform
-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0"
JAVA_OPTIONS_LINUX="-jrockit -XnoOpt -Xms512m -Xmx512m
-Dtoplink.xml.platform=oracle.toplink.platform.xml.jaxp.JAXPPlatform
-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0"
JAVA_OPTIONS_SOLARIS="-Xms256m -Xmx256m -XX:MaxPermSize=256m
-Dtoplink.xml.platform=oracle.toplink.platform.xml.jaxp.JAXPPlatform
-Dcom.sun.xml.namespace.QName.useCompatibleSerialVersionUID=1.0"
```

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

In a multi-server domain, the platform-specific versions of the JAVA_OPTIONS environment variable that appear in the setEnv script apply only to managed servers. The administration server doesn't use any of these variables, but it assumes default JVM heap size values of `"-Xms256m -Xmx256m"`.

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

```
JAVA_OPTIONS_ADMINSERVER="-Xms128m -Xmx256m"
```

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

Related Topics

- Managing JVM Heap Size
  This section discusses how to:

Determining the Service Pack Level

A summary of installed products and their versions and service pack levels is maintained in the *BEA_HOME*\inventory\registry.xml file. (In default DPK installations, *BEA_HOME* is *BASE_DIR*/pt/bea.) However, to confirm version information, it's more accurate to check the WebLogic log. A failed service pack install may be indicated in the log, but not found at runtime.

This section discusses how to:

- Check the WebLogic log
- Query WebLogic

Checking Version Information in the WebLogic Log

The WebLogic log is located in:

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

For version information, look for an entry including the WebLogic Server version, similar to:

```
####<Feb 12, 2025, 1:52:17,668 PM Pacific Standard Time> <Info> <Management>
<server1> <> <Thread-10> <> <> <> <1739397137668> <[severity-value: 64] >
<BEA-141107> <Version: WebLogic Server 14.1.2.0.0  Tue Nov 26 02:40:45 GMT
2024 2171472>
```

Checking Version Information at the Command Line

To query a *running* WebLogic for version information from the command line, follow these steps:

1. Go to the directory that contains weblogic.jar (*WebLogic_HOME*/wlserver/server/lib).

   > **Note**
   >
   > If you performed the PeopleSoft installation using the default DPK initialization process, weblogic.jar will be found in *BASE_DIR*/pt/bea/wlserver/server/lib.

   For example, on Microsoft Windows, open a command prompt and enter this command:

   ```
   cd C:\psft\pt\bea\wlserver\server\lib
   ```

   For example, on UNIX, open a terminal window and enter this command:

   ```
   cd /opt/oracle/psft/pt/bea/wlserver/server/lib
   ```

2. Run the following command:

   ```
   java -cp weblogic.jar weblogic.version
   ```

   You see the version:

   ```
   WebLogic Server 14.1.2.0.0  Tue Nov 26 02:40:45 GMT 2024 2171472
   ```

3. If you want to get subsystem information, run the following command:

   ```
   java -cp weblogic.jar weblogic.version -verbose
   ```

4. If you want to get version information for all modules, run the following command:

   ```
   java -cp weblogic.jar weblogic.utils.Versions
   ```

Enabling HTTP Access Log

This section describes how to change HTTP logging for a single server configuration of PIA. When in production, a multi server configuration would be used to perform these steps to your managed server instance of PIA or PIA1, PIA2, and so on.

To enable or disable HTTP access log:

1. Make sure the PIA server is running.

   See Starting WebLogic.

   See Stopping WebLogic.
2. Sign in to the WebLogic Remote Console.
3. Click the Edit Tree tile, and then select **Environment**, and then **Servers**.
4. Select **PIA** or your customer server name.
5. On the PIA page, select **Logging** and then **HTTP**.
6. Select the **HTTP access log file enabled** option to turn on the PIA_access.log.
7. Modify the **Log file name** field if desired.
8. Click **Save**.

   Click the Shopping Cart icon to check for pending changes and commit them if necessary.
9. Restart the WebLogic Server.

Configuring WebLogic to Modify HTTP Logging

The PIA_access.log includes a record of all HTTP transactions. Use this configuration to include SRID, Top Instance, and OPRID in the log. These values can be useful in troubleshooting and auditing to help to correlate messages from different components.

See Working with PeopleSoft Server Process Logs for information about SRID, Top Instance, and OPRID in logs.

To enable this enhanced logging you must configure it in the WebLogic Remote Console.

1. Sign in to the WebLogic Remote Console and select **Edit Tree**, then **Environments**, then **Servers**, then **PIA**, then **Logging**, and then **HTTP**.
2. Select Show Advanced Fields.
3. Select extended from the **Format** drop-down list.
4. Add the following text in the Extended Logging Format Fields box. x-psft.pt8.psftCorrelationFields

   For example, x-psft.pt8.psftCorrelationFields cs-method cs-uri sc-status
5. Restart the server.

8

Using PeopleSoft Configuration Manager

Understanding PeopleSoft Configuration Manager

PeopleSoft Configuration Manager simplifies Windows workstation administration by enabling you to adjust PeopleSoft registry settings from one central location. It contains a variety of controls that let you set up Windows workstations running development and administrative applications, such as Application Designer and Data Mover.

You can set up one workstation to reflect the environment at your site, and then export the configuration file, which can be shared among all the workstations at your site. You can also define separate profiles for connecting to different PeopleSoft databases. PeopleSoft configuration parameters are grouped on the Configuration Manager tabs according to the function, feature, or tool that they control.

> **Note**
>
> The changes you make within PeopleSoft Configuration Manager do not take effect until the next time a user signs on to PeopleSoft.

> **Note**
>
> When referencing install locations in your environment using Configuration Manager, keep in mind that some files may be located in PS_APP_HOME or PS_CUST_HOME instead of PS_HOME.

> **Note**
>
> PeopleSoft supports a number of versions of UNIX and Linux in addition to Microsoft Windows. Throughout this discussion, we make reference to operating system configuration requirements. Where necessary, this topic refers to specific operating systems by name (Solaris, Linux, and so forth). However, for simplicity the word UNIX refers to all UNIX-like operating systems, including Linux.

Related Topics

- Setting Up the PeopleTools Development Environment
  This section provides overviews of the PeopleTools development environment and the client setup process.

Common Elements in PeopleSoft Configuration Manager

Field or Control	Description
OK	Saves your settings and exits PeopleSoft Configuration Manager.
Cancel	Closes PeopleSoft Configuration Manager without saving any changes that you have made.
Apply	Saves your changes without exiting.

Starting PeopleSoft Configuration Manager

This section discusses how to launch Configuration Manager and manage its settings.

Launching Configuration Manager

You can start PeopleSoft Configuration Manager by one of two methods:

- Double-click the Configuration Manager shortcut in your PeopleSoft program group.
- At a command prompt, enter:

  PS_HOME\bin\client\winx86\pscfg.exe

> **Important**
>
> Due to changes in the registry structure beginning with PeopleTools 8.50, if you intend to run PeopleTools 8.50 and later versions of Configuration Manager and previous versions of Configuration Manager (PeopleTools 8.49, 8.48, and so on), on the same development workstation, pre-PeopleTools 8.50 values stored in the registry will be deleted.

> **Important**
>
> Certain PeopleSoft utilities require setting an environment variable, PS_SERVER_CFG, to run properly. However, PeopleSoft Configuration Manager isn't compatible with PS_SERVER_CFG being set. Before you start Configuration Manager, you must ensure that PS_SERVER_CFG is not set. You can make this convenient by using a DOS batch file to unset PS_SERVER_CFG, launch Configuration Manager, then after you exit Configuration Manager, reset PS_SERVER_CFG to its previous value.

Setting Configuration Managers for Multiple PeopleTools Clients

If you want to access multiple PeopleTools releases from a single client workstation, then you may need to run Configuration Manager and re-enter the Connect ID and Connect Password. Doing this creates a Windows registry key in the format required by the specified range of releases.

In addition, if you use a different Connect ID and Connect Password for the different releases, you must run Configuration Manager for each release and re-enter the credentials.

Whether or not this is required depends on the releases that you want to access with PeopleTools client software (Application Designer, Data Mover, and so on) running from the same client workstation. If you need to run PeopleTools clients from more than one of the releases or range of releases in the following list, launch Configuration Manager, then enter (or re-enter if necessary) and save the Connect ID and Connect Password.

- 8.54
- 8.55 — 8.56
- 8.57

For example, if you want to run clients from PeopleTools 8.55.14 and 8.57.04:

1. Run PeopleTools 8.55.14 Configuration Manager.

   For example, pshome_8.55.14/bin/client/winx86/pscfg.exe.
2. On the Startup page, enter values for Connect ID and Connect Password.
3. Click OK to save.
4. If the Connect ID and Connect Password for PeopleTools 8.57.04 are different from the values for PeopleTools 8.55.14, run PeopleTools 8.57.04 Configuration Manager.

   For example, pshome_8.57.04/bin/client/winx86/pscfg.exe.
5. On the Startup page, enter values for Connect ID and Connect Password.
6. Click OK to save.

Managing PeopleTools 8.4x and PeopleTools 8.5x Configuration Manager Settings

PeopleTools 8.4x and PeopleTools 8.5x releases manage registry settings differently. This is due, in part, so that PeopleTools complies with security policies introduced with Microsoft Windows Vista and Microsoft Windows 7.

PeopleTools 8.4x Configuration Manager uses both HKEY_CURRENT_USER and HKEY_LOCAL_MACHINE to store various settings. PeopleTools 8.5x Configuration Manager stores all settings under HKEY_CURRENT_USER. For example, in PeopleTools 8.4x, profile settings are stored in HKEY_LOCAL_MACHINE, whereas, in PeopleTools 8.5x profile settings are stored in HKEY_CURENT_USER.

Due to these changes in the registry structure, the first run of the PeopleTools 8.5x Configuration Manager will delete any PeopleTools 8.4x configuration settings stored in the registry in the HKEY_CURRENT_USER hive. PeopleTools 8.4x and PeopleTools 8.5x registry settings can coexist, but the PeopleTools 8.4x settings must be preserved prior to the first run of the PeopleTools 8.5x Configuration Manager so that they can be restored afterwards.

To preserve PeopleTools 8.4x registry settings:

1. Use the Registry Editor to export and save the contents of HKEY_CURRENT_USER\Software\PeopleSoft\PeopleTools\Release8.40.
2. Run PeopleTools 8.5x Configuration Manager, and modify settings as needed.
3. Import the .reg file containing the PeopleTools 8.4x exported registry contents.

   After completing these steps, PeopleTools 8.4x and PeopleTools 8.5x will then use the same registry values with the exception of the Profiles registry settings. PeopleTools 8.4x versions will use the Profiles settings under HKEY_LOCAL_MACHINE. PeopleTools 8.5x versions will use the Profiles settings under HKEY_CURRENT_USER.

> **Note**
>
> If the PeopleTools 8.4x settings are not saved or they are deleted when the PeopleTools 8.5x Configuration Manager runs, you can recreate the PeopleTools 8.4x settings by running the PeopleTools 8.4x Configuration Manager and reentering the required settings.

Specifying Startup Settings

Select the Startup tab.

Use the Startup tab to customize the default values that appear on the PeopleSoft Signon dialog box, for example when you start Application Designer.

Signon Defaults

Field or Control	Description
Database Type	Select the database type to appear as a default on the PeopleSoft Signon dialog box. Select *Application Server* to sign in to an application server instead of a database. To enable users to change their database type selection in the PeopleSoft Signon dialog box, you must select the **Database Type** option in the **User Can Override** group. **Note:** When you select *Application Server* from the **Database Type** drop-down list, the **Server Name** and **Database Name** fields are disabled. The system obtains these values from the application server.
Application Server Name	If you selected *Application Server* from the **Database Type** drop-down list, specify the application server's name in this field. You must have already configured your application server and registered it on the Profile tab.
Server Name	Enter the name of the default database server. This parameter is only enabled for Microsoft SQL Server, and refers to the instance to which the user connects.
Database Name	Enter a default database name. You can choose any valid PeopleSoft database name. As with the database type, you must select the appropriate option in the **User Can Override** group to enable users to override the default database name selection when they sign in.

Field or Control	Description
User ID	Specify the default user ID to sign in to PeopleSoft. You can use the user ID in conjunction with a PSUSER module containing a user-defined sign-in process. The PSUSER code, if present, can evaluate and modify the user ID value before you attempt to sign in to the selected database.

Field or Control	Description
Connect ID and **Connect Password**	PeopleSoft uses the connect ID for the initial connection to the database. Use the **Connect Password** field to define a default connect ID password. **Note:** The connect ID edit box must contain a value, or the user can't sign in to the system in a two-tier environment.

See Security Administration: PeopleSoft Authorization IDs.

See *PeopleSoft 9.2 Application Installation* for your database platform.

Numeric Keypad - Enter Key Tabs to Next Field

In Microsoft Windows applications, pressing the **Enter** key in a dialog box selects the default action button. For example, in the PeopleSoft Signon dialog box, pressing **Enter** is the same as clicking the **OK** button. Selecting the **Numeric keypad** check box overrides this default behavior for the **Enter** key on the numeric keypad; instead of selecting the action button, pressing the **Enter** key moves the cursor to the next field in the dialog box.

> **Note**
>
> This check box affects the **Enter** key on the numeric keypad, but not the **Enter** key on the main keyboard.

User Can Override

Some PeopleSoft sites use multiple database types and names. Using the check boxes in the **User Can Override** group box, you can enable users to enter a database type, database name, or user ID other than the default provided at sign-in. In most cases, you use these controls to prevent users from attempting to sign in onto any database other than the default.

Field or Control	Description
Database Type	When selected, users can choose a database other than the default. Selecting this check box selects the **Database Name** and **User ID** options automatically. You cannot clear **Database Name** or **User ID** without first clearing **Database Type**. When configuring a workstation to connect in both two-tier and three-tier, you must select this box. The user needs to specify a two-tier or three-tier connection from the PeopleSoft Signon dialog box.
Database Name	When selected, the **User ID** check box is automatically selected, although you can clear it. To clear **Database Name,** you must clear the **Database Type** check box.
User ID	Select to enable users to users override only the user ID submitted at when they sign in. You cannot clear **User ID** if **Database Type** is selected.

Cache Files Directory

Enter a parent directory to hold your client cache file directories. The directory you specify must be writable for a user with Standard User privileges. For example, %USERPROFILE%\psft\pt\client\cache.

Cache files store database object information locally and are automatically downloaded the first time you open a PeopleSoft database object. They are also downloaded automatically if the master copy of the object on the database server has changed. For each PeopleSoft database to which you connect, the system creates a single child cache files directory to store the cache files for that database.

Clicking **Purge Cache Directories** brings up a dialog box with your existing cache file directories.

You can select a single directory and click **Delete,** or you can click **Delete All** to remove all directories. If a cache file directory is missing (after you delete it), the system automatically rebuilds it the next time that cache files are downloaded. After you delete the appropriate cache directory, click **Close** to return to the Startup tab.

Specifying Display Settings

Select the Display tab.

Use the Display tab to configure the appearance of the PeopleSoft development environment graphical user interface. For instance, you can adjust page width and height to fit in with the other elements on your desktop.

> **Note**
>
> These settings affect the interface display for PeopleTools Windows applications used for development and administration, such as Application Designer and Data Mover. These settings have no effect on PeopleSoft applications displayed to end users on browsers.

Language

In the **Language** drop-down list box, specify which language you want to display. The default setting is US English.

> **Note**
>
> You select from the languages that PeopleSoft delivers. Although you can implement applications to appear in other languages, you cannot switch to custom languages using PeopleSoft Configuration Manager. Switch to these languages by manually changing the registry setting.

Page Display

You can adjust page display size or the window height and width.

Field or Control	Description
Display Size, Width, and **Height**	Specify display size in pixels. This setting affects the default size of the PeopleSoft window as displayed in the corresponding **Width** and **Height** fields. Select from: • *640 X 480:* The default window size is 640 pixels by 448 pixels. • *800 X 600:* The default window size is 800 pixels by 576 pixels. • *1024 X 768:* The default window size is 1024 pixels by 744 pixels. • *Custom:* You can manually set the default window size by specifying width and height values. **Note:** Changing these parameters does not affect open windows. If either value is either blank or zero, the values are reset to 640 by 480 pixel resolution.

Field or Control	Description
Page Sizing	Use this field to specify how pages that were designed for a different-size window should be displayed. Select from: • *CLIP:* Interface controls are always displayed in their normal size. If a window is too large for the allotted size, the window information is clipped along the right and bottom edges of the window. Use scroll bars to view the remainder of the interface. • *SCALE:* Windows are scaled to fit the allotted space as necessary. For example, if your display size is set to 640 by 480 pixels, and you open a window designed to display in an 800 by 600 pixel window, the page controls are scaled down so that all page information appears. Conversely, if you open a page designed for 640 by 480 pixel resolution in a larger window, the interface controls are scaled to fill the window completely.
Show Page in Navigator	Select to see the navigator tree view and the page view at the same time.
Highlight Popup Menu Fields	Select to highlight fields with associated pop-up menus. The box is clear by default. In most cases, it's a good idea to indicate which fields contain pop-up menus. Pop-up menus are indicated by a black rectangle surrounding the perimeter of a page control.
Show Database Name	Select to display the name of current database in the status bar at the bottom of the window. **Note:** The database name may be abbreviated to fit on the screen. **Note:** This setting controls the display of the database name in the bottom right of the window. It does not affect the display of the database name in the Application Designer title bar.

Font

Use the Font options to configure the way that text appears on the screens.

Click the **Font** button to bring up a standard font selection pop-up menu.

Business Process Display

Select from:

- *On:* The navigator appears with each menu group that you open.
- *Off:* You must open the navigator manually.
- *First:* The navigator appears on the first instance of PeopleSoft only. Subsequent instances do not display the navigator.

Specifying Business Interlink and JDeveloper Settings

Select the Bus.Interlink/JDeveloper tab.

Business Interlink Driver Options

In the **Business Interlink Directory** box, enter the complete path to the directory that contains the drivers that PeopleSoft Business Interlinks uses to communicate with external systems.

> **Note**
>
> PeopleSoft Business Interlink is a desupported product. These options currently exist for upgrade compatibility and transition.

JDeveloper Home Directory

Oracle JDeveloper is used when developing transformations using Oracle XSL Mapper. You need to specify the high-level installation directory as well as the appropriate Classpath string. These settings are discussed in detail in the Integration Broker documentation.

Related Topics

- Integration Broker: Developing Transforms Using Oracle XSL Mapper

Specifying Trace Settings

Select the Trace tab.

Use the Trace tab to select tracing options for various parts of the PeopleTools system, such as SQL statements, PeopleCode, and Application Engine. If you work on tuning your PeopleSoft system and improving online performance, familiarize yourself with this tab. When you update the Trace tab, the new settings take effect the next time you launch PeopleTools.

> **Note**
>
> The Trace tab in PeopleSoft Configuration Manager traces only Microsoft Windows client (two-tier) interactions. Use these settings only when you require tracing on the client.

You can override some of the trace options on this tab from the Trace SQL and Trace PeopleCode pages in PIA.

See Configuring SQL Trace, Configuring PeopleCode Trace.

SQL Informational Trace

Select this check box to trace information messages from the Runstats command in DB2 UDB for z/OS executed as a result of an %UpdateStats meta-SQL command.

PeopleTools Trace File

The default filename for the PeopleTools trace file is DBG1.TMP. The system writes the file to the following directories:

- In Microsoft Windows: %TEMP% directory
- In UNIX: $*PS_HOME*/log/dbname

> **Important**
>
> The PeopleTools trace file stores elapsed times for PeopleCode and SQL events to a precision of one microsecond (six decimal places). However, due to limitations of the operating system, Windows precision is actually in milliseconds (three decimal places), so the last three digits in a Windows trace will always be zero. Elapsed times in UNIX are accurate to one microsecond.

To specify a different PeopleTools trace file:

1. Click the button on the right side of the **PeopleTools Trace File** edit box.

   A standard Open dialog box appears.
2. Navigate to and select the new trace file.
3. Click **Open.**

   The **PeopleTools Trace File** field displays the path and filename.

Related Topics

- Application Engine: Setting Client Options in PeopleSoft Configuration Manager

Specifying Workflow Settings

Select the Workflow tab.

Use the Workflow tab to specify the options and locations related to the PeopleSoft Workflow implementation at your site.

Field or Control	Description
Maximum Worklist Instances	Enter a number to limit the number of worklist instances or entries that appear when viewing worklists. The default value is *250.* If you do not want any rows returned, leave the field blank.
SMTP Server	Specify the SMTP settings for email routings.

Related Topics

- Workflow Technology

Specifying Remote Call/AE Settings

Select the Remote Call/AE tab.

Some PeopleSoft applications use the Tuxedo Remote Call feature, which invokes data-intensive transactions, such as COBOL processes, on a remote server.

Field or Control	Description
Timeout	Enter the amount of time after which Remote Call terminates the child COBOL process. The default is *50* seconds.
Redirect Output	Select to specify whether the standard out or standard error of the child COBOL process is directed to a file. This check box is clear by default.
Support COBOL Animation	Select to save the COBOL input file so that you can reuse it with COBOL animator. This check box is clear by default.
Normal, Minimized, and **Hidden**	Specify how the window state of the child COBOL process appears on the desktop. • Select **Normal** to have the window state appear like a DOS window on the desktop. • Select **Minimized** to have the window state appear as an icon on the task bar. • Select **Hidden** to have the window state run unseen in the background.
Disable DB Stats (disable database statistics)	Select to turn off the %UpdateStats meta SQL construct. This setting applies to Application Engine programs. See Application Engine: Meta-SQL.

Configuring Developer Workstations

Select the Client Setup tab.

As part of the PeopleSoft installation process, you need to configure developer workstations (also called the PeopleTools development environment) to run successfully with your PeopleSoft system. You use developer workstations for development and administrative tasks that require the use of a Windows workstation. Such tasks include, updating record definitions with Application Designer and running scripts using Data Mover. Development workstations can access the PeopleSoft system using both two-tier and three-tier connections.

Use the Client Setup tab to configure developer workstations and invoke the Client Setup process. Although this tab is specifically for developer settings, all of the PeopleSoft Configuration Manager settings may affect developers, especially the Startup tab and the Process Scheduler tab for the default profile.

Shortcut Links

Here are the various shortcut links:

Field or Control	Description
Application Designer	Adds a shortcut for the main PeopleTools development environment.
Configuration Manager	Adds a shortcut for PeopleSoft Configuration Manager, which enables you to edit registry settings relevant to PeopleSoft.

Field or Control	Description
Data Mover	Adds a shortcut to launch PeopleSoft Data Mover.
Uninstall Workstation	Adds a shortcut for Uninstall Workstation, which uninstalls the most recent client setup.
PeopleTools RPT Converter	Adds a shortcut to a standalone program that converts RPT files from the format PeopleSoft used in previous releases to the currently supported reporting format. You only need to run this program if you are upgrading from previous versions of PeopleTools.
nVISION	Adds a menu item for PS/nVision to the PeopleSoft 8 menu group in the Microsoft Windows Start menu. **Note:** Back up RPT files before you run the converter program, which significantly alters them.

Install Workstation

Select the **Install Workstation** check box to run the Client Setup process. Only select the check box after specifying all the appropriate selections on all PeopleSoft Configuration Manager tabs. If you do not select this box, the Client Setup process will not run.

After you select this check box, click either **OK** or **Apply.**

Enable Dirty-Read for 2Tier Query

(Applies only to DB2 systems.) Select this option if you need to run "dirty read" queries, through PeopleSoft Query during a two-tier connection.

See Query.

Related Topics

- Setting Up the PeopleTools Development Environment
  This section provides overviews of the PeopleTools development environment and the client setup process.

Importing and Exporting Environment Settings

Select the **Import/Export** tab.

Use this tab to export, or save to file, the specified environment settings, and to import previously exported settings. This feature is useful when you plan to configure multiple workstations with similar settings.

Field or Control	Description
Export to a File	Click to write current configuration settings to a file. A **Save** dialog box appears. Note the file name that you give the configuration file. **Note:** Click **Apply** before you export a file. This ensures that the exported configuration file reflects the current settings.

Field or Control	Description
Import from a File	Click to import previously saved configurations on another workstation. Importing a configuration file overrides all the current environment settings on the machine that you import to. When you click this button, an **Open** dialog box appears. Navigate to the directory containing the appropriate configuration file, select the file, and click **Open.** **Warning!** In addition to overwriting environment settings, this function also overwrites all existing settings made in Application Designer.

Configuring User Profiles

This section discusses how to configure user profiles

Note

The term "user profiles" is used here to refer to user configurations for a workstation. These profiles are not to be confused with PeopleSoft security user profiles.

Defining a Profile

Select the Profile tab.

Use this tab to define one or more user profiles, each of which specifies connection parameters and file location information for a PeopleSoft installation.

Many PeopleSoft installations include multiple databases. For example, there may be one database for tracking financial information, such as expense reports, and another database for human resources processes, such as benefits enrollment. Each of these databases has its own set of supporting files, SQR reports, COBOL processes, and so on. PeopleTools locates these files by referring to the Microsoft Windows registry. By defining multiple profiles, you can tell PeopleTools to use different directory paths depending on the database.

When you first open PeopleSoft Configuration Manager, the Profile tab displays a single profile named Default. To set the parameters for this profile, make sure that it's selected, and click the **Edit** button. The Edit Profile dialog box appears.

Each workstation must have a default profile, which is used when the user signs in to a database or application server that isn't listed in any profile. If the workstation requires only one set of profile settings, you can use the default profile. You can also set up multiple PeopleSoft Configuration Manager profiles. The profiles are set for Microsoft Windows workstations and are specific to each user and not shared by all workstation users.

> **Note**
>
> You can use profiles to easily switch between applications.

Specifying Databases and Application Servers

From the Edit Profile dialog box, select the Database/Application Servers tab.

Use this tab to specify the configured databases and application servers associated with this profile. When a user enters one of these databases or application servers in the **PeopleSoft Signon** dialog box, PeopleTools uses the registry settings associated with this profile.

> **Note**
>
> You can assign multiple databases and application servers to a single profile. However, each database and application server must be assigned to only one profile. If you try to add a database to a second profile, PeopleSoft Configuration Manager asks you if you want to remove it from the previous profile and add it to the current one.

> **Note**
>
> Before you enter a database or application server on this tab, you should have already installed and configured it as documented in the PeopleSoft installation documentation for your database platform.

Application Server Name

Enter a name for an application server that you have configured. This name will appear in the drop-down list box on the PeopleSoft Signon dialog box. Choose a name that's intuitive for your site.

> **Note**
>
> Application server names cannot exceed 24 characters.

Machine Name or IP Address

Enter the IP address or the resolvable server name of the application server in the **Application Server Name** field. This is listed in the [Workstation Listener] section of the PSAPPSRV.CFG file. For example, you could enter *192.0.2.20* or *srv-hp32.*

See Workstation Listener Options

Port Number

Enter the port number for the application server in the **Application Server Name** field. A port number is an arbitrary number between 0 and 9999 that is determined by site specifications. This is listed in the [Workstation Listener] section of the PSAPPSRV.CFG file.

See Workstation Listener Options

> **Note**
>
> To configure SSL for Workstation Listener, you must enter the port number specified for the WSL SSL Port property in PSADMIN.
>
> See Modifications in PSADMIN for more information.

TUXEDO Connect String

Use this field to support dynamic load balancing. You can specify a free-form connect string that allows a client to connect to another application server in case another is either down or being used to full capacity.

> **Note**
>
> The Tuxedo connect string cannot exceed 1000 characters.

When configuring load balancing, you might choose from the following approaches:

- Round robin load balancing.

  With this approach, you specify multiple application servers, and each client picks a server randomly. This approach assumes that application server will receive an equal number of connections. To specify round robin load balancing, use the following syntax for the connect string:

  (//*IP address 1*:*port 1*|//*IP address 2*:*port 2*|//*Ip address n*:*port n*)

  You can specify the IP address using either dotted notation or by using the server's DNS name. Either way, the slashes (//) preceding the IP address are required.

  If the selected application server is unavailable, your connection attempt fails, and the system does not try to connect you to the other application servers defined within the parentheses.

  Spaces are not allowed in any part of the connection string. The system automatically removes embedded spaces before storing the value in the registry.

- Round robin with failover.

  With this approach, you define a failover connection string. Use the following syntax:

  (//*IP address 1*:*port 1*|//*IP address 2*:*port 2*),(//*IP address 3*:*port 3*)

  If the application server selected from the first group of parentheses (IP addresses 1 and 2) is unavailable, the system automatically attempts to connect to an application server defined in the second group (IP address 3). If that application server fails, the system attempts to connect to the next group to the right, sequentially.

If multiple application servers are defined within any group, the system round-robins between them. If the selected application server fails, the system attempts to connect to the next application server to the right, if any. The following list shows three examples of connect strings that use this approach:

– (//sp-ibm01:8000|//sp-ibm02:8000),(//sp-nt01:8000)
– (//192.0.2.88:8000|//192.0.2.88:8050|//192.0.2.88:8080)
– (//sp-sun01:8000),(//sp-sun02:8000),(//sp-sun03:8000)

Domain Connection Password

Specify for the password for establishing a connection to the domain. The domain configuration contains the DomainConnectionPwd parameter, which provides an additional layer of security for the domain by preventing connections from any unauthorized clients. Three-tier workstation connections must submit the same password as the specified DomainConnectionPwd value.

For a three-tier Windows workstation connection, you enter the password in the Domain Connection Password field.

Wallet Location

Specify the location of the Oracle Wallet for client.

See Modifications in Configuration Manager for more information.

Wallet Name

Specify the name of the Oracle Wallet for client.

See Modifications in Configuration Manager for more information.

Set and Delete Buttons

When you click **Set,** your application server information is displayed in the grid at the top of the dialog box. You can enter a new application server name and set up a different server if you like.

> **Note**
>
> The settings in the grid are not saved until you click **Apply** or **OK.** If you click **Cancel** without first clicking **Apply** or **OK,** you lose all the information in the grid.

To remove an application server configuration, select its application server name in the grid and click **Delete.**

Configuring Process Scheduler

Access the Process Scheduler tab.

Use this tab to specify the directories that are associated with PeopleSoft Process Scheduler jobs, such as SQR and COBOL directories.

General

Field or Control	Description
PeopleSoft Home Directory	Enter your high-level PeopleSoft directory, such as *N:\HR920*, or *N:\PT857*. This value is referred to as your PS_HOME.
PeopleSoft Apps Home Directory	Specify the location where you installed the PeopleSoft application software, such as *N:\HR920*. This value is referred to as PS_APP_HOME. **Note:** The PS_APP_HOME location must be different from the PS_HOME location for the current release.
PeopleSoft Customizations Home Directory	If you have created custom files or modified delivered files, you can store them in PS_CUST_HOME to simplify system resource sharing as well as upgrades and updates. See Working with PS_CUST_HOME for more information.
Output Directory	(Optional) Enter the directory used with the **Output Destination** field when scheduling a PeopleSoft Process Scheduler request.
Log Directory	Enter the directory for SQR, COBOL, and PeopleSoft Process Scheduler log files.
Temp Directory	Enter the path to your temporary directory, for example, *C:\TEMP.* This directory stores log files and other output files.
Database Drivers	Enter the path to the directory where your database drivers reside. **Note:** The PeopleTools Development Environment running on the Windows workstation (Application Designer, Data Mover, and so on) is a 64–bit application, requiring 64–bit connectivity software. See *PeopleSoft 9.2 Application Installation* for your database platform, for more information.
Word Executables Directory	Enter the directory containing Microsoft Word executable.
Redirect Output	Select to redirect on-screen COBOL Display statements to a log file. (If this check box is clear, you see the on-screen messages only.) Sending the messages to a log file is useful for debugging purposes. The log file is created in the %TEMP%*PS_HOME*\DBNAME directory. In addition to the output generated by COBOL Display statements, the log file contains errors generated by the COBOL runtime system. **Note:** To use the Application Engine debug feature, clear **Redirect Output.**

Application Engine

Field or Control	Description
Debug	Select to enable the Application Engine command-line debugger. **Warning!** Select the **Debug** check box only when you are testing and troubleshooting client-side processes. If you select**Debug** and submit a process request to the server, the process hangs, waiting for a user command.
Disable Restart	Select to disable the Application Engine restart feature, which lets you restart an abnormally terminated Application Engine program. When selected, Application Engine programs start from the beginning. This option is useful during debugging. Do not select it in a production environment.

SQR

Field or Control	Description
SQR Executables	Enter the path to the directory where SQR executables reside.
SQR Flags	Enter the SQR parameters that PeopleSoft Process Scheduler should pass on the command line to the SQR executables. The following SQR flags are required for launching SQR reports: • -i: Specifies the path to SQC files. • -m: Specifies the path to the ALLMAXES.MAX file. • -f: Specifies the output path. • -o: Directs log messages to the specified file. • -ZIF: Sets full path to the and name of the SQR initialization file, SQR.INI.
PSSQR Path	Enter the directory paths that the SQR executable should use to locate SQR reports. For example, *C:\PT8.57\SQR;C:\Apps\sqr*.

COBOL

Field or Control	Description
COBOL Executables	Enter the path to \CBLBIN, where COBOL executables reside. **Note:** When specifying COBOL locations, consider the location of your PS_APP_HOME.

Configuring nVision

Access the nVision tab.

Use this tab to specify where PS/nVision should look for files and how it should operate. PeopleSoft Query Link, the feature that enables you to send PeopleSoft Query output to a spreadsheet, also uses these settings.

Space between Query Columns

This parameter sets the number of blank Microsoft Excel characters that PeopleSoft Query Link places between query output columns. To eliminate column spacing, set **Space between Query Columns** to zero.

Directory Paths

Specify the locations of directories associated with PS/nVision jobs.

> **Note**
>
> When specifying nVision directory locations, consider the location of your PS_APP_HOME.

Field or Control	Description
Customization Macros	Enter the directory containing macros for PS/nVision and PeopleSoft Query Link. It is usually *PS_HOME*\excel.
Report Layouts	Enter the location of PS/nVision layout fields.
Drilldown Layouts	Enter the location of PS/nVision drilldown files, for example, *c:\user\nvision\layout\drilldn.*
Report Instance	Enter the directory in which PS/nVision places report instances; for example, *c:\user\nvision\instance.*
Query Templates	Enter the directory to look for the QUERY.XLT file, which defines the Microsoft Excel styles used to format output. The default is *PS_HOME*\excel.
Style Sheets	Enter the directory where the NVSUSER style wizard locates nPlosion style sheets.

<table>
<tr><th>Field or Control</th><th>Description</th></tr>
<tr><td>Trace Level</td><td>Indicate whether you want PS/nVision to generate independent trace log files of two-tier activity, and at what level, for each nVision process. Select one of the following values:

0: Disable tracing. This is the default value.
1: Generate basic high level information.
This setting can be used to check whether nVision has successfully launched and is able to connect to Excel and process the request. Some of the key entries in a level 1 trace log are:

Command Line Arguments.
Trace Level.
Excel Pid.
Run Control Name.
Report Id.
Business Unit.
Drill Layout.
Report Id.
Instance Name.

2: Generate level 1 tracing plus high level code flow.
3: Generate level 2 tracing plus runtime SQL statements.
4: Generate level 3 tracing plus most function calls and output values.
Use this setting to identify problems that are intermittent and hard to predict.

The trace log files are generated in the c:\temp directory, named with the format psnvs_process_id.nvt, for example, psnvs_1024.nvt. You can view these log files in a text editor.

See Process Scheduler: Viewing Process Detail Actions.

Note: Extensive tracing will affect PS/nVision performance. Two-tier log files aren't automatically purged by PS/nVision. Users must manually delete them from the temp directory to save disk space.</td></tr>
</table>

Related Topics

- PS/nVision

Specifying Common Settings

Select the Common tab.

Application Designer Image Conversion

When you upgrade to newer version of PeopleTools, you'll need to convert images to a new format, which may require more storage space. If the images exceed the record size limit of your platform, you can shrink the images to conform to this limit.

Field or Control	Description
Convert and Shrink Images to Platform Limit	Select to convert and shrink images to fit your selected database platform limit, as shown in the **Image Size Limit field.**
Convert and Shrink Images to Image Size Limit	If you are upgrading to a different database platform, select this option and specify the correct value in the **Image Size Limit** field.
Don't Convert, but Shrink Images to Image Size Limit	Select for images that have already been converted, but need to be converted so they meet the platform size limits.

Data Mover Directories

You can control several PeopleSoft Data Mover environment variables through PeopleSoft Configuration Manager.

> **Note**
>
> When specifying Data Mover directory locations, consider the location of your PS_APP_HOME.

Field or Control	Description
Input Directory	Enter the directory where PeopleSoft Data Mover should search for its input data files. If no path is specified for SET INPUT in the Data Mover script, Data Mover searches directories for the database file in the following order. 1. Specified output directory. 2. C:\TEMP.
Output Directory	Enter the directory where PeopleSoft Data Mover scripts and .DAT files will be created. Data Mover must have write access to the location. The default is *PS_CFG_HOME*\data.
Log Directory	Enter the location of PeopleSoft Data Mover log files. This location must allow Data Mover write access in the case of a read-only PS_HOME configuration. The default is C:\Documents and Settings\admin\Local Settings\Temp.

Related Topics

- Lifecycle Management Guide: Understanding PeopleSoft Data Mover

Specifying Command Line Options

In addition to its GUI interface, PeopleSoft Configuration Manager offers command line options. Syntax for PeopleSoft Configuration Manager command line options is as follows:

`pscfg -command`

For example:

```
pscfg -import:n:\config\hr900.cfg
```

Import File

To import configuration settings from a named file, enter `-import: filename`.

Export File

To export the current configuration settings, enter `–export: filename`.

Run Client Setup

To run the Client Setup process, enter `-setup`.

> **Note**
>
> You must use the `-setup` command in conjunction with the -import command if you are setting up a new workstation.

Run Client Setup Without Displaying Messages

To run the Client Setup process without displaying messages or dialog boxes, enter `-quiet`.

> **Note**
>
> Output messages are written to a log file called %temp%\PSINSTAL.LOG.

Install MSS DSN

To install MSS DSN, enter `-dsn`.

> **Note**
>
> For Microsoft SQL Server, the server name value is used to automatically create your ODBC data source name.

Uninstall Workstation

To clear the PeopleSoft settings from the registry or uninstall the PeopleSoft workstation, enter `-clean`.

The `-clean` command removes the following items from the workstation:

- PeopleSoft registry settings.
- All cache files from the current \CACHE directory.
- Shortcut links.
- PeopleSoft program group.

Make sure that removing all of these items is acceptable before issuing the -clean command.

Help

To view PeopleSoft Configuration Manager command-line options online, enter `-help` or a question mark (?).

Setting Up the PeopleTools Development Environment

This section provides overviews of the PeopleTools development environment and the client setup process.

Understanding the PeopleTools Development Environment

Most user workstations are equipped with supported web browsers, but with no special PeopleSoft software installed. The traditional Microsoft Windows client is supported for application developer and administrator use. The PeopleTools development environment runs on a supported version of Windows.

This topic describes how to configure these Windows-based clients using PeopleSoft Configuration Manager. Such clients can connect to the PeopleSoft database directly using client connectivity software (a two-tier connection), or through a PeopleSoft application server (a three-tier connection).

Understanding the Client Setup Process

Before running the Client Setup process, create all the profiles you need.

The Client Setup process installs a PeopleSoft program group on the workstation.

If the **Install Workstation** check box on the Client Setup tab is selected, these Client Setup functions are performed when you click **OK** or **Apply** from PeopleSoft Configuration Manager.

See Configuring Developer Workstations.

> **Note**
>
> Any files installed by the Client Setup process on the workstation from the file server use the paths specified in the default profile.

Verifying PS_HOME Access

To use the PeopleTools development environment, each workstation must have access to the file server *PS_HOME* directory (the high-level directory where PeopleSoft executables were installed) and have a drive mapped to the directory. Workstation users must have read access to the *PS_HOME* directory.

Verifying Connectivity

Database connectivity is required on all Microsoft Windows-based clients that make two-tier connections to the database. A two-tier connection is required if any of the following is true:

- You sign in to the Application Designer in two-tier.

- You run PeopleSoft Data Mover scripts.
- You run COBOL and SQR batch processes on the client.

> **Note**
>
> PeopleSoft Windows-based development tools, such as Application Designer, are 64–bit applications, and require 64–bit connectivity software.

Verify Supporting Applications

Supporting applications must be installed on any Microsoft Windows-based client on which batch processes are run locally.

SQR

On Microsoft Windows-based clients, you can install SQR locally, or you can map to a copy installed on the file server. Because SQR does not require local registry settings, you can execute SQR from any Windows-based client once SQR has been installed to a shared directory. Installing SQR locally results in improved performance; over a slow network connection, the improvement is significant.

Microsoft Office

Install Microsoft Office on any two-tier client that runs PS/nVision or Microsoft Word batch processes. Microsoft Office must be installed locally, because it requires registry settings.

Using the Configuration Manager

PeopleSoft Configuration Manager enables you to manage registry settings on a Windows workstation that control a variety of interface options, such as sign-in defaults, display settings, and environment profiles.

Related Topics

- Understanding PeopleSoft Configuration Manager
  PeopleSoft Configuration Manager simplifies Windows workstation administration by enabling you to adjust PeopleSoft registry settings from one central location. It contains a variety of controls that let you set up Windows workstations running development and administrative applications, such as Application Designer and Data Mover.

Running the Client Setup Process

To run the Client Setup process:

1. In Configuration Manager, select the Client Setup tab.
2. In the **Group Title** text box, enter the name of the program group for the icons you want on the client workstation.
3. Select check boxes to create shortcut links for PeopleSoft applications that you want to access from the workstation.

   When you run the Client Setup process, it removes existing shortcuts in the PeopleSoft 8.x program group and installs shortcuts for the applications that you have selected. If you later want to install or uninstall shortcuts, you can always run Client Setup again.

4. Select the **Install Workstation** check box.

   Client Setup runs when you click **Apply** or **OK** in PeopleSoft Configuration Manager. If this check box is not selected, the Client Setup process creates or updates settings in the registry, but it doesn't set up the PeopleSoft 8.x program group or install local DLLs.

5. Click **Apply** to run the Client Setup process and apply other PeopleSoft Configuration Manager settings.

6. To view a list of the files installed and actions taken by the Client Setup process, open the psinstal.log file in your Temp directory.

Related Topics

- Configuring Developer Workstations
  Select the Client Setup tab.

9

Using PeopleTools Utilities

Understanding the PeopleTools Utilities

As you work with the PeopleSoft system, you find that there are some administrative tasks that you only need to perform occasionally. These tasks include such things as maintaining error messages and setting DDL model defaults. The PeopleTools Utilities menu is where you find tools for accomplishing some of these more infrequent tasks.

The documentation of the utilities matches the menu structure of the Utilities interface. For example, the PeopleTools Options utility is under the Administration menu in the Utilities interface; therefore, the documentation for PeopleTools Options is in the Using Administration Utilities section in this topic. Also, in many cases this book refers to other PeopleBooks for the detailed documentation of a utility.

Using Administration Utilities

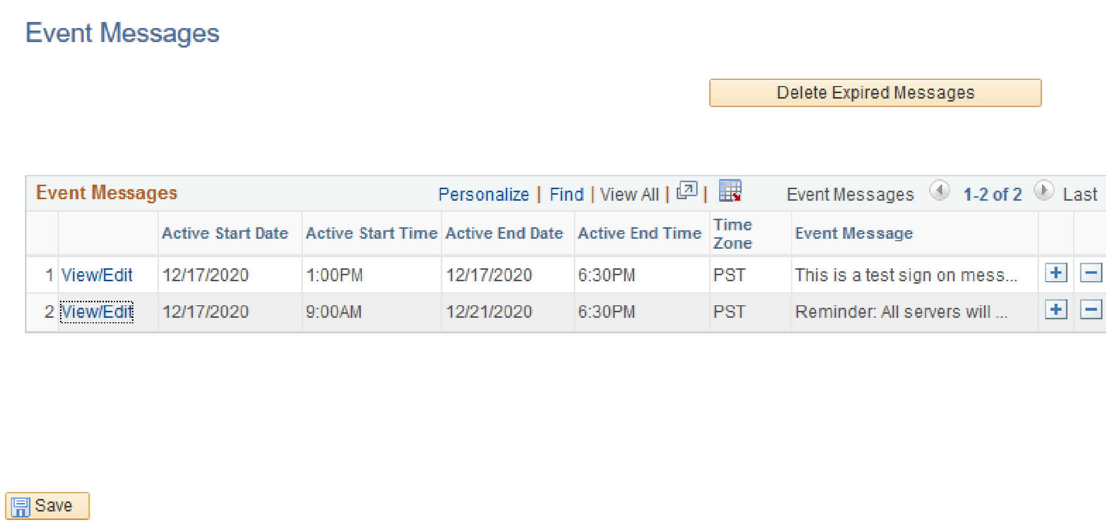
This section provides an overview of the executables for using administration utilities.

Signon Event Messages

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Signon Event Messages** to access the Event Messages page.

Use this page to define messages to display to end users when they sign in to the PeopleSoft system. As long as the signon event message is active, each time a user signs in to PeopleSoft, the event message will appear after the user clicks Sign In on the PeopleSoft system sign-in page.

This example illustrates the fields and controls on the Event Messages page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
View/Edit	Click to view or modify a sign-in message.
Delete Expired Messages	Click to delete messages that have an End Time and End Date value earlier than current time. **Note:**To delete individual messages at any time, use the Delete row (-) button.

> **Note**
>
> Only users with the role of PeopleSoft System Administrator will have access to the pages used to view and define signon event messages.

Adding Signon Event Messages

To add a new signon event message:

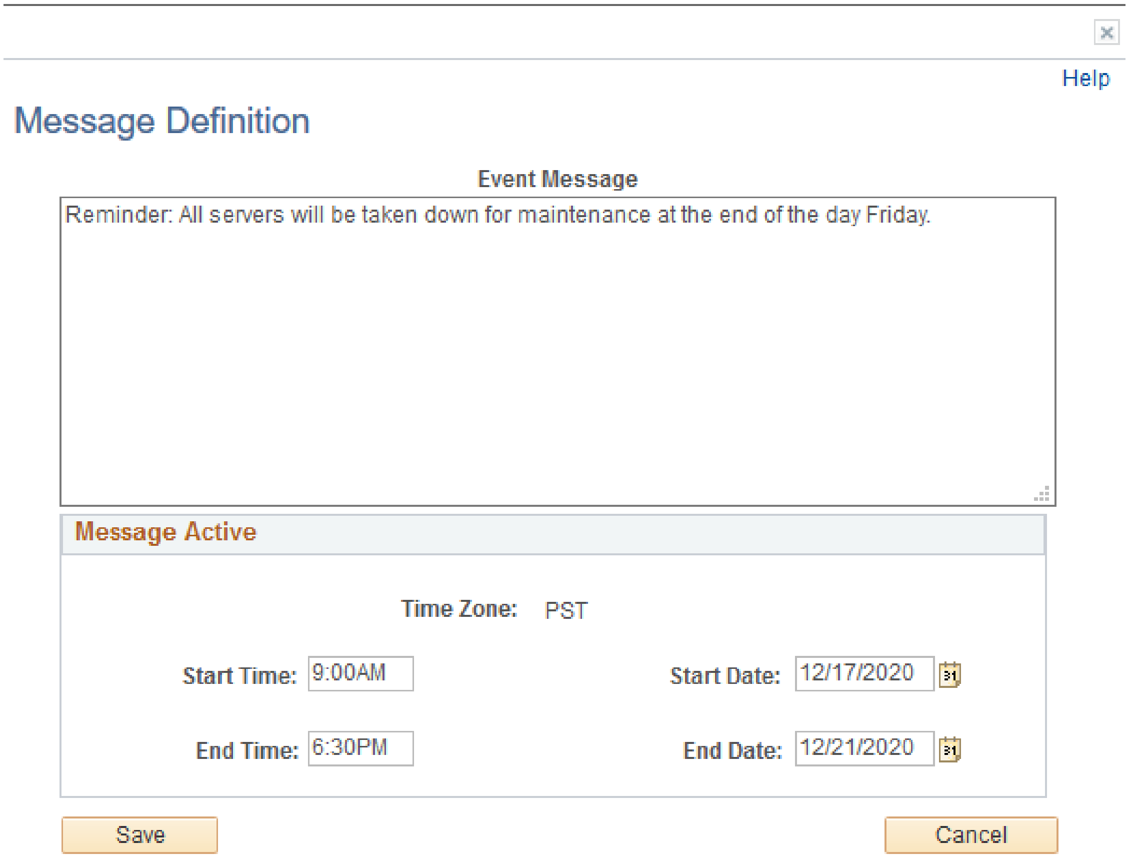
1. Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Signon Event Messages**.
2. Add a new row to the Event Messages grid.

Defining Signon Event Messages

Use the Message Definition dialog box to define signon event messages.

Click View/Edit in the Event Messages grid to display the Message Definition page.

This example illustrates the fields and controls on the Message Definition page. You can find definitions for the fields and controls later on this page.



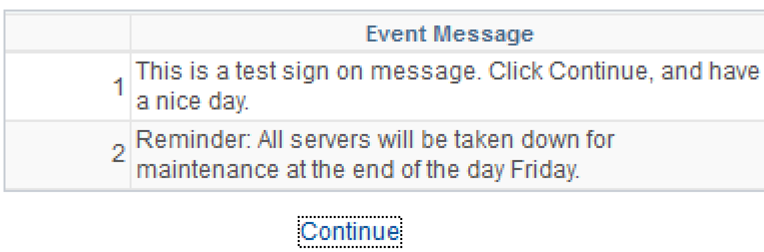
Field or Control	Description
Event Message	Compose the signon message in this edit box. The system will display to the user all text entered into this edit box. When creating messages, consider: • Message size limit is 32 KB. • Only *plain* text is supported (not HTML, XML, Rich Text, or text including links, bold, underline, and so on). • Translation of the message is the responsibility of the system administrator creating the message. The system does not translate the message depending on the language of the user. For example, in Quebec, Canada, you might consider displaying an English and French version of the message. • The message will be seen by all users signing on to the database through a browser, regardless of role, department, time zone, and so on. As such, care should be taken to ensure the message has broad relevance. **Note:**The message will not be displayed to Windows Development Environment users.

Field or Control	Description
Message Active	Specify the duration for which the message will be active by setting a start time and date and an end time and date. The system only displays messages where the current time is after the start time and date and before the end time and date. **Note:**The Time Zone is read-only, and displays the time zone in which the database runs. It does not necessarily reflect the time zone that would be applicable to the text in the message. If there are any specifics related to the time zone that users need to be aware of, that should be communicated in the event message text.

Working With Signon Event Messages

The Event Messages grid can contain multiple signon event message definitions. However, only those messages that are currently active will be displayed to the user.

This example illustrates the fields and controls on the Event Message display page. You can find definitions for the fields and controls later on this page.



Multiple signon event messages are displayed to the user by start time and date, with the earliest appearing first.

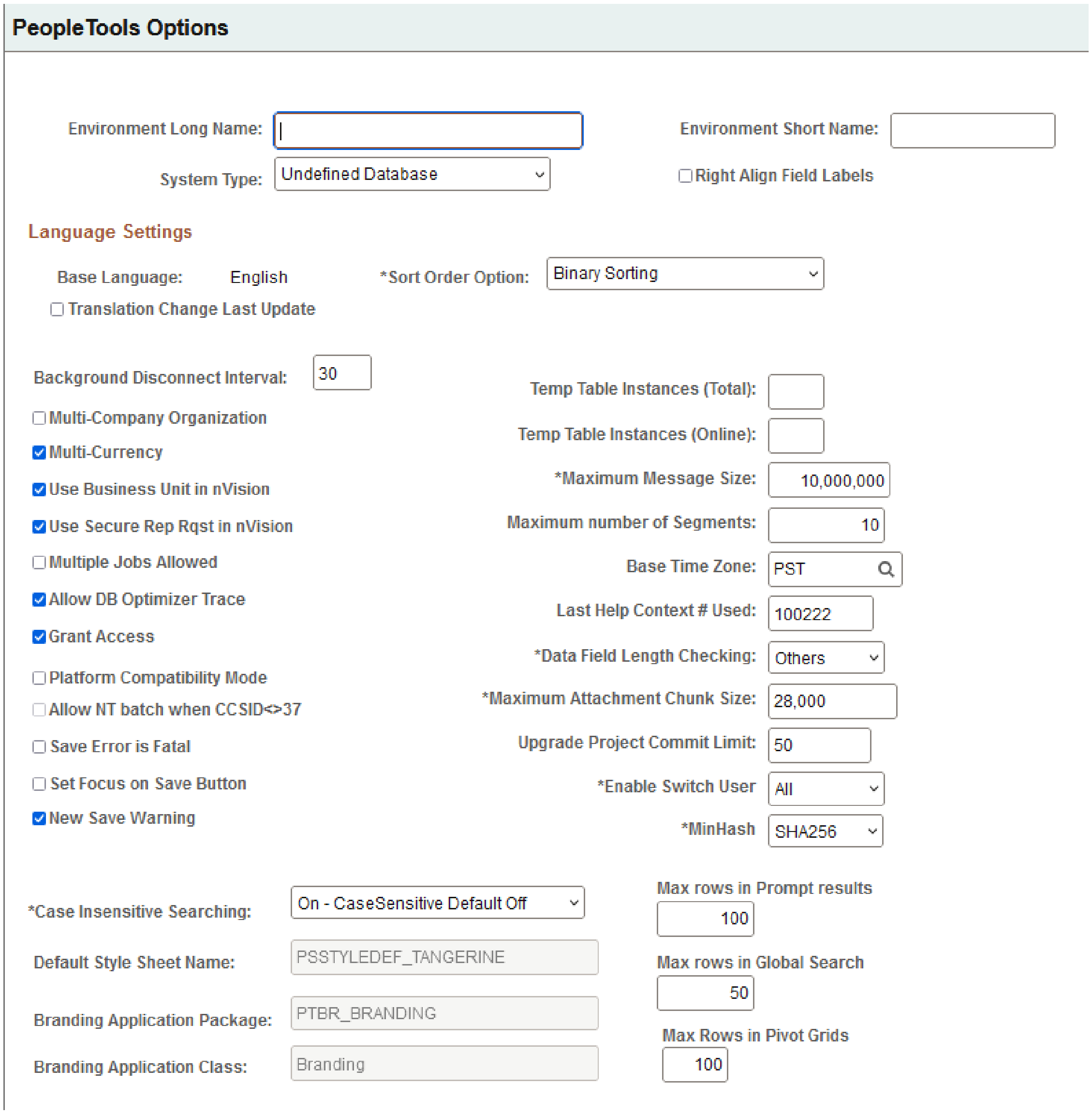
After reading the message, the user clicks Continue, and accesses the application as usual.

> **Important**
>
> The event message display uses the component PT_EVENTDISPLAY. If you intend to use signon event messages, all users of your system need to be able to access the PT_EVENTDISPLAY component. By default, the Permission List PTPT1000 ships providing access to PT_EVENTDISPLAY. If you use custom permission lists, make sure that all users have access to the PT_EVENTDISPLAY component through their permission lists.

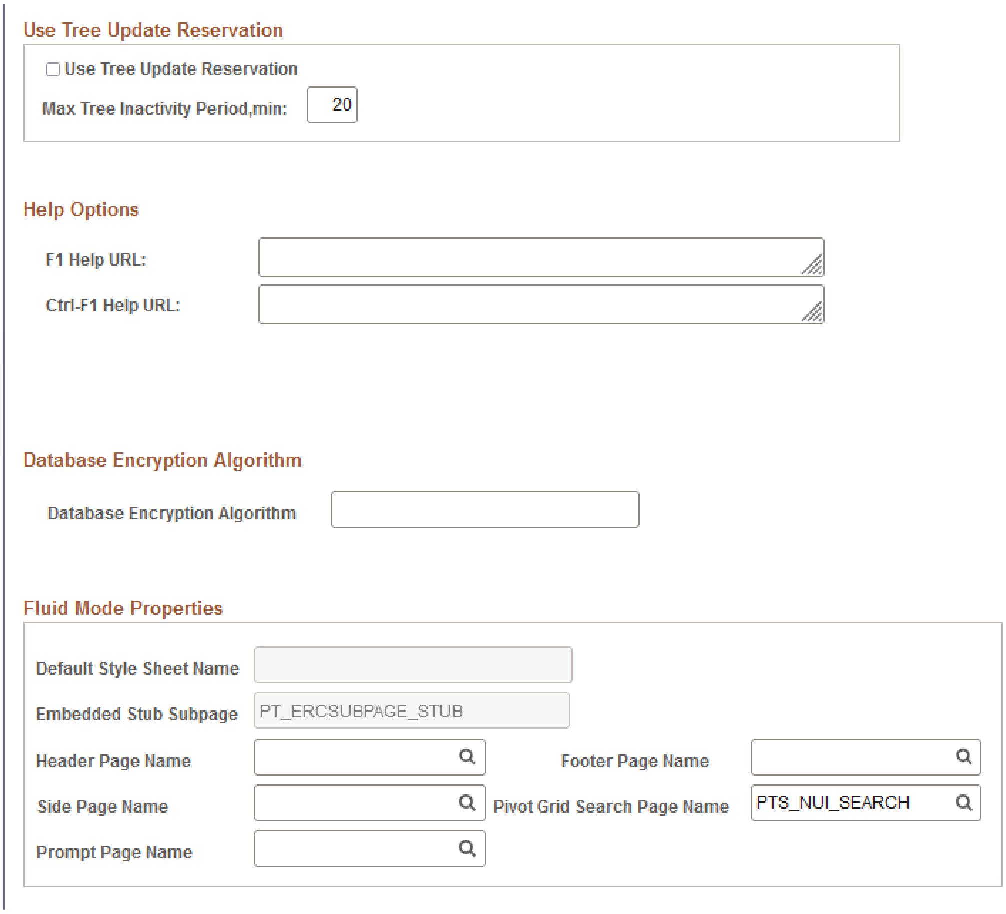
PeopleTools Options

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **PeopleTools Options** to access the PeopleTools Options page. Use this page to set a number of options that affect multiple PeopleTools and applications, such as language options and change control settings.

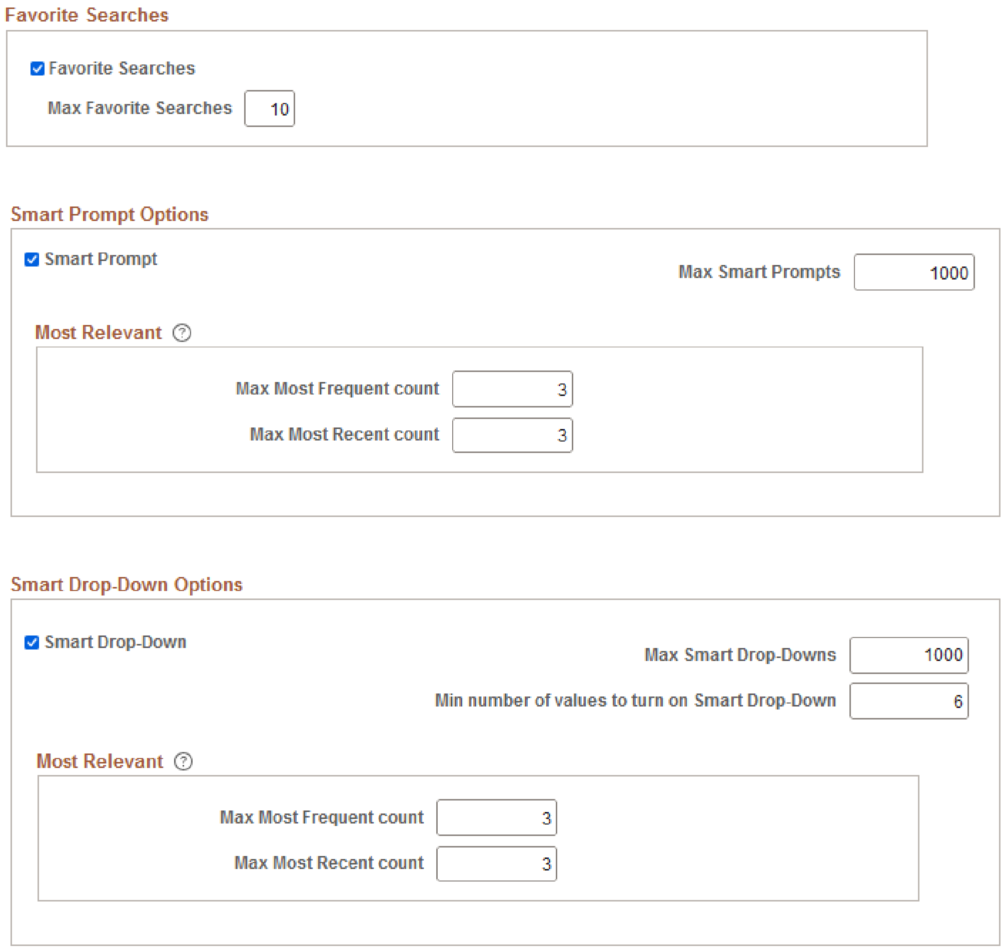
This example illustrates the fields and controls on the PeopleTools Options page (1 of 3). You can find definitions for the fields and controls later on this page.



This example illustrates the fields and controls on the PeopleTools Options page (2 of 3). You can find definitions for the fields and controls later on this page.



This example illustrates the fields and controls on the PeopleTools Options page (3 of 3). You can find definitions for the fields and controls later on this page.



Field or Control	Description
Environment Long Name and **Environment Short Name**	Enter a long name and a short name for the current PeopleSoft environment. PeopleSoft software update tools use this information to identify the database when searching for updates. For example, enter *HR Demo Environment* for the long name, and *HR Demo DB* for the short name.
System Type	Select an appropriate system type from the drop-down list, for example, *Demo Database*. This information helps to further identify the current environment for the purpose of searching for and applying software updates.
Right Align Field Labels	By default, the system uses left label alignment when adding new fields to pages in Application Designer. To make all new page field labels use right label alignment (at design time), set the **Right Align Field Labels** check box on the PeopleTools Options page. See Application Designer Developer's Guide: Setting Page Field Properties for Controls.

Language Settings

Field or Control	Description
Language Code	Identifies the database's base language. The base language of an application is the application's primary language, normally the language that is used most commonly throughout the enterprise. A database can have only one base language. All other language translations that are stored in the database are referred to as non-base languages (or sometimes as foreign languages). You can't change the base language setting on this page. This field is for display purposes only. To change the base language, use the SWAP_BASE_LANGUAGE Data Mover command.
Translations Change Last Update	If you select the **Translations Change Last Update** check box, and you use the PeopleTools translate utilities to translate objects, the system updates the Last Updated information of the translated object to the date/time/userid of the translation. If it's turned off, then the date/time/userid of the object does not change when it's translated. **Note:**This only applies when you're using the page-based PeopleTools translation utilities; the Translation Workbench always updates the last updated information.
Sort Order Option	Select the sort order that is appropriate for the site. See Global Technology: Understanding Sort Orders

General Options

Field or Control	Description
Background Disconnect Interval	This option is no longer in use.
Multi-Company Organization	Turn on Multi-Company Organization if more than one company makes up the organization. This option affects how Application Processor displays company-related fields in search dialogs and pages. See the PeopleSoft Human Capital Management product documentation for more details.

Field or Control	Description
Multi-Currency	The Multi-Currency setting is a system-wide switch that enables automatic formatting of currency amount fields that have associated currency control fields. Another function of this setting is to globally display currency control fields. If you turn off this option, automatic formatting based on currency control fields is no longer active and all currency control fields are thus hidden. When the Multi-Currency setting is on, it also validates user-entered currency data against the currency's defined decimal precision. This validation causes the system to issue an error if a user attempts to enter a decimal precision that is greater than that which is allowed by the currency code definition. Under most circumstances, leave Multi-Currency selected.
Use Business Unit in nVision	Deselect the Use Business Unit in nVision option if you're using an HCM database. Otherwise, select it.
Use Secure Rep Rqst in nVision	Select this check box if you want the report request in nVision to be secure. The default setting is selected.
Multiple Jobs Allowed	Selecting Multiple Jobs Allowed enables HCM systems to support employees holding concurrent jobs with more than one set of enrollments. This option affects how Application Processor displays employee-record-number-related fields in search dialogs and pages. See the HCM documentation for more details. You must select this option in order to display the fields EMPL_RCD and BENEFIT_RCD_NBR on a page.
Allow DB Optimizer Trace	Typically, you turn on this trace only during periods in which you are collecting detailed performance metrics. When you are not tuning your performance, the DB Optimizer trace should be turned off.
Grant Access	When adding a new user by using PeopleTools Security, the system automatically grants the new user select-level access to a set of PeopleTools SQL tables used for security. If you are using a SQL security package and do not want PeopleTools Security to perform any SQL grants, turn off Grant Access.

Field or Control	Description
Platform Compatibility Mode	Enables you to add the capability to set a database compatibility mode as an overall database setting, forcing developers to create applications by using all platforms as the least common denominator. This option enables developers, who create applications for multi-platform deployment, to catch platform-specific issues at design time rather than during testing. **Note:**This option is used mainly by PeopleSoft development teams that need to develop applications to run on all supported database platforms. To support numerous database platforms, PeopleSoft needs to have a tablespace for each physical table record definition. If platform compatibility is enabled for a database, the system forces developers to enter a tablespace name when saving a record definition regardless of the current platform. If this option is disabled, you are only prompted for a tablespace name if you are developing on a platform that utilizes tablespaces. This prevents table record definitions being added to the database without a tablespace name.
Allow NT batch when CCSID <>37	Enables you to override non-z/OS COBOL batch restrictions. If the DB2 z/OS database's CCSID is NOT 37, PeopleSoft blocks batch COBOL from running against z/OS Databases on Windows unless you choose this override. **Note:**Even if you choose this override, if you use %BINARYSORT() in the COBOL, the system issues an error on Windows. RemoteCall COBOL can run on Windows and UNIX regardless of this option setting, even if CCSID is NOT 37, but the system issues an error.

Field or Control	Description
Save Error is Fatal	Select this option when you have non-repeatable PeopleCode logic in your application's SavePreChange or Workflow. In previous releases, PeopleSoft applications were coded to assume that errors during save are always fatal, but the current PeopleTools release no longer behaves this way. Use this option to ensure predictable behavior with your application without having to modify your older application code. This check box is cleared by default. If you get an error during save processing, the transaction continues and you're allowed to attempt to save again. When this option is selected, if you get an error during save processing the transaction is aborted and all changes are lost. This applies to errors that occur between and including the SavePreChange event to the SavePostChange event. It also includes the component processor save processing. It doesn't include errors from the SaveEdit event. For example, suppose you have some calculations that occur in SavePreChange which are based on the buffers and also modify the buffers. If there's an error during the save and you attempt to save again, the calculations are repeated, but this time based on the buffers that were already modified by the first time the calculations where done. Therefore the second time the calculations are done they will be incorrect, which could lead to incorrect data being saved to the database. In this case you would want to turn on the **Save Error is Fatal** option, because a fatal error on save is more desirable than incorrect data being put into the database.
Set Focus on Save	If selected, focus is set on the Save button when a user saves a component. If not selected, focus is set on the first control on the page that can assume focus when a user saves a component. By default, the option is not selected. **Note:**This setting has a system-wide effect.
Temp Table Instances (Total):	The value that you specify in the Temp Table Instances (Total) edit box controls the total number of physical temporary table instances that Application Designer creates for a temporary table record definition when you perform the Build process. This value indicates the total number of undedicated temporary table instances. The maximum number of temporary table instances that you can specify is 99.

Field or Control	Description
Temp Table Instances (Online)	Enter the available online instance values. When you invoke a process online, PeopleTools randomly allocates a single temporary table instance number to programX for all of its dedicated temp table needs. The higher the number of online instances that is defined, the less likely it is for two online processes to get the same value.
Maximum Message Size	Use this parameter to set the maximum message size being handled by Integration Broker. There is a practical limit to how large a message can be. This limit can be determined by a variety of factors, including network speed, available memory, processing load, and so on.
Base Time Zone	Although you can display time data a number of different ways, PeopleSoft databases store all times relative to a system-wide base time zone. You can adjust the display of the time that an end user sees using the Use Local Time Zone (LTZONE) setting in PeopleTools, Personalizations. This base time zone is the one that the database server uses. In order for PeopleSoft to properly manage time data, the system needs to know which time zone that is. Set the Base Time Zone to the time zone that the database server's clock uses. **Note:**After changing this setting, reboot any application servers that are connected to the database. It is critical for the correct operation of the system that this time zone match the time zone in which the database is operating. Any discrepancy in the base time zone as defined in this page and the time zone in which the database system is operating leads to inaccurate time processing.
Last Help Context # Used	This field is no longer used.

Field or Control	Description
Data Field Length Checking	Normally, field length validation is based on the number of characters that are allowed in a field. For example, a field defined as CHAR(10) in Application Designer holds ten characters, regardless of which characters you enter. In a Unicode database, double-byte characters, such as those found in Japanese, are counted the same as single-byte characters, such as those found in the Latin alphabet. If you create a non-Unicode database, the field length in Application Designer represents the number of bytes that are permitted in the field, not the number of characters. When the non-Unicode database uses a single-byte character set (SBCS), you can only enter single-byte characters, so the number of characters and the number of bytes are the same. However, because double-byte character sets (DBCS) typically allow a mix of single- and double-byte characters, the number of characters that are allowed in a field in a non-Unicode DBCS database varies. This is true for both shifting and non-shifting double-byte character sets. For example, a if a user enters ten Japanese characters into a field that is defined as CHAR(10) in Application Designer, this string needs 20 bytes of storage in a non-shifting double-byte character set and 22 bytes of storage in a shifting double-byte character set. This ten-character input fails insertion into both these databases. Use the Data Field Length Checking option to ensure field length validation appropriate to the database's character set. Values are MBCS and Others. Choose Others if you are using a Unicode-encoded database or a non-Unicode single-byte character set database. This prevents special field length checking. As discussed above, these types of databases do not require such checking. Choose MBCS if you are running a Shift-JIS Japanese database. Japanese is the only language that PeopleSoft supports in a non-Unicode DBCS encoding. All other languages requiring double-byte character sets are only supported by PeopleSoft by using Unicode encoded databases.

Field or Control	Description
Maximum Attachment Chunk Size	Controls the size of the file attachment chunks that are stored in the database. This allows you to upload a file attachment that is larger than could be stored in a single row of the database. The system divides the data into chunks to store in different rows of the database table. The default is 28,000 bytes, and maximum attachment chunk size allowed by PeopleTools is 50 MB. Application Designer data types control how the system manages "long data." If you want to store your files in the database in a form that allows for portability across the database platforms supported by PeopleTools, you must set the **Maximum Attachment Chunk Size** option to a value that would be permitted by all of the database platforms supported by PeopleTools — no more than 28,000 bytes. However, if you do not want to consider the portability of files across database platforms supported by PeopleTools, you may set the **Maximum Attachment Chunk Size** option to the maximum value that your database supports. For example, if you are using the Oracle database, you can set the value to 2 GB. **Note:**The Maximum Attachment Chunk Size determines the size of the chunks *stored* in the database. This setting does not affect the maximum size of the file chunks that can be *read* from that table. The maximum size of the file chunks that the system reads from the database table is determined by the Maximum Length property defined for that field in Application Designer. **Note:**When working with file attachments it is imperative that you understand how the file processing PeopleCode constructs behave, such as the AllowLargeChunks parameter, used with the AddAttachment built-in function. See Application Designer Developer's Guide: Specifying Long Character Field Attributes. See PeopleCode Language Reference: Functions by Category. See PeopleCode Developer's Guide: Understanding the File Attachment Functions.
Upgrade Project Commit Limit	Sets the limit on how many rows can be modified by an upgrade project before the system issues a COMMIT statement.

Field or Control	Description
Enable Switch User	The Enable Switch User option enables you to limit the users who can change identities in a PeopleSoft system. The feature applies only when accessing PeopleSoft using a browser; it has no effect on two-tier or three-tier connections. Most sites have no reason for individual users to change their PeopleSoft system identity during a session. For those sites that do require this capability, the number of users who need to switch to another user profile typically is fairly small. For example, the users who can switch identities are usually limited to the system's GUEST user (if the system has one) and, perhaps, a few system administrators or power users. Options for Enable Switch User are: • *All:* Use this value to indicate that all users are permitted to change their identities during their browser session. This is the default value. • *None:* Use this value to indicate that no user is permitted to change identities during a browser session • *Some:* Use this value to indicate that some, selected users are permitted to change their identities during a browser session. When you select this value, the Allow Switch User check box appears on the General page of the user profile definition in the PeopleTools Security interface. Select Allow Switch User to indicate that the individual user is permitted to change identities within a PeopleSoft session. User profiles that have the Allow Switch User option unselected will be immediately logged out when the system detects a change in the user's identity. Once you specify which users can switch identities, if an identity change is attempted (switch user) by an unauthorized user, the system: • Logs that user off the system, immediately. • Displays the following message on the sign-out page: "Illegal Identity switch has been detected by the System. Please re-login." • Writes an entry in the web server logs indicating that an illegal switch user was attempted. **Note:** Assume that the user profile to which a user switches is the “destination” user profile, and the user profile from which a user switches is the “source” user profile. If a user is allowed to switch identities during a session, it is allowed by the source user profile, not the destination user profile. For example if you allow UserA to switch to UserB, it is UserA's user profile that must have the Allow Switch

Field or Control	Description
	User option selected. The setting for UserB's profile is not relevant in determining whether a switch user from UserA to UserB is allowed. This feature does not control the user profile to which a user can switch; it only addresses which users are permitted to switch identities during a browser session. See Security Administration: Setting General User Profile Attributes.
MinHash	The MinHash option enables you to set the minimum hash algorithm that all sites participating in single sign-on (SSO) must use in order to talk to each other. The default value is SHA256. Options for MinHash are: • SHA1 — The site is hashed with the SHA1 algorithm. • SHA256 — The site is hashed with the SHA256 algorithm. **Note:**PeopleSoft recommends that you do not change the default value. During the PS_TOKEN validation at the Web tier, PeopleSoft uses the MinHash value to prevent Web sites with weaker hashing algorithm from talking to other sites participating in single sign-on. The MinHash option is not used for PS_TOKEN generation. See Security Administration: Implementing PeopleSoft-Only Single Signon for more information on PS_TOKEN.

Field or Control	Description
Case Insensitive Searching	Enables you to provide case-insensitive searching for the PeopleSoft search records when searching for PeopleSoft definitions. The options are: • *Off :* Enables case sensitive searching and sets it to "on" *without* displaying the Case Sensitive check box on the search page. • *On - Case Sensitive Default Off:* (Default) Displays the Case Sensitive check box on the search page unselected, providing the user the option of enabling case-sensitive searching for a particular search by selecting the Case Sensitive check box. • *On - Case Sensitive Default On :* Displays the Case Sensitive check box on the search page selected, providing the user the option of switching to *case-insensitive* searching for a particular search by *deselecting* the Case Sensitive check box. **Note:**This option is not associated with Search Framework.

Field or Control	Description
Default Style Sheet Name **Branding Application Package** **Branding Application Class**	Display the default, system-wide branding settings, which are defined on the Branding System Options page: • The style sheet for classic components. • The application package that contains the application class that defines system-wide branding functions. • The application class that defines system-wide branding functions. See Portal Technology: Configuring Branding System Options for more information on setting these values.
Max Rows in Prompt results	Sets the maximum number of rows seen in a prompt (lookup) result. The default value is 100. See Applications User’s Guide: Using Lookup Prompts.
Max rows in Global Search	Sets the maximum number of rows in the result list that an end user sees from a Global Search. The default value is 50. See Search Technology: Working with Search Results in Global Search.
Max Rows in Pivot Grids	Sets the maximum number of rows displayed in a pivot grid.

Help Options

Field or Control	Description

F1 Help URL	Specify the URL that points to an installation of PeopleTools online help. When users press the F1 key in Application Designer or other Windows client applications, the appropriate context-sensitive PeopleTools documentation is displayed.
	You can point to either a local installation of PeopleTools online help or to an installation on Oracle's hosted documentation website:
	• To point to a local installation of PeopleSoft online help, construct the URL similar to the following:
	`http://`*`webserver`*`:`*`port`*`/help_folder/help.html?ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%`
	For example:
	`http://myserver.example.com:8080/help-PB/help.html?ContextID=%CONTEXT_ID%&LangCD=%LANG_CD%`
	• To point to an installation on Oracle's hosted documentation website, construct a URL similar to the following:
	`https://docs.oracle.com/pls/topic/lookup?id=%CONTEXT_ID%&ctx=`*`library_cd`*
	For example:
	`https://docs.oracle.com/pls/topic/lookup?id=%CONTEXT_ID%&ctx=pt858latest`
	Note: For more information on current library codes, see Configuring Context-Sensitive Help Using the Hosted Online Help Website or Configuring Context-Sensitive Help Using a Locally-Installed Online Help Website.
	Important: This setting applies only to F1 help accessed through Application Designer and other Windows client applications. Therefore, you must configure this link to point to a PeopleTools library, and not to some other PeopleSoft application. Use the Web Profile Configuration page to configure the Help link displayed on PeopleSoft pages accessed through

Field or Control	Description
	a browser. See Portal Technology: Configuring General Portal Properties for more information.
Ctrl-F1 Help URL	This setting only applies to the Windows environment (such as Application Designer). The Ctrl+F1 URL allows you to provide an alternate location for help. For example, you may set the main F1 Help URL to PeopleTools online help and the Ctrl+F1 for the company's help site.

Database Encryption Algorithm

(Applies to Oracle databases only.) Enables you to select the encryption algorithm the system uses to encrypt database fields to additional security.

See Data Management: Implementing Oracle Transparent Data Encryption.

Fluid Mode Properties

These properties apply to fluid applications only.

See Fluid User Interface Developer’s Guide: Configuring PeopleTools Options for Fluid Applications.

Favorite Searches

Field or Control	Description
Favorite Searches	Select this check box to enable all users to save configurable searches to their My Favorite Searches page. The feature is enabled by default.
Max Favorite Searches	Enter the maximum number of searches that a user can save and add to their My Favorite Searches page. The default is 10.

See Search Technology: Working With Saved Searches.

Setting Smart Prompt and Smart Drop-Down Options

You can configure prompt fields and drop-down fields to respond to user choices and display prompt values or drop-down list values based on recently used or frequently used values. Such prompts and drop-down lists are called Smart Prompt and Smart Drop-Down respectively. Smart Prompts and Smart Drop-Down are enabled by default.

You can configure the smart prompt and smart drop-down feature on:

- Classic and fluid pages.
- Search pages and transaction pages.
- Standard and keyword search.
- All fields at all levels.

> **Note**
>
> Smart-prompt feature is not supported on date fields.

- Smart drop-down feature is supported on system-generated drop-down fields and PeopleCode-generated drop-down fields.

> **Important**
>
> The smart prompt and smart drop-down feature is not available in screen reader mode.

For information on end users interacting with smart prompt and smart drop-down lists, refer to Application User's Guide: Using Smart Prompts and Applications User's Guide: Using Drop-Down List Boxes. Also, refer to the Considerations for Using Smart Prompt and Smart Drop-Down in Grids and Scroll Areas section in Applications User's Guide: Adding Rows.

Consider the following regarding the smart prompt and smart drop-down feature:

- Browser cache is used for data storage.
- Browser cache stores data per user basis, that is, data stored for one user cannot be accessed by another user.
- Browser cache is updated when a user selects a value from the prompt values or from the drop-down list. The page does not have to be saved.

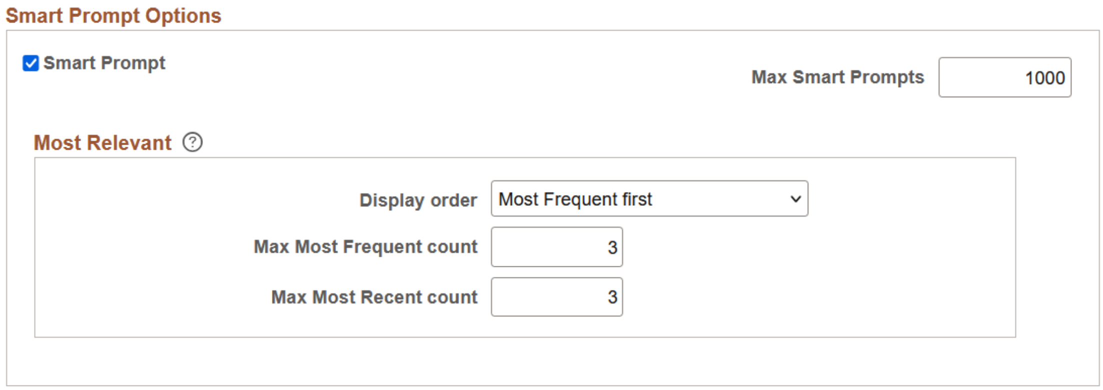
You can disable or enable smart prompts and smart drop-down at four levels, namely:

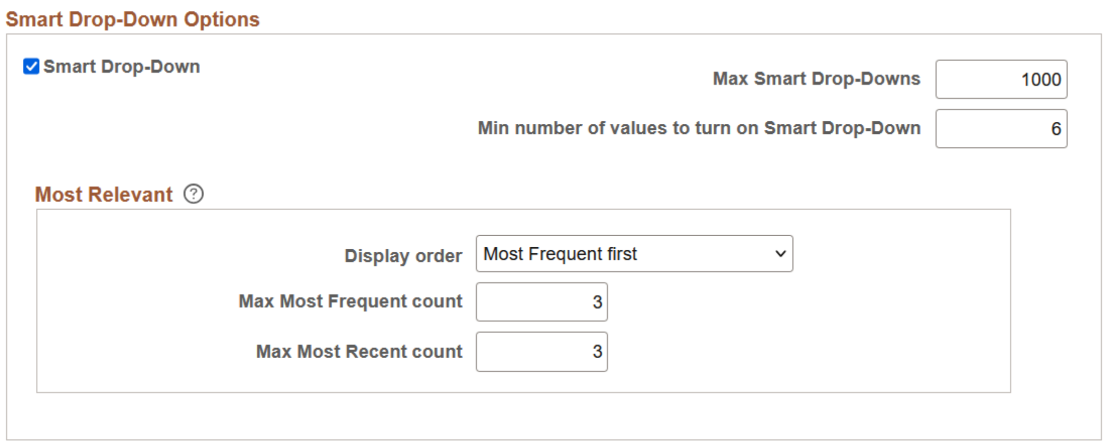
- At system-wide level (on the PeopleSoft Options page) by system administrators, which is described in this topic.
- At the user level (on the My Preferences page) by an end-user, for more information, see Applications User's Guide: Specifying General Settings in My Preferences.
- At the record-field level (in Application Designer) by application developers, for more information, see Application Designer Developer's Guide: Setting Record Field Use Properties.
- At the page-field level (in Application Designer) by application developers, for more information, see Application Designer Developer's Guide: Setting Use Properties.

Other conditions and factors also affect the availability of the feature are described later in this topic.

On the PeopleTools Options page, use the Smart Prompt Options and the Smart Drop-Down Options group boxes to disable or enable the Smart Prompt and Smart Drop-Down Options system-wide.

This example illustrates the fields and controls on the PeopleTools Options page: Smart Prompt Options and Smart Drop-Down Options. You can find definitions for the fields and controls later on this page.





The following table describes the Smart Prompt Options:

Field or Control	Description
Smart Prompt	Smart Prompt is enabled by default. Deselect the **Smart Prompt** check box to disable the display of smart prompt system-wide. That is, the prompt field displays the list of values devoid of sections. To enable smart prompt feature, select the **Smart Prompt** check box. See the table describing the configuration at the four levels that control the availability or non-availability of the smart-prompt option.
Max Smart Prompts	The maximum number of smart prompt values is 1000, by default. This is a configurable value, so based on your business requirements, you may increase or decrease the maximum number of smart prompt values. If you want to increase the maximum number of smart prompt values beyond the default value, you should be aware that it may impact storage, performance, and usability.

Field or Control	Description
Display Order	Select whether you want to display frequently viewed values first or recently viewed values first in the Most Relevant section of smart prompt. The default is **Most Frequent first**. When you select **Most Frequent first**, then in the Most Relevant section of smart prompt, the most frequently used values are listed first followed by the most recently used values. Similarly, if you select **Most Recent first**, then in the Most Relevant section of smart prompt, the most recently used values are listed first followed by the most frequently used values.
Max Most Frequent Count	Specify the maximum number of frequently used values you want to display in the Most Relevant section of smart prompt.
Max Most Recent Count	Specify the maximum number of recently used values you want to display in the Most Relevant section of smart prompt.

The following table describes the Smart Drop-Down Options:

Field or Control	Description
Smart Drop-Down	Smart Drop-Down is enabled by default. Deselect the Smart Drop-Down check box to disable the display of smart drop-down lists system-wide. That is, the drop-down field displays the list of values devoid of sections. To enable smart drop-down feature, select the **Smart Drop-Down** check box. See the table describing the configuration at the four levels that control the availability or non-availability of the smart-prompt option.
Max Smart Drop-Downs	The maximum number of smart drop-down list values is 1000, by default. This is a configurable value, so based on your business requirements, you may increase or decrease the maximum number of smart drop-down list values. If you want to increase the maximum number of smart drop-down list values beyond the default value, you should be aware that it may impact storage, performance, and usability.
Min number of values to turn on Smart Drop-Down	Specify the minimum number of drop-down list values to activate the Smart Drop-Down list. When the specified threshold is reached, the drop-down list values are displayed based on recently used or frequently used values and the list is divided into Most Relevant and Others sections. The default minimum number is 6.

Field or Control	Description
Display Order	Select whether you want to display frequently viewed values first or recently used values first in the Most Relevant section of smart drop-down list. The default is **Most Frequent first**. When you select **Most Frequent first**, then in the Most Relevant section of smart drop-down list, the most frequently used values are listed first followed by the most recently used values. Similarly, if you select **Most Recent first**, then in the Most Relevant section of smart drop-down list, the most recently used values are listed first followed by the most frequently used values.
Max Most Frequent count	Specify the maximum number of frequently viewed values you want to display in the Most Relevant section of smart drop-down list.
Max Most Recent count	Specify the maximum number of recently used values you want to display in the Most Relevant section of smart drop-down list.

The following table illustrates the configuration of smart prompts and smart drop-down options at the four levels and its resultant availability or non-availability.

System Level	User Level	Record-Field Level	Page-Field Level	Feature Availability
Enabled	Enabled	Enabled	Enabled	Yes
Not enabled	Enabled	Enabled	Enabled	No The feature is disabled for all users.
Enabled	Not enabled	Enabled	Enabled	Not available for the specific user; no impact to other users.
Enabled	Enabled	Not enabled	Enabled	Smart prompt feature is disabled in every page field that uses this record field and impacts all users. No impact to other fields in those pages.
Enabled	Enabled	Enabled	Not enabled	Smart prompt feature is disabled only for that page field and impacts all users. Does not impact other fields in the same page, similarly no impact to other pages.

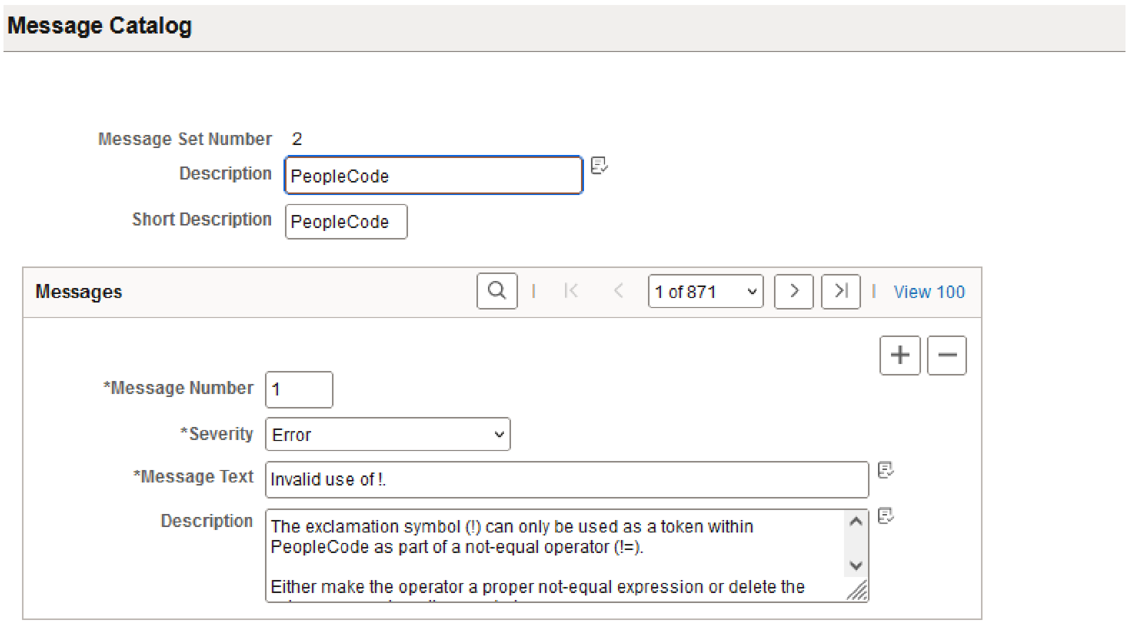
In addition to the configuration of smart prompt and smart drop-down options at the four levels, several factors or conditions also affect the availability or non-availability of the smart prompt and smart drop-down feature. When the following conditions or factors are true, the smart prompt and smart drop-down feature is not available.

Conditions or Factors	Feature Availability
User logs in screen reader mode.	No
Browser cache is not available.	No
A user switches from one browser to another browser.	Smart prompt and smart drop-down feature is not available till the user choices are built in the browser cache. Browser cache is not imported from other browsers.
User interface elements that look like drop-down list, for example, search category on global search, and so on.	No
User interface elements that look like prompt fields, such as, push buttons with prompt action.	No
Lookup field is a search record field, which does not have a prompt table.	No
Browser cache is cleared.	Smart prompt and smart drop-down feature is not available till the user choices are built again in the browser cache.
PeopleTools version is upgraded.	Smart prompt and smart drop-down feature is not available till the user choices are built in the browser cache.
PIA URL is changed. The PIA URL may change due to various reasons such as, when Web servers are recreated, system downtime due to maintenance, or servers are moved.	Smart prompt and smart drop-down feature is not available till the user choices are built in the browser cache.

Message Catalog

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Message Catalog** to access the Message Catalog page.

This example illustrates the fields and controls on the Message Catalog page. You can find definitions for the fields and controls later on this page.



You add and maintain system messages by using the Message Catalog page. PeopleSoft error messages are stored in the Message Catalog, and organized by message set number. Each message set consists of a category of messages, ranging from PeopleTools Message Bar Items and PeopleCode Runtime Messages to PeopleSoft Payroll and PeopleSoft General Ledger application messages.

Field or Control	Description
Message Set Number	Identifies the message set.
Description	The Message Set Description is a reference that is used on reports and pages for easy identification.
Short Description	The Message Set Short Description is a reference that is used on reports and pages for easy identification.
Message Number	Each message set consists of one or more rows of messages that are identified by a message number.

Field or Control	Description
Severity	You assign each message a severity, which determines how the message appears and how the component processor responds after the user acknowledges message. The severity levels are: Cancel: This severity should be reserved for the most severe of messages, as when a critical error occurs and the process must be aborted or a machine needs to be shut down. To indicate how rarely this severity level is appropriate, of all PeopleTools messages only five or so have a severity level of Cancel. In almost all cases, you use one of the other severity levels. Error: Processing stopped, and data cannot be saved until the error is corrected. Message: This is an informational message and processing continues normally. Warning: User can decide to either stop or continue processing despite the error.
Message Text	In the Message Text edit box, you see the message text. Any reference to the characters %n, as in %1 or %2, is replaced by parameter values that the system provides.
Explanation	The Explanation text provides a more in-depth explanation of why the message is generated and how to fix the problem. This text appears below the Message Text when the message appears.

PeopleTools uses some messages, but the applications use the other messages, which get called by the Error, Warning, Message Box, MsgGet, and MsgGetText built-in PeopleCode functions.

Note

You can create messages and message sets to support new or customized functionality in the system. You can also edit the messages that PeopleSoft delivers if required. In both of these cases, remember that Oracle reserves all message sets up to 20,000 (that is, 1 to 19,999). If you add a message set or edit a message set with a number that is less than 20,000, it may be overwritten in future upgrades. Message sets must not exceed 32,767, and PeopleSoft recommends that you use message sets from 20,000 to 30,000 to create new messages.

To add a message set:

1. Select Utilities, Administration, Message Catalog, and on the search page click **Add New Value**.
2. Enter the value of the new Message Set Number and click OK.
3. Enter a description and short description of the type of messages that this message set contains.

Try to group the messages logically. For instance, create one message set for the new budgeting application and a different one for the customized billing pages.

4. Add messages.
5. Save your work.

To add a message:

1. Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Message Catalog**.
2. Open the desired message set.
3. In the Message Catalog page, click the plus sign button to add a new row.

   The Message Number value is automatically set to the next unassigned number in the message set.
4. Select a Severity level, enter message text and a detailed explanation.
5. Save your work.

Spell Check System Dictionary

PeopleSoft PeopleTools provides personal and system-level dictionaries. End users and system administrators can add words to the dictionary for use with the spell check feature. Typically, system administrators add words to the system-level dictionary that are used company-wide; end users add additional role-specific terminology to their personal dictionaries.

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Spell Check System Dictionary** to access the system-level dictionary.

Select the All Languages page to enter words that are valid across all languages. Select the Language Specific page for those words that are valid to a specific language:

To add words to the system dictionary by language:

1. Select Language Specific on the Spell Check System Dictionary page.
2. Select the desired language from the Spell Check Language drop-down list box.
3. Select Use session language from the Spell Check Language drop-down list box to add a word to the current session's spell check dictionary. After saving this word, the language field refreshes to the current spell check language.
4. Enter the word (maximum 40 characters) that is to be added in the Spell Check Word field.
5. Save your changes.

Case Sensitivity for Spell Check

The words that you add to your personal dictionary are case-sensitive and are validated by the following rules:

1. If the added word is all lower case, such as worklist, then the following are considered valid:
   - Exact match, all lower case (worklist).
   - All uppercase (WORKLIST).
   - Initial capitals (Worklist), regardless of its position in the sentence. Mixed case (WorkList) is considered incorrect.

2. If the added word is all uppercase, such as CRM (customer relationship management), then only an exact match is valid.
3. If the added word is in initial capitals, such as California, then only an exact match and all upper case (CALIFORNIA) are considered valid.
4. If the added word contains an embedded capital letter, such as PeopleSoft, then only an exact match is valid. Therefore, if case is not relevant to the validity of the word, use all lower case.

Table Structure for Word Storage

System and personal words are stored in the database in the PSSCWORDDEFN table with the following fields:

- SCOPRID indicates whether a word is a system word or a user's personal word.
- SCLANG stores the dictionary language for which the word is considered valid. If the system administrator chooses to store the word for all languages, this field is left blank.
- SCWORD stores the actual word, with a maximum length of 40 characters.
- SCNEGWORDFLG is a flag used to determine if a word is negative (incorrect) or not. Values can be 'N' or 'Y'. PeopleSoft does not currently use this feature, so the value should always be set to 'N'.

To load values in bulk into PSSCWORDDEFN:

1. Using the method of your choice (as in a SQL script), issue SQL similar to the following:

```
insert into PSSCWORDDEFN (SCOPRID, SCLANG, SCWORD, SCNEGWORDFLG)
values ('SYSTEM', 'SC00', 'nnn', 'N')
```

> **Note**
>
> For each word you want to add to the library, you need a separate insert command, and the value 'nnn' will be changed in each of those insert statements to be the next value in the list of words you want to add.

2. Add a value (any value) to the Language Specific tab and click Save.

   This alerts the runtime system to update the cached version of the PSSCWORDDEFN table.

> **Note**
>
> In the current release, the maximum number of rows in the PSSCWORDDEFN table should not exceed 2,850.

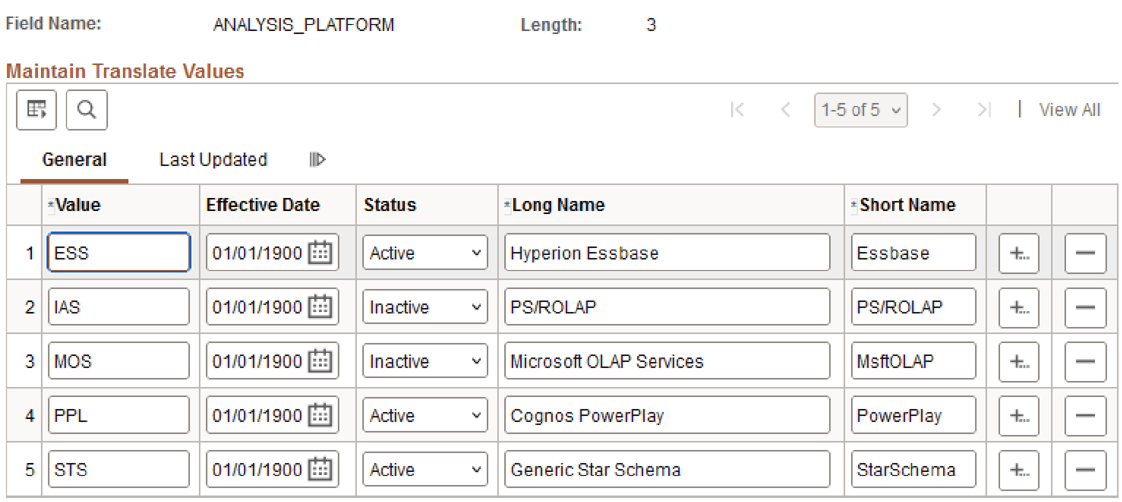
Maintain Translate Values

You use the Maintain Translate Values interface to maintain the values in the translate table. If it's allowed by site security administrators, power users can now learn to add their own pick lists (translate values) to an application.

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Maintain Translate Values** and search for a Field Name to access the Maintain Translate Values page.

This example illustrates the fields and controls on the maintain translate values page. You can find definitions for the fields and controls later on this page.

Use this page to maintain translate values



Field or Control	Description
Value	Enter the value for the translate selection.
Effective Date	Specify a date for the value to become active. **Note:**If you are adding a second row for the same translate value, you must enter a unique effective date.
Status	Specify whether the value is active or not.
Long Name	Enter a long description for identification. There is a 30-character limit.
Short Name	Enter a shorter description for identification. There is a 10-character limit.

Related Topics

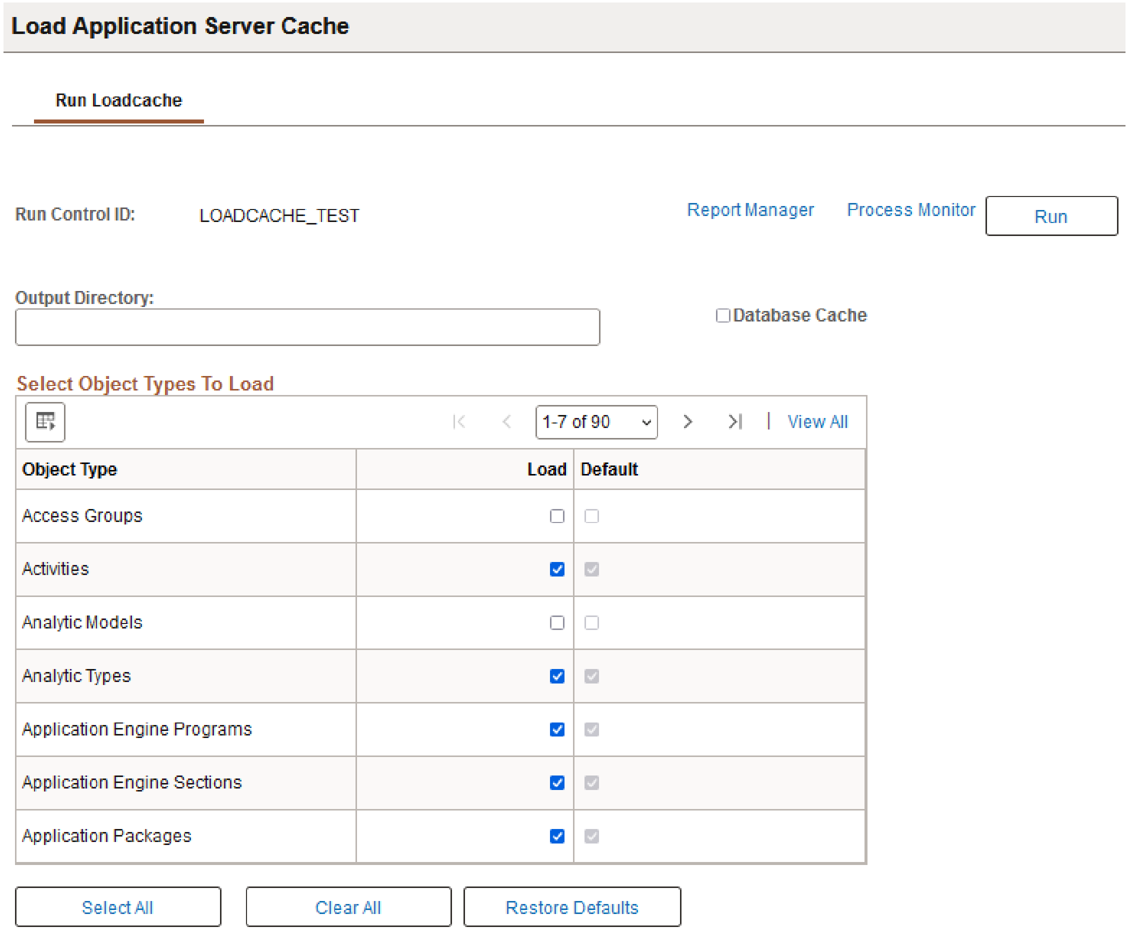
- Application Designer Developer's Guide: Using the Translate Table

Load Application Server Cache

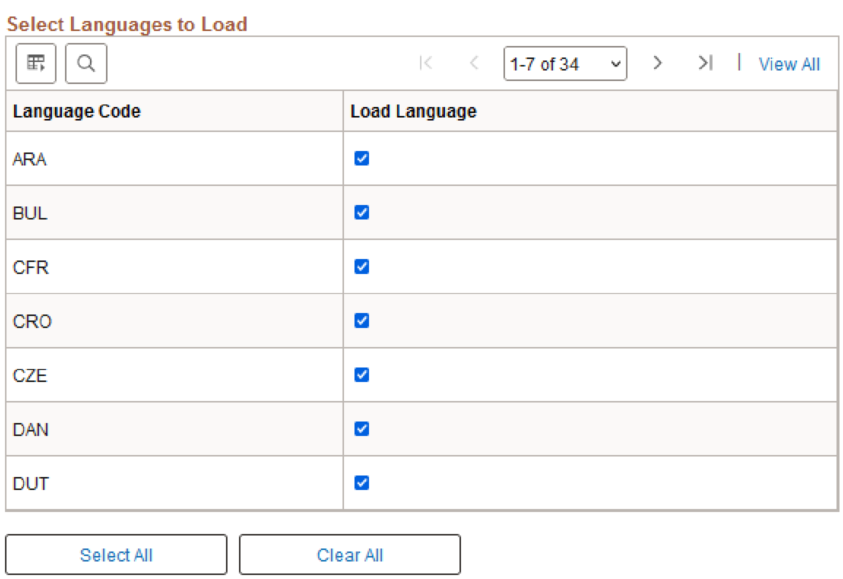
The Load Application Server Cache page enables you to run Application Engine programs that load the *shared* file cache or the database cache for application server domains.

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Load Application Server Cache**. Add a Run Control ID or select an existing one to access the Load Application Server Cache page.

This example illustrates the fields and controls on the Load Application Server Cache page (1 of 2).



This example illustrates the fields and controls on the Load Application Server Cache page (2 of 2).



The usage of the fields and controls on the Load Application Server Cache page is described in the section Loading Shared File Cache or Database Cache.

You need to run a cache loading program described in this section only if you intend to implement shared file caching on the application server or database caching.

To load *non-shared* file cache or database cache with specific cache projects, you use a separate process, which is documented elsewhere in this guide.

See Configuring an Application Server Domain to Preload Cache.

See ServerCacheMode.

Understanding Application Server Caching

Each server process running within a domain has these caching features:

Cache Type	Description
Memory cache	Stores PeopleTools definition metadata to improve system performance. Memory cache is always enabled for each server process in a domain.
File cache	Stores PeopleTools definition metadata, such as the metadata for pages, locally on the application server. File caching is controlled by the EnableServerCaching parameter in PSAPPSRV.CFG.
Database cache	Stores PeopleTools definition metadata, such as the metadata for pages, in database tables. Database caching is controlled by the EnableDBCache parameter in PSAPPSRV.CFG. Any domain that has EnableDBCache enabled and connects to the database uses the cache information stored in the database.

With file caching, you have the option of implementing shared or non-shared file caching.

File Cache Option	Description
Shared	With shared file caching, all of the server processes within a single domain share the same file cache location.
Non-shared	With non-shared file caching, each server processes within a single domain maintains its own file cache. For example, with non-shared caching configured, each PSAPPSRV server process uses its own file cache.

Oracle provides cache loading programs designed specifically to load cache prior to users accessing the system, which you run from the Load Application Server Cache page. The cache loading programs retrieve all of the PeopleTools definition metadata from the database and builds either file cache that you can deploy locally on an application server or it populates the database cache tables, depending on what type of cache you are implementing.

Running a cache loading program is the equivalent of having a user access every page in the system once so that all the metadata associated with each page is retrieved from the database and stored in the local cache. With all the metadata already stored in cache, the end users experience more predictable performance.

With cache already loaded, end users don't have to wait for the system to retrieve a definition from the database and cache it, each time a definition is accessed for the first time. The system retrieves all of the required objects from the cache as opposed to querying the database for each request. This provides significant performance benefits for first-time and large transactions.

If you elect to run a cache loading program, consider the following items:

- You need to run one of the cache loading programs at least once. Once the cache is loaded, the cache loading programs download only new or changed metadata to the cache.

  > **Note**
  >
  > Items that are outdated in the cache are marked invalid but are not deleted. This includes design time changes, upgrades, patches, and so on. However, the system uses only the most current cache definitions.

- The first time that you run a cache loading program, it can take a significant amount of time to complete, for example 2 to 30 hours. The duration of the program run depends on a number of factors, including the number of active languages selected, the size of the database, and the power of the server machine. Subsequent program runs complete in less time if there is already valid cache data in the target cache directory or the database cache tables. The programs are designed to update only the changed objects.
- You can copy the output of a cache loading program to other machines (for shared file caching).
- You can select the object types and languages for cache loading.

> **Note**
>
> Shared file cache output is not portable to different operating systems. For instance, if you generate the cached metadata on a Windows server, you can't copy the cache files to a UNIX server.

> **Warning**
>
> If for any reason you update PSSTATUS.LASTREFRESHDTTM, the system marks all items in the shared cache as invalid, and you will need to rerun one of the cache loading programs again.

Oracle provides these options for loading your cache:

Preloading Cache Option	Process Scheduler Process Type	Description
Serial cache loading	Process	Serial cache loading refers to running an Application Engine program named LOADCACHE. The LOADCACHE program, adds each applicable cache object to the cache, sequentially. **Note:** If you use the Run Control ID PT_LOADCACHE_ALL when running on the command line, it will load cache for all managed objects. Otherwise, it will load cache for the default set of objects, as in previous releases.
Parallel cache loading	PSJob	Parallel cache loading is intended to reduce the time required to build the cache. Parallel cache loading splits the work of the LOADCACHE program into two separate Application Engine programs that run simultaneously within the PSJob PLCACHE. The PLCACHE job runs a basic setup program, and then runs these programs: • LCACHE_INDEP: Loads all the cacheable objects that are independent of other cached objects. • LCACHE_DEP: Loads all the cacheable objects that are dependent on other cached objects. Objects that are dependent on other cached objects are those objects that, when loaded into cache, cause instances of other object to be loaded as well.

Loading Shared File Cache or Database Cache

To load shared file cache or database cache:

1. Make sure that the database that the application server runs against produces a clean SYSAUDIT report.

   If SYSAUDIT is not clean, the cache loading programs can fail.

2. Access the Load Application Server Cache page, entering an appropriate Run Control ID.

   Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Load Application Server Cache** to access the Load Application Server Cache page.

3. (For shared file cache) In the **Output Directory** field, enter a valid location.

> **Note**
>
> In almost all situations, the system ignores any custom value entered in the Output Directory and generates output into the default cache-loading output directory: *PS_CFG_HOME*\appserv\prcs*ProcessScheduler_domain*\CACHE\CACHE\stage\stage. However, if a cache loading program runs through a PSNT Process Scheduler server running on a remote drive, *and* the application server cache directory and Process Scheduler cache directory are located on the local drive, then the value entered in the Output Directory field supersedes the default output directory.

4. (For database caching) Select the **Database Cache** check box.

   This disables the output directory field. With database caching enabled, the cache loading program inserts the cache directly into the database cache table, PSOBJCACHE.

5. Select the **Load** check box beside the object type for cache loading.

   The list includes all cacheable object types, with display-only check marks for those that are defaults. The default set includes the minimum list of object types included for this functionality in previous releases.

   Click **Restore Defaults** to discard your selections and return to the default set of cacheable objects.

6. Select the languages to load.

   If not all possible languages are required to be loaded for a domain, you can improve performance of the program run by selecting only those languages that are required.

7. Click Run.

   The Process Scheduler Request page appears.

8. From the Server Name drop-down list, select the name of the server to run the cache-loading process.

9. From the Process List, select *Serial LoadCache* to run the single LOADCACHE Application Engine program, or select *Parallel LoadCache* to run the PLCACHE PSJob to run multiple Application Engine jobs in parallel.

10. (For Parallel LoadCache) Ensure that the Process Scheduler server definition is set to run the LOADCACHE process category and that the Max Concurrent value is set appropriately.

    Parallel LoadCache is assigned to the LOADCACHE process category. It will run only on servers that are configured to run the LOADCACHE process category.

    For the LOADCACHE process category, set the Max Concurrent value greater than 0. The recommended value is *2*.

    If you modify the server definition, restart the server for your changes to take effect.

> **Note**
>
> When running the cache loading programs, only one server at a time should have the MaxConcurrent value set greater than 0. This guarantees that the processes running within the Parallel LoadCache job will run on the same server. If processes within the job run on different servers, this could result in each server having an incomplete set of generated cache files.

See Process Scheduler: Creating Server Definitions.

11. Click **OK** to launch the cache loading program.

    After you invoke the process, you can use the **Report Manager** and **Process Monitor** links to monitor the progress.

    The first time that you run the program, the process may take numerous hours to complete (less time for the Parallel LoadCache option).

12. When the cache loading program completes, verify the existence of the cache contents.

    For shared caching, the cache loading program creates the shared file cache in the following directory:

    - Windows: *PS_CFG_HOME*\appserv\prcs*ProcessScheduler_domain*\CACHE\CACHE\stage\stage
    - For UNIX or Linux: *PS_CFG_HOME*/appserv/prcs/*ProcessScheduler_domain*/CACHE/STAGE/stage

    Where *ProcessScheduler_domain* is the Process Scheduler domain in which you ran the process.

    If database caching is enabled, the program loads the cache directly into the database. To verify the existence of the database cache, run the following SQL statement using your SQL editor:

    ```
    SELECT * FROM PSOBJCACHE;
    ```

13. Once you've determined that the cache has been successfully loaded, shut down the application server domain(s), and modify the domain configuration to recognize the generated cache.

    For shared file caching, edit the PSAPPSRV.CFG file, and enable shared caching, by uncommenting and setting `ServerCacheMode=1` for all appropriate domains.

    For database caching, edit the PSAPPSRV.CFG file, and enable database caching, by uncommenting and setting `EnableDBCache=Y` for all appropriate domains.

14. (For shared file cache) Create a \SHARE directory within the CACHE directory for the appropriate application server domains, and copy the output of the load caching program into the \SHARE directory.

    For example, copy the output of the load caching program to *PS_CFG_HOME*\appserv*domain*\CACHE\SHARE directory for the appropriate application server domain(s).

    > **Note**
    >
    > UNIX and Linux systems expect an uppercase \SHARE directory. For Windows, the \SHARE directory can be mixed case.

15. Boot (or reboot) the application server domain(s).

Working with Database Cache

Database caching provides another implementation option. Determining which caching option works best for your site requires analysis and testing. For many cases, file caching will continue to be the preferred method.

The benefits of database caching include:

- Cache loading only needs to be run once for all domains connecting to a database as they will all share the same database cache.
- New domains can immediately take advantage of the database cache once EnableDBCache is set in the domain configuration. That is, no additional cache loading programs need to be run or additional output that needs to be copied.
- Eliminates administrative overhead of managing disk space and cache files on the file system.

> **Note**
>
> The database cache stores the cached metadata as a BLOB (binary large object) in a single table for the most efficient retrieval.

> **Note**
>
> The database cache is generated specific to the system's byte-order (little-endian or big-endian). The database cache is shared by all systems that have the same byte-order.

Maintaining the Versions of Cached Definitions

PeopleTools uses version numbers to keep cached objects synchronized with the data in the database. When a system undergoes change (such as during custom configurations, updates, or upgrades) versions can get out of sync, and they need to be reset.

Use the VERSION Application Engine program to maintain the definition version synchronization in your system. VERSION can run in these modes.

Mode	Invoked From	Run Control ID	Description
Reset All Versions	Command line only	Must start with RESETVERSIONS. For example, the run control ID could be RESETVERSIONS, RESETVERSIONS2, RESETVERSIONS3, and so on. **Note:**The run control ID is not case sensitive.	Resets all versions to 1. Before running in this mode, shut down all servers and client connections. After running, regenerate cache.
Reset Differences Only	Command line or Process Scheduler	*Any* run control ID, *not* beginning with RESETVERSIONS or REPORTONLY.	Synchronizes only the definition versions where discrepancies exist. Severs can remain running and clients can remain connected. Cache does not need to be regenerated.

Mode	Invoked From	Run Control ID	Description
Report Only	Command line or Process Scheduler	Must begin with REPORTONLY. For example, the run control ID could be REPORTONLY, REPORTONLY2, REPORTONLY3, and so on **Note:**The run control ID is not case sensitive.	Generates a report indicating where version discrepancies exist, but does not update any versions. Severs can remain running and clients can remain connected. Cache does not need to be regenerated.

When running VERSION in any mode, the system generates a report and writes it to %PS_SERVDIR%\log_output.

When running VERSION through Process Scheduler, keep these items in mind:

- Invoke VERSION from **PeopleTools**, and then **Application Engine**, and then **Submit AE Process Requests.**
- Create the appropriate run control ID (or use an existing run control ID), such as REPORTONLY.
- Set Process Frequency to *Always* so that the run control ID can be used more than once.

Working with the Dynamic Cache Repair Utility

PeopleTools provides a dynamic cache repair utility that enables you to configure the system to monitor and adjust automatically a domain's cache while it runs, requiring no intervention from system administrators, and without requiring the domain or a server process to be restarted.

For more information on enabling and using the cache repair utility, see EnableCacheRepair.

Related Topics

- Configuring an Application Server Domain to Preload Cache
  This section provides an overview and discusses how to:
- Application Engine: Using the Command Line to Invoke Application Engine Programs
- Cache Settings
  Use this section to specify how to handle caching at your site. Enabling caching on the application server improves performance, but you need to review the available cache options and determine which option best suits your site.

Manager Auto Update Settings

When managed objects are changed, it is possible for the version in the cache to get out of sync with the version in the database. PeopleTools includes the Extended Manager Update Protocol feature, which is a series of SQL statements that synchronize managed object cache with the database.

The Extended Manager Update Protocol feature is run any time a managed object is updated in:

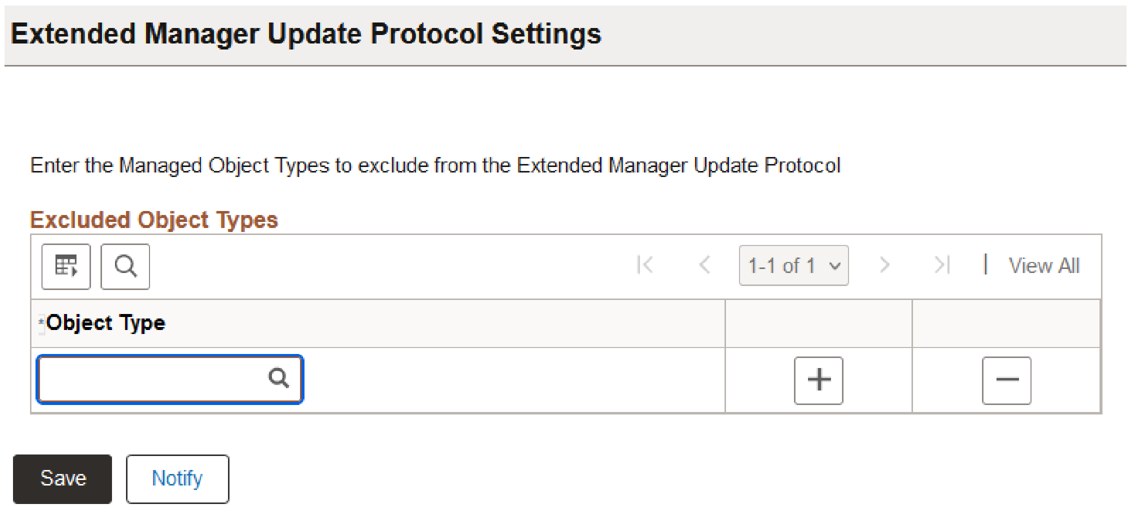
- C++ codeline outside of the manager framework
- PeopleCode SqlExec statements
- SQL objects

- Data Mover scripts

In some cases, Oracle Support may direct you to exclude managed objects from the Extended Manager Update Protocol. In that case, use the Extended Manager Update Protocol Settings page to disable the Extended Manager Update Protocol for selected managed objects.

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Manager Auto Update Settings.**

This example illustrates the fields and controls on the Extended Manager Update Protocol Settings page.



Click the Object Type find button and select the object that you want to exclude from the Extended Manager Update Protocol. Click the add button (+) to select additional objects for exclusion.

Save the page. You do not have to restart the application server or Process Scheduler in order for the changes to take effect.

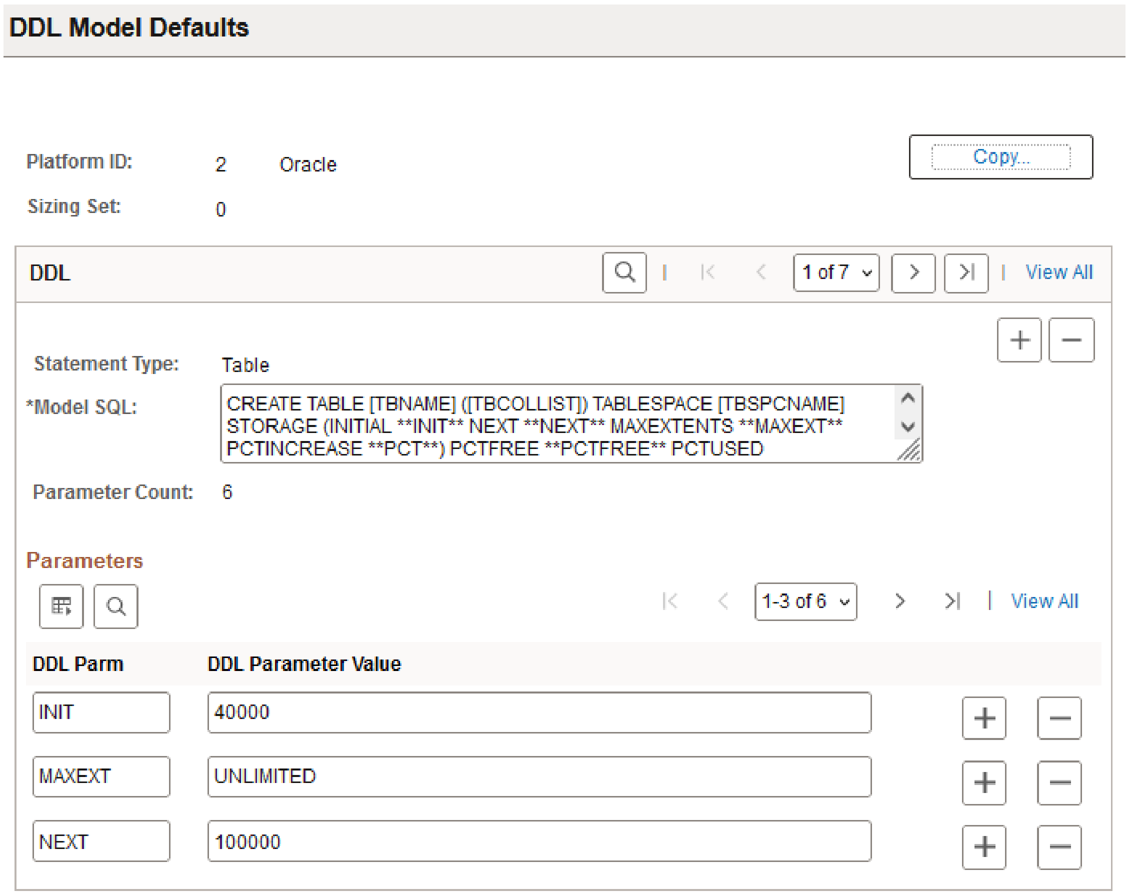
DDL Model Defaults

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **DDL Model Defaults** and search for your database platform name to access the DDL Model Defaults page.

This page is used to view and edit the DDL for creating tablespaces, indexes and tables. Any changes that you make here are global.

> **Important**
>
> Oracle recommends that you do not use the DDL Model Defaults page to edit the model SQL because if you edit the SQL incorrectly, DDL generation may fail at runtime and the Record DDL menu item from Application Designer may not work correctly.

This example illustrates the fields and controls on the DDL Model Defaults page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
Platform ID	Identifies the type of platform of the current database type.
Sizing Set	Indicates the number of Sizing Sets. Specify multiple Sizing Sets if needed. Sizing Sets are a way to maintain multiple versions of the DDL Model statements for a particular database platform. For example, you could have one sizing set to be used during a development phase, when tables only have test data, and you could have separate sizing set to be used during production, when tables have much more data.
Copy	Copies information from one sizing set to another.
Statement Type	Indicates the type of statement that's entered in the **Model SQL** edit box. Values for this field can be *Table*, *Index*, and *Tablespace*.

Field or Control	Description
Model SQL	This field displays the model SQL statements, which you can edit. Valid statements are CREATE TABLE, CREATE INDEX, CREATE TABLESPACE, and a platform-specific statement for updating statistics. Some platforms have all the statements, some do not. When you edit the model SQL statements, consider the following: • System parameters should not be added, that is, parameters within square brackets. • If you add a new DDL parameter, you must provide a default value. • If you add a new DDL parameter from PIA, the newly added parameter must be added to the model SQL statement within the ** markers, for example **newparameter**. **Note:**If the rules for adding new DDL parameters are not adhered to, DDL generation may fail at runtime and the Record DDL menu item from Application Designer may not work correctly.
Parameter Count	The Parameter Count is calculated based on how many nonblank DDL Parm rows that you define.
DDL Parm	Enter the DDL Parm (parameter) to which you want to assign a default.
DDL Parameter Value	The DDL Parameter Value field is where you override a DDL parameter's default value with your own for the selected statement type.

Using the DDL Model Defaults page, you can maintain DDL model statements and default parameters for Data Mover and Application Designer when DDL (PSBUILD.SQL, for example) is generated during the build process.

Using this utility, you can:

- Scroll through all the statement types and platforms that are defined in the PSDDLMODEL table.
- Change the default values or overrides for the parameters in the DDL model statements.
- Add, delete, or change DDL parameters and values.

> **Note**
>
> There is no validation performed on the Model SQL statement, the DDL Parm syntax, or the relationship between the statement and the parameters.

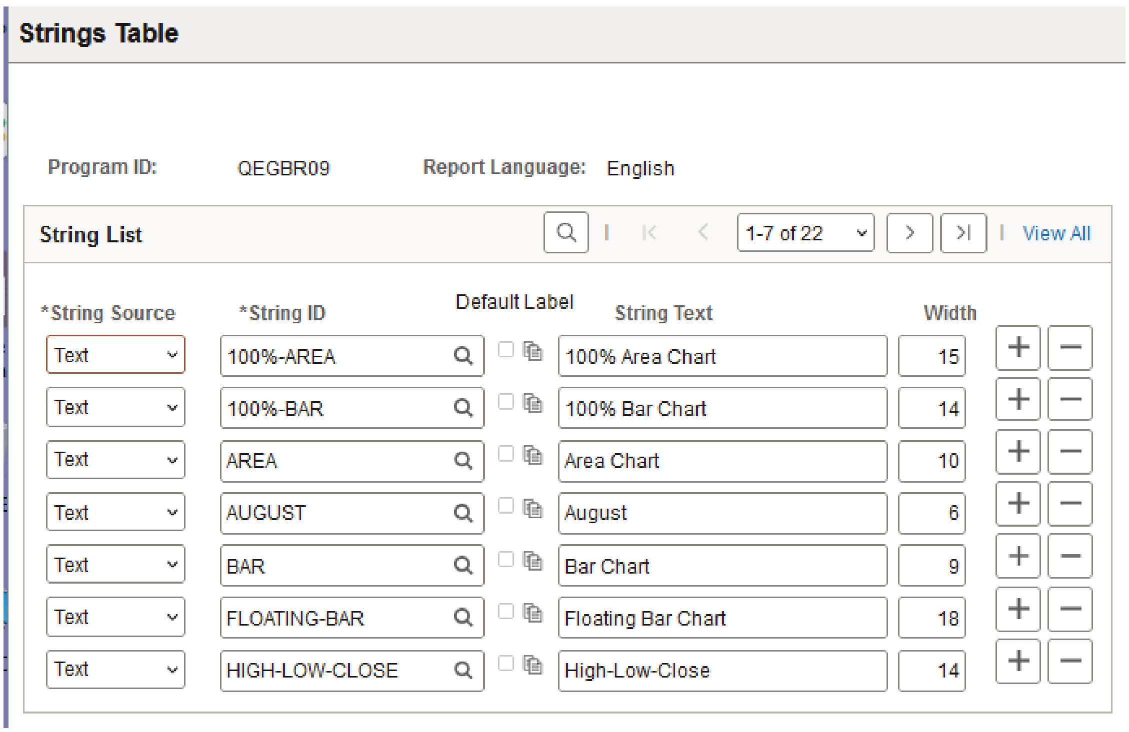
The Platform IDs are as follows:

Number	Platform
1	DB2
2	Oracle
3	Informix (no longer supported)
4	DB2/Unix (no longer supported)
6	Sybase (no longer supported)
7	Microsoft

Strings Table

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Strings Table**. Search for and select a Program Name to access the Strings Table page.

This example illustrates the fields and controls on the Strings Table page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
String Source	Options are: RFT Long: Select if you want the long description of the field to be displayed in the column heading as set in Application Designer. RFT Short: Select if you want the short description of the field as set in the Application Designer to be displayed in the column heading. Text: Select to enter a custom column heading for the report.
String ID	Use the browse button to select the string ID that is to be used for the column heading in the SQR report.
Default Label	The default label is enabled if you select the RFT Long or RFT Short string source, otherwise, the check box is disabled. Remember that fields can have multiple labels. Select the Default Label option to ensure that the default label is used. If you do not use the field's default label, you must select which of the field's labels to use using the label properties button.
String Text	Enter the text for the custom column heading, This is the text that is displayed if you set the string source to Text.
Width	The default value is the current width of the string that you enter or select. Be sure to update the width based on the actual space that is available on the report layout to avoid limiting a translator to an artificially short length, which is likely to degrade the quality of the translation.

Lookup Exclusion

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Auto Lookup Exclusion List** to access the Lookup Exclusion page.

Enter or browse to locate the record name for the tables you want to exclude from the auto lookup feature. A prompt or lookup button opens a lookup page in the user's browser populated with up to 300 available values for that field. The user can then either select the desired value or refine their search further. For extremely large tables, the system administrator has the option of excluding that table from auto prompting by adding the table to this list.

Related Topics

- Application Designer Developer’s Guide: Prompt Fields

TableSet IDs

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Maintain TableSet IDs.** Search for and select an existing Set ID or add a new value to access the TableSet ID page.

Use this utility to create and maintain setIDs. Before doing this:

- Add the SETID field (as a key field) to the record definition for that table.
- Define a Set Control Field as the field controlling the assignment of table sets.

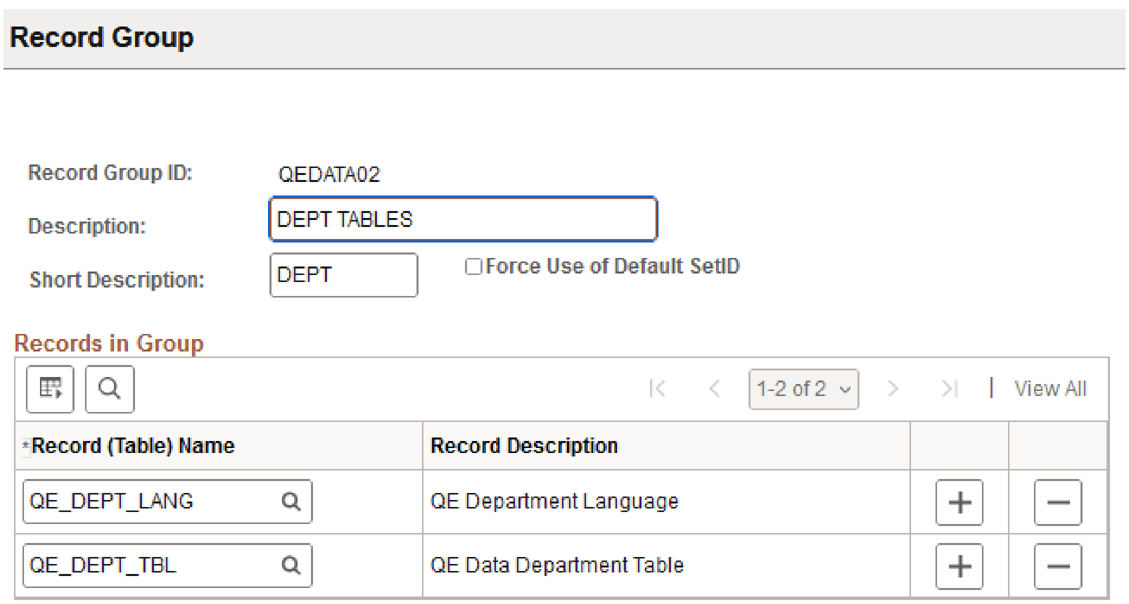
Field or Control	Description
SetID	Enter the setID as defined in the record definition.
Description/Short DescriptionComments	Add any descriptions and comments that are necessary for identification and internal documentation.

Create TableSet Record Groups

Used to group record definitions for the tables that you want to share, as well as any dependent record definitions.

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Create TableSet Record Groups.** Search for and select a Record Group ID or enter a new value to access the Record Group page.

This example illustrates the fields and controls on the Record Group page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
Description	The Record Group ID description should provide enough information to encompass a category of related tables, not just the table that you are specifically sharing.
Short Description	Enter a short description.

Field or Control	Description
Force Use of Default SetID	This overrides alternate setIDs that are entered so that the default is used.
Record (Table) Name	This prompt list comes from a SQL view of record definitions that are defined with that Set Control Field that aren't already associated with a record group.
Record Description	Automatically populated when the Record (Table) Name is selected.

Maintain TableSet Control

The following pages are used to control table sets.

Record Group Page

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Maintain TableSet Control.** Search for and select a Set Control Value to access the Record Group page.

Used to define which record groups use which table set.

Field or Control	Description
Default SetID	This is the setID that the system uses as you add additional record definition groups to be shared within this tableset.
SetID	Although this database is set up to share only one accounting-related record group, you may have multiple record groups to which you assign default unique Set IDs.

Tree Page

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Maintain TableSet Control.** Search for and select a Set Control Value, then select the Tree page.

Used to share Trees as well as tables and views.

Field or Control	Description
Default SetID	The Default setID that you assign to this field value automatically appears. If you create another tableset for sharing trees, you can change this value.
Tree Name	Use the browse button to select from a list of only the tree definitions that are defined with the same Set Control Field.
SetID	Use the browse button to select the appropriate SetID.

Convert Panels to Pages

The following pages are used to convert panels that are used in previous PeopleSoft Windows applications to pages that are used for browser access.

Scope Page

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Convert Panels to Pages.** Search for an existing Run Control ID or add a new value to access the Scope page.

This utility helps you update panels from PeopleTools 7.5x versions to reflect the pages that are used for the PeopleSoft Internet Architecture.

Field or Control	Description
Project List	Insert projects, containing panels that you want to convert, into this scroll. In addition, if you use the Apply Panel Group Defaults option, any panel group that is contained in projects in this scroll are processed. Note that exceptions may be defined see the task titled, Project Exceptions.
Page List	Insert panels that you want to convert to pages into this scroll.
Project Exceptions	If you want to ensure that a group of panels or panel groups is never processed for conversion, you can insert them into an application upgrade project and insert the project name in this scroll.
Page Exceptions	Panels that are inserted into this scroll are not be processed.

See the PeopleSoft upgrade documentation for more information.

Options Page

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Convert Panels to Pages.** Search for an existing Run Control ID or add a new value, then select the Options page.

Specify the options for the conversion process:

Field or Control	Description
Convert Scrolls to Scroll Areas	If you select this option, scroll-to-scroll area conversions take place for panels with scroll bars. If this is unchecked, no scroll-to-scroll area conversion takes place.

Field or Control	Description
Convert Scroll Action Buttons to Scroll Areas	Some scroll bars may exist with scroll action buttons that are already defined. This option determines whether these scrolls should be converted or ignored. If they are converted, the scroll action buttons are removed before the scroll bar is converted to a scroll area. If you select this option, scrolls with scroll action buttons are converted. If this options is not checked, scrolls with scroll action buttons are ignored.
Panels with Level 1 Scrolls	If you select this option, panels with level 1 scrolls are processed for scroll conversion.
Panels with Level 2 Scrolls	If you select this option, panels with level 2 scrolls are processed for scroll conversion.
Panels with Level 3 Scrolls	If you select this option, panels with level 3 scrolls are processed for scroll conversion.
Convert Level 1 Scrolls	If you select this option, level 1 scrolls are converted to scroll areas.
Convert Level 2 Scrolls	If you select this option, level 2 scrolls are converted to scroll areas.
Convert Level 3 Scrolls	If you select this option, level 3 scrolls are converted to scroll areas.
Max # Scrolls	This parameter is a general scroll count limit for scroll conversion processing. For example, if this is set to 5, any panel with more than five scrolls that are not invisible is ignored. This is a simple way of eliminating complex panels from automatic scroll conversion.
Apply Specific Page Size	This option is used to define whether a specific size should be assigned to a panel. If you select this option, the panel size that is defined in the drop-down list box is applied to the panel. If this is unchecked, no changes are made to the panel size. **Note:**When you select a specific panel size, the panel size is applied to standard panels only (secondary panels and subpanels are not sized automatically).
Apply Default Style Sheet	If you select this option, the style sheet that is associated with a panel is updated with a blank value, so that the panel's style sheet appears by default from PSOPTIONS.STYLESHEETNAME ('PSSTYLEDEF').
Apply Frame/Horz/GrpBox Styles	If you select this option, the conversion process looks for frames, group boxes, and horizontal rules that have no styles associated with them, and that appear to be associated with a specific scroll area by virtue of their position within a scroll area. It then assigns level-specific styles, based on the occurs level of the scroll area.

Field or Control	Description
Convert Frames to Horizontal	Horizontal lines were a new page object for PeopleSoft 8. This option applies only for applications upgraded from releases previous to PeopleSoft 8. If you select this option, the conversion process looks for frames on the panel with upper and lower coordinates less than nine grid units apart. These frames are then converted to horizontal lines.
Delete All Frames	If you select this option, the process removes all frames on the converted panel. **Note:**If Convert Frames to Horizontal and Delete All Frames are both checked, the conversion from frame to horizontal takes place first, then any remaining frames are deleted.
Turn On Grid 'Odd/Even Style'	This applies to grids that are on a panel being converted. If you select this option, the conversion process determines if grids on the panel have their 'Odd/Even Style' turned on. If it is not turned on, the conversion process turns on this option.
Turn On 'Show Prompt Button'	This option applies to edit box fields that are not invisible and are not display-only. If you select this option, the conversion process turns on the Show Prompt Button option for edit box fields that have it turned off.
Apply Component Defaults	Used to apply standard defaults to component definitions. The defaults that are set are dependent on the Use characteristics of the component. See Application Designer, Component Properties/Use and Component Properties/Internet tabs.
Turn Off 'Show Grid Lines'	Turns off the Show Grid Lines option for grids that have it checked on.
Language Code	Enables you to convert panels whose language code differs from that in PSOPTIONS. Select a language code from the drop-down list box.

Update Utilities

The Update utilities enable you to keep track of the PeopleSoft updates that you apply to the database.

Updates - View All

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **View All Updates** to access the Updates - View All page.

Select one of the options, to Show All Updates, or Show PS Updates Only.

Updates By Release Label

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **View Updates By Release Label** and search for a release to access the Updates by Release Label page.

The release label refers to the official release name, such as PeopleTools 8.58.00.

Updates By Update ID

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **View Updates By Update ID** and search for an Update ID to access the View Updates By Update ID page.

The update ID refers to the patch or project name that you apply to the system. The update ID is typically an incident ID or change package ID.

See Change Assistant and Update Manager: Reviewing the Updates Change Log

Remote Database Connections

Use the Remote Database Connection page to set up remote databases for use with the Remote Data Access (RDA) feature.

Defining remote database connections is documented in *PeopleTools: Database Management*. See Data Management: Understanding Remote Data Access and Data Management: Defining Remote Database Connections.

URL Maintenance

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Maintain URLs.** Search for an existing URL identifier or add a new value to access the URL Maintenance page.

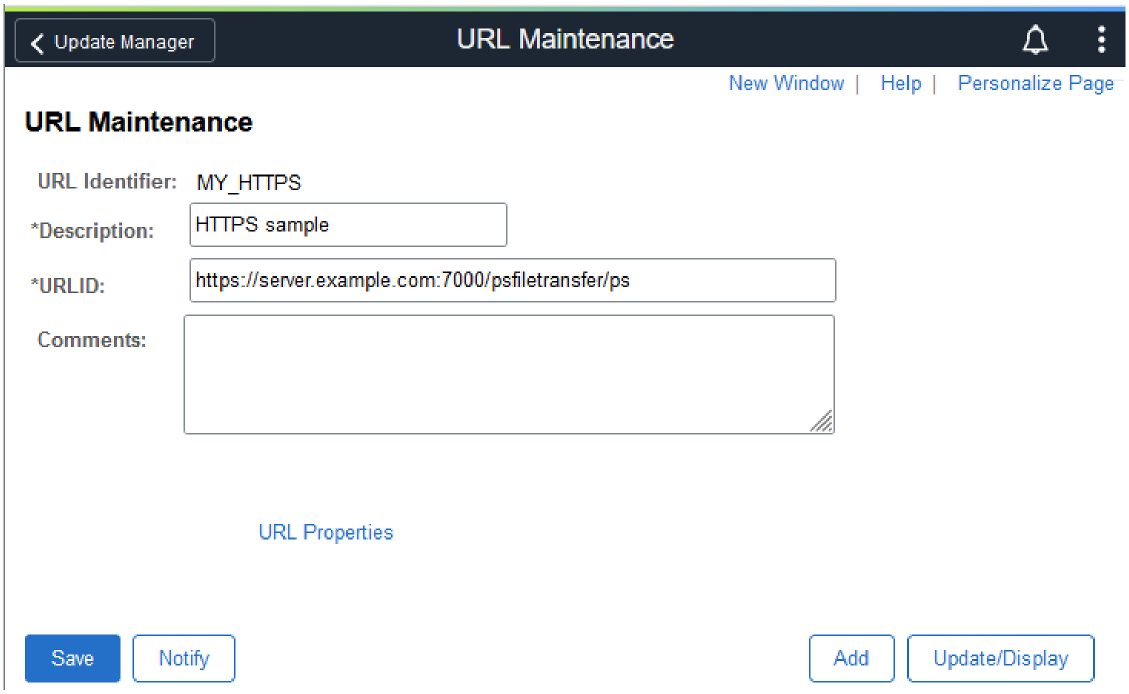
Use the URL Maintenance page to store URL addresses and to simplify specifying and updating URLs. URLs that are saved here can be referenced from page controls such as a push button or link. The associated URL can be either an internet or intranet link. Use one of these protocols:

- FTP
- FTPS
- SFTP
- HTTP
- HTTPS

The documents for file attachments can be stored in repositories based on:

- FTP
- FTPS
- SFTP
- HTTP
- HTTPS
- database tables (PeopleSoft records)
- Oracle Content and Experience Cloud (OCE)
- Oracle Cloud Infrastructure Object Storage

This example illustrates the fields and controls on the URL Maintenance page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
Description	Users can search for URLs by description.
URLID	Enter the entire URL. **Note:**For the file attachment functionality, in specifying the URL for the FTP server, the FTP server's machine name can be more than 30 characters, but the length of the full URL is limited to 120 characters. The URL format for file attachments stored in the database is *record://recordname*, as in *record:// ABSENCE_HIST_DB*.
Comments	This field can be used to make notations and comments and is not displayed elsewhere.
URL Properties	Assign properties to various URL types. See Specifying URL Properties.

Adding New URLs

To add a new URL entry:

1. Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Maintain URLs.**
2. Click **Add a New Value**.

   Enter the name that you want to use to identify the new URL address.
3. Select Add.
4. Enter the Description, URL, and Comment, if any.
5. Select URL Properties and specify the required properties.

   See the following sections for details.

6. Select Save.

   You must save the page before you can add another URL, or update or display existing URL addresses.
7. Select Add to add another URL.

Viewing and Updating Existing URLs

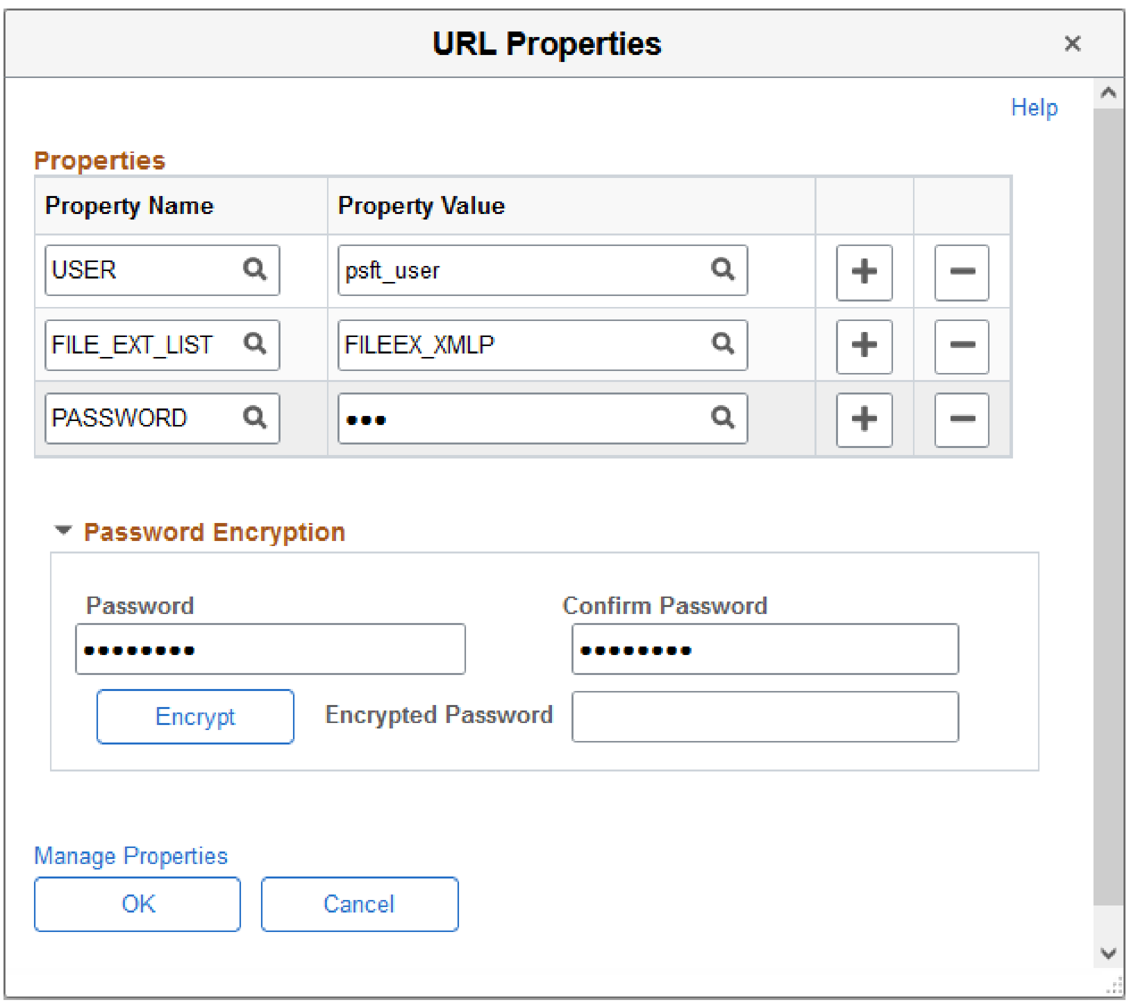
To update or display existing URLs:

1. From the URL Maintenance search page, click Search.
2. Select the URL Identifier link that you want to update from the Search Results.
3. Make changes to the page and save.

Specifying URL Properties

You access the URL Properties page by clicking the URL Properties link on the URL Maintenance page.

This example illustrates the fields and controls on the URL Properties page. You can find definitions for the fields and controls later on this page.



You create the URLID using the URL Maintenance page, but you assign various properties to the URLID using the URL Properties page. The properties you assign depend on the protocol and the type of security you are implementing.

The Password Encryption box enables you to encrypt a password value that you intend to assign to the PASSWORD property. Click Encrypt to encrypt the password, then copy the encrypted password into the edit box for the PASSWORD property. The value of this property must be the user password in an encrypted form.

Select the Manage Properties link if you need to add any LibCurl properties that are not PeopleSoft-delivered.

See Maintain URL Properties.

> **Note**
>
> If you intend to implement secure FTP, SFTP, HTTP, or HTTPS, Oracle recommends you understand the details and behavior of the attachment PeopleCode constructs, such as AddAttachment.

See PeopleCode Developer's Guide: Understanding the File Attachment Functions.

Specifying FTP/FTPS URLs

FTP over SSL (FTPS) can be achieved in two ways: implicit and explicit. Implicit SSL uses ftps:// and explicit uses ftp://.

To use passive mode (PASV) only, set URL property EXTENDEDPASSIVEMODE to *0*.

To assign properties to FTP and FTPS URLs:

1. On the URL Maintenance page, add a URL value containing *ftp://* or *ftps://* in the URLID field.
2. Click Save.
3. Click the URL Properties link.
4. Set the URL properties for the protocol and your security preference.
5. Select the Manage Properties link if you need to add any LibCurl properties that are not PeopleSoft-delivered.
6. Click Save.

The URL properties for the FTP and FTPS protocols are listed in the following table:

Properties	Description
ACTIVEMODE	To enable active mode, add the ACTIVEMODE property to the URL and set it to *Y*. The default FTP connection mode is extended passive mode.

Properties	Description
ACTIVEPORTOPTION	This property can be used along with ACTIVEMODE. When active mode is enabled, you can use ACTIVEPORTOPTION to specify the IP address and port on which the FTP server can be accessed. This is useful when the server is behind a firewall. By default, ACTIVEPORTOPTION uses the default IP address of your system. If you want to use a particular IP address, set the ACTIVEPORTOPTION value to either the full IP address, a host name to resolve to an IP address, or a local network interface name. You can also specify a port range. For example: *198.51.100.111:10000-13000*
CERTALIAS	(Certificate Alias) The Certificate Alias must be an alias name of a certificate stored in the database (using the PeopleTools Digital Certificates page). **Note:**Currently, only PEM certificates are supported for FTPS.
ENABLEEPRT	This option can be used only with Active Mode. If Active Mode is enabled and ENABLEEPRT is set to *N*, then the system will use a PORT (IPv4) Active Mode connection. By default, ENABLEEPRT is Y, if Active Mode is set to Y.
EXTENDEDPASSIVEMODE	Enables you to control whether extended passive mode (EPSV) will be used by FTP. To enable select *1*, to disable select *0*. EPSV is used by default. That is, by default, this value is considered to be 1. If the client fails to connect to the server with EPSV, then the system will try passive mode (PASV). To use PASV only, add EXTENDEDPASSIVEMODE to the URL Properties and set it to 0.
FILE_EXT_LIST	Select the name of a file extension list created using the File Extension List page. A file extension list enables you to restrict file attachment features to work only with appropriate file types, and reject those file types that you believe may be suspicious, potentially harmful, or not appropriate for a particular call to a file-processing built-in PeopleCode function. See File Extension List.
JKSPASSWORD	Enter the password to access and maintain certificates in the JKS keystore/truststore.
JKSPATH	Enter the path where the JKS keystore/truststore file is maintained.

Properties	Description
KEYSTOREPASSWORD	Required for FTPS and HTTPS repositories. For attachments transferred from the PeopleSoft system to the FTPS or HTTPS repository, the system retrieves the key pair for the client certificate from the digital certificate store and writes them to a file in PKCS12 format with password protection. The value of this property will be used as the password for the PKCS12 file. The PKCS12 file enables connection and file transfer, and it exists only temporarily in <*PS_SERVDIR*>\files\<*CERT ALIAS NAME*> for the duration of the file transfer. The system deletes the file after the file transfer transaction. **Note:**If the system fails to delete the certificate alias file, a message will be written to the application sever log. The maximum number of files that can exist at any time is equal to the total number of FTPS and HTTPS URL identifiers defined in the system.
PASSWORD	The password associated with the USER property, which identifies the FTP user ID. The value of this property must be the user password in an encrypted form.
SSLUSAGELEVEL	*0 - No SSL*: No SSL will be used. *1 - Try SSL*: Try using SSL, but proceed as normal otherwise. *2 - Control*: Require SSL for the control connection. *3 - SSL Only*: (Default) Require SSL for all communication.
USER	The FTP user ID used for authentication when accessing the FTP site.
VERIFYHOST	*0*: Do not verify the server for host name. *1*: (Default) Checks for a match with the host name in the URL with the common name or Subject Alternate field in the server certificate.
VERIFYPEER	*False*: Do not verify the peer. *True*: (Default) Verify the peer by authenticating the certificate sent by the server.

Specifying SFTP URLs

With SFTP, these authentication methods are supported:

- password.
- key authentication.

> **Note**
>
> With key authentication, you have two choices: storing the public and private keys as files within the file system, or storing the public and private keys within an SSH certificate added using the Digital Certificates page.

To set up password authentication create a URL identifier with an SFTP URL value, such as sftp://198.51.100.122/, then set these URL properties:

- AUTHTYPE: *2 - PASSWORD*
- USER: *user ID*
- PASSWORD: *encrypted user password*

For public key authentication *with key files stored in the file system,* you can copy the public key file to <PS_SERVDIR>\sshkeys\public and the private key file to <PS_SERVDIR>\sshkeys\private. Alternatively, you can copy the key files to a different location and define an environment variable PS_SSHKEYPATH.

> **Note**
>
> PS_SSHKEYPATH can replace only PS_SERVDIR. The following directory structure must exist in your custom location, .\sshkeys\public and .\sshkeys\private. For example, <PS_SSHKEYPATH>\sshkeys\public.

If storing the SSH key data *in the file system,* set these URL properties appropriately:

- AUTHTYPE: (*1 - PUBLIC KEY*)
- USER
- PRIVATEKEY
- PUBLICKEY
- PASSWORDKEY

For public key authentication *with an SSH certificate stored in the PeopleTools keystore,* save the private key and public key data to the database using the Digital Certificates page (PeopleTools, Security, Security Objects, Digital Certificates page). If your SSH certificate is stored *in the database* by way of the Digital Certificates page, set these URL properties appropriately:

- AUTHTYPE: (*1 - PUBLIC KEY*)
- USER
- SSHKEYALIAS
- PASSWORDKEY

> **Note**
>
> Oracle recommends storing the SSH public and private key data in the database by creating an SSH certificate stored in the PeopleTools keystore for increased security and ease of maintenance. If the private key and public key files are stored in the file system, Oracle recommends implementing a password-protected private key.

For more information on the Digital Certificates page, see Security Administration: Configuring Digital Certificates

To assign properties to SFTP URLs:

1. On the URL Maintenance page, add a URL value containing *sftp://* in the URLID field.
2. Click Save.
3. Click the URL Properties link.
4. Set the URL properties for the protocol and your security preference.
5. Select the Manage Properties link if you need to add any LibCurl properties that are not PeopleSoft-delivered.
6. Click Save.

The URL properties for the SFTP protocol are listed in the following table:

Properties	Description
AUTHTYPE	Select the authentication type: • Public Key • Password • Any
FILE_EXT_LIST	Select the name of a file extension list created using the File Extension List page. A file extension list enables you to restrict file attachment features to work only with appropriate file types, and reject those file types that you believe may be suspicious, potentially harmful, or not appropriate for a particular call to a file-processing built-in PeopleCode function. See File Extension List.
PASSWORD	The password associated with the USER property, which identifies the FTP user ID. The value of this property must be the user password in an encrypted form.
PASSWORDKEY	Enter the password for the private key.
PRIVATEKEY	Select the private key.
PUBLICKEY	Select the public key.

Properties	Description
SSHKEYALIAS	Select the SSH certificate saved to the database using the PeopleTools Security, Digital Certificates page (PeopleTools, Security, Security Objects, Digital Certificates). The SSH certificate added through the Digital Certificates page contains both the public and private key data, identified by the Alias column value on the Digital Certificates page. If using the SSHKEYALIAS URL property, the Property Value prompt displays only the list of SSH certificates that have been added to the Digital Certificates page. If you have added the SSH certificate using the Digital Certificates page, and you have assigned an SSH certificate to the SSHKEYALIAS URL property, the system ignores the PUBLICKEY and PRIVATEKEY properties, regardless if they refer to valid key files in the file system. If you provided a password (or passphrase) when generating your SSH certificate, specify that value using the PASSWORDKEY URL property. See Security Administration: Configuring Digital Certificates
USER	Specify the user ID to be authenticated.

Specifying Properties for HTTP URLs

To assign properties to HTTP URLs:

1. On the URL Maintenance page, add a URL value containing *http://* in the URLID field.
2. Click Save.
3. Click the URL Properties link.
4. Set the URL properties for the protocol and your security preference.
5. Select the Manage Properties link if you need to add any LibCurl properties that are not PeopleSoft-delivered.
6. Click Save.

The available URL properties for the HTTP protocol are listed in the following table:

Properties	Description
CLOUDPROVIDER	If CLOUDREPOSITORY is set to Y, select one of the available cloud service providers.
CLOUDREPOSITORY	Enter Y if the URL is a cloud repository, and N if it is not.

Properties	Description
FILE_EXT_LIST	Select the name of a file extension list created using the File Extension List page. A file extension list enables you to restrict file attachment features to work only with appropriate file types, and reject those file types that you believe may be suspicious, potentially harmful, or not appropriate for a particular call to a file-processing built-in PeopleCode function. See File Extension List.
FOLDERPATH	If CLOUDREPOSITORY is set to Y, enter the location in cloud storage where you want the report to be stored. The folder will be created if it does not exist.
GENERATEACCESSKEY	If CLOUDREPOSITORY is set to Y: • Set to Y if you want the ShareAttachment API to generate an access key and set it back, if an access key is not passed in to the API. • Set to N if you don't want the ShareAttachment API to generate an access key, when an access key is not passed in to the API.
PASSWORD	The password associated with the USER property, which identifies the user ID. The value of this property must be the user password in an encrypted form. If CLOUDREPOSITORY is set to Y, enter the encrypted password for the cloud account where the reports will be stored.
PROXY	Specify the proxy server (if used).
PROXYPORT	Specify the proxy server port (if used)
USEAUTHTOKEN	Uses the PS_TOKEN authentication cookie, if single sign-on is configured between source and target destination. **Note:**This must be set to *Y* if a PeopleSoft web server is used for the HTTP repository.
USER	Specify the user ID to be authenticated. **Note:**USER must be set if a PeopleSoft web server is used for the HTTP repository. If CLOUDREPOSITORY is set to Y, enter the user name for the cloud account where the reports will be stored.

Specifying Properties for HTTPS URLs

To assign properties to HTTPS URLs:

1. On the URL Maintenance page, add a URL value containing *https://* in the URLID field.
2. Click Save.
3. Click the URL Properties link.

4. Set the URL properties for the protocol and your security preference.
5. Select the Manage Properties link if you need to add any LibCurl properties that are not PeopleSoft-delivered.
6. Click Save.

The available URL properties for the HTTPS protocol are listed in the following table:

Properties	Description
ACCESSTOKEN	Enter a password that you encrypted using the Password Encryption utility on the URL Properties page.
ACCOUNTID	Enter your Twilio account ID Use this property for push notifications. See Fluid User Interface Developer's Guide: Managing Global Settings
BUCKETNAME	If you are using Oracle Cloud Infrastructure as a file repository, enter the name of the bucket you created in the Object Storage service. See Specifying URLs for Oracle Cloud Infrastructure.
CERTALIAS	(Certificate Alias) The Certificate Alias must be an alias name of a certificate stored in the database (using the PeopleTools Digital Certificates page). **Note:**Currently, only PEM certificates are supported for FTPS.
CLIENTID	Use when defining URL properties for SharePoint. See Defining SharePoint Repositories
CLIENTSECRET	Use when defining URL properties for SharePoint. See Defining SharePoint Repositories
CLOUDPROVIDER	If CLOUDREPOSITORY is set to Y, select one of the available cloud service providers.
CLOUDREPOSITORY	Enter Y if the URL is a cloud repository, and N if it is not.
DOCUMENTLIBRARY	Use when defining URL properties for SharePoint. See Defining SharePoint Repositories
FILE_EXT_LIST	Select the name of a file extension list created using the File Extension List page. A file extension list enables you to restrict file attachment features to work only with appropriate file types, and reject those file types that you believe may be suspicious, potentially harmful, or not appropriate for a particular call to a file-processing built-in PeopleCode function. See File Extension List.

Properties	Description
FINGERPRINT	If you are using Oracle Cloud Infrastructure as a file repository, enter the fingerprint for the API signing key. See Specifying URLs for Oracle Cloud Infrastructure.
FOLDERPATH	If CLOUDREPOSITORY is set to Y, enter the location in cloud storage where you want the report to be stored. The folder will be created if it does not exist.
GENERATEACCESSKEY	If CLOUDREPOSITORY is set to Y: • Set to Y if you want the ShareAttachment API to generate an access key and set it back if an access key is not passed in to the API. • Set to N if you don't want the ShareAttachment API to generate an access key, when an access key is not passed in to the API.
KEYSTOREPASSWORD	Required for FTPS and HTTPS repositories. For attachments transferred from the PeopleSoft system to the FTPS or HTTPS repository, the system retrieves the key pair for the client certificate from the digital certificate store and writes them to a file in PKCS12 format with password protection. The value of this property will be used as the password for the PKCS12 file. The PKCS12 file enables connection and file transfer, and it exists only temporarily in <PS_SERVDIR>\files\<CERT ALIAS NAME> for the duration of the file transfer. The system deletes the file after the file transfer transaction. **Note:** If the system fails to delete the certificate alias file, a message will be written to the application sever log. The maximum number of files that can exist at any time is equal to the total number of FTPS and HTTPS URL identifiers defined in the system.
KEYPASSPHRASE	If you are using Oracle Cloud Infrastructure as a file repository, enter the encrypted passphrase for the API signing key pair. See Specifying URLs for Oracle Cloud Infrastructure.
NAMESPACENAME	If you are using Oracle Cloud Infrastructure as a file repository, enter the namespace that you created in Object Storage service. See Specifying URLs for Oracle Cloud Infrastructure.

Properties	Description
PASSWORD	The password associated with the USER property, which identifies the user ID. The value of this property must be the user password in an encrypted form. If CLOUDREPOSITORY is set to Y, enter the encrypted password for the cloud account where the reports will be stored.
PRIVATEKEYPATH	If you are using Oracle Cloud Infrastructure as a file repository, enter the full path to the location where you saved the API signing key pair. See Specifying URLs for Oracle Cloud Infrastructure.
PROXY	Specify the proxy server (if used).
PROXYPORT	Specify the proxy server port (if used).
SHAREPOINTREPOSITORY	Use when defining URL properties for SharePoint. See Defining SharePoint Repositories
SHAREPOINTSITE	Use when defining URL properties for SharePoint. See Defining SharePoint Repositories
SSLUSAGELEVEL	*0 - No SSL*: No SSL will be used. *1 - Try SSL*: Try using SSL, but proceed as normal otherwise. *2 - Control*: Require SSL for the control connection. *3 - SSL Only*: (Default) Require SSL for all communication.
TENANCYOCID	If you are using Oracle Cloud Infrastructure as a file repository, enter the OCID for your tenancy.
TENANTID	Use when defining URL properties for SharePoint. See Defining SharePoint Repositories
TEXTFROM	Enter the phone number associated with your Twilio account. Use this property for push notifications. See Fluid User Interface Developer's Guide: Managing Global Settings
USEAUTHTOKEN	Uses the PS_TOKEN authentication cookie, if single sign-on is configured between source and target destination. **Note:**This must be set to *Y* if a PeopleSoft web server is used for the HTTP repository.
USER	Specify the user ID to be authenticated. **Note:**USER must be set if a PeopleSoft web server is used for the HTTP repository. If CLOUDREPOSITORY is set to Y, enter the user name for the cloud account where the reports will be stored.

Properties	Description
USEROCID	If you are using Oracle Cloud Infrastructure, enter the user OCID for your account.
VERIFYHOST	*0*: Do not verify the server for host name. *1*: (Default) Checks for a match with the host name in the URL with the common name or Subject Alternate field in the server certificate.
VERIFYPEER	*False*: Do not verify the peer. *True*: (Default) Verify the peer by authenticating the certificate sent by the server.

Specifying Properties for URLs for Database Records

To assign properties to URLs for files in the database:

1. On the URL Maintenance page, add a URL value containing *record://* in the URLID field.
2. Click Save.
3. Click the URL Properties link.
4. Set the URL properties for the protocol and your security preference.
5. Click Save.

The available URL properties for the record protocol are listed in the following table:

Property	Description
FILE_EXT_LIST	Select the name of a file extension list created using the File Extension List page. A file extension list enables you to restrict file attachment features to work only with appropriate file types, and reject those file types that you believe may be suspicious, potentially harmful, or not appropriate for a particular call to a file-processing built-in PeopleCode function. See File Extension List.
IMG_DATA_COLUMN	Enter information about the database record. For information on this property, see PeopleCode Language Reference: ResizeImage function.
IMG_FILE_FORMAT	Enter information about the database record. For information on this property, see PeopleCode Language Reference: ResizeImage function.
IMG_NAME_COLUMN	Enter information about the database record. For information on this property, see PeopleCode Language Reference: ResizeImage function.

Using HTTP/HTTPS File Transfers

HTTP/HTTPS file transfer requires that the http/https repository is available (file storage location). The repository can either be on a PeopleSoft web server or non-PeopleSoft server. The PeopleSoft Internet Architecture provides a handler servlet (psfiletranfer) to handle the requests. If you use a non-PeopleSoft web server you need to create a handler for non-PeopleSoft servers/repositories.

Once the repository is created, the configuration.properties file for the appropriate PeopleSoft site needs to be modified to include the HttpRepository property. For example:

HttpRepositoryPath=D:/HttpRepository

> **Note**
>
> Use a forward slash (/).

To set up HTTP:

1. Create a URL identifier using *http://* (as in, http://<PIA URL>/psfiletransfer/<sitename>/).
2. Select these URL properties:
   - USEAUTHTOKEN: *Y*
   - USER: *user_ID*
3. Set the Authentication Option field on the default local node to *Password*.

To set up HTTPS:

1. Set up the target web server for HTTPS.
2. Add the Root CA and Client Certificate to Digital Certificate store.
3. Create an URL identifier using *https://*.
4. Select these URL properties:
   - CERTALIAS: *your certificate alias*
   - SSLUSAGELEVEL: *3-SSL Only*
   - VERIFYHOST: *1*
   - VERIFYPEER: *Y*
   - USER: *user ID*
   - USEAUTHTOKEN: *Y*

You can have the HTTP repository on a separate PeopleSoft web server too. For example, PIAsystem1 can have the URL identifier pointing to a URL on PIAsystem2. This requires the following to be in place:

- The user specified in the URL properties must exist in both the systems with the proper permissions.
- The default local nodes on both systems should have Authentication Option set to *Password* and should be pinged successfully.
- The default local nodes of each system are added to the single-sign-on list for each database.

> **Note**
>
> To upload non-ASCII file names to an HTTP Repository, the application server needs to be booted in the native language (such as Japanese or Chinese). If the application server is booted in English, file names with non-ASCII characters will not be uploaded to the HTTP Repository.

See Global Technology: Understanding Character Set Selection for more information on how character sets on various tiers of the PeopleSoft system can affect file attachment processing.

Specifying URLs When Attaching Files

When attaching files, whether you can pass a string URL or only use the URL identifier depends on the protocol and repository type being used. When attaching files, keep these guidelines in mind:

- Using the URL Identifier is mandatory for FTPS, SFTP, HTTP, and HTTPS.
- Using a URL Identifier is also recommended for FTP.
- For Record and File attachment protocols, you can use the URL Identifier or use a URL string (and not reference the stored URL Identifier).

For example, with plain FTP you could use a URL string, such as:

ftp://user:password@site/

Or, if a URL Identifier has been created, which is *ftpurl* in these examples, you can reference that in one of these formats:

- *ftp://userid:password@ftpurl/*: where the FTP user and FTP password are provided in clear text, but the URL information comes from the URL Identifier.
- *ftp://userid@ftpurl/*: where the FTP user is provided in clear text, but the FTP password and the URL information comes from the URL Identifier.

  For this format, provide the rest of the properties (password) by accessing the URL Properties link on the URL Maintenance page.
- *ftp://ftpurl/*: where all the required information comes from the URL Identifier.

  For this format, provide the rest of the properties (including user and password) by accessing the URL Properties link on the URL Maintenance page.

> **Note**
>
> Follow the requirements for the URLID at the beginning of this section.

Specifying URLs for Oracle Content and Experience Cloud Repository

You can use Oracle Content and Experience Cloud as a file repository for File Attachments and Process Scheduler Report Distribution. To define a URL ID for Oracle Content and Experience Cloud as a file repository:

1. Create a new HTTP or HTTPS URL ID, with the following format.

In this example, <hostname> is the server where the Oracle Content and Experience Cloud storage instance is running, and [port] is the Oracle Content and Experience Cloud port (the port designation is optional).

http://*<hostname>*:*[port]*/documents

or

https://*<hostname>*:*[port]*/documents

2. Select URL Properties, and enter values for these properties, using the definitions in the HTTP or HTTPS table earlier in this topic. For example:
   - CLOUDREPOSITORY: *Y*
   - CLOUDPROVIDER: *Oracle Document Cloud*
   - FOLDERPATH: */psftfolder/reports*
   - USER: *user ID*
   - PASSWORD: *encrypted password value*
   - GENERATEACCESSKEY (optional): *N*
3. Add the following properties for SSL if necessary, using the definitions in the HTTP or HTTPS table earlier in this topic. For example:
   - CERTALIAS: *your certificate alias*
   - SSLUSAGELEVEL: *3 - SSL Only*
   - VERIFYHOST: *1*
   - VERIFYPEER: *Y*

Use this URL with the file attachment APIs to enable the PeopleSoft system to upload attachments from the Oracle Content and Experience Cloud repository. You can also share the uploaded files with other users.

See PeopleCode Language Reference: AddAttachment function

See PeopleCode Language Reference: ShareAttachment function

Use this URL when you define a report node that allows reports to be uploaded to Oracle Content and Experience Cloud.

See Process Scheduler: Defining Report Nodes

For additional information on working with Oracle Content and Experience Cloud repositories, see Using Cloud Repositories.

Specifying URLs for Oracle Cloud Infrastructure Object Storage

You can use Oracle Cloud Infrastructure Object Storage as a file repository for File Attachments and Process Scheduler Report Distribution. This process requires that you have an account for Oracle Cloud Infrastructure Object Storage service. You enter account details in specifying the URL.

For information on getting an Oracle Cloud Infrastructure account, see Welcome to Oracle Cloud Infrastructure in the Oracle Cloud Infrastructure documentation.

To define a URL ID for Oracle Cloud Infrastructure Object Storage as a file repository.

1. Get the tenancy OCID and user OCID for your Oracle Cloud Infrastructure account.

   See Required Keys and OCIDs in the Oracle Cloud Infrastructure online documentation.

2. Generate an API signing key pair and get the fingerprint of the public API signing key for your Oracle Cloud Infrastructure account.

   See Required Keys and OCIDs in the Oracle Cloud Infrastructure online documentation.

   If you used a passphrase in creating the API signing key pair, as recommended, get the passphrase.

3. Create a bucket and namespace for the files in Oracle Cloud Infrastructure Object Storage.

   See Overview of Object Storage in the Oracle Cloud Infrastructure documentation.

4. In your PeopleSoft environment, install an application server-based digital certificate.

   See Security Administration: Installing Application Server-Based Digital Certificates

5. In your PeopleSoft environment, create a new URL using *https://*.

6. Enter the Object Service Storage API endpoint in the URLID field on the URL Maintenance page.

   The Object Service Storage API endpoints are available in the Oracle Cloud Infrastructure online documentation.

   See API Reference and Endpoints.

7. Select URL Properties, and enter values for these properties:

   - CLOUDREPOSITORY: *Y*
   - CLOUDPROVIDER: *OCI Object Storage*
   - BUCKETNAME: The name of the bucket you created in Object Storage.
   - TENANCYOCID: The OCID for your Oracle Cloud Infrastructure tenancy.
   - USEROCID: The OCID of the user calling the API.
   - FINGERPRINT: The fingerprint for the API key pair being used.
   - PRIVATEKEYPATH: The full path to the location where you saved the API signing key pair.
   - KEYPASSPHRASE: The encrypted passphrase for the API signing key pair.

     Use the Password Encryption fields to encrypt the passphrase. See Specifying URL Properties in this section.

8. Add the following properties for SSL if necessary, using the definitions in the HTTP or HTTPS table earlier in this topic. For example:

   - SSLUSAGELEVEL: *3 - SSL Only*
   - VERIFYHOST: *1*
   - VERIFYPEER: *Y*

Use this URL with the file attachment APIs to enable the PeopleSoft system to upload attachments from the Object Storage repository.

See PeopleCode Language Reference: AddAttachment function

Use this URL when you define a report node that allows reports to be uploaded to Object Storage.

See Process Scheduler: Defining Report Nodes

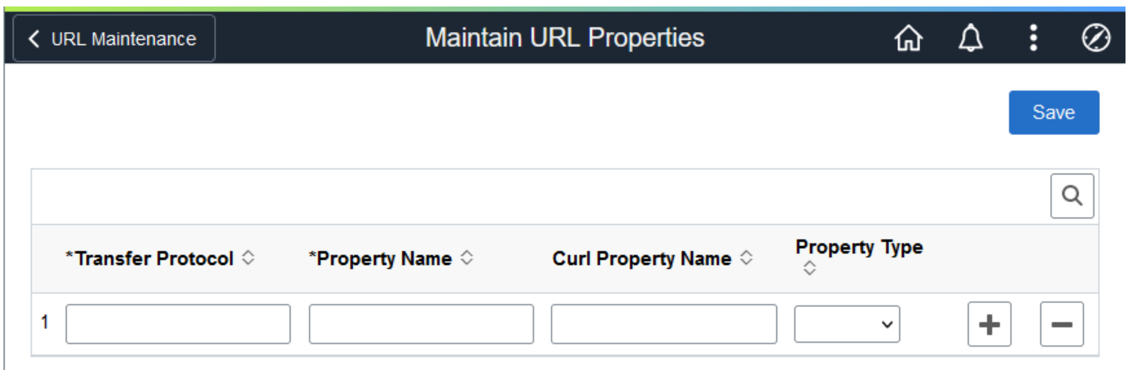
Maintain URL Properties

PeopleSoft delivers all the necessary properties for use with URL identifiers. File transfers using FTP, SFTP, HTTP, and HTTPS protocols use a third-party library LibCurl. If you need to use non-delivered LibCurl properties, use this page to define them. After defining the property, use the URL Maintenance page to associate the property with the URL.

Select **PeopleTools**, and then **Utilities** , and then **Administration** , and then **Administer File Processing** to access the Maintain URL Properties page.

You can also access the page from the URL Properties dialog box.

This example illustrates the fields and controls on the Maintain URL Properties page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
Transfer Protocol	Select the protocol to use for file transfer: • FTP • SFTP • HTTP • HTTPS
Property Name	Enter a name to be used in the URL Maintenance page. After you save this page, the property will be available when you create or edit a URL.
Curl Property Name	Enter a curl property to be associated with the Property Name. The values that you enter will be passed directly to curl. See the curl documentation for allowed values. See https://curl.se/libcurl/c/curl_easy_setopt.html Only the properties of the Integer, String, and Boolean types are supported, with some limitation. Any property (such as enums defined as values, or a property like CURLOPT_WILDCARDMATCH) that requires special handling is not supported.

Field or Control	Description
Property Type	Select one of these types from the drop-down list: • Integer • String • Boolean: Enter Y (yes) or N (no). The values that you enter are passed directly to curl. See the curl documentation for allowed values.

To make curl properties available for URL definitions:

1. Select **PeopleTools**, and then **Utilities** , and then **Administration** , and then **Administer File Processing**, and then **Maintain URL Properties**.
2. Enter one of the supported protocols for file transfer.
3. Enter a Property Name to be associated with the curl property.
4. Enter a Curl Property Name.
5. Select a Property Type.
6. Click the plus sign to add additional properties as necessary.
7. Save your changes.
8. Select **PeopleTools**, and then **Utilities** , and then **Administration** , and then **Maintain URLs**.
9. Add a URL identifier or open an existing URL.
10. Enter the necessary information on the URL Maintenance page.
11. Select URL Properties.
12. Click the lookup icon (magnifying glass) and locate the property name that you associated with the curl property.
13. Enter a value that is allowed for the curl property.

    See the curl documentation for allowed values.

Query Administration

System administrators can use Query Administration to monitor query performance and usage. Some of the conditions that you can monitor include average runtime, number of times run, and the dates last run. Using a predefined search, you can also select queries to review and report on.

Related Topics

- Query: Understanding Query Administration

Analytic Server Administration

See Analytic Calculation Engine: Administering Analytic Servers.

Manage Upgrade Driver Programs

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Manage Upgrade Driver Programs** to access the PT Upgrade Drivers page.

Search for your PeopleTools upgrade path. The PT Upgrade Drivers page lists upgrade drivers, providing details regarding Application Engine program, section, group, and calling sequence. If you need to specify any of these values, your upgrade documentation will provide the details.

See your upgrade documentation for more information.

Analytic Model Viewer

See Analytic Calculation Engine: Viewing Analytic Model Properties.

Load/Unload Analytic Instances

See Analytic Calculation Engine: Loading and Unloading Analytic Instances.

Maintain Analytic Instances

See Analytic Calculation Engine: Creating, Deleting, and Copying Analytic Instances.

QAS Administration

See Reporting Web Services: Using QAS Administration.

Chart Options

See PeopleCode API Reference: Accessibility Settings for Charts.

Manage Oracle Resource Groups

See Data Management: Working With Oracle Consumer Groups.

File Extension List

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Administer File Processing**, and then **Manage File Extensions.** Search for an existing Extension List Name or add a new value to access the File Extension List page.

A file extension list contains a list of file types, identified by file extension, that the system either accepts or rejects during specific file attachment operations. Once you have created a file extension list you specify it as the value for the FILE_EXT_LIST property of a URL object using the URL Maintenance page. This enables you to only upload the appropriate file types at runtime, and reject those file types you think may be suspicious, potentially harmful, or not appropriate for a particular call to a file-processing built-in PeopleCode function.

A file extension list applies only when performing these file attachment operations:

- Uploading files (AddAttachment, MAddAtachment PeopleCode)
- Viewing files (ViewAttachment PeopleCode)

- Detaching files (DetachAttachment PeopleCode)

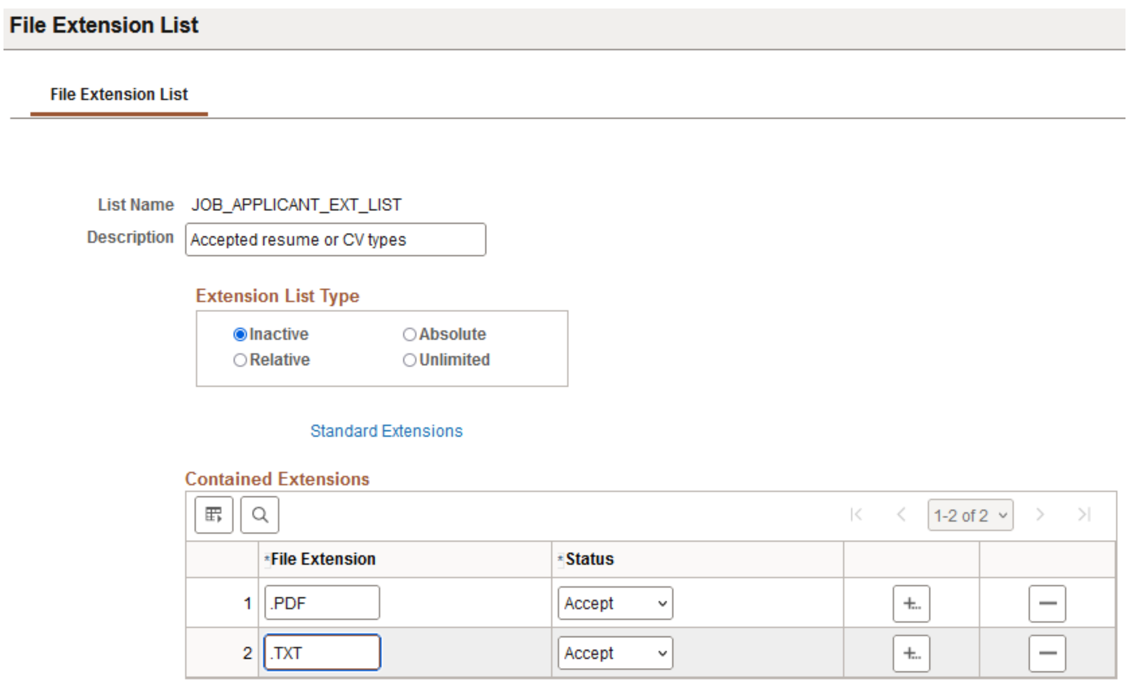
> **Note**
>
> File extension lists can be assigned only to a URL object by way of the FILE_EXT_LIST property, and they apply only to these file attachment operations.

See URL Maintenance.

See PeopleCode Developer's Guide: Understanding the File Attachment Functions.

You can create different file extension lists to restrict certain file types, depending on the needs of the application. For example, for an application that users access from outside your firewall, such as a job applicant interface, you might want to impose strict restrictions on the accepted file types to avoid allowing potentially harmful file types into your system. For an internal-facing application, you may not require such strict requirements for file attachments.

This example illustrates the fields and controls on the File Extension List page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
List Name	Displays the name you entered for the file extension list in the Add New Value tab. File extension list names cannot exceed 30 characters.
Description	Enter any descriptive information to help indicate how this list is to be used.

Extension List Type

Field or Control	Description
Inactive	When set to *Inactive*, the system ignores the file extension list. Any file extension entries in the Contained Extensions grid are ignored at runtime for an Inactive list type. From a development perspective, it may be useful to keep a working list of file extensions, but keep the list itself inactive until it is fully tested. **Note:**When troubleshooting or testing, setting a file extension list to *Inactive*, immediately disables any restrictions imposed by that file extension list.
Absolute	With an *Absolute* file extension list, you can create a list in which all the acceptable filename extensions are explicitly enumerated and have a status of *Accept*. That is, *only* the file extensions that appear in the Contained Extensions grid *and* have the Status set to *Accept* will be accepted for that file attachment operation at runtime. An Absolute file extension list is most appropriate for situations where you wish to accept a relatively small number of file types. Any file extension set to *Reject* will not be accepted at runtime. However, *not* including such a file extension in your absolute list will have precisely the same effect.

Field or Control	Description
Relative	The Relative file extension list type is based on the Standard Extensions list, and provides a flexible means to specify a finite number of files that are acceptable, as well as enabling you to extend the Standard Extensions list, if needed. A Relative file extension list is most appropriate for situations where you wish to reject a relatively small number of file types. After reviewing the Standard Extensions list, add specific file extensions that you *do not* want to authorize for file attachment operations to the Contained Extensions grid, and set the Status value to *Reject*. By doing so, you indicate to the system that you authorize all of the extensions on the Standard Extensions list, *except* for those you've added to the Contained Extensions grid and set to *Reject*. For a Relative file extension list type, you can use the *Accept* status value to effectively extend the current list of Standard Extensions for this specific relative file extension list. For example, assume that at your site you use a product that uses a file extension that does not appear in the Standard Extensions list. By adding that file extension to the Contained Extensions grid and setting its Status value to *Accept*, the PeopleSoft system will accept that file type for file attachment operations involving this specific relative file extension list. **Note:**PeopleTools does not permanently add the extension to the Standard Extensions list when it is set to *Accept* for a relative file extension list type. That is, when you view the Standard Extensions list after setting the Status to *Accept* for a particular file extension, it will not appear in the list and will not affect the contents of the Standard Extensions list when it is accessed at runtime by other file extension lists you use. However, at runtime, the system will recognize that file extension as acceptable for this specific relative file extension list.
Unlimited	Do not use the Unlimited extension list type in the current release. This type of list is intended for possible use with future functionality.

Standard Extensions

Click to display a read-only list of standard file extensions that are known and potentially acceptable to the PeopleSoft system. This list comprises the standard list of file types that a PeopleSoft web server will handle. The displayed Standard Extensions list is provided for informational purposes to help you complete your file extension lists by indicating the set of file extensions that the PeopleSoft system recognizes.

To include additional file extensions to the Standard Extensions list, refer to the Relative file extension list type description.

Contained Extensions

Field or Control	Description
File Extension	Enter the file extensions you want to accept or reject. Add one extension per row, with each extension not exceeding ten characters. The system saves the value of an individual file extension as all uppercase, regardless of the case you enter, and adds a period (".") to the beginning of the file extension unless its first character is already a period. **Note:**Files having no file extension may be indicated collectively by specifying a file extension of a period (".").
Status	For a specified file extension of type Absolute or Relative, you have these Status options: • *Accept:* Indicates that the system will accept files having the specified extension. • *Reject:* Indicates that the system will reject files having the specified extension.

Bypassing Restrict-by-File-Type Functionality for System Administrators

System administrators managing the file attachment features of your PeopleSoft system can bypass restrictions otherwise imposed by filename extension lists by using a delivered security role, *PTFX Ignore Download Ext Lists*.

During a call to ViewAttachment or DetachAttachment, if the user profile initiating the transaction is associated with the *PTFX Ignore Download Ext Lists* role, then the system ignores file download restrictions specified by any filename extension list.

> **Note**
>
> The *PTFX Ignore Download Ext Lists* role should be associated only with the appropriate user profile(s).

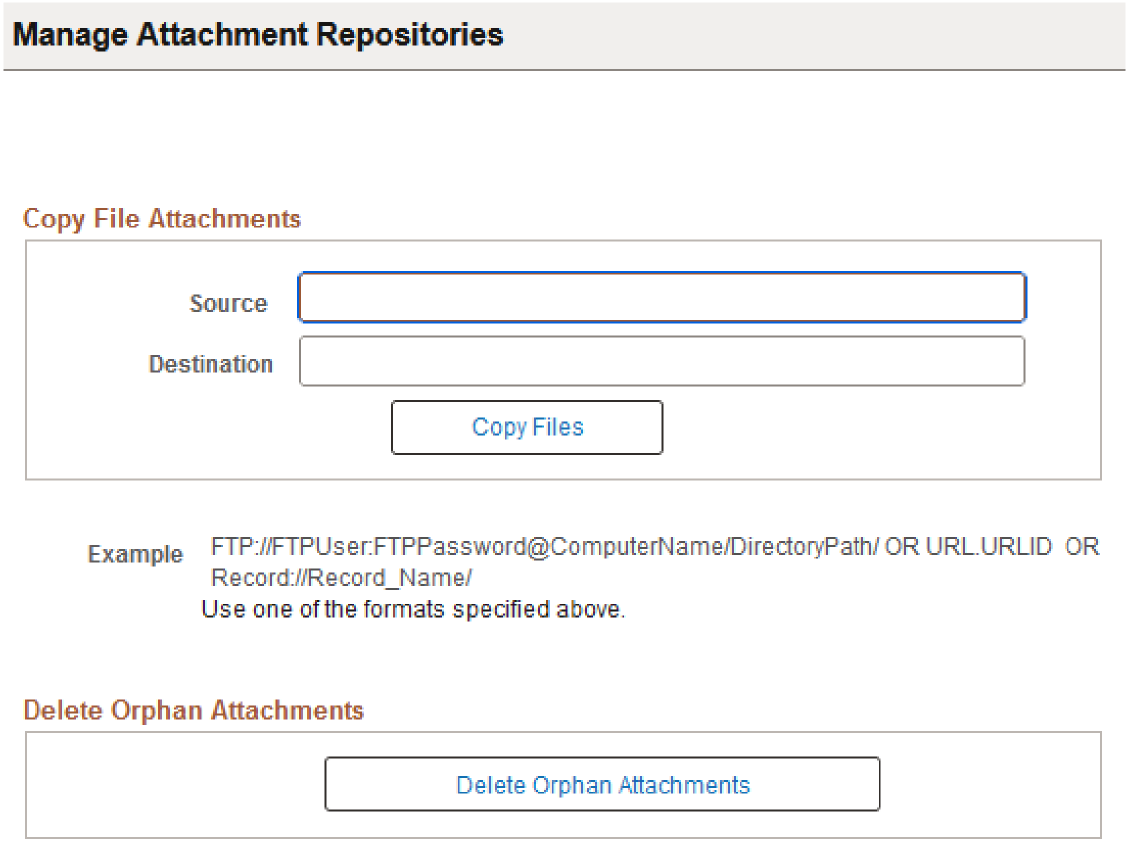
> **Note**
>
> The *PTFX Ignore Download Ext Lists* role contains no permission lists, however, its association with a user profile triggers internal PeopleTools security rules.

Manage Attachment Repositories

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Administer File Processing**, and then **Manage Attachment Repositories** to access the Manage Attachment Repositories page.

This page enables you to manage your attachment archives. You can transfer file attachment archives from one location (source) to a new location (destination), and you can perform general housecleaning by deleting orphaned attachments.

This example illustrates the fields and controls on the Manage Attachment Repositories page. You can find definitions for the fields and controls later on this page.



Copy File Attachments

Field or Control	Description
Source	Enter the URL that corresponds to the current (source) location of the file attachment archive to be copied.
Destination	Enter the URL that corresponds to the destination location of the file attachment archive to be copied.

Field or Control	Description
Copy Files	Invokes the PeopleCode function (CopyAttachment) that copies the file attachment archive from the specified source storage location to the specified destination storage location. For example, you can copy from the FTP server to a database, a database to the FTP Server, and so on. **Note:**For the file attachment functionality, in specifying the URL for the FTP server, the FTP server's machine name can be more than 30 characters. The length of the full URL is limited to 120 characters. Prior to copying files to a database storage location, you must first prepare the target record to store the attachments, which is discussed in the PeopleCode Developer's Guide. See PeopleCode Developer’s Guide: Understanding the File Attachment Functions.
Example	Displays the sample formats used for FTP, URL, or Record attachment transfers. These sample formats apply to both the Source and Destination edit boxes.

Deleting Orphan Attachments

The accumulation of file attachments can consume a significant amount of space in your database. Therefore, on a regular basis, it is recommended that you make sure that orphaned file attachments are deleted from the storage location, reclaiming disk space being used unnecessarily. *Orphan file attachments* are stored file attachments for which there is no longer a corresponding file reference in a valid file reference table.

Click the **Delete Orphan Attachments** button to complete this task. This button invokes the CleanAttachments PeopleCode function.

> **Warning**
>
> There is no way to roll back changes made by the CleanAttachments function. Review the PeopleCode Language Reference product documentation describing the behavior of CleanAttachments in order to appropriately anticipate how it will behave. Oracle suggests that you perform a database backup before deleting orphan attachments.

> **Note**
>
> When large numbers of orphaned file attachments may exist, Oracle recommends using the delivered Application Engine program, CLEANATT84, rather than the Delete Orphan Attachments button to remove the orphaned files. This avoids potential time-out issues.

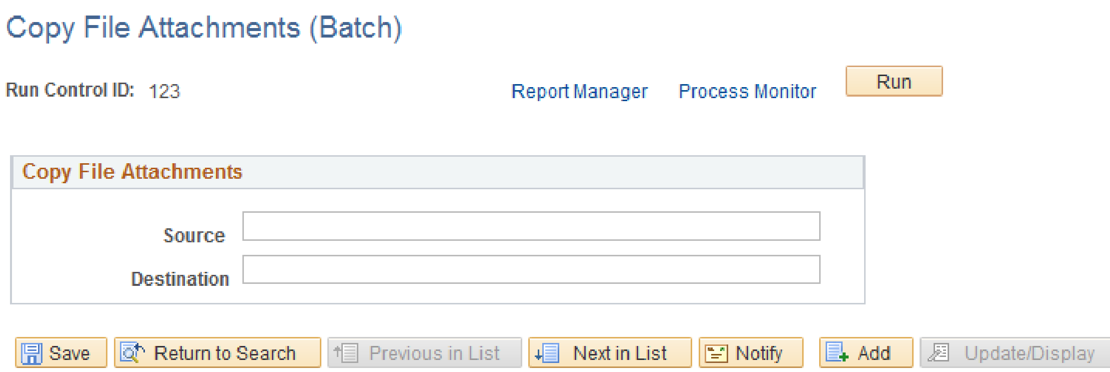
See PeopleCode Language Reference: CleanAttachments function

Copy File Attachments (Batch)

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Administer File Processing**, and then **Copy File Attachments (Batch).** Search for an existing Run Control ID or add a new value to access the Copy File Attachments (Batch) page.

This page enables you to run a CopyAttachments process via the COPYATTS Application Engine program on the Process Scheduler. This ensures that a large CopyAttachments job does not terminate prematurely due to PIA time-out.

This example illustrates the fields and controls on the Copy File Attachments (Batch) page.



Field or Control	Description
Report Manager	Click this to view reports. For more information about viewing reports, see Process Scheduler: Viewing Reports.
Process Monitor	Click this to review the status of scheduled or running processes. For more information about viewing the status of processes, see Process Scheduler: Viewing the Status of Processes
Run	Click this to run the processes. When you click Run, the Process Scheduler Request page appears, showing all of the jobs and processes that you have the security to run.
Source	Enter the URL that corresponds to the current (source) location of the file attachment archive to be copied.
Destination	Enter the URL that corresponds to the destination location of the file attachment archive to be copied.

Related Topics

- Process Scheduler: Understanding Report Manager
- Process Scheduler: Understanding Process Monitor

Delete Orphan Files (Batch)

Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Administer File Processing**, and then **Delete Orphan Attachments (Batch).** Search for an existing Run Control ID or add a new value to access the Delete Orphan Files (Batch) page.

This page enables you to run a CleanAttachments process via the CLEANATT84 Application Engine program on the Process Scheduler. This ensures that a large CleanAttachments job does not terminate prematurely due to PIA time-out.

This example illustrates the fields and controls on the Copy File Attachments (Batch) page.

Field or Control	Description
Report Manager	Click this to view all of the reports that are in the PeopleSoft system (across multiple databases) that the user is authorized to access.
Process Monitor	Click this to review the status of scheduled or running processes.
Run	Click this to access the Process Scheduler Request page. Select the process and click **OK** to delete.

Administer Cloud Providers

The Administer Cloud Providers utility is discussed elsewhere in this documentation.

See Configuring a Cloud Provider

Materialized Views

The Materialized Views utilities are documented in *PeopleTools: Data Management*.

See Data Management: Using Materialized Views for Oracle databases.

See Data Management: Working with Indexed Views for Microsoft SQL Server databases.

See Data Management: Working with Materialized Query Tables for Db2 for z/OS.

Related Topics

- Application Designer Developer’s Guide: Working with Materialized Views

Pre-Load Cache Utilities

See [Configuring an Application Server Domain to Preload Cache](#).

Gather Utility

The Gather utility facilitates communications between PeopleSoft and the customer on technical questions or issues. The Global Support Center (GSC) directs the customer to the Gather Utility when problems arise. Customers can also use a self-service website to run this utility and send in relevant information about their problems or issues.

Using a simple command line interface, the Gather utility is a small Java application that can run on any platform to collect various files from the following environments:

- Application Server.
- Web Server.
- Any additional files that the user chooses (SQL Trace files, PeopleCode Trace Files, and so on).

  The collected files are placed in a single jar file with psft.jar as the default name, in the temp directory. Subsequently, these files are sent to PeopleSoft.

> **Note**
>
> For this utility to work, the supported version of Java (JRE) must be installed on the target machine.

Getting Started

The following files reside in the starting directory:

- Gather.class: The main Java class file
- Helper.class: This class file is called by Gather.class
- Runnit.bat: A MS-DOS batch file that is used by Windows users.

  UNIX users have to run the Gather utility manually.
- Vars.sh: a UNIX shell script.

  Gather calls this automatically if the UNIX operating system is detected.

Windows Users

The following steps are used for Windows:

1. Make sure that you have the PS_HOME environment variable set.

   This saves the user from having to type it in.

2. Go to *PS_HOME*\utility.
3. Type `runnit`.
4. Follow the directions that are on the screen.

UNIX Users

Use the following steps for UNIX:

1. At a command prompt, run the following command where PeopleSoft is installed:

   ```
   ../psconfig.sh
   ```

2. Go to the *PS_HOME*/utility directory.
3. Change permissions for all files:

   ```
   chmod 777 *.*
   ```

4. Enter the following to start the utility:

   ```
   java -cp .:$CLASSPATH Gather
   ```

   > **Note**
   >
   > UNIX is case-sensitive – Gather is spelled with a capital G.

5. Follow the instructions that are on the screen.

Environmental Data

On Windows, both the set and netstat commands are invoked with the results copied to a file that is collected. On UNIX, the same thing is done with the env command.

Application Server Data

The following files are collected from the Application Server:

- PSAPPSRV.CFG
- PSAPPSRV.UBB
- LOGS/*.*

  This includes all subdirectories under LOGS. This usually includes all application server or Oracle Tuxedo logs, dump, and replay files.

Web Server Data

The gather utility collects numerous files (log files, configuration files, and so on) from each of the supported web servers. If an analyst only asks for a specific file, send that, but make sure to keep the other collected files in case they are needed.

Additional Files

There is always a need to include files that are not on the above list. These can include PeopleCode Trace files, SQL Trace files, SQL output, and so forth. The command line interface allows you to specify any file that you want to be included in the jar file.

Define Accessibility Help

You can find information on managing accessibility help in *PeopleTools: Accessibility Guide*.

See Accessibility Guide: Using the Define Accessibility Help Page

Using Audit Utilities

This section covers the utilities that are used for auditing the system's integrity.

Using the Record Cross Reference Component

Select **PeopleTools**, and then **Utilities**, and then **Audit**, and then **Record Cross Reference** and search for a Record (Table) Name.

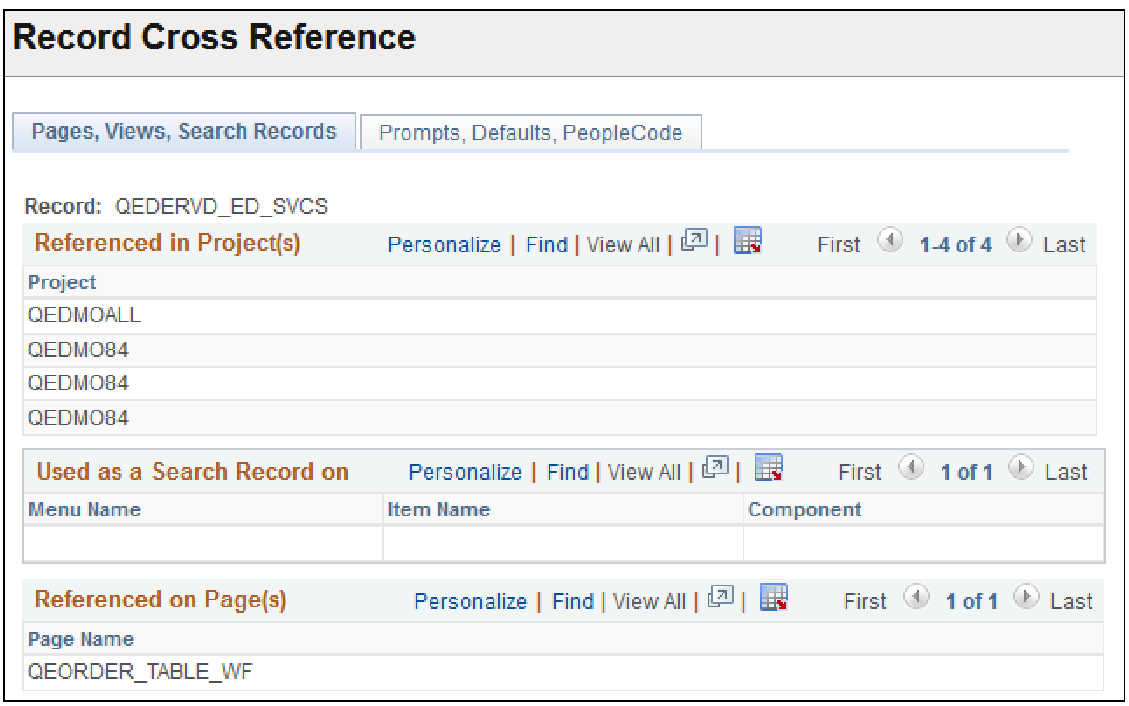
You use the Record Cross Reference component (XREF_PANEL_01) to view where a record is used throughout the application. There are two pages in this page group:

- Pages, Views, Search Records.
- Prompts, Defaults, PeopleCode.

Pages, Views, Search Records

This is a read-only page that shows which Projects, Menus, Pages, and Objects reference a particular record:

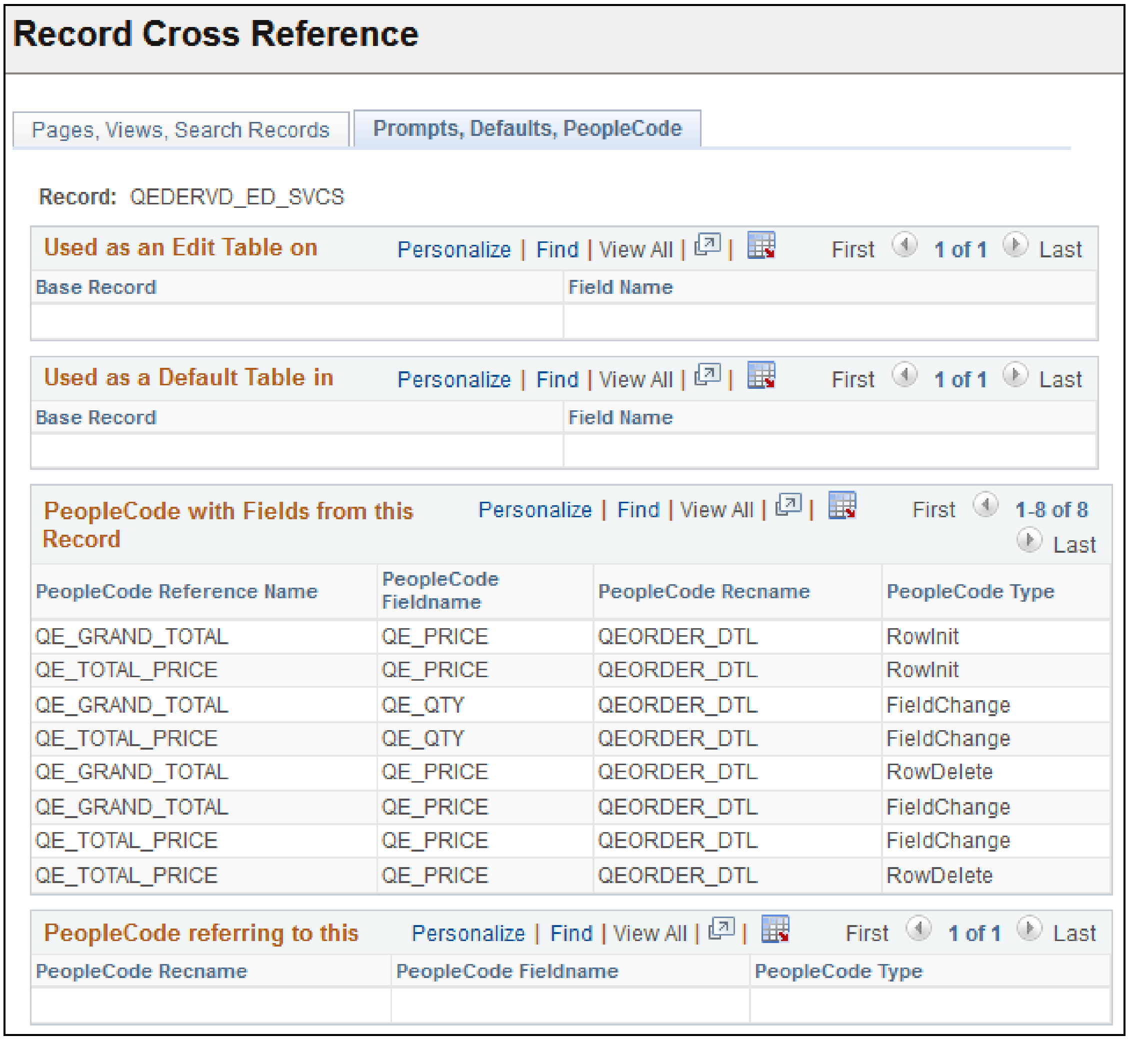
This example illustrates the fields and controls on the Pages, Views, Search Records page.



Prompts, Defaults, PeopleCode

On the Prompts, Defaults, PeopleCode page, the group boxes list the components that refer to the record.

This example illustrates the fields and controls on the Prompts, Defaults, PeopleCode page.



Field or Control	Description
Used as an Edit Table on	Lists pages that use the record as edit table.
Used as a Default Table in	Lists pages that use the record as a default table.
PeopleCode with Fields from this Record	Shows where fields from this record are used in PeopleCode.
PeopleCode referring to this	Shows all PeopleCode that references this record.

Performing a System Audit

The System Audit (SYSAUDIT) utility is documented in the Data Management PeopleBook.

Related Topics

- Data Management: Running SYSAUDIT

Performing Database Level Auditing

This utility is used to support database level auditing features, and is documented in the Data Management PeopleBook.

Related Topics

- Data Management: Understanding Database Level Auditing

Using Debug Utilities

This section provides an overview of debug utilities.

Using the HTTP Request Reporter Page

Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **HTTP Request Reporter** to access the HTTP Request Reporter page.

This page displays a report on the %Request PeopleCode object and system variables for the current user session.

Use the page to review the fields and properties of inbound HTTP requests. The information listed is not editable. The fields include the following:

- HTTP Request Headers
- HTTP Request Cookies
- HTTP Request Parameters
- HTTP Request class Properties
- Other HTTP Request Content
- PeopleCode system variables

Related Topics

- PeopleCode API Reference: Request Class Properties
- PeopleCode API Reference: Request Class Methods
- Portal Technology: Using Custom Portal Headers
- Portal Technology: Configuring Web Profiles

Using the Server Side Events Test Page

Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **Server Side Events Test Page**.

Use the Server Side Events Test pages to verify that Push Notifications have been set up correctly.

- Server event test for fields: Verify whether push notifications are configured correctly on an environment using a test event with key-value pairs as the event data type.

- Server event test for grid: Verify whether push notifications are configured correctly using a test event with PeopleSoft rowset as the event data type.
- Server event collection test: Verify that a collection of push notification events is configured correctly.

See Troubleshooting the Notifications Configuration in Fluid User Interface Developer's Guide: Configuring the System for Push Notifications.

Using Send Broadcast Messages

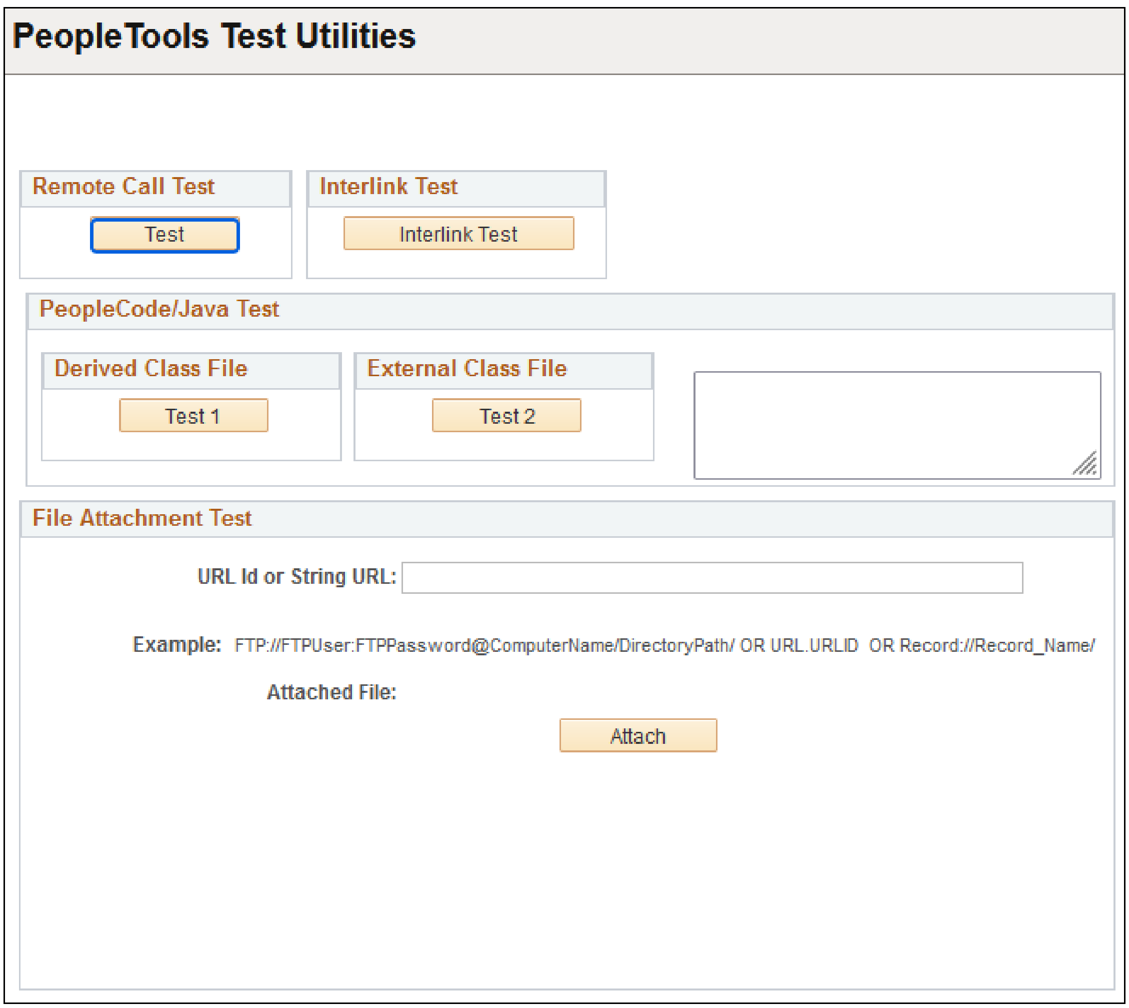
Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **Send Broadcast Messages** to access the Broadcast Message page.

See Fluid User Interface Developer's Guide: Sending Broadcast Messages

Using the PeopleTools Test Utilities Page

Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **PeopleTools Test Utilities** to access the PeopleTools Test Utilities page.

This example illustrates the fields and controls on the PeopleTools Test Utilities page.



Field or Control	Description
Remote Call Test	You use the Remote Call Test button to test the Remote Call configuration.
Interlink Test	Used to test Business Interlinks. (Business Interlinks is a desupported product).
PeopleCode/Java Test	The Derived Class File button tests Java-PeopleCode integration. It tests to see that Java is being processed correctly through PeopleCode. The result appears in the box to the right. The External Class File button tests Java PeopleCode integration.

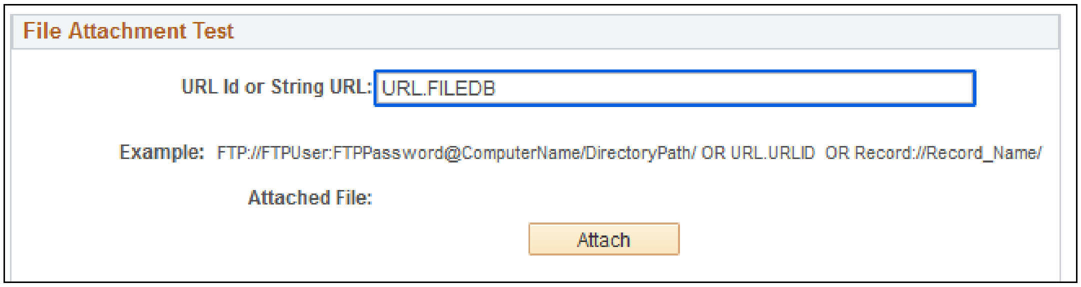
Field or Control	Description
File Attachment Test	This enables you to test the file attachment PeopleCode functions with your file storage location. You can supply the URL for testing as a URL ID or as a string. The URL ID is the recommended format for all repositories. • Enter a URL identifier (URL.*URL_ID*). For example: *URL.FILEDB* • Enter a repository path, such as: *record://psfile_attdet/* Click **Attach** to attach the file. Once the file is attached, you can View, Detach, and Delete the file. The file attachment PeopleCode functions and their usage within PeopleSoft applications are documented in detail in the PeopleCode Developer's Guide. When testing file attachment features, keep these usage stipulations in mind: • Using a URL Identifier is mandatory for FTPS, SFTP, HTTP, and HTTPS attachment protocols. • Using a URL Identifier is also recommended for the FTP attachment protocol. • Record and File attachment protocols can use the URL Identifier, *or* they can be specified as a plain string. **Note:** The File Attachment Test does not allow .JSP files to be uploaded or downloaded. **Note:** PeopleTools supports uploading, downloading, and deleting empty file attachments. An empty file refers to a file that exists, but has a size of zero bytes. **Note:** The File Attachment Test utility enables only a single file to be uploaded at a time, because the underlying PeopleCode function is AddAttachment. PeopleTools supports uploading multiple files at a time by using the MAddAttachment PeopleCode function. **Note:** For the file attachment functionality, in specifying the URL for the FTP server, the FTP server's machine name can be more than 30 characters. The length of the full URL is limited to 120 characters. See PeopleCode Developer’s Guide: Understanding the File Attachment Functions and URL Maintenance.

Example: File Attachment Test

The following shows the process of the file attachment test:

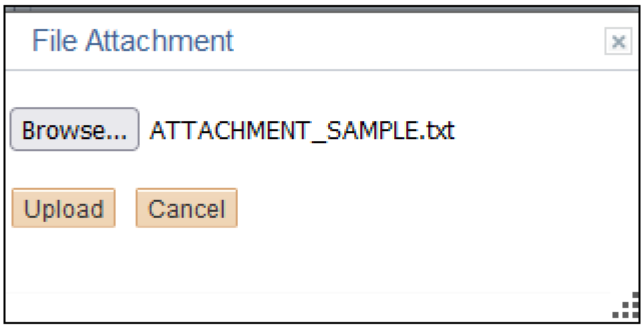
1. Enter a URL Identifier or a string URL defined in the URL Maintenance utility, and click **Attach**.

This example illustrates the fields and controls on the File Attachment Test section:



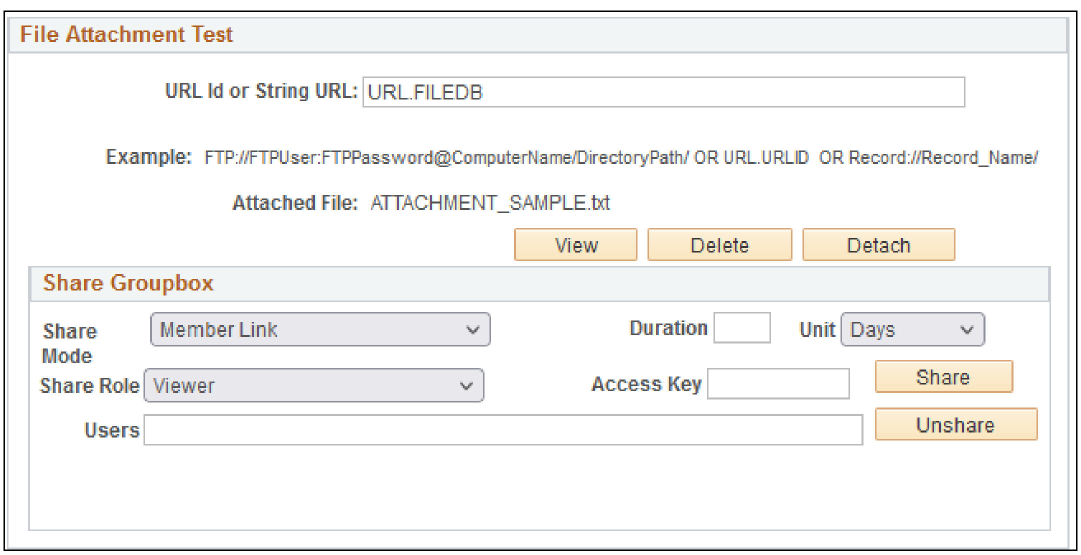
2. Browse to the location of the file to attach, and click **Upload**.

   This example illustrates the fields and controls on the File Attachment dialog box after selecting a file.



3. Confirm that the file has been stored in the appropriate repository (attached).

   This example illustrates the fields and controls on the File Attachment Test section after selecting a file attachment.



After you upload a file, you see the following fields and controls in the File Attachment Test section:

Field or Control	Description
View	Review the file. See PeopleCode Language Reference: ViewAttachment function.
Delete	Remove the file from the storage location. See PeopleCode Language Reference: DeleteAttachment function.
Detach	Download a file from the source storage location and save it locally. See PeopleCode Language Reference: DetachAttachment function.
Share Mode	Select Member Link, Public Anyone or Public Reg Users for the user access level.
Share Role	Select Viewer, Downloader, or Contributor to specify how a user is allowed to work with a file.
Duration and **Unit**	Specify the expiration time for the URL in the URL Id or String URL field. Select days, hours, or minutes from the Unit drop-down list.
Access Key	Specify an access token.
Share	Click to test the sharing with Oracle Content and Experience Cloud.
Unshare	Click to revoke sharing access to Oracle Content and Experience Cloud.
Users	Specify users who are added as members to the folder, which contains the specified file.

> **Note**
>
> Use the fields and controls in the Share Groupbox section only with files shared with Oracle Content and Experience Cloud. See Using Cloud Repositories. The underlying PeopleCode function is ShareAttachment. The functions mentioned here are described in detail in PeopleCode Language Reference: ShareAttachment function. You see an error if you try to use these fields for a non-cloud repository.

Related Topics

- PeopleCode Developer's Guide: Understanding the File Attachment Functions
- URL Maintenance
  Select **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Maintain URLs.** Search for an existing URL identifier or add a new value to access the URL Maintenance page.

Using the Replay App Server Crash Files Utility

Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **Replay App Server Crash Files** to access the Pt Replay page.

See Configuring the Application Server to Handle Cache Files and Replay Files.

Using the Trace PeopleCode Utility

The Trace PeopleCode utility is discussed elsewhere in this PeopleBook.

Related Topics

- Setting Up the PeopleCode Debugger
  This section discusses how to set up and use the PeopleCode Debugger.
- Configuring PeopleCode Trace
  Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **Set PeopleCode Trace Options** to access the Trace PeopleCode page.

Using the Trace SQL Utility

The Trace SQL utility is discussed elsewhere in this PeopleBook.

Related Topics

- Configuring SQL Trace
  Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **Set SQL Trace Options** to access the Trace SQL page.

Using Image Manipulation Utilities

Use the Image Manipulation testing pages to test the PeopleCode constructs related to managing images, such as the ResizeImage built-in method that enables users to resize an existing image as per the dimensions provided, and the CropImage built-in method that enables users to crop a source image and save it to a different location.

See PeopleCode Language Reference: ResizeImage function

See PeopleCode Language Reference: CropImage function

Using the Image Size Defn Page

Select **PeopleTools**, and then **Utilities**, and then **Image Manipulation**, and then **Define Image Dimensions** to access the Image Size Defn page.

Each dimension can be added as a row with width, height and a size label for the dimension. The label is expected to be unique.

The list defines a complete list of allowed dimensions in an application.

A duplicate entry with the same dimensions is allowed as long as the label is unique.

Using the Image Grouping Page

Select **PeopleTools**, and then **Utilities**, and then **Image Manipulation**, and then **Define Image Groups** to access the Image Grouping page.

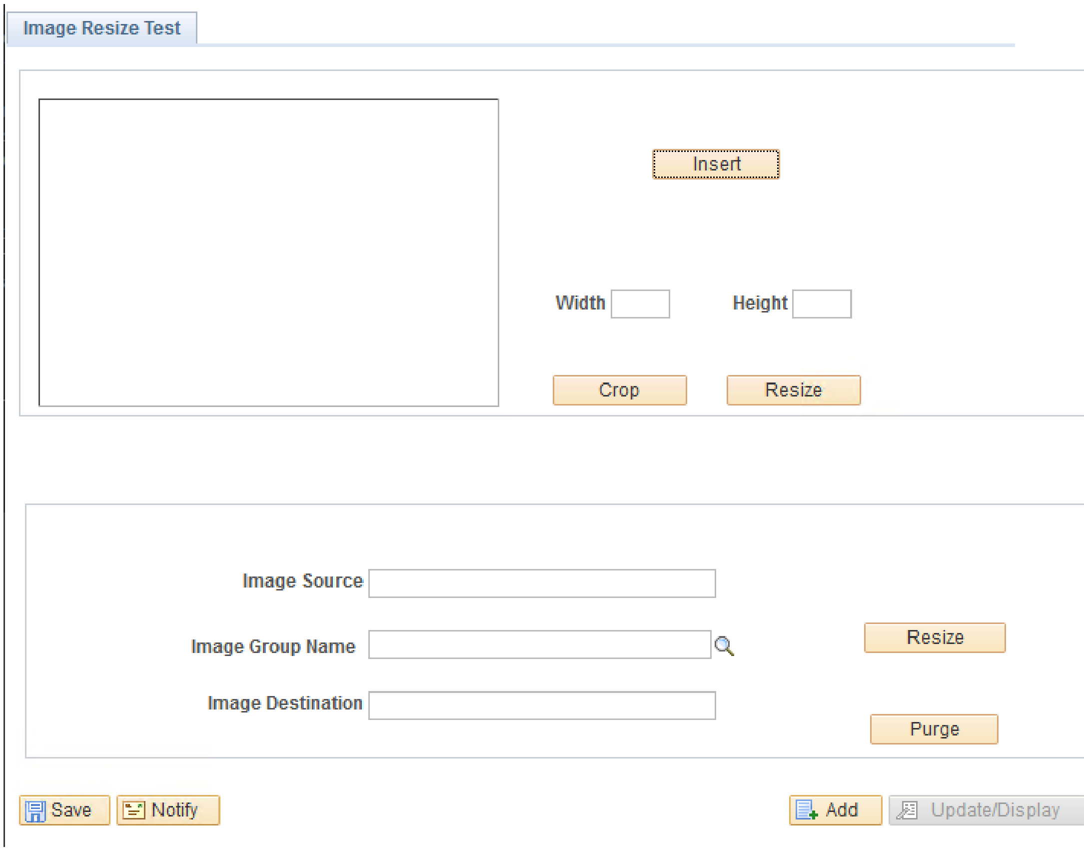
Create a list of image sizes. The list created on the Image Size Defn page acts as the basis for creating image groups.

An image group may have a minimum of one image size added to it. The maximum number of rows allowed is determined by the available entries in the image size list.

Using the Image Resize Test Page

Select **PeopleTools**, and then **Utilities**, and then **Image Manipulation**, and then **Image Resize and Crop Utility** to access the Image Resize Test page.

This example illustrates the fields and controls on the Image Resize Test page.



Use this page to test image PeopleCode constructs, such as insert, crop, resize, and so on.

Use the top section of the page to test dimensions for cropping or resizing. After inserting an image, enter the desired dimensions, and then click Crop or Resize. When you click Crop, a window appears that enables you to select the portion of the image to retain. The resulting cropped or resized image appears on this page.

Note

CropImage displays the image in actual size. If the image dimensions do not match the device dimensions, it may be difficult to crop the image even though a scroll bar is displayed. In such a case, Oracle PeopleSoft recommends that you resize the image and then crop the image.

Use the bottom section of the page to test resizing.

- Image Source corresponds to the ResizeImage parameter URL.*source_URL*.
- Image Group Name corresponds to the array of sizes in the ResizeImage API.

  You define image groups on the Image Grouping page.
- Image Destination corresponds to the ResizeImage parameter URL.*dest_URL*.
- When you use the ResizeImage API with a database table specified as the destination location, the PS_PT_IMG_TMPSTORE table is used to temporarily store the images instead of the specified database table. Use Purge to delete the entries from this PS_PT_IMG_TMPSTORE table.

Using International Utilities

This section covers the utilities that you use in globalization efforts.

Setting International Preferences

Select **PeopleTools**, and then **Utilities**, and then **International**, and then **Reset My Session Language** to access the International Preferences page.

Used to override the language that you select when you sign in to the database.

Field or Control	Description
Language Preference	Use the International Preferences page to temporarily change the session's language preference that was specified during sign in. This change lasts until you exit the PeopleSoft session or change the language preference again. Only languages that are enabled on the Languages page are available for selection.

Related Topics

- Global Technology: Changing the Session Language While Signed In

Setting International Field Size

Select **PeopleTools**, and then **Utilities**, and then **International**, and then **Resize Currency Fields** to access the International Field Size page.

If you process currency values that require large numbers, such as Italian lira, that require fields longer than those that are included in the standard application, you can use the International Field Size page to expand amount fields throughout the application.

After you create or select a run control ID, set the appropriate lengths for a list of fields, then click the **Run** button to launch the batch program that performs the field size changes.

Field or Control	Description
Field Name	Use the Browse button to select the field name.
Current Field Size	This is a read-only field indicating the current field size as stored in PSDBFIELDS.
Field Size - International	Enter the field size to expand (or contract) the field size for foreign fields.

Related Topics

- Global Technology: Resizing Currency Fields by Using the International Field Size Utility

Administering Time Zones

The Define Time Zones and Map IANA Times Zones utilities are extensively documented in the *PeopleTools: Global Technology* product documentation.

Related Topics

- Global Technology: Understanding Time Zones
- Global Technology: Maintaining Time Zones

Managing Languages

Select **PeopleTools**, and then **Utilities**, and then **International**, and then **Manage Installed Languages** to access the Manage Installed Languages page.

This utility is extensively documented in the *PeopleTools: Global Technology* product documentation.

See Global Technology: Managing Languages in the PSLANGUAGES Table

Using Maintain Database Sequences

Use the Maintain Database Sequences page to define a database sequence as a managed object.

Select **PeopleTools** , and then **Utilities**, and then **Maintain Database Sequences**. Add a name for a new sequence or search for an existing one.

You can define a database sequence to generate a series of numbers for applications such as getting new purchase order numbers or employee numbers. To use database sequences:

1. Define a database sequence on the Maintain Database Sequences (PSSEQUENCE) page.
2. Use Application Designer to insert the sequence into a project.

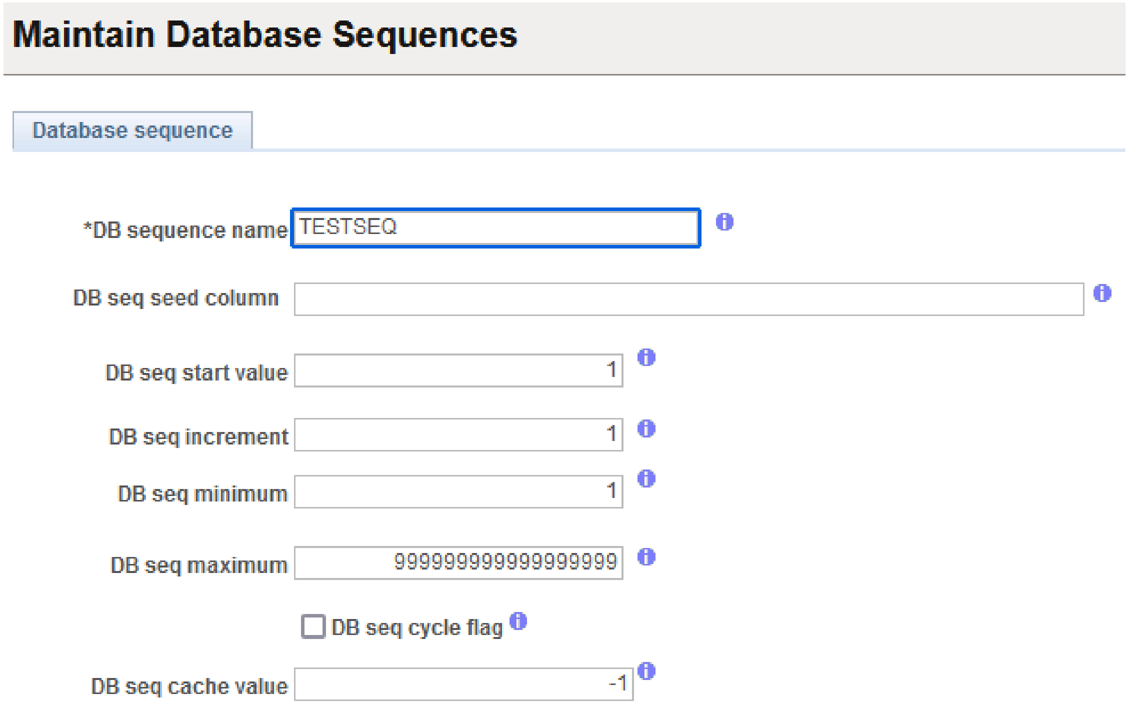
   See Application Designer Developer’s Guide: Inserting Definitions Into Projects.
3. Build the project.

   See Application Designer Developer’s Guide: Running the Build Process.

To create a database sequence using the Data Mover command CREATE_SEQUENCE, refer to Lifecycle Management Guide: Using PeopleSoft Data Mover Commands.

You can also reference the database sequence in the Meta-SQL function %GetNextSeqValue. See Application Engine: %GetNextSeqValue meta-SQL element.

This example illustrates the fields and controls on the Maintain Database Sequences page.



Field or Control	Description
DB sequence name	The field includes the DB sequence name you supplied on the search page.
DB seq seed column	Enter the name of a table and column in the format *TableName.ColumnName*. When a sequence is created the value selected from this column will be used instead of the value in **DB seq start value**. This is optional.
DB seq start value	Enter a number to begin the sequence. The default is 1.
DB seq increment	Enter a value to increment each number by. The default is 1. Enter a negative number to have the sequence values decrease.
DB seq minimum	Enter the minimum value that the sequence can reach. The default is 1.
DB seq maximum	Enter a number for the maximum value in the sequence. The default is 999,999,999,999,999,999. The maximum value is 10^{28} -1.

Field or Control	Description
DB seq cycle flag	Select the **DB seq cycle flag** if you want the sequence to continue to generate values after reaching either its maximum or minimum value. After an ascending sequence reaches its maximum value, it generates the sequence minimum value. After a descending sequence reaches its minimum, it generates the sequence maximum value. If this option is not set an error is returned when the maximum or minimum value is reached.
DB seq cache value	Specify how many sequence numbers to cache. • Enter a number greater than zero to cache that many sequence numbers in memory. • Enter zero to cache no numbers. • Enter a number less than zero to use the system default for caching. The default value is -1, meaning the system default is used.

Using Maintain Table Partitions

The Maintain Table Partitions page is documented in Data Management: Maintaining Partition Definitions.

Using Manage Configurable Search

Use Manage Configurable Search to create and manage custom search pages.

The Manage Configurable Search utility is documented in the Search Technology product documentation.

See Search Technology: Configuring a Search Page.

Using Optimization Utilities

The Optimization utilities are documented extensively in the Optimization Framework PeopleBook.

Related Topics

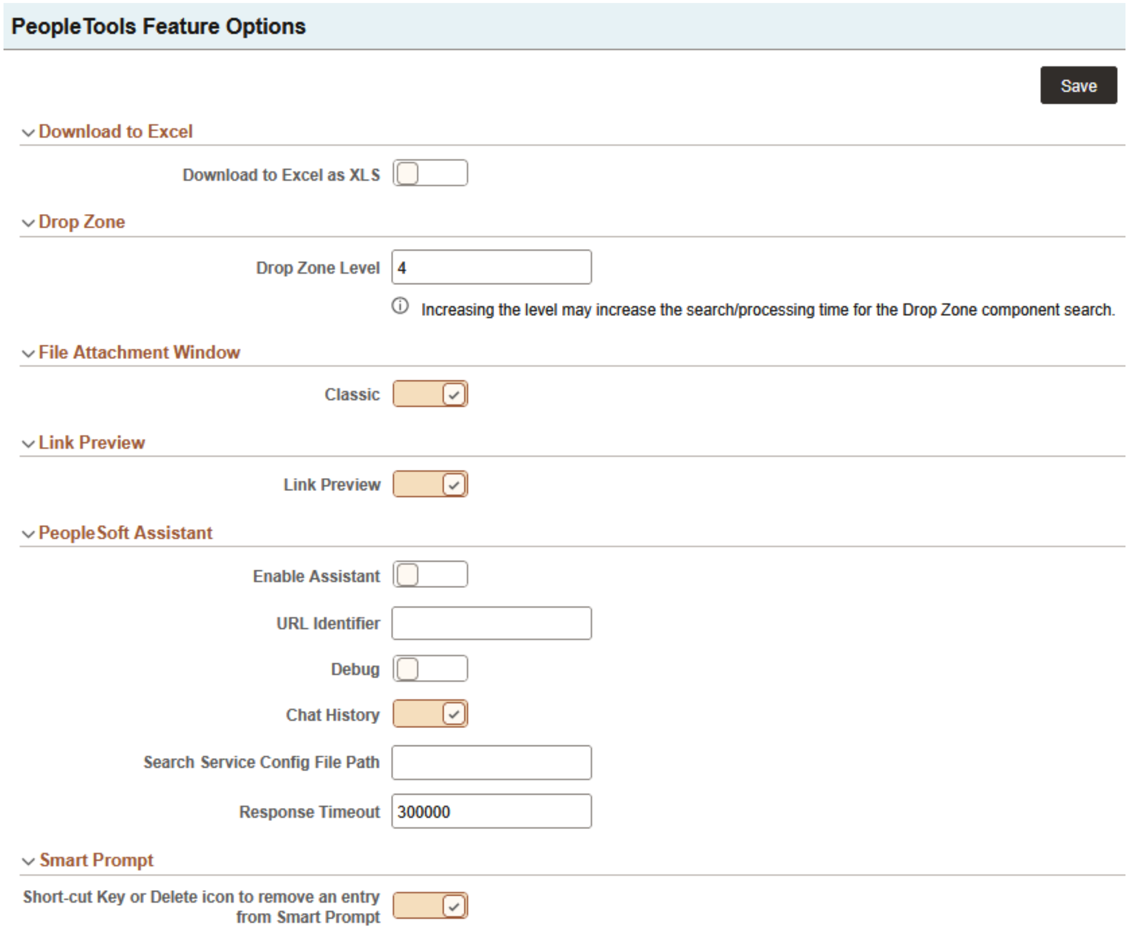
- Optimization Framework
- Optimization Framework: Updating Solver Licenses

Using PeopleTools Feature Options

Use the PeopleTools Feature Options page to configure system-level properties for PeopleTools features.

Select **PeopleTools**, then **Utilities**, and then **PeopleTools Feature Options**. Only administrators are able to access this page. Expand and collapse the sections and use the toggle switches to enable or disable the features.

This example illustrates the PeopleTools Feature Options page.



Download to Excel

Field or Control	Description
Download to Excel as XLS	Select this option to download grid data as a spreadsheet with the .xls extension. By default, this option is disabled, that is, the grid data is downloaded as a Microsoft Excel spreadsheet with the .xlsx extension. This option should be enabled only as a fallback mechanism.

For more information on using this option, see Viewing Data in Original Layout Grid, Downloading Grid Data and Viewing Data in Scrollable Layout Grids, Downloading Grid Data

Drop Zone

Field or Control	Description
Drop Zone Level	Use this section to specify how many drop zone levels are included in a search. The default is 4, and zero (a blank field) is not allowed. The default value of 4 means that search results on the Component Drop Zone page include components with drop zones on the main page and on pages nested up to three levels below it. **Note**. Increasing the level may increase the search or processing time for the Drop Zone component.

See Applications Designer Developer's Guide: Configuring Drop Zones.

File Attachment Window

Field or Control	Description
Classic	Control whether a fluid-style file attachment window appears on classic pages. The file attachment window is where users browse for and select files to upload. This option is enabled by default, which means that when a user performs an action to add an attachment from a classic page, the system displays the traditional modal window as seen in previous releases. If this option is disabled, when a user adds an attachment from a classic page, the system displays a fluid-style attachment window. Because the window is opened within a classic page, it includes standard classic window controls, such as Move and Resize, which are not available when the attachment window is accessed directly from a fluid page. Note that an end user can drag and drop attachments using either the classic or fluid modal window, but the fluid window has the area to drag and drop attachments more clearly defined.

See Understanding the File Attachment Functions

Link Preview

Field or Control	Description
Link Preview	Control the ability to display previews for PeopleSoft links. This option is enabled by default. If the Link Preview option is enabled, when a link is shared or posted, instead of seeing a URL, users can see a preview; that is, title, description, and image of the link content. If this option is disabled, end users see a URL when accessing a PeopleSoft link.

See Portal Technology: Creating Rich Preview for PeopleSoft Links.

PeopleSoft Assistant

This section includes controls for the PeopleSoft chat feature. The feature is available in many locations in PIA by default.

Field or Control	Description
Enable Assistant	Control whether the PeopleSoft Assistant will be available. It is not enabled by default.
URL Identifier	The URL identifier specifies the credentials (user ID, password, token) that the system uses to access the PeopleSoft Assistant services. See URL Maintenance for details on specifying the credentials.
Debug	Enable debugging. This is not enabled by default. When debugging is enabled, the administrator obtains the debugging information in the browser console log. Check your browser documentation for information on accessing the browser console log.
Chat History	Control whether to store the questions asked by the end user in the chat history table (PS_PTML_CHAT_HIST). It is enabled by default. If this is disabled, the questions are not saved.
Search Service Config File Path	Enter the path to the configuration file used by PeopleSoft Assistant for using the Search Service.
Response Timeout	Enter a value in milliseconds. The default is 300000 msec (5 minutes). This is the maximum amount of time the Chat User Interface (Chat UI) will wait for a back-end response before it prints a "Response Timeout" message.

See Portal Technology: Working with Chat User Interface (Chat UI)

Smart Prompt

Field or Control	Description
Short-cut Key or Delete icon to remove an entry from Smart Prompt	Select this option to display a delete (X) button next to a smart-prompt value in the Most Relevant section. This option is enabled by default. **Note**: The delete button can be enabled only for prompt values; it cannot be enabled for smart drop-down list values.

See Applications User's Guide: Using Smart Prompts.

See Setting Smart Prompts and Smart Drop-Down Options in PeopleTools Options.

Reviewing the PT_FEATUREOPT Table

The options on the PeopleTools Feature Options page are stored in the PT_FEATUREOPT table. This table follows a name-value pair model, and contains several columns that define the metadata for each system-level property. This table lists each relevant column and its purpose:

Column Name	Purpose
PT_PROPERTYNAME	Property Name —Stores the name of the individual property associated with a PeopleTools feature.
PT_PROPERTYVALUE	Property Value — Stores the current value assigned to the property.
MSGSET	Message Set Number — Defines the Message Catalog entry used to display the label for the property name in PIA (PeopleSoft Internet Architecture).
MSGNUM	Message Number — Defines the Message Catalog entry used to display the label for the property name in PIA (PeopleSoft Internet Architecture).
LABEL_ID	Field Label ID — Contains the Label ID that identifies the user-facing label associated with the property name.
PT_FEATUREMSGSET	Feature Message Set — Represents the feature name through Message Catalog entries (for example, PT_CONFIG_SRCH, PT_MYFAV_SEARCH). These are used for grouping properties under a specific feature
PT_FEATUREMSGNUM	Feature Message Number — Represents the feature name through Message Catalog entries (for example, PT_CONFIG_SRCH, PT_MYFAV_SEARCH). These are used for grouping properties under a specific feature
PT_FEATURE_SW	Feature Switch — Indicates whether the property acts as a feature switch for the feature. **Note:** There will be only one feature switch for a feature.
PT_OPTNDISP	Display type — Specifies the UI control type (for example, Drop-down, checkbox or edit box) used to render the property value in the user interface. • C — Check-box control • E — Edit-box control • D — Drop-down control
DISPLAYONLY	Display only — Specifies whether the display control is display only or not.
FIELDTYPE	Field type — Specifies the field type, *0* for character field, *2* for number field.
PT_PROMPTRECNAME	Prompt record name — Specifies the record name that should be used by the prompt. The field is blank by default (the value is " ").
PT_HELPMSGSET	Message Set Number — Contains the Message Catalog entry for the help text associated with the property, shown in PIA for user guidance.
PT_HELPMSGNUM	Message Number — Contains the Message Catalog entry for the help text associated with the property, shown in PIA for user guidance.

The table uses a caching mechanism to improve performance. The system reads from cache by default. The cache is refreshed only when the table data is modified.

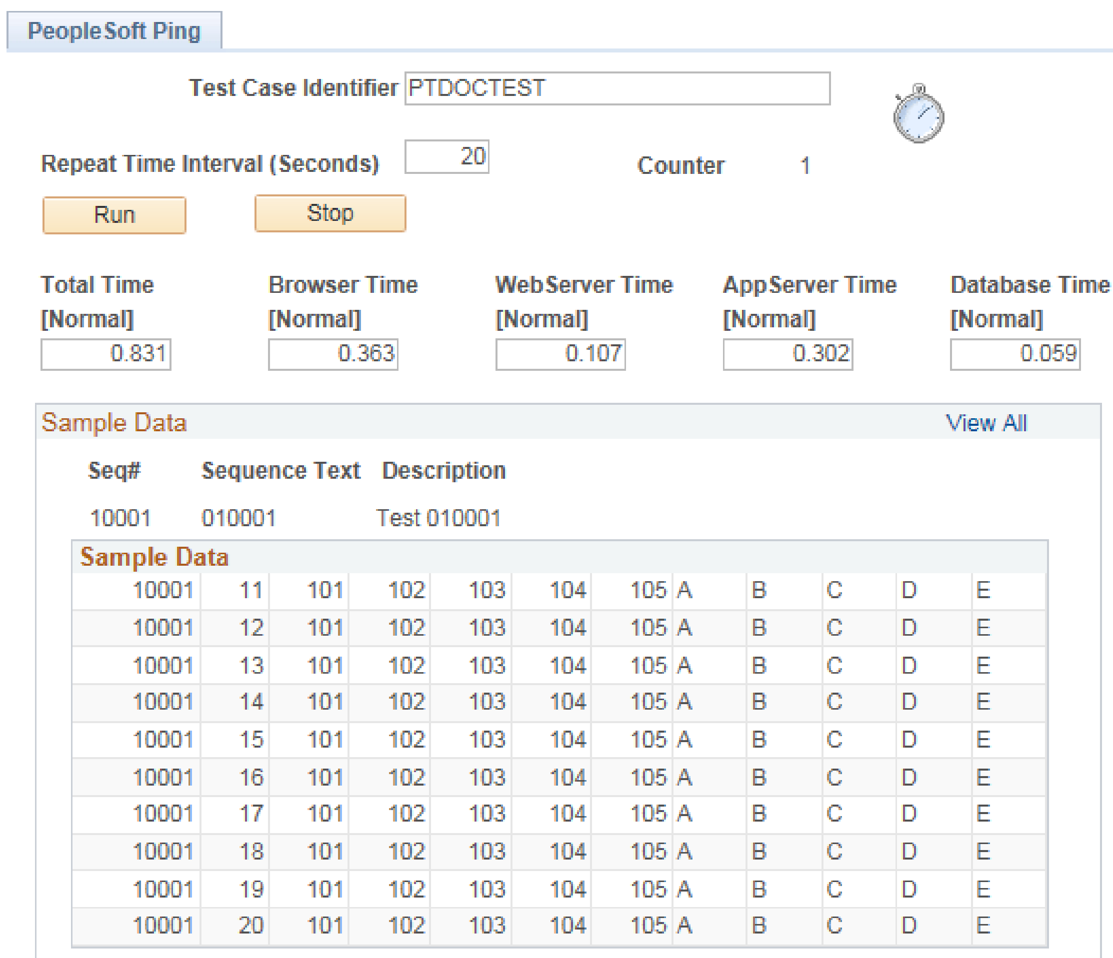
Using PeopleSoft Ping

The PeopleSoft Ping utility collects timestamps by sending a specific page to different tiers of the PeopleSoft system, starting at the browser, then going to the web server, the application server, the database and back. The timestamps that are collected are total time elapsed for the round trip, and arrival and departure time at each of the tiers.

Run PeopleSoft Ping

To use the PeopleSoft Ping feature, select **PeopleTools**, and then **Utilities**, and then **PeopleSoft Ping**, and then **Run PeopleSoft Ping.**

This example illustrates the fields and controls on the PeopleSoft Ping page. You can find definitions for the fields and controls later on this page.



Enter a Test Case Identifier to uniquely group each set of metrics. For **Repeat Time Interval**, enter an increment for the ping to run. To avoid creating unnecessary traffic and overhead to the PeopleSoft system, set the Repeat Time Interval to a relatively high value, such as 600 to 1800 seconds, during normal operations. You may need to increase the Session Timeout value accordingly.

Select View All to display all the results.

Depending on the targets set on the PeopleSoft Ping Options page, the fields for total time and tiers are marked as Normal, Alert, or Warning. The page displays results for Total Time, Browser Time, WebServer Time, AppServer Time, and Database Time.

PeopleSoft Ping Chart

Select **PeopleTools**, and then **Utilities**, and then **PeopleSoft Ping**, and then **PeopleSoft Ping Chart.** Specify a test case identifier.

PeopleSoft Ping includes a charting utility to zoom in to a specific time interval from the ping test.

You can change the displayed time interval to a subset of the full ping test period. Edit the start time and end time values, and click **Redraw** to refresh the chart display with the new time interval.

Click **Query Viewer** to query the database for the ping data. A new browser window opens, displaying the ping data for the full test period in a table.

PeopleSoft Ping Delete

To delete a ping page test case:

1. Select **PeopleTools**, and then **Utilities**, and then **PeopleSoft Ping**, and then **PeopleSoft Ping Delete.**

   The Delete page lists the current test case identifiers.
2. Select the check box next to the identifier(s) you want to delete.
3. Click Delete.

PeopleSoft Ping Options

Select **PeopleTools**, and then **Utilities**, and then **PeopleSoft Ping**, and then **PeopleSoft Ping Options.**

The PeopleSoft Ping Options page enables you to set targets for each tier (browser, web server, application server, and database) as well as overall completion time (Total time). If the ping process exceeds your targets, this affects the text qualifiers on the PeopleSoft Ping page, using Normal, Warning, and Alert. *Normal* is any time under the Warning and Alert targets, *Warning* is any time over the Warning target yet under Alert, and *Alert* is anything exceeding the Alert target. The metrics you enter depend on the typical conditions and expectations at your site.

10

Tracing, Logging, and Debugging

Working with PeopleSoft Server Process Logs

For each PeopleSoft server process, such as PSAPPSRV, the system writes an individual log file to your specified logs directory. Depending on the logging level, these log files can contain varying amounts of information. Reviewing these log files regularly can help you to identify potential issues or system trends, and they are a valuable source of information when troubleshooting. The PIA and application server domain logs include correlation fields, such as SRID, TOP Instance ID, and Operator ID that enable you to correlate log files across multiple domains when troubleshooting. To understand the log information, it is helpful to be able to identify the various log fields that the system writes to an individual line in the log file.

> **Note**
>
> Correlation data is logged when available, otherwise placeholders (hyphen) are used in place of correlation fields to maintain uniformity of format in the log files.

> **Note**
>
> Integration Broker, trace, and third-party log files do not include correlation information. However, all log files include placeholders (hyphen) for the correlation fields to maintain uniformity of format in the log files.

A typical log file contains these log fields:

[Server Process]:[Operating System Process ID] [Service Request Number] [Timestamp] [SRID] [TOP Instance ID] [Operator ID] [Log Level] [Message]

The following fields are included in the logs:

- *Server Process* — The PeopleSoft server process, such as PSAPPSRV.
- *Operating System Process ID* — The process number assigned by the operating system.
- *Service Request Number* — The number of service requests that this Tuxedo server process has sequentially executed.

  See Listeners, Handlers, and Queues.
- *Timestamp* — Date and time the message was generated.

  The format is YYYY = year, MM = month, DD = day, HH = hour, MM = minute, SS = seconds, and sss = milliseconds.
- *SRID* — A base64 encoded field uniquely associating a user session and its service request in the PeopleSoft-generated online transaction processing (OLTP) log files. For example, 17F2TbFWJw69UXCzfobVhZ.

> **Note**
>
> This is sometimes referred to as TRID.

- *TOP Instance ID* —Top correlation attribute in PPM's PMU Tree.

  It is a 64-bit integer. For example, 455266533594. It is used for correlating log entries with PPM data. For more information on instance IDs in PMU trees, see Performance Monitor: Viewing Open PMUs.
- *Operator ID and remote address* — PeopleSoft user ID and the remote address (IP address or DNS host name) that the user has at the time the message is logged.
- *Log Level* — The relative importance of this particular log message.

  Only messages whose logging levels are less than or equal to the LogFence appear in the log file.

  See the LogFence information in Domain Settings.
- *Message* — The actual log message.

The following table provides examples of content for each trace field.

Server Process:PID	Service Request	Timestamp *(YYYY-MM-DDTHH:MM-SS.sss)	SRID	TOP Instance ID	Oper-ator ID	Log Level	Message
PSAPPSRV.5648	(993)	[2012-12-12T11:54:23.778]	-	-	-	(3)	Authenticate User not needed. New User Id/Password same as current.
PSAPPSRV.5648	(1000)	[2012-12-12T11:54:26.198]	-	-	-	(0)	Recycling server after 1000 services
PSAPPSRV.28708	(0)	[2012-12-12T11:54:35.514]	-	-	-	(0)	Cache Directory being used: D:\PT_SERVER\appserv\T1B85301\CACHE\PSAPPSRV_1\
PSAPPSRV.28708	(0)	[2012-12-12T11:54:38.339]	-	-	-	(0)	Server started
PSAPPSRV.24208	(1344)	[2015-07-14T00:28:15.317]	17F2TbFWJw69UXCzfobVhZ	455266533594	QEDMO	(3)	-

You can select how you prefer to separate the trace fields using the [Domain Settings] LogFieldSeparator setting.

Related Topics

- Domain Settings
  Use this section to specify general settings for the entire domain—not just for a specific component of the domain or server process.
- Trace Options
  This section enables you to specify the tracing options that you can enable on the application server to track the Structured Query Language (SQL) and PeopleCode of the domains. You can also set all of the trace parameters from the PeopleSoft sign-in page.

Just beneath the Sign In button, click the link that opens the trace flags page. This enables you to set the trace options and then sign in to the system.

- Specifying Trace Settings
  Select the Trace tab.

Reviewing Stdout and Stderr Files

Oracle Tuxedo creates multiple stdout and stderr files when PeopleSoft domains are booted.

When you boot application server and Process Scheduler domains, Oracle Tuxedo generates stdout and stderr files for each server process. The files are located in the LOGS directory for the application server domain (<*PS_CFG_HOME*>/appserv/<*domain*>/LOGS) or Process Scheduler domain (<*PS_CFG_HOME*>/appserv/prcs/<*domain*>/LOGS).

The files names include the Oracle Tuxedo server ID (SRVID) and process ID (PROCID), in the format stdout_SRVID.PROCID and stderr_SRVID.PROCID. There will be one unique set of files for each process. You can use the files to narrow down the server and process when troubleshooting. Information about server IDs is available in the TUXLOG.* files, which are found in the same LOGS directory. For example, in this sample from a TUXLOG file, the SRVID is *2*.

```
215.53.sample!BBL.610.399.0: LIBTUX_CAT:541: WARN: Server APPSRV/2 terminated
215.56.sample!BBL.610.399.0: LIBTUX_CAT:557: INFO: Server APPSRV/2 being
restarted
```

These files are generated each time a domain is booted. Review the files after each reboot, and either back them up or delete them.

Setting Up the PeopleCode Debugger

This section discusses how to set up and use the PeopleCode Debugger.

> **Note**
>
> PeopleCode debugging is not supported on z/OS.

You can debug the PeopleCode program configurations of a two-tier connection to the database or a three-tier connection to the database.

> **Note**
>
> When you debug PeopleCode with an application server, Application Designer should be run in three-tier mode. PeopleCode debugging by using a two-tier PSIDE and an application server is not supported on multi-homed (multiple Internet Protocol address) workstations.

> **Note**
>
> Application Designer (PSIDE.EXE) will not connect to the PeopleCode Debugger through a firewall using Network Address Translation (NAT).

Debugging for a Two-Tier Connection

Debugging in two-tier connections involves connecting directly to the database, not through the application server. Use this method to debug two-tier Windows applications.

> **Note**
>
> By default, the port number that the PeopleCode debugger uses is 9500. Unless this port number is being used by another application, you do not need to alter any environment settings, and after you sign on to the database, you are able to debug PeopleCode.

If you need to change the PeopleCode Debugger port settings, complete the following procedure.

To change the default PSDBGSRV listener port number:

1. Open PeopleSoft Configuration Manager.
2. Select the Trace tab.
3. Locate the PeopleCode Debugger section, and make sure that the default value for the **Local PSDBGSRV Listener Port** is suitable for the system.

   For example, make sure that no other applications are configured to listen on the default port number (9500). If so, you must assign a port number that is not being used.

> **Note**
>
> If you're using a personal firewall, you must configure it to enable data packets to flow through the PSDBGSRV listener port. If you can't configure your firewall appropriately, you must shut it down while performing PeopleCode debugging.

Debugging for a Three-Tier Connection

Use three-tier debugging to debug three-tier Windows applications and PeopleSoft Internet Architecture (PIA) applications. For three-tier debugging, use PSADMIN to ensure that the following items are set:

- The appropriate PSDBGSRV listener port is specified in the PeopleCode Debugger section of PSADMIN.
- At least two PSAPPSRV processes are configured to boot in the domain with the service timeout parameter set to zero.
- Enter y for yes at the Enable PSDBGSRV Server Process prompt at the end of the PSADMIN interface.

Debugging on a Multi-Homed System

If you're debugging on a multi-homed (multiple IP address) system, you must explicitly specify an IP address in the Workstation Listener section of the PSADMIN configuration, rather than %PS_MACH%. The address you specify must be one by which the application server identifies the machine on which you're doing the debugging. This ensures that the workstation listener monitors requests from the correct location.

See Workstation Listener Options.

Setting the PSDBGSRV Listener Port

In the PeopleCode Debugger section of PSADMIN make sure that the value that is assigned to the PSDBGSRV listener port is not already in use by another application or listener on the application server. The default value is 9500. If the default is not acceptable, assign a suitable value to the parameter. If it is acceptable, no changes are required.

For example,

```
Values for config section - PeopleCode Debugger
    PSDBGSRV Listener Port=9500

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

Consider the following when debugging PeopleCode:

- If multiple application server domains are running on a single, physical machine, each domain needs to use different debugging port numbers.

  Otherwise, there is contention for the PSDBGSRV listener port value. This is the same principle that requires each application server domain on a server to have unique workstation listener port numbers.
- When you are not debugging, turn off (set to 0) the Enable Debugging parameter.

  The debugging mode results in an unavoidable amount of overhead, which can degrade performance.
- Regarding performance, do not perform debugging on a production domain.

  Debugging should be performed on a designated testing domain only.

Enabling Multiple PSAPPSRV Server Processes

The minimum requirements for PeopleCode debugging are:

- Two PSAPPSRV server processes are configured to boot in the domain.
- The Service Timeout value in the PSAPPSRV configuration section must be set to 0.

For the debugger to work, it has to run in parallel with the application that it's debugging. Suppose that the domain has only one PSAPPSRV server process running. In this case, the PSAPPSRV can process the requests of only one component at a time, and therefore debugging is not possible. Debugging involves two items, the debugger (PSDBGSRV) and the PSAPPSRV server process that is running the application PeopleCode.

Provided that you have two PSAPPSRV server processes configured; one PSAPPSRV handles the debugger program, while the other handles the application that you're stepping through with the debugger. In this case, the two programs run in parallel, which enables interactive debugging.

The configuration templates that PeopleSoft delivers all have at least two PSAPPSRV processes. However, if you are using a custom template, make sure that you configure the domain to start two PSAPPSRV processes prior to debugging. To do this, in PSADMIN set the Min Instances parameter in the PSAPPSRV section to 2.

The following example shows a sample PSAPPSRV section properly configured for debugging PeopleCode:

```
Values for config section - PSAPPSRV
    Min Instances=2
    Max Instances=2
    Service Timeout=0
    Recycle Count=0
    Allowed Consec Service Failures=0
    Max Fetch Size=5000

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

When configuring the PeopleCode debugger:

- PeopleSoft recommends using the Developer configuration template because this template, by default, provides two PSAPPSRV server processes and has service timeout set to zero.
- PeopleSoft recommends using a simple configuration where you are assured that the server that Application Designer connects to is the same server that the application you are debugging is running on.

> **Note**
>
> If you do not set the settings for PSAPPSRV correctly (at least two PSAPPSRV processes), PSADMIN automatically sets these values to comply with the minimum requirements when you enable PeopleCode Debugging (as discussed in the next section).

> **Note**
>
> When you enable the PeopleCode Debugger (PSDBGSRV), service timeout settings for server processes are set to zero (0) by the system, overriding any previous settings you may have made in PSADMIN. For example, if the service timeout settings for the PSAPPSRV service process are set to 300 prior to enabling the debugger, after you enable the debugger, the service timeout value will be zero (0). After using the debugger, you need to reset your service timeout settings to the desired values. Before enabling the debugger, it is recommended that you make a backup of your configuration file or make note of your service timeout settings.

Requesting a PSDBGSRV Server Process

After you specify the settings by using PSADMIN, the system prompts you with a series of options, such as setting up messaging server processes, enabling Jolt, and so on.

When you're prompted to enable the PSDBGSRV, enter y. Y appears in the Developer template by default.

Using the PeopleCode Debugger

After the system is configured properly, using the PeopleCode debugger is just a matter of signing on to the PeopleSoft system and entering the PeopleCode Debugger mode in Application Designer.

> **Note**
>
> You must use a unique user ID when you're performing PeopleCode debugging, as opposed to using a shared user ID, such as those that PeopleSoft delivers, for example QEDMO, PS, or VP1. Shared IDs are likely to be used by others that are connecting to the same test database, which can affect debugging.

Configuring PeopleCode Trace

Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **Set PeopleCode Trace Options** to access the Trace PeopleCode page.

You use this page to change the PeopleCode tracing options while online. Use Trace PeopleCode to create a file displaying information about PeopleCode programs processed from the time that you start the trace. This page does not affect trace options that are set in PeopleSoft Configuration Manager.

You can also access the PeopleCode trace settings (with slightly different labels) when you sign in to the PeopleSoft application in a browser, by selecting the Set Trace Flag link below the Sign In button. You enable the Set Trace Flag link to appear on the sign-in page, by accessing the Debugging tab of the Web Profile Configuration page for your web profile.

Field or Control	Description
Trace Evaluator Instructions	Select to show a line-by-line trace of the program
List Evaluator Program	Select to show the code of the PeopleCode program.
Show Assignments to Variables	Select to show variable assignments.
Show Fetched Values	Select to show values that are from PeopleCode Fetch call.
Show Stack	Select to display the PeopleCode evaluator's stack after each PeopleCode (internal) instruction.
Trace Start of Programs	Select to show the starting and ending points of the program.
Trace External Function Calls	Select to show calls to application written functions.
Trace Internal Function Calls	Select to show the calls to PeopleTools built-in function calls.
Show Parameter Values	Select to show function parameter values.
Show Return Parameter Values	Select to show function return parameter values.
Show Each	Select to trace each statement in the program.

Field or Control	Description
Suppress Logs in Loop	Select to suppress the trace logs when the trace program involves a large number of for, repeat, or while loops. The trace will print the PeopleCode executed during just the first pass. Note that if the for, repeat, or while loop contains any branching statements (that is, if/else or evaluate/when) which evaluate differently between the first pass and subsequent passes, only the PeopleCode executed in the first pass will be reflected in the trace logs. For that reason it is not recommended to use this flag when debugging for, repeat, and while loops that contain branching statements. **Note:** You see this option only when you sign in to the PeopleSoft application in a browser. It is not available from the Trace PeopleCode page, or on the Configuration Manager - Trace tab.
Disable Value Truncation	Select to prevent the system from truncating PeopleCode trace values. When you select this setting, it is applicable if TracePC includes any of the following values: • Show assignments to variables (4). • Show fetched values (8). • Show parameter values (512). • Show function return value (1024). The default trace setting is to truncate PeopleCode trace values. In addition to the Trace PeopleCode page, the **Disable Value Truncation** option is available on: • Sign-in page (PeopleCode trace settings). • Configuration Manager - Trace tab. • Selective Trace page (**PeopleTools**, and then **Application Engine**, and then **Selective Tracing**). • Application Designer - PeopleCode Log Options dialog box (**Debug**, and then **Options**, and then **Log Options**).

Note

The Trace PeopleCode utility decreases system performance because of the overhead that occurs during the monitoring and recording of all PeopleCode actions.

The check boxes on this page correspond to the options on the Trace tab in Configuration Manager. However, the selections that appear on this page do not necessarily reflect those that are made in Configuration Manager. While the Configuration Manager settings are stored in the Windows registry and used at each sign in, the settings in the Utilities page only apply to the current online session, and, once set, they override the Configuration Manager's settings.

The benefit of using this page to control PeopleCode tracing is that you can turn it on and off without having to restart PeopleTools, and without resetting the Configuration Manager settings. Keep in mind, though, your selections are not enabled until you save the page.

To enable/disable PeopleCode tracing while online:

1. Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **Set PeopleCode Trace Options**.

   The Trace PeopleCode page appears.
2. Select or deselect the desired Options.
3. Save the page.

   If you selected any of the check boxes, the system starts writing to the trace file.

Related Topics

- Specifying Trace Settings
  Select the Trace tab.
- PeopleCode Language Reference: SetTracePC function
- Portal Technology: Configuring Web Profiles

Masking Sensitive Data in PeopleCode Trace File

Prior to this release, PeopleCode trace files displayed the value of any record field including passwords, sensitive data, or personally identifiable (PII) data through PeopleCode operations, such as variable assignments, function parameters and other operations. From this release, PeopleTools enables you to prevent displaying passwords and sensitive or PII data in PeopleCode trace files.

Applications use the Data Privacy Framework to identify and mark applications-owned record fields as sensitive or PII data. For a description on using data privacy framework, see "Understanding Data Privacy" in PeopleSoft HCM Enterprise Components online help.

PeopleTools does not have access to the Data Privacy Framework provided by applications. Therefore, for PeopleTools-owned record fields, PeopleTools provides the Do Not Trace Value check box on the Record Field Properties dialog in Application Designer. Select this check box to identify and ensure proper masking of passwords and any potential sensitive or PII record fields owned by PeopleTools. However, Oracle recommends that application developers and customers mainly use the Data Privacy Framework and use the Do Not Trace Value check box in rare situations.

See Application Designer Developer’s Guide: Setting Record Field Use Properties.

When you select the Do Not Trace Value check box for a record field or identify a record field as sensitive or PII through Data Privacy Framework, the value of the record field is masked with a single asterisk (*) on the PeopleCode trace file.

To unmask the values of a PeopleTools-owned record field, deselect the Do Not Trace Value check box. To unmask application-owned record fields, refer to the documentation on Data Privacy Framework.

Configuring SQL Trace

Select **PeopleTools**, and then **Utilities**, and then **Debug**, and then **Set SQL Trace Options** to access the Trace SQL page.

You use this page to change the SQL tracing options while you're online. Your Configuration Manager settings are not affected:

Field or Control	Description
Trace SQL Statement	Select to show the SQL statement.
Trace SQL Bind	Select to show bind values for SQL statements that have parameter markers.
Trace SQL Cursor	Select to show connect, disconnect, commit and rollback calls.
Trace SQL Fetch	Select to show fetch call for Select Statement.
Trace SQL API	Select to show other API calls (Execute, Describe, and so on.)
Trace SQL Set Select Buffer	Select to show Binds for Select columns.
Trace SQL -- Database Level	Select to specify low-level tracing at the database API (ODBC, ct-lib, and so on.)
Trace SQL -- Manager Level	Select to show calls for Cache calls.

The check boxes on the Trace SQL page correspond to options on the Trace tab in the Configuration Manager. However, the selections that appear on this page do not necessarily reflect those that are made in the Configuration Manager. The displayed page selections are not enabled until you save the page.

To enable or disable SQL tracing while online:

1. Select or deselect the desired trace options.
2. Save the page.

   If you select any of the check boxes, the system starts writing to the trace file.

You can also access the SQL trace settings (with slightly different labels) when you sign in to the PeopleSoft application in a browser, by selecting the Set Trace Flag link below the Sign In button. You enable the Set Trace Flag link to appear on the sign-in page, on the Debugging tab of the Web Profile Configuration page for your web profile.

Related Topics

- Specifying Trace Settings
  Select the Trace tab.
- PeopleCode Language Reference: SetTraceSQL function
- Portal Technology: Configuring Web Profiles

Configuring Component Processor Trace

The component processor trace settings that are available when you click Set Trace Flags on the PeopleSoft application sign-on page.

You enable the Set Trace Flag link to appear on the sign-in page, on the Debugging tab of the Web Profile Configuration page for your web profile.

See Portal Technology: Configuring Web Profiles

Field or Control	Description
Page Structures at Init	Trace how page is constructed when page is initialized.
Component Buffers at Init	Dump component buffer when page is initialized.
Component Buffers before/after service	Dump component buffer when executing a service, for example, ICPanel.
Component Buffers after scrollselect	Dump component buffer after a ScrollSelect PeopleCode is executed.
Component Buffers after modal page	Dump component buffer after displaying a modal page.
Component Buffers before Save	Dump component buffer before save action is done.
Component Buffers after row insert	Dump component buffer after inserting a row.
Default Processing	Trace the default values processing of all fields if there are default values.
PRM Contents	Dump PRM information. PRM is an internal manager object of PeopleTools.
Internal counters (debug build only)	Trace internal object counters information. **Note:** This option is not supported for end user.
Memory stats at Init	Dump memory usage statistics after page is initialized.
Keylist Generation	Trace keylist generation processing.
Work Record Flagging	Trace the usage of Work Record Flagging. You can see which records are flagged as work records. Work Records Flagging is a performance feature of PeopleTools.
Related Displays	Trace the processing of related display fields.

Configuring Page Generation Trace

The page generation trace settings that are available when you click Set Trace Flags on the PeopleSoft sign-on page.

You enable the Set Trace Flag link to appear on the sign-in page, by accessing the Debugging tab of the Web Profile Configuration page for your web profile.

See Portal Technology: Configuring Web Profiles

Field or Control	Description
Log page generation errors	Log page generation errors.
Show table layout	Show table layout via cell borders in a generated page.
Source annotation for overlap	Annotate field overlap in HTML source.
Detailed table gen trace	Detailed trace of table generation algorithm.

Field or Control	Description
Inline stylesheet	Inline stylesheets are inserted into generated pages. **Note:** No inline stylesheets will be generated on fluid homepages or other fluid pages.
Inline javascript	Unminified javascript.
Extra markup needed for QA	**Note:** Required for internal QA testing.
Format source	Format source so that HTML is readable.
Save source in files	Save each generated page in log directory.
Include javascript debugging	Log all request parameters on each service.
Log unknown parameters	Write log message for unrecognized query parameters.

Enabling IDDA Logging

This section provides an overview of how to enable IDDA logging and discusses how to work with and view log results.

Understanding IDDA Logging

System administrators typically use numerous monitoring and logging utilities to diagnose system issues and internet applications. Such logging utilities include:

- TCPMON
- ieHttpHeaders
- Access log
- Heap dump
- Thread dump

All of these utilities provide different information, but each helps an administrator gain insight and detailed information related to specific system behavior. PeopleSoft provides a variety of logging and tracing mechanisms as well, enabling you to gather vital information at various levels of the architecture, including database server, application server, web server, and so on.

The PeopleSoft Instrumented Development Diagnostic Aid (IDDA) logger, enables you to gather specific information about various areas within the PeopleSoft Internet Architecture and PeopleSoft Interaction Hub, including:

- PeopleSoft Internet Architecture processing.
- Integration Broker.
- Reporting, Report Repository.
- Portal.
- Caching.
- File processing.
- Security, authentication.
- Performance Monitor.

- Jolt.
- Email attachment viewing.

Typically, administrators only run IDDA traces when instructed to do so by Oracle support contacts, looking for specific information. However, it can also be a useful troubleshooting tool.

> **Note**
>
> Whenever tracing or logging is enabled, you should always expect a certain degree of performance degradation as the system incorporates the overhead involved with logging. Disable logging when you finish troubleshooting.

Enabling IDDA Logging

To enable IDDA logging:

1. Select **PeopleTools**, and then **Web Profile**, and then **Web Profile Configuration**, and open the current web profile.
2. Select the Custom Properties page.
3. Add a new row, and enter these values:

Column	Value
Property Name	*IDDA*
Validation Type	*Number*
Property Value	The sum of the bit values of the functional area(s) you want to log. For example, if you wanted to log PIA (1) and Portal (8), you enter 9.

4. Click **Save**.
5. Restart the PeopleSoft site.

Working with IDDA Functional Categories

The IDDA logger, gathers information for a wide range of technology within the PeopleSoft Internet Architecture. Depending on the needs of your troubleshooting, you can select different areas to log.

The different areas of technology are referred to as functional categories in the context of the IDDA logger. Each functional category is assigned a bit value (in a 32-bit space). When you enable IDDA logging, you enter specific bit values *or* the sum of the bit values of different areas.

The IDDA functional categories are:

Bit Value	Functional Category
1	PeopleSoft Internet Architecture
2	Integration Broker
4	Report repository

Bit Value	Functional Category
8	Portal
16	Web server caching
32	File processing (attachments)
64	Authentication
128	Performance Monitor
512	Jolt
1024	Email attachment viewing

The type of information included in the log message differs per category. The log message is free format and can display any information that helps troubleshooting for that particular area, such as error codes, exception messages, and so on.

Configuring Logging Options

Once you've enabled IDDA logging, you can modify configuration options in the logging.properties file, which you can find in this location:

<*PS_CFG_HOME*>/webserv/<*domain_name*>/piaconfig/properties

The relevant configuration properties are:

Property	Description
.level	Sets the global logging level. • OFF: Disables all IDDA logging. • SEVERE: Displays server issues, such as premature termination and failure. • WARNING: Displays less severe issues, such as configuration issues. • INFO: Displays basic operational information, such as starting and stopping. • FINE, FINER, FINEST: Displays internal non-critical informational messages. • ALL: Enables all logging levels. Default value is *INFO*. WARNING and SEVERE messages are always logged, unless IDDA logging is set to OFF. INFO or FINE, FINER, FINEST messages are only logged if the IDDA value is greater than zero.
java.util.logging.FileHandler.pattern	Sets the naming style for the log file and the output directory location. Default value is: *./servers/PIA/logs/PIA_servlets%u.log* If you are running a multi-server, distributed environment (for clustering and failover purposes), the default value for java.util.logging.FileHandler.pattern is not applicable. You must set this property to point to a valid location in order to collect logging messages for a particular server. Logging output is tab-delimited.

Property	Description
java.util.logging.FileHandler.limit	Limits the size of the output file in bytes. When set to 0, there is no size limit. Default value is *0*.
java.util.logging.FileHandler.count	Sets the total number of log files. Default value is *5*. If the value is greater than 1, the system writes to a rotating set of log files. When the file reaches a given size limit or the web server restarts, the system ends writing to the current file and begins writing to a new file. The system names each file in the sequence it is saved by adding "0", "1", "2", (and so on) into the base filename.

Viewing IDDA Logging Output

The output log files contain:

- Machine header information
- Log message information

Working with Machine Header Information

Each log file displays the following machine environment information at the top of each log file.

Header Entry	Description
Timestamp	Date, time, time zone.
PeopleTools Release	PeopleTools version number.
os.name	Operating system.
os.version	Operating system version.
os.arch	Operating system architecture (such as x86 for Windows servers).
java.version	Java version number.
java.vendor	Java vendor.
java.vm.info	Mode of the Java virtual machine (JVM), such as "compiled mode."
java.home	Java installation directory.
user.dir	The value of the user.dir property.
java.class.path	CLASSPATH setting on the server.
.level	Logging level, such as INFO, SEVERE, WARNING, and so on.
java.util.logging.FileHandler.pattern	Logging output directory and file naming convention.
Trace	Current IDDA functional group(s) being logged (integer reflecting the sum of the bit values assigned to each group).

Working with Log Message Information

Each entry in the log file contains this information.

Log Message Data	Description
Timestamp	Date, Time and Time zone. For example, 2/7/09 4:00:39 PM PST
Sequence	Tracks the sequential order of the messages. Starting at 1, the system increments by 1 per each log message.
Thread ID	Java thread ID.
Logging group	Bit value representing the functional grouping, as in 1 for PeopleSoft Internet Architecture.
SRID	A base64 encoded field uniquely associating a user session and its service request in the PeopleSoft-generated online transaction processing (OLTP) log files. For example, 17F2TbFWJw69UXCzfobVhZ.
TOP Instance ID	Top correlation attribute in PPM's PMU Tree. It is a 64-bit integer. For example, 455266533594. For more information on PMU Tree, see Performance Monitor: Viewing Open PMU Trees.
Operator ID	PeopleSoft user ID.
Source class	Class name of where the message is logged. For example, `psft.pt8.auth.PSAuthenticator`
Source method	Method name of where the message is logged. For example, `SetCookie`
Log message	The actual log message.

Viewing Log Contents

The log file is a tab-delimited text file and can be opened in any standard text editor, such as Notepad or Textpad. Because the output is tab-delimited, you can also use a spreadsheet application, such as Microsoft Excel, for more efficient analysis. For example, viewing the output within a spreadsheet enables you to apply filters to columns and only view specific log messages, which can be helpful with large files.

11

Working with Jolt Configuration Options

Configuring Jolt Failover and Load Balancing

This section provides an overview of how to configure Jolt failover and load balancing.

Configuring Weighted Load Balancing

With weighted load balancing, you can set the “weight” of the load, or amount of requests, being directed to a particular server. Weight values are integers 1–10, with 1 being low and 10 being a heavy load. Servers that can handle extra work can take heavy loads, while servers that are either less powerful or are being used in other capacities can take lower loads. You specify weighted load balancing by modifying the server values in the psserver property in the PeopleSoft Internet Architecture configuration.properties file, using the following format.

```
psserver=Host1:Port1#Wt,Host2:Port2#Wt
```

For example,

```
psserver=appserver1:9000#3,appserver2:9010#1
```

In this case, appserver1 would receive 3x more requests than appserver2.

Configuring Jolt Failover

You an also specify strict failover assignments with weighted load balancing, with the following options:

- Strict failover with weighted backup.
- Strict failover with sequential backup.

You add the failover string within curly brackets at the end of the server entry.

```
psserver=<host>:<port>#wt{failover servers}
```

With the failover string, you can set weighted backup by separating failover server with a comma (,).

```
psserver=Host1:Port1#Wt{Host3:Port3#Wt,Host4:Port4#Wt},Host2:Port2#Wt
```

In this case, Host 3 and Host 4 are failover servers when Host 1 is down. You can assign weighted load balancing to the backup servers just as you would a primary server.

You can also set a sequential backup with your failover string. To set sequential backup, you separate multiple backup servers using a semicolon (;).

```
psserver=Host1:Port1#Wt{Host3:Port3;Host4:Port4},Host2:Port2#Wt
```

In this case, the system assigns Host 4 the requests when both Hosts 1 and 3 are down.

Configuring Dynamic Reloading of the Application Server Connect String and Web Profile Attributes

This section describes how to manage changes to the psserver string (application server connect string) and the web profile attributes.

Configuring Dynamic Reloading of the Application Server Connect String

The DynamicConfigReload property enables the PIA domain to reload the psserver string (application server connect string) dynamically. That is, after modifying the psserver string to include new servers, replace existing servers, or remove existing servers from the string, you do not need to restart the PIA domain.

With DynamicConfigReload enabled, the system automatically detects when the psserver string has been updated and ensures the PeopleSoft site maintains seamless connections with the application server domains listed in the psserver string.

If the Jolt session connection pool was initially started with a single application server, for example App1, all pooled connections are established to App1. When a new application server (App2) is later added to the connection string using dynamic reloading, the pool does not immediately rebalance or redistribute existing connections. Instead, it must first exhaust all current connections to App1. Only after all those connections are used up will the pool begin establishing a new set of connections to App2.

To enable dynamic psserver reloading:

1. Using a text editor, open the configuration.properties file in the web server deployment folder located here:

   <*PS_CFG_HOME*>\webserv\<*web_server*>\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs\<*sitename*>\configuration.properties
2. Set DynamicConfigReload to *1*. For example:

   ```
   DynamicConfigReload=1
   ```

To disable dynamic psserver reloading:

1. Using a text editor, open the configuration.properties file in the web server deployment folder located here:

   <*PS_CFG_HOME*>\webserv\<*web_server*>\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs\<*sitename*>\configuration.properties
2. Set DynamicConfigReload to *0*. For example:

   ```
   DynamicConfigReload=0
   ```

3. Restart the PIA domain manually for it to recognize any changes made to the psserver string.

Reloading Web Profile Attributes without Restarting the Web Server

Use the reloadWebProfile script to reload all web profile attributes without restarting the web server. The script applies any changes to the web profile for the web site.

Oracle recommends that you run the reloadWebProfile script at a point in time when there are the least number of users logged in to PIA, to minimize any potential impact.

1. Change directory to <*PS_CFG_HOME*>\webserv\<*web_server*>\piabin.
2. Run reloadWebProfile.cmd (Windows) or reloadWebProfile.sh (UNIX).

Configuring Jolt Session Pooling

Jolt session pooling is enabled by default. Jolt session pooling enables web server connections to be shared between user sessions, which reduces the usage of system resources, such as threads and file descriptors.

You control session pooling by modifying the joltPooling property in the configuration.properties file per site.

```
joltPooling=true
```

To enable Jolt session pooling, set the property value to *true*, and to disable Jolt session pooling set the property value to *false*.

Configuring Parallel Pagelet Loading

You can use the parallelLoading property in the configuration.properties file to control how your portal loads pagelets. Pagelets can be loaded sequentially or in parallel.

Loading pagelets sequentially refers to a single-threaded process, downloading multiple pagelets one at a time. Loading pagelets in parallel refers a multi-threaded process where multiple pagelets can be loaded at the same time. To enable parallel pagelet loading, set the parallelLoading property to true.

For example:

```
parallelLoading=true
```

> **Note**
>
> To load pagelets in parallel, Jolt pooling must be enabled.

To disable parallel loading of pagelets, set the parallelLoading property to false.

Pagelets with global variables cannot be loaded in parallel; such pagelets can only be loaded sequentially. If you have set the parallelLoading property to true and if any of the pagelets contain global variables, you must mark the pagelet with a content reference attribute BGLOBALVALUES set to true. When the pagelet using global variable is marked with the content reference attribute BGLOBALVALUES set to true, then the pagelet will be loaded sequentially and other pagelets will be loaded in parallel. For example, if the pagelet with global variables is part of a Dashboard or WorkCenter, then the pagelet marked with BGLOBALVALUES will be loaded sequentially and other pagelets within the Dashboard or WorkCenter will be loaded in parallel.

Configuring Domain Connection Password

The DomainConnectionPwd setting in the [Security] section of the application server domain, sets the domain connection password for all PeopleSoft Internet Architecture Jolt connections to that application server domain. It is not a required password setting, but *if set*, for the Jolt session to connect successfully to the application server domain, the DomainConnectionPwd property value in the configuration.properties file needs to match the DomainConnectionPwd parameter set for the application server domain.

You can encrypt the password value using the PSCipher utility.

Related Topics

- DomainConnectionPwd
  The domain connection password adds an extra layer of security between the application server domain and any connections made to it. This password enables you to further prevent unauthorized clients from establishing connections to an application server domain. It is recommended to use PSADMIN to update this value. The value can be up to 30 characters.

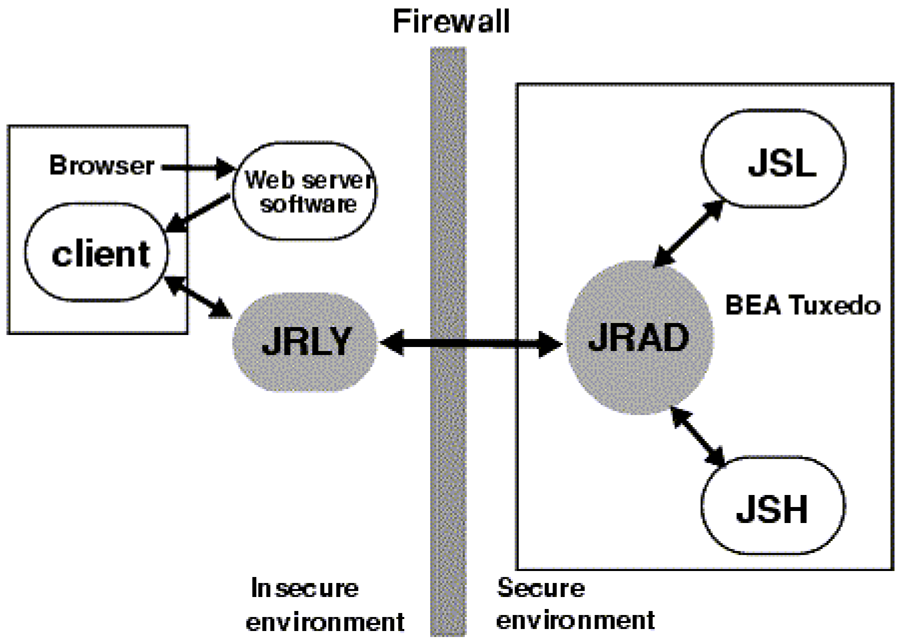
Using Jolt Internet Relay

This section provides an overview of the Jolt Internet Relay architecture and how to use it.

> **Note**
>
> Jolt Internet Relay was used in previous releases at some sites for specific security requirements. Currently, it is not a widely used feature and should only be implemented in specific circumstances or by suggestion of Oracle support staff.

Jolt Internet Relay Architecture

Jolt Internet Relay provides an environment in which the PeopleSoft web server and application server can be further decoupled. This provides greater security at sites where security is of utmost importance. Jolt Internet Relay routes messages from a Jolt client to a Jolt Server Listener (JSL) or Jolt Server Handler (JSH), and eliminates the need for the JSL, JSH, and Tuxedo application to run on the same machine as the web server. Communication takes place between the JRLY and JRAD elements rather than between the Jolt client and JSL/JSH processes. Traditionally an application server domain opens between 2 and 6 ports for such communications. The use of Jolt relay restricts this to one port per domain on the application server machine. This enables an administrator to open just one port on the application server machine. The following diagram illustrates this feature:



Jolt Internet Relay consists of two elements: Jolt Relay (JRLY) and Jolt Relay Adapter (JRAD). It's important to understand the difference between these two elements.

JRLY consists of a standalone program and configuration file; the program runs on the same machine as the web server. JRLY receives Jolt messages from a PeopleSoft web application and routes those messages to JRAD on the application server. It receives the Jolt message through one port, the LISTEN port, and connects to the JRAD by using another port, the CONNECT port. JRLY is sometimes referred to as a front-end relay.

JRAD runs on the same machine as the application server. It's configured on the application server domain as part of the PeopleSoft PSADMIN domain configuration procedure. JRAD listens for JRLY messages on its LISTEN port and transfers the message to the JSL or JSH. JRAD is sometimes referred to as a back-end relay.

> **Note**
>
> Implementing Jolt Relay can impede performance. Always perform testing with typical production system load to ensure it will meet your service level requirements.

Implementation Considerations

Keep the following points in mind as you configure the Jolt Internet Relay components:

- The jrly binary and its corresponding jrly.config file must exist in the same directory. To start multiple Jolt Relays on a machine, copy the jrly binary and jrly.config into each subdirectory, modify the parameters in the jrly.config file, and start Jolt Relay. On Windows, you can define multiple Jolt Relay services on a machine.
- You can start the JRLY process before or after you start JRAD. The JRLY attempts to connect to JRAD on the client request. If the JRLY is unable to connect to the JRAD, the

client is denied access and disconnected. The connection will be retried upon the first use of PeopleTools..

- If you're installing Jolt Internet Relay on UNIX and anticipate a large number of concurrent connected clients, increase the file descriptors limit before running the JRLY executable.
- At runtime, if you get the following message:

```
[FRi JUNl 6 20:25:11 1997] JRLY:accept():accept failed,⇒
 err no: 23, strerror: File table overflow
```

  PeopleSoft recommends that you increase the MaxUSERS kernel parameter and regenerate the kernel.
- If you're unable to connect, make sure that you check the following items:
  - Port numbers do not match.

    Print out the jrly.cfg file and the psappsrv.cfg file and compare the port numbers that you specified.
  - Make sure that the application server is running.
  - Make sure that JRLY is running.
- Jolt Internet Relay can be installed on an intermediate machine rather than the web server machine if necessary. This extra level of indirection can cause performance degradation.
- Make sure that JRAD is running on the application server and that you configure JRAD using PSADMIN.

Configuring JRLY

Configuring JRLY is identical on UNIX and Windows.

To configure JRLY, navigate to *TUXDIR*\udataobj\jolt\relay and open jrly.config in a text editor.

> **Important**
>
> On UNIX, you can edit this configuration file by using VI or an equivalent editor. However, on Windows, you must edit the file using an editor that preserves the file's UNIX line feeds. WordPad is valid for this purpose, but Notepad is not.

Modify the parameters in the configuration file to reflect the site specifications, as follows:

Parameter	Description
LOGDIR	*LOGDIR* specifies the directory where JRLY creates access and error log files. This directory must exist; the JRLY program does not start if it can't find this directory. The path that you specify for LOGDIR should be an absolute path (starting from / on UNIX systems, starting from a drive letter on Windows systems). The JRLY accepts relative path names, but LOGDIR is relative to the directory from which the JRLY program is started, unless you specify it as an absolute.

Parameter	Description
ACCESS_LOG	*ACCESS_LOG* specifies the name of the file where JRLY records access information. This log file is created in *LOGDIR*. If the log file already exists, the most recent information is appended to it. This parameter can be any valid file name. Everything after the equals sign (=) to the end of the line is considered as part of the file name, but leading and trailing blanks are ignored. **Note:** If the JRLY program can't create this file or open it for appending, the program exits.
ERROR_LOG	*ERROR_LOG* specifies the name of the file where JRLY records error information. This file follows all the rules that apply to the *ACCESS_LOG* parameter. JRLY_error_log is created in /tmp.

LISTEN	*LISTEN* specifies the host and port on the current machine (that is, the machine where you're installing Jolt Relay). JRLY listens for client connections. The following formats are acceptable:
	`LISTEN=192.0.2.100:9000`
	`LISTEN=//192.0.2.100:9000`
	`LISTEN=server_name:9000`
	`LISTEN=//server_name:9000`
	Specify the port number in decimal; it must match the port number that is specified by the *psserver* parameter in the configuration.properties file for the PIA web application.
	Note: If a machine has multiple network interfaces, you should use the IP address notation, because specifying the host name could be ambiguous (the result is OS dependent). If the JRLY program can't establish a network listening end-point at the host and port specified, it prints an error and exits.
	The host name that's specified for this parameter must be the name of the host on which the program is running.
	Note: You can create multiple configuration files to run multiple instances of JRLY. Each configuration file must specify a different port number for this parameter.

Parameter	Description

CONNECT	*CONNECT* specifies the location of the JRAD machine and process port on the application server machine to which the JRLY program connects. A JRLY program communicates only with a single JRAD. The address you specify for this parameter must match the JRAD listener address that's on the application server machine (check the PSAPPSRV.CFG file in *PS_CFG_HOME*/appserv/*domain*). The JRAD doesn't have to be running when you start the JRLY program. JRLY attempts to connect to the JRAD when it first starts, and if the JRAD is not available, JRLY tries again whenever a new client connects to it. You can use any of the following formats for this parameter:
	`CONNECT=192.0.2.100:9100`
	`CONNECT=//192.0.2.100:9105`
	`CONNECT=server_name:9105`
	`CONNECT=//server_name:9105`
	Note: PeopleSoft has found that machine address formats are operating system and environment dependent. If one fails to connect to the application server, try another format.

SOCKETTIMEOUT	*SOCKETTIMEOUT* specifies the duration (in seconds) for which the Jolt Internet Relay Windows service blocks the establishment of new socket connections to allow network activity (new connections, data to be read, closed connections) to complete. It's valid only on Windows machines. *SOCKETTIMEOUT* also affects the Service Control Manager (SCM). When the SCM requests that the service stop, the SCM needs to wait at least the number of seconds specified by this parameter.

Configuring JRAD

The JRLY connect port connects to the JRAD listener port that is specified on the application server machine. JRAD then routes the message to Jolt, either using the JSL for initial connection from a web client, or using the JSH for all subsequent connections from a web client. The return message follows the same path in reverse.

To configure JRAD:

1. Launch the PSADMIN utility.
2. Navigate to the PeopleSoft Domain Administration menu and select *Configure this domain*.
3. In the Quick Configure menu, select the number for the Jolt Relay option, to enable Jolt Internet Relay.
4. Select the JRAD Port option, and enter the appropriate port number for the JRAD Port.

> **Note**
>
> The JRAD (listener) port number must match the JRLY connect port that you previously configured.

Running Jolt Internet Relay

This section discusses how to:

- Use the JRLY administration program.
- Run Jolt Relay on Windows.
- Run Jolt Relay on UNIX.

Using the JRLY Administration Program

You use the jrly command located in *TUXDIR*\udataobj\jolt\relay to administer Jolt Relay on all platforms. You can use the following jrly command options at any time:

- jrly -version

  Display the current version of the JRLY binary.
- jrly -help

  Display a summary of command-line options with brief descriptions.

Running Jolt Relay on Windows

On Windows, you set up Jolt Relay to run as a service. On other platforms you must run Jolt Relay directly.

If you want to install multiple Jolt Relay services, you must specify a string to be used as a *display suffix* that uniquely identifies each additional service you install. You subsequently use the suffix to identify each service it commands. An additional service with the suffix MyJoltRelay, for example, is called *Oracle Jolt Relay_MyJoltRelay*, but you refer to it using only the suffix. You can omit the suffix when installing only one of these services, which becomes the default Jolt Relay service, called *Oracle Jolt Relay*.

> **Note**
>
> All administrative commands in the following table except -start and -stop require that you have write access to the Windows registry. The -start and -stop commands require that you have Windows service control access. These requirements are based on Windows user restrictions.

Command	Description
jrly -install [*display_suffix*]	Install JRLY as a Windows service.
jrly -remove [*display_suffix* | -all]	Remove one instance, all instances, or the default JRLY Windows service.

Command	Description
jrly -set [-d *display_suffix*] -f *config_file*	Update the registry with the full path of a new configuration file for the specified JRLY service. **Note:** You can run multiple Jolt Relay services by specifying a different display suffix along with the name of a different configuration file for each installed service. Each configuration file must contain a unique value for the LISTEN parameter that specifies a different port. This is essential to avoid port clashes when running the services concurrently. You must run this command before the service starts.
jrly -manual [*display_suffix*]	Set the start/stop to manual. This command sets the specified JRLY service to be manually controlled, using either the command-line options or the Service Control Manager (SCM).
jrly -auto [*display_suffix*]	Set the start/stop to automatic. This command sets all the operations for a specified JRLY service to be automatically started when the OS boots and stopped when the OS shuts down.
jrly -start [*display_suffix*]	Start the specified JRLY service.
jrly -stop [*display_suffix*]	Stop the specified JRLY service.

Running Jolt Relay on UNIX

This section discusses how to start and stop Jolt Relay directly from a command line on UNIX.

To start Jolt Relay on UNIX:

1. Change directories to the Jolt Relay directory within your Tuxedo installation:

   ```
   cd $TUXDIR/udataobj/jolt/relay
   ```

2. Run the following command:

   ```
   jrly -f jrly_config &
   ```

   Where *jrly_config* is the name of a Jolt Relay configuration file.

   You can run multiple instances of Jolt Relay by using a different port for each instance. You run JRLY once for each instance, and specify a different configuration file each time. Each configuration file must contain a value for the LISTEN parameter that specifies a different port.

   The ampersand character (&) causes JRLY to run in the background.

To shut down Jolt Relay on UNIX, use the UNIX kill -9 command.

12

Setting Up Oracle Wallet

Understanding Oracle Wallet

Oracle Wallet is a container that stores authentication and signing credentials.

Trusted certificates are stored in the Oracle Wallet when the wallet is used for security credentials.

PeopleSoft enables you to create an Oracle Wallet in two ways:

- ORAPKI command line — The ORAPKI tool is available with Oracle database, so this tool can be used only by those users who have a license for Oracle database.
- OpenSSL utility — Users who do not have a license for Oracle database can use this utility to create their own certificates.

After creating an Oracle Wallet, you must configure SSL for the Workstation Listener and Jolt Listener ports to ensure secure client and server communications.

Setting Up Oracle Wallet Using ORAPKI

To set up Oracle Wallet using ORAPKI command line in Oracle database, complete the following steps:

1. Create a server wallet for Application server domain.
   a. Create server wallet with the following command:

      ```
      orapki wallet create -wallet <wallet_name> -pwd <password>
      ```

      For example:

      ```
      C:\oracle\product\12.1.0\dbhome_2\BIN\orapki wallet create -wallet
      wallet.psft -pwd password
      ```

   b. Add self-signed certificate to the wallet.

   ```
   orapki wallet add -wallet <wallet_name> -dn "user_dn" -keysize 1024 -
   self_signed -validity 3650 -pwd <password>
   ```

   For example:

   ```
   C:\oracle\product\12.1.0\dbhome_2\BIN\orapki wallet add -wallet
   wallet.psft -dn "CN=psft,C=CN" -keysize 1024 -self_signed -validity 3650 -
   pwd password
   ```

2. Create client wallet for Workstation clients.

a. Create Workstation wallet with the following command:.

```
orapki wallet create -wallet <client_wallet_name> -pwd <password>
```

For example:

```
C:\oracle\product\12.1.0\dbhome_2\BIN\orapki wallet create -wallet
wallet.wscpsft -pwd password
```

b. Add self-signed certificate to the wallet.

```
orapki wallet add -wallet <client_wallet_name> -dn "user_dn" -keysize
1024 -self_signed -validity 3650 -pwd <password>
```

For example:

```
C:\oracle\product\12.1.0\dbhome_2\BIN\orapki wallet add -wallet
wallet.wscpsft -dn "CN=not_used,C=CN" -keysize 1024 -self_signed -
validity 3650 -pwd password
```

3. Export public key from server certificate.

```
orapki wallet export -wallet <server_wallet> -dn "user_dn" -cert
<server_wallet>/<certificate> -pwd <password>
```

For example:

```
C:\oracle\product\12.1.0\dbhome_2\BIN\orapki wallet export -wallet
wallet.psft -dn "CN=psft,C=CN" -cert wallet.psft/psft.cer -pwd password
```

4. Add public key from server certificate to client certificate.

```
orapki wallet add -wallet <client_wallet_name> -trusted_cert -cert
<server_wallet>/<certificate> -pwd <password>
```

For example:

```
C:\oracle\product\12.1.0\dbhome_2\BIN\orapki wallet add -wallet
wallet.wscpsft -trusted_cert -cert $PWD/wallet.psft/psft.cer -pwd password
```

5. You can remove client wallet password using the following steps:

   a. `cd wallet.wscpsft`

   b. Linux: `cd wallet.wscpsft`

      Windows:`xcopy ewallet.p12 oldwallet.p12`

   c. `openssl pkcs12 -clcerts -nokeys -in oldwallet.p12 -out certificate.crt -password pass:`*`password`*` -passin pass:`*`password`*

   d. `openssl pkcs12 -cacerts -nokeys -in oldwallet.p12 -out ca-cert.ca -password pass:`*`password`*` -passin pass:`*`password`*

e. `openssl pkcs12 -nocerts -in oldwallet.p12 -out private.key -password pass:`*`password`*` -passin pass:`*`password`*` -passout pass:temp`

f. `openssl rsa -in private.key -out NewKeyFile.key -passin pass:temp`

g. Linux: `cat certificate.crt ca-cert.ca >PEM.pem`

   Windows: `type certificate.crt ca-cert.ca >PEM.pemtype certificate.crt ca-cert.ca >PEM.pem`

h. `openssl pkcs12 -export -nodes -in PEM.pem -inkey NewKeyFile.key -out ewallet.p12 -passout pass:TrustedCertsOnlyNoPWNeeded`

6. Add the server's certificate to the PIA's truststore.

   a. Export public key from server certificate.

```
 orapki wallet export -wallet <server_wallet> -dn "CN=psft,C=CN" -cert
<server_wallet>/<certificate> -pwd <password>
```

   For example

```
orapki wallet export -wallet wallet.psft -dn "CN=psft,C=CN" -cert
wallet.psft/psft.cer -pwd password
```

   b. Add the server's certificate to the client's truststore.

```
cd %ORACLE_HOME%\jdk\bin
keytool -import -file <server_wallet>/<certificate> -alias srv_cert -
keystore <PIA_HOME>\webserv\<DOMAIN_NAME>\piaconfig\keystore\pskey -
storepass password -noprompt
```

   For example:

```
keytool -import -file wallet.psft\psft.cer -alias srv_cert -keystore
<PIA_HOME>\webserv\<DOMAIN_NAME>\piaconfig\keystore\pskey -storepass
password -noprompt
```

Setting Up Oracle Wallet Using OpenSSL

To set up Oracle Wallet using OpenSSL, use the following command:

```
openssl pkcs12 -export -out ewallet.p12 -inkey server.key -in server.crt -
chain -CAfile caCert.crt -passout pass:password
```

This table lists the command options:

Field or Control	Description
`-export`	Indicates that a PKCS 12 file is being created.
`-chain`	Specifies that an attempt is made to include the entire certificate chain of the user certificate.
`-inkey`	Specifies the private key file.

Field or Control	Description
`-in`	Specifies the file that contains the user certificate and any other certificates in the certificate chain. **Note:** If the private key and the certificate chain are in the same file, the `-inkey` and `-in` parameters can specify the same file.
`-CAfile`	Specifies a file containing trusted certificates.
`-out`	Specifies the output file name, which must be *ewallet.p12* for an Oracle Wallet.
`-passin`	Specifies the password for the private key file.
`-passout`	Specifies the password for the newly created wallet.

To create a self-signed SSL certificate using OpenSSL, complete the following steps:

1. Create the server wallet.
   a. Create a new directory and change to the directory:

      ```
      mkdir wallet.server
      cd wallet.server
      ```

   b. `openssl genrsa -out server.key 4096`
   c. `openssl req -new -key server.key -out server.csr -subj <subject>`

      For example:

      ```
      openssl req -new -key server.key -out server.csr -subj '/C=CN/CN=psft'
      ```

   d. `openssl genrsa -out caCert.key 4096`
   e. `openssl req -new -x509 -days 1826 -key caCert.key -out caCert.crt -subj <subject>`

      For example:

      ```
      openssl req -new -x509 -days 1826 -key caCert.key -out caCert.crt -subj
      '/C=US/OU=Class 2 Public Primary Certification Authority/O=VeriSign'
      ```

   f. `openssl x509 -req -days 730 -in server.csr -CA caCert.crt -CAkey caCert.key -set_serial 01 -out server.crt`
   g. `openssl pkcs12 -export -out ewallet.p12 -inkey server.key -in server.crt -chain -CAfile caCert.crt -passout pass:`*`password`*

      For example:

      ```
      openssl pkcs12 -export -out ewallet.p12 -inkey server.key -in
      server.crt -chain -CAfile caCert.crt -passout pass:password
      ```

2. Create the Workstation wallet.

a. Create a new directory and change to the directory:

```
mkdir wallet.client
cd wallet.client
```

b. `openssl genrsa -out client.key 4096`

c. `openssl req -new -key client.key -out client.csr -subj '/C=CN/CN=wscpsft'`

3. Copy server's truststore to client:

   a. Change directory:

   ```
   cd ..
   ```

   b. `cp wallet.server\caCert.crt wallet.client`

   c. `cp wallet.server\caCert.key wallet.client`

   d. `cd wallet.client`

   e. `openssl x509 -req -days 730 -in client.csr -CA caCert.crt -CAkey caCert.key -set_serial 01 -out client.crt`

   f. `openssl pkcs12 -export -out ewallet.p12 -inkey client.key -in client.crt -chain -CAfile caCert.crt -passout pass:`*`password`*

   For example:

   ```
   openssl pkcs12 -export -out ewallet.p12 -inkey client.key -in
   client.crt -chain -CAfile caCert.crt -passout pass:password
   ```

4. To remove the client wallet password:

   a. `cp ewallet.p12 oldwallet.p12`

   b. `openssl pkcs12 -clcerts -nokeys -in oldwallet.p12 -out certificate.crt -password pass:`*`password`*` -passin pass:`*`password`*

   c. `openssl pkcs12 -cacerts -nokeys -in oldwallet.p12 -out ca-cert.ca -password pass:`*`password`*` -passin pass:`*`password`*

   d. `openssl pkcs12 -nocerts -in oldwallet.p12 -out private.key -password pass:`*`password`*` -passin pass:`*`password`*` -passout pass:temp`

   e. `openssl rsa -in private.key -out NewKeyFile.key -passin pass:temp`

   f. `cat certificate.crt ca-cert.ca >PEM.pem`

   g. `openssl pkcs12 -export -nodes -in PEM.pem -inkey NewKeyFile.key -out ewallet.p12 -passout pass:TrustedCertsOnlyNoPWNeeded`

5. Add the server's certificate to the PIA's truststore.

   a. `cd %ORACLE_HOME%\jdk\bin`

   b. `keytool -import -file <server_wallet>/<certificate> -alias srvcert -keystore <`*`PIA_HOME`*`>\webserv\<`*`DOMAIN_NAME`*`>\piaconfig\keystore\pskey -storepass `*`password`*` -noprompt`

For example:

```
keytool -import -file wallet.server\caCert.crt -alias srvcert -keystore
<PIA_HOME>\webserv\<DOMAIN_NAME>\piaconfig\keystore\pskey -storepass
password -noprompt
```

Configuring SSL for Workstation Listener and Jolt Listener

SSL encryption establishes data privacy for messages moving between the machines in an application. The industry-standard TLS 1.0 protocol is used for SSL encryption. You can use 56-bit, 128-bit, or 256-bit SSL ciphers.

Before you can configure SSL on Workstation Listener or Jolt Listener, you should be familiar with the SSL min and max notation. The defaults for these parameters are:

- For min: 0.
- For max: Number of bits that indicates the highest level of encryption possible for the installed SSL version. For example, 56, 128, or 256.

  You can change the default values and assign new values to min and max in the psappsrv.cfg file for your application server.

To configure SSL for Workstation Listener and Jolt Listener, you should make changes in:

- PSADMIN.
- Configuration Manager.
- PIA.

Modifications in PSADMIN

In PSADMIN, to configure SSL on Workstation Listener and Jolt Listener, follow these steps to provide the SSL port:

1. Select option **1** - Application Server.
2. Select option **1** - Administer a domain, and select the domain, such as APPDOM.
3. Select option **4** - Configure this domain.

   The Quick-Configure menu appears.
4. Select option **28** - WSL SSL Port to provide the SSL port for Workstation client configuration.
5. Select option **27** - WSL Port and provide the same port as for WSL SSL.
6. Select option **30** - JSL SSL Port to provide the SSL port for Jolt configuration.
7. Select option **29** - JSL Port and provide the same port as for JSL SSL.

Additionally, you should provide wallet information in the Oracle Wallet section.

1. Select option **15** - Custom configuration.
2. As each section appears, enter n to move to the next. When you reach the section you want to edit enter y.
3. In the Oracle Wallet section, enter:
   a. The wallet location for SEC_PRINCIPAL_LOCATION.

b. The wallet name for SEC_PRINCIPAL_NAME.
c. The wallet password for SEC_PRINCIPAL_PASSWORD.

For example:

```
#Values for config section - Oracle Wallet
SEC_PRINCIPAL_LOCATION=%PS_SERVDIR%\security
SEC_PRINCIPAL_NAME=psft
SEC_PRINCIPAL_PASSWORD=password
```

4. In the Workstation Listener section, enter 128 for WSL Min Encryption.

```
WSL Min Encryption=128
```

5. In the Jolt Listener section, enter 128 for JSL Min Encryption.

```
JSL Min Encryption=128
```

6. Load the configuration.

Modifications in Configuration Manager

To configure SSL on Workstation Listener, you must specify Workstation wallet information on the Profile tab of Configuration Manager.

1. Open Configuration Manager (pscfg.exe).
2. Select the Profile tab, and edit the user profile.
3. On the Database/Application Server tab, select a server of type Application Server or configure a server of type Application Server.
4. In the Port Number field, enter the SSL port specified for WSL (as specified on PSADMIN).
5. In the Wallet Location field, enter the location of the Oracle Wallet for client. The default value is *<PS_HOME>/bin/client/winx86/security*.
6. In the Wallet Name field, enter the name of the Oracle wallet for client.

See Configuring User Profiles.

Modifications in PIA

To configure SSL for Jolt Listener, set SSL port in PIA configuration.

1. In PSADMIN, shut down the web server (PIA) domain.
2. Edit the configuration.properties file (<*PIA_HOME*>\webserv\<*DOMAIN_NAME*>\applications\peoplesoft\PORTAL.war\WEB_INF\psftdocs\<*SITE_NAME*>).
3. Enter the SSL port for psserver.

```
psserver=<machine_name>:<SSL_port_for_JSL>
```

4. Enter the Java Keystore password.

```
#Keystore password for ssl connection
KeyStorePwd=encrypted_password
```

5. In PSADMIN, boot the Application Server and web server (PIA) domains.
6. Sign in to PIA to verify that the connection between the Application Server and PIA is working.

13 Working with Cloud Repositories

Using Cloud Repositories

Use supported cloud service providers as file repositories for File Attachments and Process Scheduler Report Distribution. The cloud storage providers that are supported for report distribution in the current release are:

- Oracle Cloud Infrastructure Object Storage service — an internet-scale, high-performance storage platform that offers reliable and cost-efficient data durability

  See Overview of Object Storage, https://docs.oracle.com/en-us/iaas/Content/Object/Concepts/objectstorageoverview.htm.

- Oracle Content and Experience Cloud — a cloud-based content hub to drive omni-channel content management and accelerate experience delivery

  See Oracle Content and Experience Cloud in the Oracle Help Center, https://docs.oracle.com/en/cloud/paas/content-cloud/.

Subject	Reference
Specify the URL for the cloud repository.	See URL Maintenance
Use a cloud storage service as a file repository for file attachments.	See Understanding PeopleSoft Support for Cloud Attachments
Configure the APIs to share or unshare file attachments.	See PeopleCode Language Reference: ShareAttachment function See PeopleCode Language Reference: UnshareAttachment function
Set up a report node to distribute reports to Oracle Cloud Infrastructure Object Storage service.	See Process Scheduler: Defining Oracle Cloud Infrastructure Report Nodes
Set up a report node to distribute reports to Oracle Content and Experience Cloud.	See Process Scheduler: Defining Report Nodes for Oracle Content and Experience Cloud

Understanding PeopleSoft Support for Cloud Attachments

PeopleSoft enables users to attach files from a cloud storage to PeopleSoft applications. Users can configure a cloud service provider and directly upload files from the cloud storage to PeopleSoft applications.

For example, if an HR recruiter wants to upload the resume of a potential candidate from cloud storage to PeopleSoft, he can do so without having to move away from his current browser page. If configured, the cloud service provider is displayed as an option in the File Attachment window. From here, the HR recruiter can choose the cloud service provider, attach the resume directly, and upload it to PeopleSoft applications.

This feature enables you to upload a single file or multiple files. PeopleSoft provides file attach functionality through the AddAttachment and MAddAttachment APIs.

> **Note**
>
> This feature is only available from Fluid pages.

Related Topics

- PeopleCode Language Reference: AddAttachment function
- PeopleCode Language Reference: MAddAttachment function
- PeopleCode Developer's Guide: Understanding the File Attachment Functions

Configuring Access to SharePoint for File Attachments and Report Repository

Use Microsoft SharePoint as a repository for file attachments and Process Scheduler Report Distribution.

For information on Process Scheduler reports, see Process Scheduler: Defining SharePoint Distribution Nodes.

1. Register an application in Microsoft Entra ID, formerly Azure Active Directory, as described in this section.
2. Use the URL Maintenance page to define an URLID pointing to SharePoint.
   See URL Maintenance.
3. Modify file attachment functions.
   See PeopleCode Developer's Guide: Understanding the File Attachment Functions.

Registering an Azure Application for SharePoint Repositories

Before you configure a PeopleSoft URLID for a SharePoint repository, you must register an application in Microsoft Entra ID, formerly Azure Active Directory. PeopleSoft uses this application registration to authenticate with Microsoft Graph and access the SharePoint site and document library that store files.

The registration process provides the values that you enter later on the PeopleSoft URL Maintenance and URL Properties pages, including the client ID, client secret, tenant ID, and SharePoint-related connection properties. You must have access to the Microsoft Azure portal and the required permissions to register applications and grant administrator consent for Microsoft Graph API permissions. See Defining SharePoint Repositories.

Registering the Application

To register an application for PeopleSoft access to SharePoint:

1. Sign in to the Microsoft Azure portal.
2. In the left menu, select Azure Active Directory or Microsoft Entra ID.
3. From the Manage section, select App registrations.
4. Select New registration.
5. Enter the application registration details.
   - Name

- Supported account types
- Redirect URI
  For PeopleSoft server-to-server (client credentials flow), you don’t need this. You can leave it blank or use https://localhost as a placeholder.

6. Select Register.
7. Make a note of the Application (client) ID and Directory (tenant) ID for use in the next process.

Creating a Client Secret

PeopleSoft uses the client secret value with the client ID and tenant ID to obtain an access token from Microsoft Graph. To create a client secret for the registered application:

1. In the registered application, select Certificates & secrets.
2. Under Client secrets, select New client secret.
3. Enter the client secret details.
   Enter a description and expiration period.
4. Select Add.
5. Copy the client secret Value immediately.

Assigning Microsoft Graph API Permissions

An Azure administrator must grant admin consent before PeopleSoft can use the application registration to access SharePoint.

To assign Microsoft Graph permissions for allowing registered application to access SharePoint:

1. In the registered application, select API permissions.
2. Select Add a permission.
3. Select Microsoft Graph.
4. Select Application permissions.
5. Expand Sites.
6. Select the required permission.
   For example, Sites.Selected is recommended as giving the least access.

   Files.ReadWrite.All gives broader access and is not recommended.
7. Select Add permissions.
8. Select Grant admin consent.

Values Needed for PeopleSoft Configuration

After completing the Azure application registration, collect these values before configuring the PeopleSoft URLID:

- CLIENT ID
- CLIENT SECRET
- TENANT ID

Defining SharePoint Repositories

You can define a URLID that points to a Microsoft SharePoint repository. PeopleSoft uses the URLID and its URL properties to identify the SharePoint repository and to authenticate with Microsoft Graph.

SharePoint repositories can be used with file attachment APIs, such as AddAttachment, PutAttachment, ViewAttachment, DetachAttachment, GetAttachment, and DeleteAttachment, when the URLID used by the API points to SharePoint. The SharePoint repository is accessed through Microsoft Graph REST APIs.

Before defining the SharePoint repository in PeopleSoft, register an application in Microsoft Entra ID and collect the client ID, client secret, and tenant ID information. See Registering an Azure Application for SharePoint Repositories.

Navigation:

PeopleTools, then **Utilities**, then **Administration**, and then **URLs**.

Defining the SharePoint URLID

Use the URL Maintenance page to create a URLID for the SharePoint repository.

See URL Maintenance

Field	Description
URL Identifier	Enter a unique URLID for the SharePoint repository.
Description	Enter a description for the SharePoint repository.
URL	Enter the Microsoft Graph endpoint for the SharePoint site. Use this format: `https://graph.microsoft.com/v1.0/sites/<tenant-name>` The URL consists of the following parts: • Microsoft Graph endpoint. • Microsoft Graph API version. • Microsoft Graph resource path for SharePoint sites. • SharePoint tenant name, such as <tenant>.sharepoint.com.
Comments	Optionally enter information that identifies the SharePoint site or document library.

Defining URL Properties for SharePoint

After you create the URLID, use URL Properties to define the authentication and repository details required for SharePoint. The URL properties for the SharePoint URLID are listed in this table:

Field	Description
CERTALIAS	Specify the certificate alias used for client certificate authentication, if required.

Field	Description
CLIENTID	Specify the Microsoft Entra ID application client ID. Supply a unique identifier (for example, a user name) for the registered app in Azure.
CLIENTSECRET	Specify the client secret generated for the Microsoft Entra ID application. This value should be encrypted, and should be stored securely.
DOCUMENTLIBRARY	Specify the SharePoint document library where files are stored.
KEYSTOREPASSWORD	(Required) For attachments transferred from the PeopleSoft system to the SharePoint repository, the system retrieves the key pair for the client certificate from the digital certificate store and writes them to a file in PKCS12 format with password protection. The value of this property will be used as the password for the PKCS12 file. The PKCS12 file enables connection and file transfer, and it exists only temporarily in <PS_SERVDIR>\files\<CERT ALIAS NAME> for the duration of the file transfer. The system deletes the file after the file transfer transaction. **Note**: If the system fails to delete the certificate alias file, a message will be written to the application sever log. The maximum number of files that can exist at any time is equal to the total number of FTPS and HTTPS URL identifiers defined in the system.
PROXY	Specify the proxy server (if used).
PROXYPORT	Specify the proxy server port (if used).
SHAREPOINTREPOSITORY	*Y*: Identifies the repository as a SharePoint repository. *N*: Specifies that the URLID does not point to SharePoint.
TENANTID	Enter the Microsoft 365 tenant ID.
VERIFYHOST	*0*: Do not verify the server for host name. *1*: (Default) Checks for a match with the host name in the URL with the common name or Subject Alternate field in the server certificate.
VERIFYPEER	*False*: Do not verify the peer. *True*: (Default) Verify the peer by authenticating the certificate sent by the server.

Configuring a Cloud Provider

This topic discusses registering a cloud service provider and administering the cloud service provider.

Registering a Cloud Service Provider

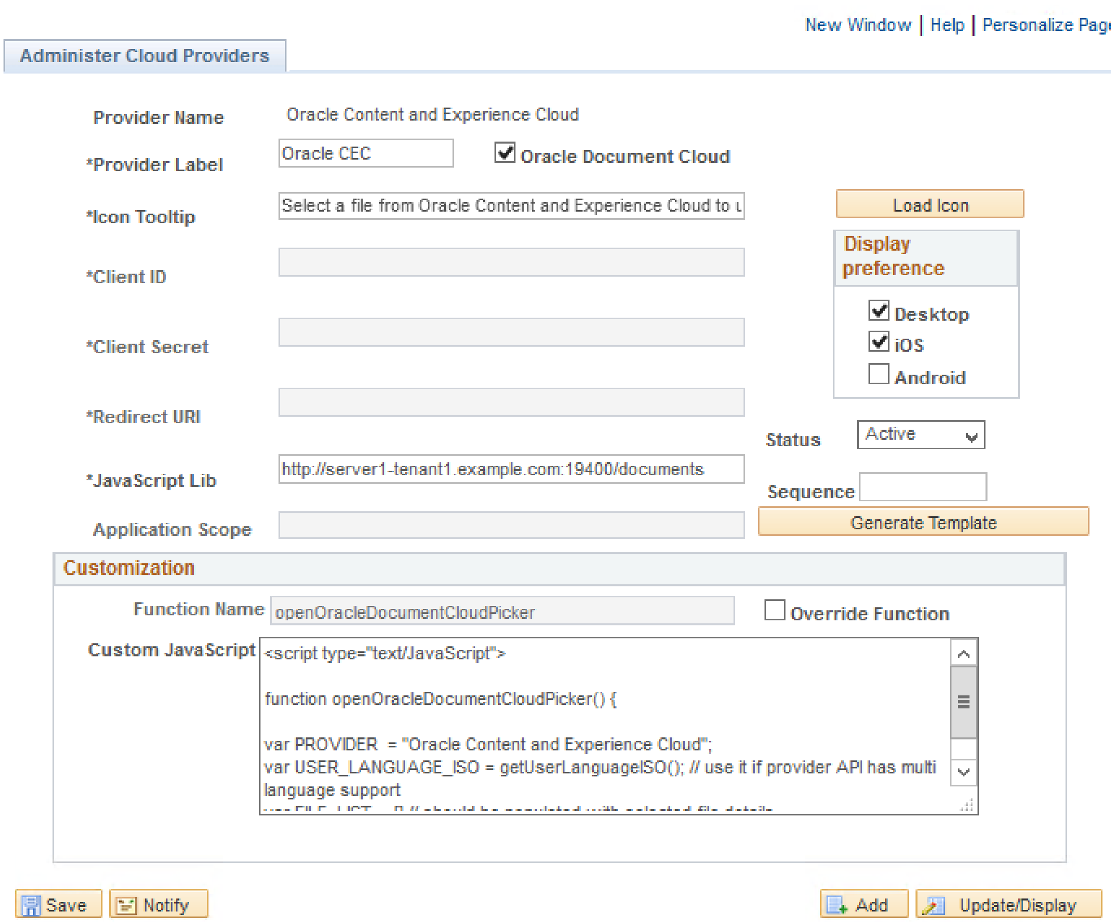
To register a cloud service provider for supporting cloud attachments in PeopleSoft, you must complete the following steps:

1. Register your PeopleSoft application with a cloud service provider.
2. Add the credentials, generated in Step 1, of the cloud service provider in the Administer Cloud Providers page.
3. If the cloud service provider is in Active status, it appears in the File Attachment window.

Using the Administer Cloud Providers Page

Access the Administer Cloud Providers page by selecting **PeopleTools**, and then **Utilities**, and then **Administration**, and then **Administer File Processing**, and then **Administer Cloud Providers**. Add the cloud provider name.

This example illustrates the fields and controls on the Administer Cloud Providers page. You can find definitions for the fields and controls later on this page.



Field or Control	Description
Provider Name	Name of the cloud service provider. It should contain characters a-z/A-Z/0-0 only because it is used as part of a JavaScript function.
Provider Label	A language specific label for the cloud service provider to be displayed in the File Attachment window.
Oracle Document Cloud	Select to use Oracle Document Cloud as your cloud storage. **Note:** Other third party personal Cloud Accounts can also be configured as a cloud storage.
Icon Tooltip	A language specific tool-tip for the cloud service provider icon in the File Attachment window.
Client ID	A technically specified value generated by the cloud service provider as per OAuth 2 standards. Refer to the OAuth 2 documentation for more details.
Client Secret	A technically specified value generated by the cloud service provider as per OAuth 2 standards. Refer to the OAuth 2 documentation for more details. **Note:** Though the Client Secret value is embedded in the generated template, it is recommended to remove it from the generated code if the provider's API does not use it.
Redirect URI	A page in PeopleSoft domain to which the authorization code is sent as a parameter. It should be a static page in the user’s domain which should include the javascript library scripts. There is no need for any content in the page. It acts as a safe location where the token for the OAuth 2 protocol is sent. **Note:** The domain must be a trusted domain.
JavaScript Lib	A JavaScript library file accessible over the web from the PeopleSoft page where AddAttachment is invoked. **Note:** If an invalid value is entered, the AddAttachment API will fail.
Application Scope	The code defined for user authorization. Refer to the OAuth 2 documentation for more details.
Load Icon	Adds a new icon for the cloud service provider, which is displayed in the File Attachment window. If a new icon is not uploaded, a default icon is used.

Field or Control	Description
Display Preference	Sets the display preference for the File Attachment window. Allows administrators to set the display preference to: • Desktop. • Mobile devices using iOS. • Mobile devices using Android. By default, cloud service providers are always displayed in Desktop mode.
Status	Enables or disables a cloud service provider. It can be used to temporarily stop using a cloud service provider due to business or security reasons. • Active - Set the status to Active if you want the cloud service provider to be available in the File Attachment window. • Inactive - Set the status to Inactive if you do not want the cloud service provider to be available in the File Attachment window.
Sequence	Sets the sequence of how the cloud service providers are displayed in the File Attachment window. The provider with the sequence number of 1 appears first and providers with sequence numbers of subsequent values follow. If two or more providers have the same sequence number, the icons appear alphabetically by the provider name.
Generate Template	Generates the template function for the cloud service provider using the configurations added.
Function Name	The JavaScript function name to be used inside the custom JavaScript. This field is automatically populated based on the entry in the Provider Name field.
Override Function	Enables you to name the JavaScript function in case the default function conflicts with any of the existing ones. When selected, you can override the automatically generated function name and manually edit the Function Name field. In such a situation, template generation explicitly uses the typed function name instead of generating it. The function name should be entered in the following format: `open<provider name>Picker` There should be no spaces in the value displayed in the Function Name field. The function name should qualify as a JavaScript function name.

Field or Control	Description
Custom JavaScript	Holds the custom JavaScript API calls to the cloud provider's JavaScript library. A template function should be generated before the APIs are added. The following JavaScript functions should be used after retrieving the URL of the files: • AddFileToArray(Array, fileName, fileSize, fileURL, provider, icon, thumbnail Link) - This function adds the files to the files listed in the File Attachment window. • CloudAttachFile(fileListArray) - This function completes the final loading of the fileAttach object to be sent to the Web server when the Upload button is selected.

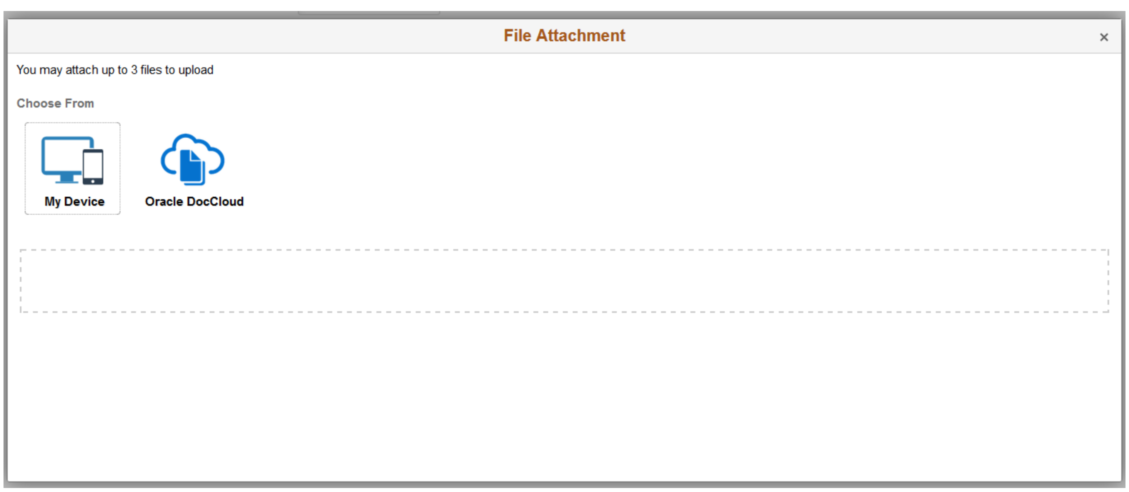
Attaching a File from Cloud Storage

PeopleSoft provides file attach functionality through the AddAttachment and MAddAttachment APIs. This functionality enables you to attach and upload files to PeopleSoft applications from any configured cloud storage. The AddAttachment API enables you to attach a single file and the MAddAttachment API enables you to attach multiple files.

> **Note**
>
> Use CTRL + click to select more than one file at once.

If the status of the cloud service provider is active, the cloud service provider will be listed on the File Attachment window. In the following example, Oracle Document Cloud is the configured cloud service provider with status as active.

This example illustrates a File Attachment window that displays the configured cloud service provider.



14

Securing PS_HOME and PS_CFG_HOME

Understanding PS_HOME and PS_CFG_HOME Security

With the separation of the PS_HOME and PS_CFG_HOME directories, system administrators can implement more secure PeopleSoft deployments by restricting access within each of these directory structures.

This section describes the procedures and considerations involved in configuring these additional security options.

> **Note**
>
> Each site can elect to implement these security measures as needed according to individual security policies.

> **Note**
>
> Each PeopleSoft application you have licensed may have specific instructions regarding the implementation of these security measures. Always check your application-specific documentation for any information you need to consider to ensure both a secure environment and a properly functioning application.

Understanding PS_HOME Security

Because the configuration files, by default, do not reside in PS_HOME, the PS_HOME installation can be locked down to prevent unauthorized access, by user or system process. By making the PS_HOME directory ‘Read-Only’, processes running in the domain cannot write to PS_HOME or any of the subdirectories therein. Likewise, any users with malicious intent are unable to delete or modify executable files in PS_HOME.

Securing PS_HOME involves making the directory read-only, yet making sure that the following system elements have sufficient access.

System Element	Description
Application server	Application server domains need read access to the executable and binary files of PS_HOME to process requests and run application logic.

System Element	Description
Process Scheduler	Process Scheduler domains need read access to the executable and binary files of PS_HOME to run batch processes. Plus, keep in mind these points: • The user creating a Process Scheduler domain on Windows needs read and write access to the Windows Registry. • The restricted OS user needs to have full privilege access to the psreports folder (and its children). Process Scheduler and the Report Distribution agent inherit the restricted user's security settings they will need to create folders in psreports. • Oracle ProcMGR (Tuxedo) should be started with the restricted OS user ID. • Configure Process Scheduler with an Admin OS user for Windows. While logged into Windows with the full privilege OS user ID, create and configure the Process Scheduler domain, so that ODBC and nVision DLLs register properly.
File server/Windows workstations	The file server and/or Windows workstations running Application Designer need read-only access to PS_HOME to facilitate three-tier connections.

Note

These instructions do not apply to the PS_HOME residing on the web server for PIA or the PS_HOME on the database server.

Note

When implementing a read-only PS_HOME, consider that environment locations to which processes write files can't be in a read-only location. Settings for "temporary" directories and "output" directories should not be located within the PS_HOME directory structure. For example, the default temporary directory locations are C:\Documents and Settings\<user>.PEOPLESOFT\Local Settings\Temp (Windows) and %root%\TMP (UNIX).

Note

All elements of your PeopleSoft implementation, such as Process Scheduler and SQR can operate within a secure PS_HOME configuration.

Understanding Minimum Access Required by the User Starting Domains

The bare minimum that needs write access at the time a domain boots includes:

- The domain directory: it must be possible to write content to the domain directory although most of the configuration files in this directory can be read-only.
- The domain LOGS directory: by default this is the LOGS directory beneath the domain directory. If this location is overridden in the configuration file, the relevant location must also be read-write.
- The .adm directory: this subdirectory within the domain (if present) must also be read-write. This is required by Oracle Tuxedo.
- The Archive directory: located within the domain directory, a copy of the .cfg is archived to this directory each time it is updated. This directory is also used by the Purge Cache PSADMIN option for application servers.

All other files in the PS_CFG_HOME directory tree can be made read-only to the user starting the domain.

Understanding PS_CFG_HOME Security

Some administrators may want to implement additional security and restrict access to PS_CFG_HOME. For example, in some cases you may want take further steps to limit privileges of the user starting a domain, or lock down configuration files to prevent unintended configuration changes during runtime.

Securing PS_HOME on UNIX

The UNIX operating system lends itself to a read-only configuration for PS_HOME because of the way that Inter-process Communication (IPC) resources are allocated and managed. UNIX was designed to allow multiple users concurrent access to the same physical hardware and file system while enforcing a strong privileges model.

> **Note**
>
> It is necessary to have access to at least two user accounts in order to setup a true and complete read-only environment on UNIX.

To illustrate the procedure, two user accounts are used.

User Account	Description
InstallAdmin	User responsible for installing PeopleTools.
DomainAdmin	User responsible for creating, configuring, and booting domains. **Note:** It is under this account that domain processes will run and therefore should have the most stringent permissions.

To setup read-only PS_HOME on UNIX:

1. Install PeopleTools using the InstallAdmin account.

2. Verify that PS_HOME is read-only.

   After installing PeopleTools, attempt to delete both a directory and file from PS_HOME using the DomainAdmin account.

   If it is not read-only to the DomainAdmin account, login as the InstallAdmin account and use the chmod command to make PS_HOME read-execute to the world.

   If the DomainAdmin account is a member of the same group as the InstallAdmin account you will need to apply the read-execute restriction to the group also. For example,

   ```
   chmod -R 755 $PS_HOME
   ```

3. Sign in as the DomainAdmin account, open a shell, and change directory to PS_HOME.
4. Invoke psconfig.sh to set the environment.
5. Create and configure a new domain.

   This can be an application server or Process Scheduler domain.

6. Start the new domain and verify that all of the domain processes have started.

   For application server domains, ensure that you can sign in through PIA and make successful page requests.

Managing a Secure PS_HOME on UNIX

When deploying a secured PS_HOME environment on UNIX, keep the items in this section in mind.

Working with User Accounts

The user account that boots the domain must be the same user who configures the domain. This is a Tuxedo requirement, not a PeopleTools requirement. This means that the user account under which the domain processes will run must have read-write access to the domain directory.

The owner of the domain processes is the user account who starts the domain. This is different from Microsoft Windows, where the domain processes are booted by the account that starts the Oracle ProcMGR service. If you use both Windows and UNIX servers to deploy PeopleSoft, keep this subtle distinction in mind between the two operating systems.

Configuring Partial PS_HOME Access

In some cases, user accounts may need to access specific parts of the PS_HOME directory tree. This is recommended through the addition of a “hybrid” user to the same group to which the “InstallAdmin” account (the user who installed PeopleTools) belongs. The InstallAdmin can then choose to allow group access to the specific parts of the PS_HOME directory tree to which the hybrid user is permitted read-write access.

For example, consider a scenario where you have installed PeopleTools at your site, but have hired a consultant to help with various implementation tasks. The InstallAdmin only wants to allow the consultant access to specific parts of the PS_HOME directory tree. The account that the consultant uses is therefore a hybrid account. It is has read-write access to PS_HOME, but only to the specific subdirectories deemed necessary.

To achieve this hybrid privilege model, allow group access to those specific directories under PS_HOME to which the consultant requires write access.

Securing PS_HOME on Windows

This section provides an overview of how you can secure PS_HOME on Windows through different types of user accounts.

Multiple Administrator User Accounts

This method of securing PS_HOME on Windows offers the ability for many administrator user accounts to share the same PS_HOME while managing separate PS_CFG_HOMEs. This method is most appropriate for a production environment.

In this configuration, you install PeopleTools on a server machine, and share the PS_HOME installation location as read-only. Domain administrators may then map to this network drive, and invoke PSADMIN to create and start domains.

Once you have set up a secure PS_HOME, domain creation and the various prerequisites for setup are the same as before. For example, database connectivity must be available on the machine on which the domain will boot. The server where PS_HOME is located acts as a read-only file server for the domains.

To illustrate this procedure, two user accounts are used.

User Account	Description
User1	An administrator on Machine, having read-write access to PS_HOME.
User2	An administrator on Machine2, having read-only access to PS_HOME. A restricted user.

In this scenario, one administrator installs PeopleTools, and a second user, with a more limited set of privileges, creates and administers domains.

The steps for User2 on Machine2 can be applied to multiple users. Any number of users can map to a single read-only PS_HOME.

> **Note**
>
> This information applies to PeopleTools installations on drives assumed to be formatted as NTFS.

To install PS_HOME and restrict privileges:

1. While logged in as User1 on Machine1, run the PeopleTools installation program.
2. Choose Application Server and Batch Server installation options.

   File Server is optional, depending upon whether or not it is required.
3. Ensure that you install PeopleTools such that PS_HOME is not the top-most directory on the drive.

   For example,

   D:\PTInstalls\PT8.58

> **Note**
>
> This is essential if you plan on accessing PS_HOME as a mapped drive.

4. After the installation has completed, set privileges on the PS_HOME directory tree, using Windows File Sharing.
   - Using Windows Explorer, select the high-level directory (as in D:\PTInstalls), right-click, and select Sharing and Security.
   - On the Properties, Sharing tab, click Share this folder.
   - Click Permissions, and for the Group of Everyone, select the Allow check box for Read, make sure the Allow check box for Change is not selected, and click Apply or OK.
5. Verify that the folder has been shared by making sure the 'hand' icon appears on the folder in Windows Explorer.

To set up access to a secure PS_HOME:

1. Sign in as User2 on Machine2.
2. Map a network drive to the shared PS_HOME.
3. Verify that you cannot add, modify, or delete any content below the mapped location.

   This ensures that PS_HOME is read-only. If you can delete or change content in the mapped location, it is possible that User2 is an administrator on Machine1. User2 must not be an administrator on Machine1 for these security measures to be effective.

   If User2 cannot see the shared location there may be a problem with the share or the local network. Make sure the machine can be pinged.
4. Configure Oracle ProcMGR service to allow the User2 account to access PS_HOME.

   This is necessary because by default the Oracle ProcMGR service uses the Local System account.

   a. Select Start, Programs, Administrative Tools, Services, and double-click on the Oracle ProcMGR service.
   b. On the General tab, stop the service, and set the Startup type to Manual.
   c. On the Log On tab, select the This account radio button, and enter the logon information for User2.
   d. Click OK.

   > **Note**
   >
   > Do not start the service yet.
5. Set the TM_TUXIPC_MAPDRIVER user variable for User2.

   This environment variable must be set to contain any mapped drives required by the domain processes, such as the drive where PS_HOME is located. Use the following format:

```
drive1:=\\machine_name1\dirpath1[;drive2:=\\machine_name2\dirpath2[...]]
```

For example,

```
N:\\198.51.100.200\PTInstalls
```

If multiple mapped drives are required, use a semicolon to separate the values, similar to the way directories are expressed in the PATH environment variable.

> **Note**
>
> Depending on your network, use either the DNS name or the IP address to specify the machine name.

> **Note**
>
> If the Oracle ProcMGR needs to run unattended, where no user is signed in, set the TM_TUXIPC_MAPDRIVER environment variable as a system environment variable instead of a user environment variable.

6. Start the Oracle ProcMGR service.

   Once started, you can start PSADMIN as you normally would.

Local User Accounts

Using local user account to secure PS_HOME is a machine-bound solution that you may consider during an initial demo, development, or testing environment, where PS_HOME and PS_CFG_HOME reside on the same machine. In this method, only one machine and one domain account is required.

In this scenario, PeopleTools is installed by a user with administrative privileges. In the context of this scenario, this refers to the network domain user, a user that is a member of an existing network domain of users.

Application server domains are administered by a second user with a more limited set of privileges. This second user is created on the local system and only has access to resources on that machine.

In the following procedure, these user types are represented as:

User	Description
COMPANY\User1	User1 is an administrator on Machine1 and belongs to the network domain named *COMPANY*.
LOCAL_MACH\Guest2	Guest2 does not have administrative privileges on Machine1 but can sign on to Machine1. Guest2 is a Restricted User.

Setting up a secure PS_HOME and restricting privileges:

1. While logged in as User1 on Machine1, install PeopleTools to the server using the install program.

Choose Application Server and Batch Server installation options. File Server is optional depending upon whether or not it is required.

Ensure that you install PeopleTools such that PS_HOME is not in the top directory level on the drive. For example, an ideal location would be C:\PTInstalls\PT8.53.

2. While logged in as User1 on Machine1, create the Guest2 user account.

   Create a new user with at least the following attributes:

   - User name
   - Password
   - Select Restricted user on the Group Membership tab.
   - De-select the User must change password at next logon check box.

     See Microsoft Windows documentation for more information related to creating users on Microsoft Windows.

3. Verify that the user is a Restricted User by highlighting the user ID and clicking on the Properties tab.

   You may need to exit and re-enter the User Accounts dialog to see the new user added.

4. Make the PS_HOME directory tree read-only.

   a. Open Windows Explorer and navigate to and select the PS_HOME directory location.
   b. Open the Properties dialog and click on the Security tab.
   c. Deny Write access to all Local Users.

To configure access to PS_HOME:

1. Set up the Oracle ProcMGR Service to allow the IPC resources to be created using the restricted user ID (Guest2).

   This is necessary because by default the Oracle ProcMGR service uses 'Local System' account which provides greater access to the PS_HOME than desired.

   a. Select Control Panel, Administrative Tools, Services.
   b. Double-click Oracle ProcMGR and change the Startup Type to Manual.

2. Change the user and password with which the service is started to match the new local user that you created earlier (Guest2).

   > **Note**
   >
   > Do not start the service, just click OK.

3. Log off and sign on to Machine1 as Guest2.

   In most cases, any error messages that you see when re-signing on can be ignored as they are associated with signing on with restricted permissions.

4. Verify that you cannot delete, update, or add any content to PS_HOME.
5. Before creating a domain your database connectivity must be setup within the restrictions of the guest2 local user account that you have created.
6. Start PSADMIN, and create a new domain, and confirm that the domain boots.
7. Install PIA on a separate machine, and verify that you can sign in through PIA.

Because PIA is not within the scope of the secure PS_HOME, PIA should be installed on an additional machine. This is necessary because PIA needs write access to various locations within PS_HOME.

Managing a Secure PS_HOME on Windows

When deploying a secured PS_HOME environment on UNIX, keep the items in this section in mind. Depending upon your domain configuration and usage pattern, you may need to unlock specific subdirectories of PS_HOME.

Working With Mapped Drives, UNC Paths, and TM_TUXIPC_MAPDRIVER

This page explains mapped drives (Windows share drives) and the use of UNC paths for PeopleTools application server and Process Scheduler server. When a PeopleTools domain is started on a Windows machine, it runs under the user for whom the ProcMGR Windows service has been configured. As such, the domain processes inherit the privileges of that user and not the user logged on to the system.

By default, the ProcMGR runs as the Local System account. While the Local System account has most privileges on the local host, it can't usually access UNC paths or mapped drives.

> **Note**
>
> ProcMGR is the Windows service that is responsible for allocating resources to Tuxedo domains.

Accessing UNC Paths and Mapped Drives

To allow PeopleTools domain processes to access UNC paths or mapped drives, it is necessary to start the ProcMGR service using an account that has access to these resources. This is typically a Windows *domain account*. A domain account refers to an account that logs a user ID on to both a machine and the corporate network. This account has been created by the network administrator. An example of such an account would be COMPANY_NAME\psftuser.

The ProcMGR should be configured to start as the domain account. On the Log On tab of the service configuration dialog, select the option **This account:** and enter the credentials of the domain account.

UNC Paths

If you plan to use UNC paths to access PS_HOME you must start PSADMIN using a UNC Path. For example:

```
\\ptinstalls\pt853\APPSERV\psadmin.exe
```

With the ProcMGR service set to a domain account, you can use PSADMIN to create and configure domains as if PS_HOME were on the local file system.

Mapped Drives

Additional steps are required if you plan on using a mapped drive for your PS_HOME or PS_CFG_HOME. These additional steps are required because Windows services do not recognize mapped drives. However, Oracle Tuxedo provides a mechanism by which the ProcMGR service is permitted to access network drives, which involves defining any mapped drives using an additional environment variable, TM_TUXIPC_MAPDRIVER. If this environment variable has not been set, the domain processes will be unable to recognize the network drives.

To make sure that the TM_TUXIPC_MAPDRIVER variable is visible to the ProcMGR service, it is necessary to set it globally, as a System environment variable. For example, set TM_TUXIPC_MAPDRIVER to:

N:=\\198.51.100.123\PTInstalls

Important

The mapped drive cannot point *directly* to PS_HOME. The mapping must point to the parent directory above PS_HOME. For example, if PS_HOME is N:\\198.51.100.123\PTInstalls\pt850, TM_TUXIPC_MAPDRIVER should point to N:\\198.51.100.123\PTInstalls. PTInstalls is the parent directory of \pt850, which would resolve to N:\pt850.

Note

When mapping network drives to the PS_HOME is located, make sure to select Reconnect at logon.

Note

The working directory for psadmin.exe, must always be mapped to a letter drive, such as C:, D:, N:, and so on. When starting psadmin.exe, you must do so from the command line (or script) by referencing the full mapped working directory, such as \\ptinstalls\pt851\APPSERV\psadmin.exe. Launching psadmin.exe outside the current working directory (as in, using Start, Run) will cause psadmin.exe to function incorrectly. This restriction is imposed by Tuxedo.

Working With Oracle ProcMGR Windows Service

When logging on with user accounts Oracle ProcMGR service should be set to *Manual* instead of *Automatic*. Failure to do so may result in your domain account becoming locked. If set to *Automatic* the service may continually attempt to start with an expired password causing the network to lock out the domain user account due to successive failed retries.

Note

On Windows servers, it is recommended to have the Windows user logged in running PSADMIN be the same user that runs the ProcMGR service.

See PeopleSoft Installation guide for Oracle.

Managing TM_TUXIPC_MAPDRIVER

The TM_TUXIPC_MAPDRIVER environment variable needs to be maintained consistent with the mapped drives upon which you access PS_HOME. If the drive mappings change, then you need to make sure the new values are specified in TM_TUXIPC_MAPDRIVER. If the drive mapping represented by the TM_TUXIPC_MAPDRIVER does not exist, the ProcMGR service will fail to start.

Resolving Initialization Timeout Issues

If the PS_HOME used by your domains is on a network drive, you may notice a delay with starting a domain. This is a result of the binaries being loaded from across the network versus from the local disk. This can cause an initialization timeout.

If you notice domain start failures, check for the following message in the Tuxedo log for the domain:

```
tmboot.16020.15792.-2: CMDTUX_CAT:1859: ERROR: Server process ID 12668
failed to initialize within 60 seconds
```

In such cases, increase the timeout values in the domain's environment variables to accommodate for the slower start time. Use the `Edit environment settings` option on the Quick Configure menu to modify environment variables.

For example,

```
TM_BOOTTIMEOUT           :[300]
TM_RESTARTSRVTIMEOUT     :[300]
```

For more information on editing domain environment variables, see Setting Domain-Level Environment Variables.

Working With the Tuxedo TM_CPAU Setting

For PeopleSoft systems, the Oracle ProcMGR service (tuxipc.exe) is responsible for starting Tuxedo domain processes on Windows. By default, domain processes run as the same user ID that the service is running as. In a default installation, the service is configured to log on to Windows as the Local System user. Microsoft does not support assigning network privileges to the Local System user for security reasons, but the Local System user otherwise has full administrative access to the local system.

In this configuration, PeopleTools domain processes also run as the Local System user, which presents several potential issues, including:

- PeopleTools domain processes are unable to access network resources.
- PeopleTools domain processes run with more privileges than are necessary. A compromised PeopleTools process will have full access to the local system and could potentially be used to gain unauthorized access to the local system.
- All PeopleTools domain processes on the system run as the same user ID

These problems are not present on UNIX systems where domain processes are always started as the user that runs tmadmin (by way of PSADMIN for PeopleSoft installations) to boot the domain. UNIX systems therefore support multiple domains, each running under different user

IDs, with only the desired local privileges, and with no undesirable restrictions to network resources.

For Windows platforms, you can use the Oracle Tuxedo TM_CPAU (Create Process As User) environment variable to achieve behavior similar to UNIX systems. If TM_CPAU is set to YES before tuxipc is started, tuxipc creates an Oracle Tuxedo process that belongs to the user who initiated tmboot. If tuxipc boots as a service, go to Start, Control Panel, System, Advanced, Environment Variables and set system variable TM_CPAU =YES on the dialog box and reboot the Oracle ProcMGR service. If the Oracle ProcMGR service (tuxipc.exe) is started with the TM_CPAU=YES environment variable set, then domain processes will run as the user ID used to run tmadmin (PSADMIN) to boot the domain.

Using the TM_CPAU environment variable enables a variety of configuration options, including:

- The ProcMGR service can be run as the Local System user, but domain processes can be run using a minimally privileged user. This reduces the chance of a compromised PeopleTools process being used to gain unauthorized access to the system.
- The ProcMGR service can be configured to log on to Windows using a minimally privileged user ID and PeopleTools processes can run as a user with more privileges than the Tuxedo user ID. For example, the Tuxedo user ID could have read-only access to PS_CFG_HOME, but the PeopleTools user could have read-write access. The Tuxedo user ID does not actually require read-access to PS_HOME. When CreateProcessAsUser runs, access to the executable to start is evaluated using the user ID that the process will run as.
- A single Windows system can be used to host multiple PeopleTools installations that are each administered by a different user. A non-administrative user ID used to boot one domain will have no privileges to processes started with a different user ID.
- Domain processes can be identified and managed in Windows Task Manager by a non-administrative user.

For more information, see Oracle Tuxedo documentation: "File Formats, Data Descriptions, MIBs, and System Processes Reference."

Implementing PS_CFG_HOME Security

The steps in this section describe techniques for applying more stringent access to a PeopleTools environment by restricting access to PS_CFG_HOME. If you intend to secure PS_CFG_HOME, it is assumed that you have also secured PS_HOME. Securing PS_CFG_HOME enables you to prevent malicious access to content and configuration files located in PS_CFG_HOME and in domain directories.

These steps describe a security implementation where you configure a user account(s) that can create and configure domains, and user account(s) for domain administrators, who can start and stop domains.

It is possible to limit the permissions to PS_CFG_HOME, such that the domain administrator account can:

- Create files and sub-directories in PS_SERVDIR.

  This is necessary for creating log files and temporary files. Tuxedo also requires read-write access to the domain directory.
- Read (but not change or delete) any existing configuration or template files in PS_SERVDIR.

  These files include .cfg, .ubb, .ubx, .val files, and so on.

> **Note**
>
> To apply these permissions, you must do so after the domain has been configured but before it has been started.

> **Note**
>
> Once these permissions have been implemented, only a user account with the appropriate privileges can reconfigure the domain.

Securing PS_CFG_HOME on UNIX

You have these approaches when implementing a secure PS_CFG_HOME on UNIX:

Security Approach	Description
Use a root account, or use *super user do* (sudo), instead, to perform root-level operations with a non-root account:	This technique involves locking a user account's access to a file from the root account.
Use two administrator user accounts:	The two administrator accounts approach involves using two user accounts that are members of the same group. This approach permits all users in the account in the group read access to the domain configuration. This approach works best if the number of users in the group is kept to a minimum. Because there are multiple users involved, it is necessary to override the default PS_CFG_HOME environment variable such that both users will see the same domains.

To set up a secure PS_CFG_HOME using sudo:

1. Create and configure the domain.

   For this procedure, assume the user who does this is DomainAdmin.
2. Use `sudo` to restrict write access to the sensitive configuration files.

   For example, with the `sudo` command include:

   ```
   chmod 555 <filename>
   ```

   In this case, only `sudo` can change the configuration files or restore write access to DomainAdmin.
3. Log in as DomainAdmin, and verify that none of the restricted files can be changed or deleted by the DomainAdmin session.
4. Start the domain as DomainAdmin.

To set up a secure PS_CFG_HOME using two administrator accounts:

1. Create and configure a domain.

   For this procedure, assume the user who does this is DomainAdmin.

2. As the DomainAdmin user, change the permissions on the domain configuration to allow write access to only those files and directories needing to be written to by the user starting the domain.

3. Sign in as the DomainBootAdmin user, start PSADMIN, navigate to the Domain Administration menu, and re-configure the domain without making any changes.

   This results in only the TUXCONFIG file being updated.

4. Star the domain as the DomainBootAdmin user.

Securing PS_CFG_HOME on Windows

To secure PS_CFG_HOME on Windows:

1. Ensure that your PS_CFG_HOME is created in a location that is accessible to the user account that needs to start the domain.

   By default, the PS_CFG_HOME location will not be visible to the user account starting the domain because it is created in the user home of the user account creating the domain. Use one of these options to make the PS_CFG_HOME visible to the user account that needs to start the domain:

   - Override the default location for PS_CFG_HOME, by manually setting the PS_CFG_HOME environment variable to a custom value.
   - Enable the restricted user to view the PS_CFG_HOME in the domain creator's user home. Set the PS_CFG_HOME as a system-level environment variable so that the restricted user will be able to see it when logged on.

2. While signed in as the domain administrator, create and configure your new application server or Process Scheduler server domain.

3. While signed on as the domain administrator, apply the necessary read-write restrictions.

   It is recommended to apply read-only privileges to the entire directory path to the domain.

   a. Using Windows Explorer select the domain directory and open the Properties dialog.
   b. Select the Security tab. If the restricted user is not displayed, click Add to append the user to the list.
   c. Once added, highlight the restricted user ID and ensure that the following actions are checked in the Allow column: Read & Execute, List Folder Contents, Read, and Write.

4. Apply read-only privileges to the sensitive domain files that should be protected from read-write at runtime.

   This typically includes the Tuxedo binary configuration (PSTUXCFG and PSBDMCFG), ASCII configuration files and templates (*.cfg, *.ubb, *.env, *.ubx, *.lst).

   Using Windows Explorer, select each of these files in the domain directory and open the Properties dialog.

> **Note**
>
> At a minimum, it must be possible for the user starting the domain to write to the .adm and LOGS directory within PS_SERVDIR. Additionally, this user must be permitted to create new files in the domain directory.

5. Sign in as the restricted user, boot the domain, and verify that it is not possible to delete or modify any of the restricted domain files.

> **Note**
>
> Sign in using the same user account as the one entered in the Oracle ProcMGR service.

> **Note**
>
> If subsequent configuration changes are required it is necessary to configure the domain while signed in as the administrator account that created and configured the domain.

15

WebLogic Managed Server Architecture

PeopleSoft Internet Architecture Servlets and Applications

PIA is packaged as a JEE Enterprise Archive and is comprised of a collection of JEE web applications, commonly referred to as webapps or servlets. For the most part, in the context of PeopleSoft, the term *servlets* is used. The PeopleSoft servlets are:

Term	Definition
PORTAL	PeopleSoft Interaction Hub
PSIGW	Integration gateway
PSEMHUB	PeopleSoft Environment Management Framework
PSINTERLINKS	PeopleSoft Business Interlinks **Note:** PeopleSoft Business Interlinks is a desupported product. These options exist for upgrade compatibility and transition.

In addition, these servlets are added when you install PIA on a WebLogic server machine. These elements are not part of the PeopleSoft Enterprise Archive, but instead are defined individually.

Term	Definition
HttpProxyServlet	Reverse Proxy Server – Proxy to a single content server per URL. Each URL can provide unique content.
HttpClusterServlet	Reverse Proxy Server – Proxy to multiple WebLogic servers. All content servers provide access to the same content for load balancing.

WebLogic Domain Types

This section provides an overview of WebLogic domain types and discusses the factors associated with them.

Understanding WebLogic Domain Types

During the PeopleSoft Pure Internet Architecture (PIA) setup, you can choose between two different WebLogic domain configurations: a single-server domain and a multi-server domain. A multi-server domain can be expanded across multiple machines using the *distributed managed server* option. So, a distributed managed server implementation is a variation of the multi-server domain. Each of these domain configurations has a specific purpose but is fully customizable beyond that purpose.

This section discusses:

- single-server domains.
- multi-server domains.
- distributed managed servers.

Single-Server Domains

In a single-server configuration, the WebLogic domain’s remote console for administration and the J2EE components of PIA are all provided on a single instance of WebLogic Server. This configuration is intended for single-user or very small scale, non-critical production environments. It can be used as a starting point for you to familiarize yourself with WebLogic Server.

In a single-server domain, the resources used to administer WebLogic Server and your PeopleSoft application are not isolated from one another, therefore you don't administer each element individually. While each element must compete for the same resources, the low resource requirements of this configuration make it ideal for small scale and non-production usage.

The single-server domain in a PeopleSoft implementation consists of only one server: PIA.

> **Note**
>
> Do not create a web server domain using the names PIA, PIA1, PIA2, or weblogic. The name of the WebLogic domain and the server and cluster names within it must be unique. The names PIA, PIA1, and PIA2 are reserved for use by the PeopleSoft system. The name weblogic is reserved for use by the Oracle WebLogic software. These restrictions apply to single server and multi-server domain creation.

Single-Server Deployment

Some of the servlets deployed in a single-server domain configuration must be accessed using a modified URL:

http://*server*:*port*/*servlet_name*/...

The single-server domain configuration deploys servlets as follows:

Application	Deployed to Server	Servlet Name in URL
PORTAL	PIA	(not needed)
PSIGW	PIA	PSIGW
PSEMHUB	PIA	PSEMHUB
PSINTERLINKS	PIA	PSINTERLINKS
Remote Console	PIA	rconsole
HttpProxyServlet	Defined but not deployed.	(not needed)
HttpClusterServlet	Defined but not deployed.	(not needed)

Single-Server Domain Specific Settings

Use the WebLogic Remote Console to configure the single-server domain specific settings.

Click the Edit Tree tile, then select **Environment**, and then **Servers**, and then **PIA**. The default web application for the PIA server is PORTAL. The single-server domain specific default settings for the PIA server are as follows:

Setting	Default Value
Listen address	* (all local IPs).
Listen Port	8000 (set during PIA setup).
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
SSL Listen Port	8443 (set during PIA setup).

> **Note**
>
> To configure SSL, you must also define SSL certificates.

See Implementing WebLogic SSL Keys and Certificates.

Example: Single-Server Domain

The following table includes process types and elements for a typical single-server domain:

Process Type	WebLogic and PIA Elements
WebLogic administration	WebLogic Administration Server
Server instance	PIA
Servlet or application	Remote Console PORTAL PSEMHUB PSIGW

Multi-Server Domains

The multi-server domain configuration is intended for production environments. This configuration takes advantage of WebLogic's administration server and managed server architecture. In a multi-server configuration, multiple instances of WebLogic server are used, each contributing a specific function. The WebLogic console is provided on the domain's administration server, WebLogicAdmin, and the J2EE components of PIA are provided on individual or shared WebLogic managed servers.

A production application warrants process and resource pool isolation for greater stability and optionally tighter security controls, which this configuration provides. In a multi-server configuration, the resources used for WebLogic domain administration and monitoring are isolated from similar resources used to support the PIA application. A server process named *WebLogicAdmin* performs nothing but WebLogic administration, which includes domain administration and monitoring. Continuing that separation, the individual PIA servlets are (usually) isolated from each other. The PIA servlets are targeted and deployed across a portion of the six remaining server definitions, all of which are classified as *managed servers,* which are delivered in the multi-server configuration.

> **Note**
>
> Do not create a web server domain using the names PIA, PIA1, PIA2, or weblogic. The name of the WebLogic domain and the server and cluster names within it must be unique. The names PIA, PIA1, and PIA2 are reserved for use by the PeopleSoft system. The name weblogic is reserved for use by the Oracle WebLogic software. These restrictions apply to single server and multi-server domain creation.

A multi-server domain in a PeopleSoft implementation creates the following servers:

Term	Definition
WebLogicAdmin	Administration server for WebLogic domain administration.
PIA	Server for the PeopleSoft Interaction Hub and integration gateway.
PIA1	Server for the PeopleSoft Interaction Hub and integration gateway.
PIA2	Server for the PeopleSoft Interaction Hub and integration gateway.
PSEMHUB	Server for the PeopleSoft Environment Management Framework application.
RPS	Server for WebLogic reverse proxy server applications.

Multi-Server Servlet Deployment

Some of the servlets deployed in a multi-server domain configuration must be accessed using a modified URL:

http://*server*:*port*/*servlet_name*/...

The multi-server domain configuration deploys servlets as follows:

Application	Deployed to Server, Cluster (members)	Servlets Name in URL
PORTAL	PIA, PeopleSoftCluster (PIA1, PIA2)	(not needed)
PSIGW	PIA, PeopleSoftCluster (PIA1, PIA2)	PSIGW
PSEMHUB	PSEMHUB	PSEMHUB
PSINTERLINKS	PIA, PeopleSoftCluster (PIA1, PIA2)	PSINTERLINKS
Console	WebLogicAdmin	console
HttpProxyServlet	RPS	(not needed)
HttpClusterServlet	Defined but not deployed.	(not needed)

Multi-Server Domain Specific Default Settings

To configure the multi-server domain specific settings:

1. Sign in to the WebLogic Remote Console.

2. Click Edit Tree, and then select Environment > Servers.
3. Click on the appropriate servlet, server, or application, and then select the General page.

The domain specific default settings for the WebLogicAdmin server are as follows:

WebLogicAdmin Setting	Default Value
Listen Address	* (all local IPs)
Listen Port	9999
SSL Listen Port Enabled	Disabled

The domain specific default settings for the PIA server are as follows:

PIA Setting	Default Value
Listen Address	* (all local IPs)
Listen Port	HTTP port set during PIA setup.
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
HTTPS Listen port	HTTPS port set during PIA setup.

The domain specific default settings for the PIA1 server are as follows:

PIA1 Setting	Default Value
Listen Address	Locally determined hostname.
Listen Port	HTTP port set during PIA setup.
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
SSL Listen Port	HTTPS port set during PIA setup.

The domain specific default settings for the PIA2 server are as follows:

PIA2 Setting	Default Value
Listen Address	127.0.0.1
Listen Port	HTTP port set during PIA setup.
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
SSL Listen Port	HTTPS port set during PIA setup.

The default web application for the PSEMHUB server is PSEMHUB. The domain specific default settings for the PSEMHUB server are as follows:

PSEMHUB Setting	Default Value
Listen Address	* (all local IPs)
Listen Port	8081
SSL Listen Port Enabled	Disabled

In the WebLogic Remote Console, click the Edit Tree tile, then select **Environment**, and then **Servers**, and then **RPS** to configure the RPS server. The default web application for the RPS server is HttpProxyServlet. The domain-specific default settings for the RPS server are as follows:

RPS Setting	Default Value
Listen Address	* (all local IPs)
Listen Port	8080 (set during PIA setup)
SSL Listen Port Enabled	Enabled with demonstration self-signed digital certificates.
SSL Listen Port	8443 (set during PIA setup)

Note

To configure SSL, you must also define SSL certificates.

See Implementing WebLogic SSL Keys and Certificates.

Example: Multi-Server Domain

The following table lists the elements for processes running within a typical multi-server domain:

Process Type	WebLogic and PIA Elements
WebLogic administration	WebLogic Administration Server (WebLogicAdmin) WebLogic Node Manager
Managed Server	PIA1, PIA2 for PeopleSoft Interaction Hub
Managed Server	PIA for PSIGW
Managed Server	RPS for httpproxyservlet to proxy content for PIA1 and PIA2

Distributed Managed Servers

The *distributed managed server* configuration, although listed alongside the single-server and multi-server domain types, is not a *true* domain type. It's an optional extension for an existing multi-server configuration. As with the multi-server domain type, this configuration takes advantage of WebLogic's managed server architecture. The distributed managed server configuration is intended for production environments encompassing multiple machines.

A distributed managed server configuration enables you to spread a *logical* WebLogic domain configuration *physically* across multiple machines in a heterogeneous network. One server machine might act as the domain administration server, while the other server machines act as distributed managed servers, running various manages servers, such as PIA1, PIA2, and so on. Typically, you would also take advantage of the WebLogic Node Manager, which enables a system administrator to control the managed servers remotely.

See your WebLogic Node Manager documentation.

In a distributed, multi-server configuration you can have multiple machines, multiple collections of resources and program files, multiple web server processes, and a replicated domain configuration file. In this model, if any of system resources or server instances becomes unavailable, work shifts to the next instance of that resource. In the multi-server configuration, an increase in PeopleSoft Interaction Hub usage, for example, can be accommodated by configuring an additional WebLogic server instance or machine to also serve the PeopleSoft Interaction Hub. Using distributed managed servers provides flexibility in shaping your hardware allocation to meet your system demands.

Benefits of the Distributed Managed Server

A distributed managed server configuration provides the same benefits as a multi-server configuration with the added benefit of hardware isolation. This option requires a multi-server installation to be performed to some other location, which will contain the configuration for this distributed managed server.

The fundamental benefits of a multi-server configuration include:

- Dedicated service providers: Web servers can be dedicated to providing PeopleSoft Interaction Hub and are isolated from other portions of the PeopleSoft Pure Internet Architecture such as PeopleSoft Integration Gateway.
- Redundant service providers: Multiple web servers can be used to serve different aspects of PIA, providing load balancing and failover support.
- Distributed resources: Multiple web server machines can be used, each capable of serving different or redundant aspects of PIA.
- Centralized and replicated configuration: Primary domain configuration is centralized and distributed server information is replicated locally.

The server configuration settings for a distributed managed server are maintained by that domain's administration server and are stored locally on that administration server. Configuration settings are replicated to a managed server during its startup, but are only maintained as a read-only backup copy for that individual managed server in the event that the administration server isn't available the next time this particular managed server needs to be started.

> **Note**
>
> Only one managed server can be run per distributed managed server domain directory. If you intend to run multiple distributed managed servers on a single machine, perform the PIA installation and create unique distributed managed server domain directories, one for each distributed managed server that you intend to run on that machine.

> **Note**
>
> Do not create a web server domain using the names PIA, PIA1, PIA2, or weblogic. The name of the WebLogic domain and the server and cluster names within it must be unique. The names PIA, PIA1, and PIA2 are reserved for use by the PeopleSoft system. The name weblogic is reserved for use by the Oracle WebLogic software. These restrictions apply to single server and multi-server domain creation.

Example: Distributed Managed Server

The following illustrates the multiple servers, their roles, and the various elements running on the multiple server machines within a sample distributed managed server configuration:

Server	Role	WebLogic and PIA Elements
Server 1	Domain Administration Server	WebLogic Administration Server (WebLogicAdmin)
Server 2	Distributed Managed Server	PIA1 (for PeopleSoft Interaction Hub) PIA2 (for PeopleSoft Interaction Hub) WebLogic Node Manager
Server 3	Distributed Managed Server	PIA3 (for PeopleSoft Interaction Hub) PIA4 (for PeopleSoft Interaction Hub) WebLogic Node Manager
Server 4	Distributed Managed Server	PIA (for Integration Gateway) WebLogic Node Manager
Server 5	Distributed Managed Server	RPS (for httpproxyservlet) WebLogic Node Manager

Common Default Settings

Single-server and multi-server domain configurations have many settings in common.

Domain Defaults

Many of these common settings can be configured in the WebLogic Remote Console, but some are configured in other environments. Default values are listed when available

Setting	Default Value	Where To Configure in the WebLogic Remote Console
SSL functionality	Enabled with demonstration self-signed digital certificates.	Click the Edit Tree tile, then select **Environment**, and then **Servers**, and then **PIA (or custom server name)**, and then **Security**. Select the Keystores tab and SSL tab. Alternatively, on the command line. use pskeymanager.
Server logs	*Weblogic domain*\logs*server name_**.log	Click the Edit Tree tile, then select **Environment**, and then **Servers**, and then **PIA (or custom server name)**, and then **Logging**.
HTTP access log	Disabled	Click the Edit Tree tile, then select **Environment**, and then **Servers**, and then **PIA (or custom server name)**, and then **Logging**, and then **HTTP**.

Setting	Default Value	Where To Configure in the WebLogic Remote Console
HTTP keep-Alive	30 seconds	Click the Edit Tree tile, then select **Environment**, and then **Servers**, and then **PIA (or custom server name)**, and then **Protocols**, and then **HTTP**.
HTTPS keep-Alive	60 seconds	Click the Edit Tree tile, then select **Environment**, and then **Servers**, and then **PIA (or custom server name)**, and then **Protocols**, and then **HTTP**.
Max Stuck Thread Time	600	Click the Edit Tree tile, then select **Environment**, and then **Server**, and then **PIA**, and then **Server Failure Trigger**. **Note:** Click the question mark icon for information.
Stuck Thread Timer Interval	60	Click the Edit Tree tile, then select **Environment**, and then **Servers**, and then **PIA (or custom server name)**, and then **Advanced**, and then **Tuning**.
Enable Administration Port	Disabled	Click the Edit Tree tile, then select **Environment**, and then **Domain**.
PORTAL HTTP session monitoring	On (applies only to servers running PORTAL)	Click the Monitoring Tree tile, then select **Deployments**, and then **Application Runtime Data**, and then **peoplesoft**, and then **Component Runtimes**.
System administrator user ID	system (set during PIA setup)	Click the Security Data Tree tile, then select **Realms**, and then **myrealm**, and then **Authentication Providers**, and then **Default Authenticator**, and then **Users**, and then **system**.
System administrator password	*password* (set during PIA setup)	Click the Security Data Tree tile, then select **Realms**, and then **myrealm**, and then **Authentication Providers**, and then **Default Authenticator**, and then **Users**, and then **system**.
System operator user ID	operator	Click the Security Data Tree tile, then select **Realms**, and then **myrealm**, and then **Authentication Providers**, and then **Default Authenticator**, and then **Users**, and then **operator**.

Setting	Default Value	Where To Configure in the WebLogic Remote Console
System operator password	*password*	Click the Security Data Tree tile, then select **Realms**, and then **myrealm**, and then **Authentication Providers**, and then **Default Authenticator**, and then **Users**, and then **operator**.
System monitor user ID	monitor	Click the Security Data Tree tile, then select **Realms**, and then **myrealm**, and then **Authentication Providers**, and then **Default Authenticator**, and then **Users**, and then **monitor**.
System monitor password	*password*	Click the Security Data Tree tile, then select **Realms**, and then **myrealm**, and then **Authentication Providers**, and then **Default Authenticator**, and then **Users**, and then **monitor**.

Script and Environment Defaults

Modify these settings by editing a setEnv script or applying command line parameter overrides to WebLogic control scripts.

The following settings specify the names and structure of various directories on the web server machine.

Setting	Default Value	Description/Override
PS_HOME	(none)	PeopleSoft home directory (set during the DPK installation or PIA setup).
BASE_DIR	(none)	High-level install directory (set during the DPK installation or PIA setup) where Tuxedo and WebLogic may be installed.
WLS_HOME	For a default DPK installation, *BASE_DIR*/pt/bea/wlserver	WebLogic home directory (set during the DPK installation or PIA setup).
DOMAIN_NAME	peoplesoft	Name of this WebLogic domain (set during the DPK installation or PIA setup).
JAVA_HOME	(Depends on the operating system platform.)	Location of Java. Set during PIA setup or with a call to WebLogic's CommEnv script.

> **Note**
>
> You configure Java VM options including JVM memory size using the JAVA_OPTIONS_*OSplatform* parameter, during PIA setup.

The following are miscellaneous settings.

Setting	Default Value	Description/Override
HOSTNAME	*Local hostname*	Set during PIA setup.
PRODUCTION_MODE	TRUE	Enable WebLogic production mode (set during PIA setup).
DISCOVERY_MODE	FALSE	Disable auto detection of unregistered applications. Script: setEnv
WLS_USER	Operator	Use to stop WebLogic with stop scripts and run it as a Windows service.
WLS_PW	*password*	Use to stop WebLogic with stop scripts and run it as a Windows service.
ADMINSERVER_PROTOCOL	HTTP	Protocol used for managed server to connect to administration server (not used in single-server domain).
ADMINSERVER_HOSTNAME	Single-server: *local hostname*. Multi-server: *local hostname*. Distributed server: (none — set manually).	Administration server's hostname that managed servers attempt to connect to by default when started. Set during PIA setup (except distributed server).
ADMINSERVER_PORT	Single-server: *HTTP port of PIA server*. Multi-server: 9999. Distributed server: (none — set manually).	Administration server's Listen port that managed servers attempt to connect to by default when started. Set during PIA setup (except distributed server).
ADMINSERVER_SERVERNAME	Single-server: PIA. Multi-server: WebLogicAdmin. Distributed server: WebLogicAdmin.	WebLogic server instance name of this domain's administration server, used for stopping and starting the server.
WL_VERSION	*Detected major WebLogic version.*	WebLogic major version, such as 14.
WL_SERVICEPACK	*Detected minor WebLogic version.*	WebLogic service pack level.
WL_PATCH	*Detected WebLogic patch version.*	WebLogic patch level.
BACKGROUND_PROCESS	TRUE	Run WebLogic server as a background process. On UNIX you can force foreground execution using the start script's -foreground option.

The following are debugging output settings.

Setting	Default Value	Description/Override
SET CAPTURE_STDOUT_STDERR	FALSE	(Windows only) Capture standard output and standard error of a WebLogic server running as a foreground process. You can also set this with the start script's –capture option.
ENABLE_JDPA_DEBUG	FALSE	(PeopleSoft development only) Enable JDPA debug support. You can also set this with the start script's –debug option.
ENABLE_VERBOSE_GC	FALSE	Enable verbose output of Java's garbage collector. You can also set this with the start script's –verbose:gc option.
ENABLE_VERBOSE_SSL	FALSE	Enable SSL debug support. Produces verbose SSL output. You can also set this with the start script's –verbose:ssl option.
ENABLE_VERBOSE_WL	FALSE	Enable verbose output for the core WebLogic server (not verbose output of PIA). You can also set this with the start script's –verbose:wl option.
MAX_FILE_DESCRIPTORS	16384	The number of open file descriptors set for any WebLogic server process. **Note:** In UNIX, you need to set the environment variable to an optimum value for the number of open file descriptors in your web server setEnv.sh file. The file is located at <PS_CFG_HOME>/webserv/<domain>/bin/setEnv.sh .

The following are HTTP forward proxy support settings.

Setting	Default Value	Description/Override
ENABLE_HTTP_PROXY	FALSE	Enable the use of the forward http proxy.
HTTP_PROXY_HTTPHOST	(none)	IP address or hostname of the forward HTTP proxy server for HTTP requests.
HTTP_PROXY_HTTPPORT	(none)	HTTP Port number of the forward HTTP proxy server for HTTP requests.

Setting	Default Value	Description/Override
HTTP_PROXY_HTTPSHOST	(none)	IP address or hostname of the forward HTTP proxy server for HTTPS requests.
HTTP_PROXY_HTTPSPORT	(none)	HTTP Port number of the forward HTTP proxy server for HTTPS requests.
HTTP_PROXY_NONPROXY_HOSTS	*localhost*, *local hostname*, and *domainname*.	Host names and domain names of content servers that will not be proxied.

WebLogic Domain Directory Structure and Files

This section provides an overview of the WebLogic domain directory structure and files.

WebLogic Domain Directory Structure

In a PeopleSoft implementation, the WebLogic domains are installed into your *PS_CFG_HOME* directory. The major components of the directory structure layout of a PIA install on WebLogic Server are:

Directory	Description
PS_CFG_HOME\webserv	Parent, high-level WebLogic domain directory.
PS_CFG_HOME\webserv\peoplesoft	Default WebLogic domain directory.
PS_CFG_HOME\webserv\peoplesoft\applications\peoplesoft	PeopleSoft application directory.
PS_CFG_HOME\webserv\peoplesoft\bin	WebLogic domain's bin directory, containing numerous administration scripts.
PS_CFG_HOME\webserv\peoplesoft\config	WebLogic domain's configuration directory.
PS_CFG_HOME\webserv\peoplesoft\servers\PIA\logs	WebLogic domain's log directory.
PS_CFG_HOME\webserv\peoplesoft\piaconfig\keystore\pskey	WebLogic key store location for storing keys for configuring SSL.
PS_CFG_HOME\webserv\peoplesoft\piaconfig	Includes configuration files in subdirectories diagnostics, jcmdprop, keystore, and properties.
PS_CFG_HOME\webserv\peoplesoft\piabin	Includes various scripts.

WebLogic Domain File Listing by Type

This section lists the WebLogic domain files installed by the PeopleSoft installation, organized by file type. Where necessary, each table includes columns that indicate whether a given file is used in a single-server, multi-server, or distributed server configuration.

This listing does not include Java classes or PIA configuration files. On UNIX an equivalent Bourne shell script is provided where a Windows script is listed.

The following table lists WebLogic server administration scripts. All the life cycle scripts are stored in <*PS_CFG_HOME*>\webserv\<*web_server_domain*>\bin.

Script	Single-Server	Multi-Server	Distributed Server	Description
setEnv.cmd	Yes	Yes	Yes	Use this script to set required environment variables for the WebLogic server, for example: CLASSPATH, PATH, UNIX Library Path, and JVM options.
startPIA.cmd	Yes	No	No	Use this script to start the WebLogic domain's administration server (the PIA server) in a single-server configuration. On Windows this starts WebLogic as a foreground process. On UNIX this starts WebLogic as a background process. Run the script with the –help option for usage.
startWebLogicAdmin.cmd	No	Yes	No	Use this script to start the WebLogic domain's administration server (the WebLogicAdmin server) in a multi-server configuration. On Windows this starts WebLogic as a foreground process. On UNIX this starts WebLogic as a background process. Run the script with the –help option for usage.

Script	Single-Server	Multi-Server	Distributed Server	Description
startManagedWebLogic.cmd	No	Yes	Yes	Use this script to start a WebLogic managed server. All WebLogic servers in a WebLogic domain other than the administration server are WebLogic managed servers. Run the script with the –help option for usage.
stopPIA.cmd	Yes	No	No	Use this script to stop the WebLogic PIA server. Run the script with the –help option for usage.
stopWebLogic.cmd	No	Yes	Yes	Use this script to stop WebLogic servers. Run the script with the –help option for usage.
InstallNTservicePIA.cmd	Yes	No	No	(Windows only) Use this script to install the WebLogic PIA server as a Windows service. The service name is *WebLogic_domain*-PIA. Run the script with the –help option for usage.
InstallNTservice.cmd	No	Yes	Yes	(Windows only) Use this script to install a WebLogic server as a Windows service. The service name is *WebLogic_domain-server_name*. Run the script with the –help option for usage.

Script	Single-Server	Multi-Server	Distributed Server	Description
uninstallNTServicePIA.cmd	Yes	No	No	(Windows only) Use this script to uninstall the WebLogic PIA server Windows service. Run the script with the –help option for usage.
uninstallNTService.cmd	No	Yes	Yes	(Windows only) Use this script to uninstall a WebLogic server Windows service. Run the script with the –help option for usage.
pskeymanager.cmd	Yes	Yes	Yes	Use this script to manage the JKS keystore used by WebLogic Server, which is in *WebLogic_domain*\keystore\pskey. SSL certificates for WebLogic Server are stored in this keystore. PeopleSoft Integration Gateway can also share this keystore. Run the script with the –help option for usage.
startWebLogicBuilder.cmd	Yes	Yes	Yes	Use this script to start WebLogic Builder, which is used to change local application deployment descriptors.
createThreadDump.cmd	Yes	Yes	Yes	Use this script to create a JVM Thread dump. Run the script with the –help option for usage.

The following table lists WebLogic server configuration files stored in *<PS_CFG_HOME>*/webserv/*<web_server_domain>*/config.

File	Single-Server	Multi-Server	Distributed Server	Description
config.xml	Yes	Yes	No	This file stores the WebLogic domain configuration, including information about server names, ports, IP addresses, webapps, and SSL. Edit these settings using the WebLogic Remote Console.
msi-config.xml	No	Yes	Yes	This is a version of config.xml that's copied for use with a distributed managed server configuration. It's automatically replicated from the original config.xml after a managed server successfully starts.
boot.properties	Yes	Yes	Yes	This file contains the WebLogic system ID and password used for administering the WebLogic domain.
fileRealm.properties	Yes	Yes	Yes	This file is used by WebLogic's internal LDAP server for system administration.
DefaultAuthenticatorInit.ldift	Yes	Yes	Yes	This file is used by WebLogic's internal LDAP server for system administration.
DefaultRoleMapperInit.ldift	Yes	Yes	Yes	This file is used by WebLogic's internal LDAP server for system administration.
SerializedSystemIni.dat	Yes	Yes	Yes	This file is used by WebLogic's internal LDAP server for system administration.

The following table lists PeopleSoft J2EE application scripts, which are all used with Integration Broker, and can be used with every WebLogic server configuration. The scripts are stored in <*PS_CFG_HOME*>/webserv/<*web_server_domain*>/piabin.

Script	Description
BatchProjectExecutor.bat	Use this script for Integration Broker batch EIP testing.
HashKeyGenerator.bat	Use this script to generate a hash key used for Integration Gateway playback.
MessageExport.bat	Use this Integration Broker script for extracting transaction data from request and response data.
StartSendMaster.bat	This is a Integration Broker test utility.

The following table lists miscellaneous files, which can be used with every WebLogic server configuration.

File	Description
Businterlink.txt	This file is used by PeopleSoft's Business Interlinks servlet for loading PeopleSoft libraries when needed. **Note:** PeopleSoft Business Interlinks is a desupported product. These options exist for upgrade compatibility and transition.
PSCipher.bat	Use this script for encrypting Integration Broker passwords. See Security Administration: Using the PSCipher Utility.

Related Topics

- Integration Broker Administration

J2EE Application Files

In addition to WebLogic domain configuration files, application descriptors are installed with the PeopleSoft J2EE enterprise application. The following table lists these descriptor files. The path shown for each file is relative to <*PS_CFG_HOME*>\webserv\<*web_server_domain*>\applications\.

File	Description
peoplesoft\META-INF\application.xml	This file contains a list of the webapps that comprise the PeopleSoft J2EE enterprise application.
peoplesoft\PORTAL\WEB-INF\web.xml	This file is the web application descriptor for the PORTAL webapp. It lists all of the servlets deployed as part of that application.
peoplesoft\PORTAL\WEB-INF\weblogic.xml	This file is the PORTAL web application extension descriptor. It specifies, among other things, the HTTP session cookie name, optional cookie domain, and context path of this application.

File	Description
peoplesoft\PSIGW\WEB-INF\web.xml	This file is the web application descriptor for the PeopleSoft Integration Gateway (PSIGW) webapp. It lists all of the servlets deployed as part of that application.
peoplesoft\PSIGW\WEB-INF\weblogic.xml	This file is the PSIGW web application extension descriptor. It specifies the context path of this application.
peoplesoft\PSEMHUB\WEB-INF\web.xml	This file is the web application descriptor for the PeopleSoft Environment Framework (PSEMHUB) webapp. It lists all of the servlets deployed as part of that application.
peoplesoft\PSEMHUB\WEB-INF\weblogic.xml	This file is the PSEMHUB web application extension descriptor. It specifies the context path of this application.
peoplesoft\PSINTERLINKS\WEB-INF\web.xml	This file is the web application descriptor for the PeopleSoft Business Interlinks (PSINTERLINKS) webapp. It lists all of the servlets deployed as part of that application.
peoplesoft\PSINTERLINKS\WEB-INF\weblogic.xml	This file is the PSINTERLINKS web application extension descriptor. It specifies the context path of this application.
HttpProxyServlet\WEB-INF\web.xml	This file is the web application descriptor for the WebLogic Server Reverse Proxy Server (RPS) webapp that's used to proxy content from a single WebLogic server. It lists all of the servlets deployed as part of that application.
HttpProxyServlet\WEB-INF\weblogic.xml	This file is the single-server RPS web application extension descriptor. It specifies the context path of this application.
HttpClusterServlet\WEB-INF\web.xml	This file is the web application descriptor for the WebLogic Server Reverse Proxy Server (RPS) webapp that's used to proxy content from a cluster of WebLogic servers. It lists all of the servlets deployed as part of that application.
HttpClusterServlet\WEB-INF\weblogic.xml	This file is the multi-server RPS web application extension descriptor. It specifies the context path of this application.

PIA Install and Reinstall Options

The PeopleSoft Internet Architecture (PIA) installation process enables you to create a new web server (PIA) domain or update a valid existing web server domain. A valid domain is a domain built by the PIA installation process in the *PS_CFG_HOME* directory that you specify.

> **Note**
>
> You can create a web server domain as part of the DPK setup script process, or using the PIA silent installation process.

When creating a new web server (PIA) domain as part of the DPK setup script process, the process creates a single-server domain by default.

When creating a new domain using the PIA silent installation process, you may select from three configuration types: single-server, multi-server, or distributed managed server. The silent installation also includes various options for updating an existing domain. These options are described in detail in the PeopleSoft installation documentation.

For more information, see *PeopleSoft 9.2 Application Installation* for your database platform, "Setting Up the PeopleSoft Pure Internet Architecture in Silent Mode."

Administering a WebLogic Server Life Cycle

This section provides an overview of the WebLogic server life cycle and discusses how to work with server processes.

Related Topics

- Starting WebLogic
  This section discusses how to start WebLogic in various operating systems.
- Stopping WebLogic
  This section discusses how to stop WebLogic.

Understanding the WebLogic Server Life Cycle

You control a WebLogic server's life cycle primarily using a collection of scripts provided in that server's WebLogic domain directory. Each instance of a WebLogic server runs in an isolated Java Runtime Environment (JRE), regardless of whether you're testing with a single-server configuration or implementing a multi-server configuration for production. All scripts must be launched from the WebLogic domain directory; and provide usage syntax if run with –help.

Starting and Stopping Single-Server Processes

In a single-server configuration, there's only one server to administer: PIA. You can control the life cycle of the PIA server using scripts, using PSADMIN, or in the WebLogic console.

Scripts

For all platforms:

Field or Control	Description
startPIA	Use this script to start the WebLogic server locally.
stopPIA	Use this script to connect to a locally running WebLogic server and issue a shutdown command through WebLogic APIs.

> **Note**
>
> When you shut down the server, a warning is displayed since the shutdown command uses a non-SSL http connection to connect to the WebLogic Server. This shutdown command can be changed to use the SSL connection by editing the stopPIA.sh script.

For Microsoft Windows only:

Field or Control	Description
installNTservicePIA	Use this script to register the PIA WebLogic server as a Windows service that runs as a background process. The service is named as *WebLogicDomainName*-PIA, for example: peoplesoft-PIA.
uninstallNTservicePIA	Use this script to deregister the PIA Windows service.

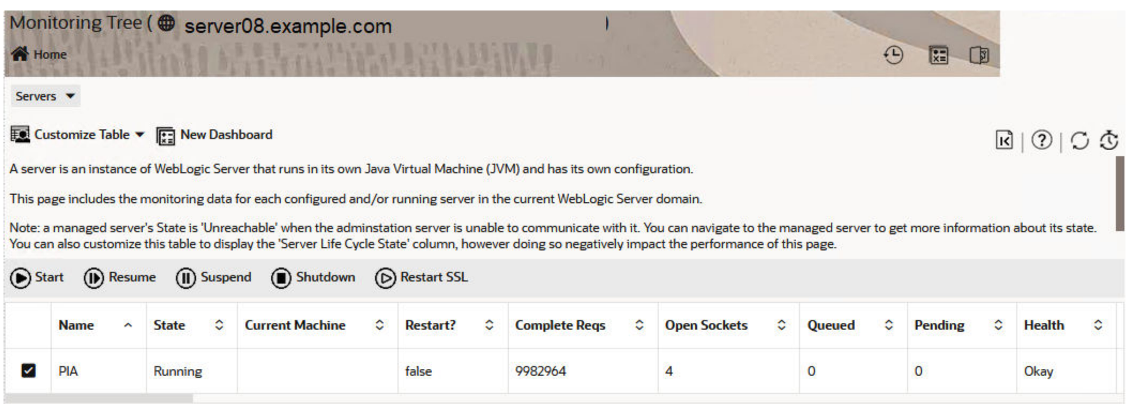
WebLogic Remote Console

A WebLogic server can also be shut down from the WebLogic Remote Console. See Using WebLogic Remote Console.

Click the Monitoring Tree tile, and then select **Environment** , and then **Servers**. Select PIA to enable these controls:

- Start
- Resume
- Suspend
- Shutdown
- Restart SSL

This example illustrates a portion of the Monitoring, Servers, PIA page . Several columns in the grid are not shown.



Starting and Stopping Multi-Server Processes

In a Multi-server configuration, as the title implies there are multiple servers to administer. Controlling the life cycle of these servers can be done by using scripts, the WebLogic console and the WebLogic Node Manager.

Scripts

For all platforms:

Field or Control	Description
startWebLogicAdmin	Use this script to start the WebLogicAdmin server.
startManagedWebLogic	Use this script to start a WebLogic managed server. All of the servers defined in a multi-server domain, except the WebLogicAdmin server, are controlled as managed servers. For example, to start PIA1 as a managed server run startManagedWebLogic PIA1.
stopWebLogic	Use this script to connect to a locally running WebLogic server and issue a shutdown command using WebLogic APIs. A remote distributed managed server can be shut down using a local administration server.

For Windows only:

Field or Control	Description
installNTservice.cmd	Use this script to register a WebLogic server as a Windows service that runs as a background process. The service is named as *WebLogicDomainName-ServerName*. For example, to define the PIA1 managed server as a Windows service, run installNTservice PIA1. To define the WebLogicAdmin server as a Windows service, simply run installNTservice.
uninstallNTservice.cmd	Use this script to deregister a WebLogic server that's defined as a Windows service.

Consider the following when using scripts with managed servers:

Operating System	Consideration
All	When starting a WebLogic managed server it will attempt to connect to its administration server. A managed server's administration server is specified either as a command line parameter when starting the managed server, or using the three administration server environment variables in setEnv, specifically ADMINSERVER_PROTOCOL, ADMINSERVER_PORT, and ADMINSERVER_HOSTNAME. The first time a managed server starts, it *must* connect to its administration server. If on subsequent startups the administration server is not available, the managed server starts up in Managed Server Independence (MSI) mode by using its locally replicated msi-config.xml. A managed server running in MSI mode can't be administered from a console, so this situation should only be encountered when it is imperative that the managed server be started even though the administration server is not running. Once the administration server is back online, running managed servers that were not previously known by the administration server to be running may be rediscovered using WebLogic's command line utility *java weblogic.Admin DISCOVERMANAGEDSERVER*, or you can just restart the managed server. To use WebLogic's java command line utility classes run setEnv to set up your environment, then run *java weblogic.Admin* for usage.
Windows	When running a WebLogic managed server as a Windows service, the managed server's administration server *must* be running. When installing a managed server as a Windows service, the managed server service can be configured to be dependent on its local administration server. To configure a managed server service to be dependent on its local admin server service use the *–depends* option of installNTservice.cmd when defining the Windows service for the managed server. In addition, when the administration server is also a Windows service, you must define it using the following command: installNTservice.cmd –delay *interval* Where *interval* is a period in milliseconds, for example, *6000*. This allows the administration server sufficient time to start before the managed server starts.

WebLogic Remote Console

A WebLogic server can also be shut down from its administration Remote Console in the same way as single server environments.

WebLogic Node Manager

The WebLogic Node Manager provides the ability to start a WebLogic managed server from the WebLogic Remote Console. In addition, the console provides a way to automatically restart a failed server. As with all WebLogic servers, the WebLogic Node Manager runs isolated in its own JRE, and on Windows it can also run as a Windows service. The WebLogic Node Manager binds to a unique IP address and port at startup and accepts lifecycle commands from a WebLogic administration server.

Multiple WebLogic domains running on a singe machine can have their managed servers administered by a shared WebLogic Node Manager, as long as each WebLogic domain uses the same version of WebLogic.

The following table lists the WebLogic Node Manager files that are provided with WebLogic server, not the PIA install. These files are located in *WLS_HOME*\wlserver\server\bin\, not your WebLogic domain directory created within PS_CFG_HOME. The default location for *WLS_HOME* that is installed by the DPK setup script is *BASE_DIR*\pt\bea\.

File	Description
startNodeManager.cmd	Use this script to start the WebLogic Node Manager as a foreground process.
installNodeMgrSvc.cmd	Use this script to define the WebLogic Node Manager as a Windows service that runs as a background process. The service is called WebLogic Platform NodeManager.
uninstallNodeMgrSvc.cmd	Use this script to uninstall the WebLogic Node Manager as a Windows service.
nodemanager.properties	This is the WebLogic Node Manager configuration file.

WLS_HOME\wlserver\common\nodemanager\NodeManagerLogs\ is the default logs directory for WebLogic Node Manager.

Additional configuration files are located in ...\common\nodemanager.

> **Note**
>
> The log and configuration files are created when Node Manager is installed.

See Oracle WebLogic Remote Console Online Help, [Configure Machines](#).

To configure Node Manager:

1. Sign in to the WebLogic Remote Console.
2. Click the Edit Tree tile, then select **Environment**, and then **Machines**.
3. Click **New**.
4. Enter a name for the new machine.

   Create one local machine (as in MachineLocal) for all of the managed servers running in the local server and one remote machine (as in MachineRemoteX) for each of the managed servers running in remote servers.

(Optional) If you are creating a machine that will run on a UNIX platform, select *UNIX Machine* from the Type drop-down list.

5. Click **Create**.
6. Configure the settings of your new machine under Environment, then Machines, then MachineLocal.
7. Select the Node Manager tab, and specify the attributes required to connect to the Node Manager process:
   - Type — Select the Node Manager type for the clients to use when connecting to the NodeManager Instance.

     Select Plain, SSH, RSH, or SSL.

     (Optional) If you set type to SSH or RSH, you should also specify values in the NodeManager home and shell command fields.
   - Listen Address — Enter the host name or IP address of the server for the server or clients on the server to use when connecting to the Node Manager.
   - Listen Port — Enter the port number for the server or clients on the server to use when connecting to the Node Manager.

     The default port number is 5556.
8. Add server instances to the machine.

   For example, you may have two managed servers running locally, with PIA1 and RPS assigned to MachineLocal.

   > **Note**
   >
   > You cannot change the machine of a server that's running, therefore you cannot use WebLogic Remote Console to change the machine of the Administration Server. Use an offline tool, such as WLST Offline instead.

9. a. Shut down the Managed Server.

      You cannot change the machine of a running server.

   b. Under Edit Tree, select **Environment**, and then **Servers**.

   c. Select the Managed Server that you want to assign to a machine.

   d. On the General tab, from the Machine drop-down list, select the machine to which you want to assign this server.

   e. Click **Save**.
10. Start the Node Manager.

    Start the Node Manger from your local WebLogic install directory (*WLS_HOME* for the WebLogic Server which your PeopleSoft domain is referring to). Node Manager can be started with the startNodeManager.cmd (for Windows) or startNodeManager.sh (for UNIX) script under *WLS_HOME*\wlserver\server\bin directory.

    After the Node Manager has been started for the first time, a file called nodemanager.properties appears under the *WLS_HOME*\wlserver\common\nodemanager directory.
11. Modify the nodemanager.properties file.

Stop the Node Manger and open the nodemanager.properties file in a text editor. Make the following changes and save the file.

SecureListener=false

ListenAddress= the IP of the computer where the Node Manager is running

> **Note**
>
> In the same file you can see the default port number is set as ListenPort=5556. Changing the port setting here and restarting the Node Manager will configure the Node Manager to listen on the newly configured port number. The port setting here must match with the one set in the machine configuration.

12. Restart the Node Manager.
13. Once the Node Manager is up and running, it should be reachable from the machine configured to listen to the local Node Manager (as in, MachineLocal). This can be confirmed on the Node Manager Status tab under the Monitoring tab.
14. In the remote computer hosting the managed server, follow the previous steps to configure the Node Manager.
15. From the bin directory under your PS_Domain (<*PS_CFG_HOME*>/psft/pt/8.59/webserv/<*web_server_domain*>/bin), run setEnv.cmd (for Windows) or setEnv.sh (for UNIX) to set up the environment.
16. Start WLST using the command `java weblogic.WLST`.
17. Connect the remote server to the Admin Server running on the local computer.

    Following is the syntax of the `connect` command:

    ```
    connect('username','password','t3://Local Machine's IP:port on which the
    admin
    server is running on Local machine')
    ```

18. Enroll the remote domain and Node Manager into the Admin Server running on the local computer.

    The command is `nmEnroll` with the following syntax:

    nmEnroll('full path to the distributed PeopleSoft Domain ','WLS_HOME\wlserver\common\nodemanager')

    For example:

    ```
    nmEnroll('D:\PT862\webserv\Dis4103Installed603','D:\WlsInstall\wlserver\com
    mon\nodemanager')
    ```

19. Once the domain and Node Manager have been enrolled, check the Remote Console to confirm that the Node Manager can be reached from the server that was configured to listen to the remote Node Manager.

    Once a Node Manager is reached from a "machine," the Remote Console will be able to start and stop the managed server assigned to that "machine."

    The Servers area shows the list of servers. The Control tab grants the control of those servers to the Admin Console.

Starting and Stopping a Distributed Managed Server

In a multi-server configuration, a distributed managed server is simply a managed server that isn't started from the same physical location as its domain's administration server. You can control the life cycle of these servers using scripts, the WebLogic Remote Console and the WebLogic Node Manager.

See Starting and Stopping Multi-Server Processes.

Tuning Performance and Monitoring Resources

Monitoring the performance of an Oracle WebLogic instance is primarily performed in the WebLogic Remote Console.

This section discusses how you can tune performance and monitoring resources for WebLogic.

Related Topics

- Using WebLogic Remote Console
  Use the Oracle WebLogic Remote Console to administer and monitor the WebLogic server processes.

Managing JVM Heap Size

This section discusses how to:

- Monitor JVM heap.
- Change JVM heap size.

Monitoring JVM Heap

The JVM heap size is the amount of memory that a particular JRE (Java Runtime Environment) gives to the JVM (Java Virtual Machine) that it creates. The java.exe command on Windows, java on UNIX and beasvc.exe when running WebLogic as a Windows service is the JRE, and the JVM exists within the JRE's memory space. The primary sources for monitoring the amount of memory that is in use within a JVM are the WebLogic Remote Console and the WebLogic logs.

To monitor the amount of the JVM heap size available and in use with the WebLogic Remote Console:

1. Sign in to the WebLogic Remote Console.
2. Click the Monitoring Tree tile, then select **Environment**, and then **JVM Runtime**, and then **General**.
3. The General page includes the following information about the JVM heap size and other information:

Field or Control	Description
Heap Free Current	The current amount of memory (in bytes) that is available in the JVM heap.
Heap Free Percent	Percentage of the maximum memory that is free.
Heap Size Current	The current size (in bytes) of the JVM heap.

Field or Control	Description
Heap Size Max	The maximum free memory that is configured for this JVM.
Java VM Vendor	The JVM vendor.
Java Vendor	The Java vendor.
Java Version	The Java version of the JVM.
Java Vendor Version	The vendor version of Java.
OS Name	The operating system on which the JVM is running.
OS Version	The version of the operating system on which the JVM is running.
Process Cpu Load	CPU load.

Changing the JVM Heap Size

If you need to adjust any of the Java options, most commonly the JVM heap size, you must manually edit that WebLogic domain's local setEnv script. The parameters, -Xms and -Xmx, control the JVM memory minimum and maximum heap size respectively.

See the following reference for examples of the JVM heap size as specified in setEnv using the JAVA_OPTIONS_*OSplatform* environment variable. You only need to set the variables that correspond to the operating system where the WebLogic server is running.

See Adjusting the JVM Heap Size.

> **Note**
>
> If you do adjust any of the Java options, most commonly the JVM heap size, you must restart WebLogic for these changes to take effect.
>
> If you're running WebLogic Server as a Windows service you must rerun the installNTservice script to propagate this change into the Windows registry.
>
> The WebLogic Node Manager does not use the Java options set in setEnv, but instead uses Java options set from the WebLogic console.

To modify the Java options for a WebLogic instance started from the WebLogic Node Manager:

1. Sign in to the WebLogic Remote Console.
2. Select **Providers**, and then **Add Admin Server Connection Provider**.

   Supply the name and password for the WebLogic administrator and the URL for the provider in this format: http://*server*:9999.
3. Select the provider.
4. Click the Edit Tree tile, then select **Environment**, and then **Servers**, and then **PIA**, and then **Advanced**, and then **Node Manager**.
5. Update the **Arguments** field.
6. Click **Save**.

   Click the Shopping Cart icon to check for pending changes and commit them if necessary.

Related Topics

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

Monitoring HTTP Session Count for PeopleSoft Interaction Hub

In addition to memory and thread usage, it's also possible to monitor the number of established HTTP sessions used in conjunction with the PeopleSoft PORTAL application. This number, although not necessarily directly related to current performance, is an indicator of the following performance factors:

- JVM memory used to store HTTP session data.
- Current number of logged on clients.
- Peak number of logged on clients.
- Idle time of logged on clients.

To monitor the total number of HTTP sessions:

1. Sign in to the WebLogic Remote Console
2. Click the Monitoring Tree tile, then select **Deployments**, and then **Application Runtime Data**, and then **peoplesoft**, and then **Component Runtimes**, and then **PIA**.
3. Review the values for Open Sessions Current Count, Open Sessions High Count, Open Session Total Count and so on.

See the following My Oracle Support document for more information: *Clustering and High Availability for PeopleTools, Doc ID 747378.1.*

Changing Configuration Settings

This section provides an overview of the WebLogic server configuration files and discusses how to change the settings for these files.

Understanding the WebLogic Server Configuration Files

WebLogic server configuration settings are stored in a collection of files. Primarily these include: setEnv script, config.xml, and the web.xml and weblogic.xml for each webapp.

Configuration File	Description
setEnv script	SetEnv contains statically and dynamically defined environment variables. It's called from all of the WebLogic administration scripts to assist in building the Java command line. You modify this file using a text editor.
config.xml	Config.xml contains server runtime settings, such as the HTTP port. You modify this file using the WebLogic Remote Console.
web.xml weblogic.xml	Located in the WEB-INF directory of each servlet, providing web application descriptors and settings relevant to their application.

Changing the WebLogicAdmin Server’s Listen Ports

In the multi-server configuration, several parameters are set based on the environment detected and delivered defaults. One such parameter is the HTTP port of the WebLogicAdmin server. By default the WebLogicAdmin server’s HTTP listen port is 9999.

To change this value:

1. Start the WebLogicAdmin server using the startWebLogicAdmin script.
2. Sign in to the WebLogic Remote Console.
3. Click the Edit Tree tile and then select **Environments**, and then **Servers**, and then **WeblogicAdmin**, and then **General**.
4. Modify the value of the **Listen Port** field.
5. Click **Save**.
6. Click the Shopping Cart icon to check for pending changes and commit them if necessary.
7. Restart the WebLogic server.

If you can't initially start the server due to a port conflict, you can manually edit the value of the ListenPort parameter in that domain’s config.xml file. Creating a backup of config.xml is recommended before manually changing this file.

After changing the ListenPort value in your domain’s config.xml, either directly or using the console, you must also update your setEnv script. Update the ADMINSERVER_PORT environment variable to reflect the new HTTP port. This setting is used by the stopWebLogic and startManagedWebLogic scripts as the default administration server HTTP port.

16

PeopleSoft Timeout Settings

Web Server Timeouts

You specify web server timeouts using the Web Profile Configuration component (WEB_PROFILE). To access these settings in PIA, select **PeopleTools**, and then **Web Profile**, and then **Web Profile Configuration**, then select the appropriate page.

The following table provides basic information about the web server timeout settings, which are more completely documented in the *PeopleTools Portal Technology* PeopleBook.

Page Element	Page Name	Description	Default
Inactivity Warning	Security	Specify how long the portal should wait before warning users that their browser session is about to expire. They can continue with their current session by clicking the OK button in the message. If a user doesn't respond, the session ends and the expired connection page appears. Suppress this warning by setting this value to be greater than the sessionTimeout value.	1080 seconds (18 minutes)

Page Element	Page Name	Description	Default
Inactivity Logout	Security	Specify the inactivity timeout interval of the PeopleSoft application for which the user is currently authenticated. When the interval passes with no user activity, the user's browser displays the page specified by the **Expire Page - Page** field on the Web Profile Configuration - Look and Feel page. **Note:** Depending on the application implementation, authenticated users might also experience an HTTP session inactivity timeout.	1200 seconds (20 minutes)
Authenticated Users - **HTTP Session Inactivity**	Security	Specify the HTTP session inactivity timeout interval that applies to authenticated users. When the interval passes with no user activity, the web server discards all session information, including cached page states. The next time the user submits a request, the web server creates a new HTTP session. If not set, the HTTP interval for an authenticated user is the same value as the inactivity logout.	0 seconds for all profile types.

Page Element	Page Name	Description	Default
Public Users - **HTTP Session Inactivity**	Security	Specify in seconds the inactivity timeout interval that applies to public users. When the interval passes with no user activity, the web server discards all session information, including cached page states. The next time the user submits a request, the web server creates a new HTTP session. Unlike authenticated users, public users are not signed out of their PeopleSoft application when this interval expires. However, PIA releases their application states from memory. If users click a link, they regain access to the application at the search dialog. This setting prevents an overload of web server resources for inactive public users.	DEV, KIOSK profile: 1200 seconds (20 minutes). TEST, PROD profile: not set.
Disconnect Timeout	Security	Specify the amount of time to wait before disconnecting the Jolt connection. A value of *0* seconds (the default) means no limit. This means that the client connection must be retained throughout the session. If the connection becomes invalid (due to one of the other timeouts) the session will be expired. **Note:** If you specify *0* seconds, the Jolt client attempts to connect the Jolt Server Handler (JSH) in RETAINED mode. If any positive value is specified, the Jolt client attempts to connect the JSH in RECONNECT mode.	0 seconds

Page Element	Page Name	Description	Default
Send Timeout	Security	Specify the maximum time permitted between the sending of the Jolt Request by the client servlet and its full receipt on the application server. **Note:** You might need to increase this value where a large amount of data is being sent to the application server, or the network is slow.	50 seconds
Receive Timeout	Security	Specify how long the client servlet should wait after issuing a Jolt Request for a response from the application server. This value should be considerably larger than the Send Timeout. Make sure that this value is also greater than your application server online service timeouts, such as the Service Timeout setting for PSAPPSRV that appear in the PSAPPSRV.CFG configuration file on the application server. **Note:** Ideally, this timeout should also be greater than the Tuxedo SANITY_SCAN setting (BLOCKTIME * SCANUNIT).	1300 seconds

Related Topics

- Portal Technology: Configuring Portal Security

Session-Timeout

You specify the web server *session-timeout* using the Inactivity Logout property and HTTP Session Inactivity property in the web profile.

Web Server Default System Timeout

PeopleSoft Interaction Hub normally depends on a content reference timeout setting to determine how long to wait for a pagelet to load before it considers the pagelet to be unavailable. However, if the remote server is unavailable, the content reference timeout setting is ignored. If the portal can’t establish a connection to the remote host, it uses the default system timeout.

The default system timeout defaults to 20 seconds. If you expect the remote server to be slow or down for longer than 20 seconds, you should specify a longer default system timeout, by configuring your web server to set the defaultConnectTimeout JVM environment variable to an appropriate value using one of the following procedures.

For example,

```
SET JAVA_OPTIONS_WIN32=-server -Xmsnnnm -Xmxnnnm -XX:MaxPermSize=nnnm
 -Dsun.net.client.defaultConnectTimeout=default_timeout
```

Where *default_timeout* is the number of milliseconds that the portal should wait to establish the connection to the host.

See the web server documentation for instructions on modifying this JVM environment variable.

Application Server Timeouts

All configurable settings for the application server require modification in PSADMIN:

Name	In This File	Description	Default
JOLT Listener/Client CleanupTimeout	psappsrv.cfg	Specify the inactivity interval permitted for the server-side JoltSession. Specifying too low a value can cause unnecessary reinstantiation of resources for clients who surpass this inactivity interval. However, specifying too high a value can keep unnecessary server-side resources allocated. **Note:** This setting doesn't affect the user experience, but it has an impact on server-side performance.	10 minutes
JOLT Listener/Init Timeout	psappsrv.cfg	Specify the amount of time that's allowed for the JSL process to start. **Note:** It's not necessary to adjust this setting.	5 minutes

Name	In This File	Description	Default
Workstation Listener/ Client Cleanup Timeout	psappsrv.cfg	Specify the inactivity interval permitted for the server-side Workstation Listener Session. Specifying too low a value can cause unnecessary reinstantiation of resources for clients who surpass this inactivity interval. However, specifying too high a value can keep unnecessary server-side resources allocated. **Note:** This value is required only for three-tier connections.	60 minutes
Workstation Listener/ init Timeout	psappsrv.cfg	Specify the amount of time that's allowed for the WSL process to start. **Note:** It's not necessary to adjust this setting. This value is required only for three-tier connections.	5 minutes

Name	In This File	Description	Default
Spawn Threshold	psappsrv.cfg	Applies only if spawning is enabled. Specify the rates at which PSAPPSRV processes spawn and decay. The spawn rate is determined by the last two numbers, and the decay rate is determined by the first two numbers. Using the default value as an example, for the spawn rate of *1,1* an extra PSAPPSRV process is spawned if there is at least 1 outstanding service request on the application server request queue for 1 second or more. This spawning will continue until the PSAPPSRV Max Instances value is reached. For the decay rate of *1,600* a server process is decayed if less than 1 service request is in the application server request queue for 600 seconds (ten minutes) or more. **Note:** This parameter applies only if, for PSAPPSRV, the value of *Max Instances* is greater than that of *Min Instances*.	1,600:1,1

Name	In This File	Description	Default
Service Timeout	psappsrv.cfg	Each server process has its own instance of this setting in its section of the psappsrv.cfg file. Specify the maximum interval for services to run in a given process. If a service has not completed within the specified interval, Tuxedo terminates the server processing and restarts the server process. For each server process, specify the longest time that any service is expected to take. **Note:** A value of 0 produces an indefinite timeout for any service.	PSAPPSRV: 300 seconds (5 minutes) PSSAMSRV: 300 seconds PSQCKSRV: 300 seconds PSQRYSRV: 1200 seconds (20 minutes) PSBRKHND_dflt: 1200 seconds PSSUBHND_dflt: 1200 seconds PSPUBHND_dflt: 1200 seconds
Restart Period (PSBRKDSP_dflt, PSSUBDSP_dflt, PSPUBDSP_dflt)	psappsrv.cfg	Specify how long each dispatcher should wait before redispatching a message if the associated handler has not started processing it.	120 seconds
TM_RESTARTSRV TIMEOUT	psappsrv.ubx (which is the template for psappsrv.env)	Specify the time period that a domain server process (for example, PSAPPSRV, PSWATCHSRV, PSSAMSRV) is permitted to remain in restarting mode before it is killed by Tuxedo. This setting resolves processes hanging during restart. **Note:** To modify this setting, you must change the value in the .UBX template file, then recreate your domain.	60 seconds (one minute)

Process Scheduler Timeouts

All configurable settings for PeopleSoft Process Scheduler require modification through domain configuration within PSADMIN:

Name	In This File	Description	Default
Process Scheduler/ Reconnection Interval	psprcs.cfg	Specify the interval between attempts to reconnect to the database when the connection is lost.	300 seconds (5 minutes)
Process Scheduler/ Authentication Timeout	psprcs.cfg	Specify how long PeopleSoft Security has to authenticate a process that's released by PeopleSoft Process Scheduler The timer starts when Process Scheduler initiates the request.	5 minutes
RemoteCall/RCCBL Timeout	psprcs.cfg	Specify the maximum interval for a remote call from an Application Engine program to run before it's terminated. This is similar to a general Tuxedo service timeout.	300 seconds (5 minutes)

For Spawn Threshold, see the application server timeout settings.

PIA Timeouts

A number of additional timeouts may be set through PIA. These settings reflect changes at the database level that may pertain to different groups of users.

> **Note**
>
> The timeout settings in PIA are optional and are not required to run PIA. However, an understanding of how these settings can contribute to a user's session duration is important in the context of other timeout values that appear in this topic.

Name	Navigation Path	Description	Default
Authentication Token expiration time	**PeopleTools**, and then **Security**, and then **Security Objects**, and then **PeopleSoft Single Signon**	Specify the interval during which the system can trust a single sign-on token (PS_TOKEN) from the same or another content provider. **Note:** As long as users remain signed in, the expiration of PS_TOKEN does not affect them. This setting is relevant only for the GetCertificate request during single sign-on.	720 minutes (12 hours)
Permission List - Time-out Minutes	**PeopleTools**, and then **Security**, and then **Permissions and Roles**, and then **Permission Lists**	Specify an interval during which a given permission list applies. The interval starts for a user to which the permission list is assigned when that user signs in. When the timeout period elapses, the user's online session is terminated. If a user belongs to multiple permission lists, the largest timeout value from among those permission lists is applied to the user's session during sign in. The permission list timeout is effective only if its value is less than the web server session-timeout. This means that all of the permission list timeouts for a given user must be less than the web server session-timeout to be effective. However, the Inactivity Warning timeout still applies. **Note:** A value of zero (0) produces an indefinite timeout.	0 minutes

Related Topics

- Web Server Timeouts
  You specify web server timeouts using the Web Profile Configuration component (WEB_PROFILE). To access these settings in PIA, select **PeopleTools**, and then **Web Profile**, and then **Web Profile Configuration**, then select the appropriate page.

17

Ensuring Session Stickiness

Working With Session Cookie Names

Session stickiness, generally, refers to ensuring that the requests for a particular session get routed to the same physical machine that serviced the first request for that session. This helps to reduce the situation where sessions are lost as a result of requests for a session being routed to different servers.

For either of the following scenarios, all PeopleSoft web servers need to have the same session-cookie name:

- The Load-Balancer uses the PeopleSoft web server's session-cookie as the stickiness criterion.
- Any of the supported proxy servers is used to forward requests to the PeopleSoft web servers.

Ensure that the session-cookie name is the same for all web server instances to which the Load Balancer or proxy dispatches requests. The cookie name needs to be configured in the appropriate configuration section of the Load-Balancer or proxy server.

Using PeopleSoft Interaction Hub and Content Providers

Typically, PeopleTools recommends that the portal and content sites run in separate PeopleSoft Internet Architecture domains. If these sites are running in the same PeopleSoft Internet Architecture domain (as in having multiple sites within the same domain, with one site for the portal and the other sites for content), then there is no need to load-balance requests separately to the content sites, because the requests will be routed to the same JVM's.

When the portal and content sites run in the same PeopleSoft domain as multiple sites, they share the same session because the web application is the same for all sites. PeopleTools does not recommend configuring portal and content sites in the same PeopleSoft domain for performance and reliability reasons.

So assuming that portal and content sites are running in separate PeopleSoft domains, they will have separate ports and, usually, separate hosts. In this case, if portal and content sites are using the same physical load balancer, the routing rules will have to be set up for these sites separately.

When using PeopleSoft Interaction Hub and Content Providers:

- The session-cookie name of PeopleSoft Interaction Hub and the Remote Content Provider must be *absolutely* distinct. For example,

  PeopleSoft Interaction Hub's session-cookie name: PTENPRTL-8080-PORTAL-PSJESSIONID

  Content Provider's session-cookie name: PTFIN-9090-PORTAL-PSJESSIONID
- Neither of the systems' session-cookie name must be part of the other's session-cookie name. The following example illustrates a *bad choice* of a session-cookie name:

PeopleSoft Interaction Hub's session-cookie name: PORTAL-PSJESSIONID

Content Provider's session-cookie name: PTFIN-9090-PORTAL-PSJESSIONID

- If there are multiple PeopleSoft Interaction Hub or Content Provider web servers with Load-Balancer or Proxy, see carefully choose a session-cookie name.

  For example, consider an environment having a Load Balancer in front of two PeopleSoft Interaction Hub web servers, with the session-cookie's name used as Load Balancer's stickiness criterion. Suppose the name of the session-cookie is PTENPRTL-8080-PORTAL-PSJESSIONID. In this case, both PeopleSoft Interaction Hub web servers must have the same session-cookie name: PTENPRTL-8080-PORTAL-PSJESSIONID.

  If this environment has an HCM Load balancer with four HCM web servers, with the session-cookie name as stickiness criterion, each of the HCM web servers must have the same cookie, like HTR-9090-PORTAL-PSJESSIONID and this name must be distinct from the name of the session-cookie used for PeopleSoft Interaction Hub.

Working With Absolute URLs

PeopleSoft Internet Architecture constructs absolute URLs in certain scenarios (such as the Homepage URL to which the user is re-directed after successful login). Certain Load Balancers and proxy servers change the HOST header of the request to the target web server's name before forwarding the request. This causes PeopleSoft to construct the URL with the host-name of the web-server, rather than the name of the Load Balancer or proxy, which means that subsequent requests would be sent directly to the web server, instead of routing requests by way of the Load Balancer or proxy. In order to ensure that URLs can be constructed with the Load Balancer or proxy's host name, the Virtual Addressing page of the Web Profile needs to be set with the Load Balancer or proxy information.

To modify virtual addressing information:

1. Select PeopleTools, Web Profile, Web Profile Configuration, and open the web profile configured for the site.
2. Click on the Virtual Addressing tab and populate the Default Addressing section with the Load-Balancer or Proxy's host name, scheme and port.

   Example: Suppose the system has been set up to be accessed through a Load Balancer which has a host-name of example.com and accepts http requests at port 8080. Suppose the PeopleSoft site's name is ps. The URL for accessing the site would be as follows:

   http://example.com:8080/ps/signon.html

   In this case, the Default Addressing section will need to be set with Protocol: http, Port: 8080, and Host: example.com.
3. Save the Web Profile.
4. Re-start the web server.

Setting The Cookie Domain

Browsers enforce the rules stated by Request for Comments (RFC) document 2965 for sending cookies to servers. If the cookie's domain is not set when the browser receives the cookie, that cookie is associated with the server that sent the cookie. If the cookie's domain has been set but does not match the name or the domain suffix of the server, then the browser does not send the cookie. Therefore, not having the correct cookie domain can lead to failure in maintaining a PeopleSoft session.

The cookie domain value is configured in web server-specific configuration by the PeopleSoft Internet Architecture setup if the Authentication Token Domain value has been entered. If this field was left blank during PeopleSoft Internet Architecture setup, either re-install PeopleSoft Internet Architecture or configure the cookie domain manually in the web server-specific configuration.

Working with Load Balancer Timeouts

If session-stickiness is configured with a timeout on the Load Balancer, that timeout value must be higher than the PeopleSoft Inactivity Logout timeout configured in the current Web Profile.

Keep in mind the session cookie name format. The session-cookie name must not start with a digit (0-9) and must not contain dots.

Example of bad name for session-cookie: 11.12.13.14-80-PORTAL-JSESSIONID

Example of good name for session-cookie: FINTSTWEB-80-PORTAL-JSESSIONID

Configuring the Session Cookie Name for WebLogic

The session cookie name is configured in weblogic.xml of the portal servlet (web application). This file can be found in the following directory: *<PIA_HOME>*/webserv/*<DOMAIN-NAME>*/applications/peoplesoft/PORTAL/WEB-INF

Locate the tag CookieName in weblogic.xml, as in the following example:

```
<session-param>
  <param-name>CookieName</param-name>
  <param-value>PSTST-8080-PORTAL-PSJSESSIONID</param-value>
  </session-param>
```

The name of session cookie is assigned in the tag <param value>. In this example, the session-cookie name is `PSTST-8080-PORTAL-PSJSESSIONID`.

Re-start the WebLogic Server to make this change effective.

Configuring Session Cookie Domain for WebLogic

Update the CookieDomain manually in weblogic.xml file, as in the following example:

```
<session-param>
      <param-name>
      CookieDomain
      </param-name>
      <param-value>
      .example.com
      </param-value>
```

In this example, the value `.example.com` is the domain that will be assigned to the session-cookie.

Re-start the WebLogic Server to make this change effective.

18

Troubleshooting Server Issues

Uploading Files Using Non-Latin Characters

When uploading an attachment file with a filename that uses non-Latin characters, such as Hebrew or Japanese, the system uses the exact filename or UTF-8 % sequenced numbers. In previous releases, the system renamed the file and converted to strings of numbers.

When uploading non-Latin characters, the following error can occur: "Attachment Filename is too long (maximum is 64 characters)."

Solution For UNIX

To resolve this issue on UNIX servers:

1. Make sure that the corresponding language locale, such as the UTF-8 character set, is installed at the operating system level.

   Use "locale -a" to get the list of installed locales.

2. Use the LANG environment variable to set for the session that runs the application server to a valid locale for the character set.

   For example, en_US.utf8, iw_IL.utf8, ja_JP.utf8, and so on. See your "locale -a" output.

3. Use PSADMIN to set the Character Set parameter to the appropriate value, as in Character Set=utf8.

   The character set values specified in the shell locale (LANG) and PSAPPSRV.CFG (Character Set) must agree, and, in this case, PSAPPSRV.CFG requires utf8.

> **Note**
>
> If the shell locale is set to something other than the default of C, Tuxedo will issue warnings when the application server boots. In such a case, add a symlink from $TUXDIR/prod/locales/<your_shell_locale> to $TUXDIR/prod/locales/C to resolve that issue.

Solution For Windows

On a Windows-based application server, make sure the server operating system has the corresponding language installed.

To resolve this issue on Windows servers:

1. Make sure that the corresponding language locale, such as CP932 (for Japanese) character set, is installed at the operating system level.

Use the Regional and Language Options utility in the Control Panel to get the list of installed locales, and use “chcp” in the command prompt window to see the current default code page.

2. Use Control Panel/Regional and Language Options to set the system locale for the session that runs the application server with a valid language.

   For example, on the Advanced tab for the Language for non-Unicode programs option, select *Japanese* for the Japanese system locale of CP932.

3. Use PSADMIN to set the Character Set parameter so that it reflects the appropriate code set.

   Using the Japanese example: the default code page CP932 (equivalent to SJIS) and the character set value in PSAPPSRV.CFG must agree. So in this case, PSAPPSRV.CFG requires “sjis” to be set as the character set, as in Character Set=sjis.

19

Replicating an Installed Environment

Understanding Environment Replication

Environment Replication involves taking a working, well-tested server elements, and copying the PeopleTools binary and configuration files to a new location to create a new environment that required only making minor modifications to the new configurations.

The server elements that can be replicated appear in the following:

Server Element	Description
PeopleSoft Internet Architecture	A web server environment consists of one or more site directories under the *PIA_HOME*\webserv directory within a single *PIA_HOME* location. Each directory contains configurations that point to a single application server domain through a JSL port designation. Although a site may point to more than one redundant application server machine name for failover or load balancing purposes, each machine name is given a unique JSL port number. Additional *PIA_HOME* directory locations on the same web server machine are considered separate environments.
application server domains	A single application server environment consists of one or more domain directories under the \appserv directory within a single *PS_CFG_HOME* location. Each domain contains configuration settings that point to a single database. Multiple domains can be configured to point to the same database for failover or load balancing. Each domain has its own server processes and must be configured to have unique WSL and JSL port numbers. To replicate PS_CFG_HOME, see Replicating Configuration Home (PS_CFG_HOME).
Process Scheduler domains	A single Process Scheduler Environment consists of one or more domain directories under the \appserv\prcs directory within a single *PS_CFG_HOME* location. Each database that needs to schedule processes must have its own Process Scheduler server configured separately. The configuration files are kept in their own directory location under the \prcs directory.

Important

When you configure and run an Environment Management agent, hub, or viewer, files associated with that Environment Management Framework are created and updated with information that refers to the absolute directory structure of your PeopleSoft system. When you replicate your installed environment, the information in those files is no longer valid for the new location. You must reconfigure the Environment Management Framework to recognize the environment changes.

Replicating a Web Server Environment

Use the following steps to replicate the web server environment:

1. Copy the web server *PIA_HOME*\webserv directory structure to a new location, whether it is a new machine, or same machine with a new high-level *PIA_HOME* directory name.
2. Review the following configuration files to make sure they reflect the new location:
   - piainstall.xml: PS_HOME and host name
   - psconfig.sh: PS_HOME and path information
   - peopletools.properties: install location
3. Run *PS_HOME*\setup\PsMpPIAInstall\setup.exe or equivalent from the copied/cloned location.
4. Select Redeploy PeopleSoft Application.
5. Choose a site name.
6. Enter the application server name and port information.

Note

Be sure to enter a different JSL port value when replicating on the same machine.

Reconfiguring Replicated Environment Management Components

This section discusses how to reconfigure the components for replicated environment management.

Reconfiguring an Environment Management Agent

Configuring PSEMAgent in a replicated PS_HOME involves deleting the agent local cache and modifying the PS_HOME location specified within the PSEMAgent configuration files.

> **Note**
>
> *New_PS_HOME* refers to the directory where PeopleTools is located in your*replicated* environment. If the same PS_HOME location is used by your source and destination environment, you do not need to make any changes.

To reconfigure the agent:

1. Delete all subdirectories below *New_PS_HOME*\PSEMAgent\envmetadata\data\

   > **Note**
   >
   > Don't delete the matchers.xml file in this location.

2. Delete the following directories:
   - *New_PS_HOME*\PSEMAgent\envmetadata\PersistentStorage
   - *New_PS_HOME*\PSEMAgent\envmetadata\scratchpad
   - *New_PS_HOME*\PSEMAgent\envmetadata\transactions
3. Modify StartAgent.bat (Windows), or StartAgent.sh (UNIX).

   Ensure that references to the drive and path of the PSEMAgent directory are correct for the *New PS_HOME* location.
4. Modify StopAgent.bat (Windows) or StopAgent.sh (UNIX).

   Ensure that references to the drive and path of the PSEMAgent directory are correct for the *New PS_HOME* location.
5. Verify the settings in the agent configuration file: *New_PS_HOME*\PSEMAgent\envmetadata\config\configuration.properties.

   Properties that specify path locations must be valid for the replicated agent in its new location, but the same hub is still addressed by all agents.
   - hubURL
   - windowsdrivestocrawl
   - unixdrivestocrawl

Reconfiguring the Environment Management Hub

The Environment Management hub is replicated as part of replicating the web server environment. After redeploying PeopleSoft Internet Architecture, clear the cache by performing the instructions below:

> **Note**
>
> *New_PIA_HOME* is the directory where PeopleSoft Internet Architecture is located in your replicated environment.

To reconfigure the hub:

1. Delete all subdirectories below *New_PIA_HOME*\webserv\peoplesoft\applications\peoplesoft\PSEMHUB\envmetadata\data\

   > **Note**
   >
   > Don't delete the data.txt file in this location.

2. Delete the following directories:
   - *New_PIA_HOME*\webserv\peoplesoft\applications\peoplesoft\PSEMHUB\envmetadata\PersistentStorage
   - *New_PIA_HOME*\webserv\peoplesoft\applications\peoplesoft\PSEMHUB\envmetadata\scratchpad
   - *New_PIA_HOME*\webserv\peoplesoft\applications\peoplesoft\PSEMHUB\envmetadata\transactions

Reconfiguring the Environment Management Viewer

In the replicated environment, local cache files need to be cleared for the *New_PS_HOME*\PSEMViewer directory.

> **Note**
>
> *New_PS_HOME* is the directory where PeopleTools is located in your replicated environment.

To reconfigure the PSEMViewer:

1. Delete all subdirectories below *New_PS_HOME*\PSEMViewer\envmetadata\data\

   > **Note**
   >
   > Don't delete the files in this location, just the subdirectories.

2. Delete the following directories:
   - *New_PS_HOME*\PSEMViewer\envmetadata\PersistentStorage
   - *New_PS_HOME*\PSEMViewer\envmetadata\scratchpad

Updating the GUID Value of Cloned Databases

When copying databases, it is extremely important to delete the GUID value in the new (copied) database. If not deleted, the hub will assume that the old and new environments are the same, leading to confusing environment records. To resolve this, set the value of the GUID field in the PSOPTIONS table to <space> in the cloned database using the SQL tool at your site. The next time an application server connects to the database, the system generates a new, unique GUID for the cloned database.

20

Working with PSADMUTIL

Understanding PSADMUTIL

The PSADMUTIL utility runs outside of PSADMIN for the purposes of:

- preloading file or database cache.
- purging database cache.

Preload cache projects as well as stored file and database cache repositories can be very large. Because of this, it is not advantageous to have caching operations subject to Tuxedo timeout limits. PeopleTools uses PSADMUTIL so that the operations it handles run independently of PSADMIN and Tuxedo.

When running the preload cache or purge cache options from PSADMIN, the PSADMIN utility calls PSADMUTIL to perform these operations.

The PSADMUTIL utility can also be invoked outside of PSADMIN from the command line.

Using PSADMUTIL from the Command Line

The psadmutil.exe is located in the *PS_HOME*\bin\server directory for the application server installation. The command line interface for psadmutil.exe uses the following syntax:

```
psadmutil -CT dbtype [-CS server name] -CD database_name -CO user_ID
-CP user_password -CI connect_ID -CW connect_password [-Preload
[ProjectName]] [-Purge]
```

The following table describes the accepted command line parameters.

Parameter	Argument	Example
–CT	Specify the database type. Valid values are DB2ODBC, MICROSFT, and ORACLE. **Note:** Notice the spelling of MICROSFT. DB2ODBC is the database type for Db2 z/OS.	`-CT ORACLE`
–CS	(Optional) Specify the name of the database server for the database to which you're connecting.	`-CS pt-server1`
–CD	Specify the name of the database to connect to, as you would when signing in to PeopleSoft.	`-CD FSDMO`

Parameter	Argument	Example
–CO	Specify the PeopleSoft user ID you're using to sign in.	`-CO VP1`
–CP	Specify the user password for the PeopleSoft user ID you specified.	`-CP` *`password`*
-CI	Specify the connect ID used to connect to the database server.	`-CI people`
-CW	Specify the password for the Connect ID you specified.	`-CW` *`password`*
-Preload [preload cache project]	Invokes the preload cache operation. If this parameter is used, you must also specify the name of the preload project. **Note:** If you do not specify a cache project name, the system reads the value from the server configuration file (PSAPPSRV.CFG or PSPRCS.CFG file), based on the PS_SERVER_CFG file.	`-Preload PLC_HR_PAYROLL`
-Purge	Purges the cache for the specified database. **Note:** This option only applies if database caching is implemented.	`-Purge`

> **Note**
>
> If any of these parameters are not specified, the system reads them from the PSAPPSRV.CFG or PSPRCS.CFG file. For example, for security reasons, the database connection related parameters do not need to be specified on the command line, and the system can get these values from the server configuration files.

> **Note**
>
> Before running psadmutil.exe from the command line, make sure PS_SERVER_CFG and PS_SERVDIR are set correctly. When PSADMIN calls psadmutil.exe, these environment variables are already set by PSADMIN to the valid values.

21

Implementing MCP Server for Chat UI

Implementing MCP Server for Chat UI

Model Context Protocol (MCP) is an open standard that provides a consistent way for AI applications to connect with external systems.

The MCP Server covered in this topic is specific to the PeopleSoft Chat UI. It provides the server-side integration needed for Chat UI to communicate with an MCP client and an AI agent implementation. This topic describes how to configure and use the MCP Server to process chat requests and return responses in the Chat UI.

> **Note**
>
> Use this implementation only when working with the PeopleSoft-delivered Chat UI. If you want to implement a generic MCP deployment that exposes PeopleSoft APIs to external LLM agents, refer to Configuring the MCP Stream Server.

> **Note**
>
> The framework provides the interfaces required to implement AI agents for the Chat UI but does not include AI agent logic. Customers are responsible for implementing the integration with their chosen LLM provider and MCP client, including connecting to the PeopleSoft MCP Server for the Chat UI, discovering and invoking MCP tools, and formatting the responses returned to Chat UI based on their business requirements.

Follow these configuration steps to enable the MCP Server for the Chat UI:

1. Set up Application Server domain. The application server domain boot process starts the PSASSISTANTSRV server and advertises the GenAgentResponse service.
   Login as the psadm1 user and run the following commands:

   ```
   $ cd $PS_HOME/bin/ps_assistant
   $ python -m venv venv
   $ source venv/bin/activate
   $ python -m pip install -r requirements.txt
   ```

2. Enable the MCP Server feature for the Chat UI within the application server domain settings. This feature is disabled by default.
   The following image illustrates the MCP Server feature for the Chat UI in the application server domain.

```
---------------------------------------------
Quick-configure menu -- domain: CHATDOM
---------------------------------------------
      Features                      Settings
     ==========                    ==========
  1) Pub/Sub Servers    : Yes   18) DBNAME       :[CHATDOM]
  2) Quick Server       : No    19) DBTYPE       :[ORACLE]
  3) Query Servers      : Yes   20) UserId       :[]
  4) Jolt               : Yes   21) UserPswd     :[]
  5) Jolt Relay         : No    22) DomainID     :[TESTSERV]
  6) WSL                : Yes   23) AddToPATH    :[.]
  7) PC Debugger        : Yes   24) ConnectID    :[]
  8) Event Notification: Yes    25) ConnectPswd :[]
  9) MCF Servers        : Yes   26) DomainConnectPswd:[]
 10) Perf Collator      : Yes   27) PHC Remote Admin Settings:[]
 11) Analytic Servers   : Yes   28) WSL Port     :[7000]
 12) Domains Gateway    : No    29) WSL SSL Port:[7025]
 13) Push Notifications: Yes    30) JSL Port     :[9000]
 14) MCP Server         : Yes   31) JSL SSL Port:[9025]
                                32) JRAD Port    :[9100]

      Actions
     =========
 15) Load config as shown
 16) Custom configuration
 17) Edit environment settings
  h) Help for this menu
  q) Return to previous menu
```

Enter **Yes** to enable the MCP Server feature for the Chat UI.

Enable the MCP Server feature for the Chat UI on only a single application server domain.

> **Note**
>
> The MCP Server feature for the Chat UI is only available for Linux application server domains.

3. Implement the `AgentOrchestrator` class based on your AI agent, LLM provider, and MCP client requirements. The `CustomAgentOrchestrator` class acts as the integration point between Chat UI, the MCP client, and the AI agent implementation.
   The following sample shows the default `CustomAgentOrchestrator` implementation. Users can extend or replace this class to integrate their own MCP client, AI Agent, and business logic.

```
class CustomAgentOrchestrator(AgentOrchestrator):
    """
    Custom Agent Orchestrator template.

    Users can subclass or replace this implementation to integrate their
own
    AI Agent, MCP client, and business logic while keeping the server
contract
    unchanged.
    """
```

```
    def __init__(self, config_file: str):
        """
        Initialize the custom agent orchestrator.

        This constructor is invoked during the booting of the
PSASSISTANTSRV
        application server process.

        Users can implement the following initialization logic here:

        1. Read the MCP Server configuration from the config file.
        2. Initialize the MCP client based on the selected LLM provider.
        3. Connect the MCP client to the configured MCP Server.
        4. Retrieve the available MCP Server tools.
        5. Initialize the AI Agent.
        6. Associate the retrieved MCP tools with the AI Agent.
        """

        # Store the configuration file path.
        # The config file contains MCP Server details under the
"mcpServers" section.
        self.config_file = config_file

        # Example placeholders for user implementation:
        #
        # self.mcp_client = initialize_mcp_client(self.config_file)
        # self.mcp_tools = self.mcp_client.get_available_tools()
        # self.ai_agent = initialize_ai_agent()
        # self.ai_agent.register_tools(self.mcp_tools)

    def get_query_result(self, user_query: dict) -> str:
        """
        Process the chatbot user query and return the response.

        This method is invoked when a chat request is received from the
chatbot UI.

        The user_query dictionary contains the user input, authentication
details,
        chat metadata, and request-specific context.

        Sample user_query structure:

        {
            "username": "",
            "password": "",
            "token": "<PSTOKEN_REDACTED>",
            "basicauthtoken": "<BASIC_AUTH_TOKEN_REDACTED>",
            "headers": {},
            "userprompt": "My emplid is KU0046, emplrcd is 0 and
department name is CS",
            "message": "Hi",
            "chatid": "2ecadd283d1c82e597fcf2fa7e15b021",
            "authtype": "pstoken"
        }
```

```
        Field details:

        - username: User name, if available.
        - password: Password, if applicable.
        - token: PeopleSoft authentication token. This value should be
handled securely.
        - basicauthtoken: Basic authentication token. This value should be
handled securely.
        - headers: Additional request headers, if any.
        - userprompt: User-provided context or prompt details.
        - message: Current chat message entered by the user.
        - chatid: Unique chat session identifier.
        - authtype: Authentication type used for the request, such as
"pstoken".

        Users can pass the relevant details from user_query to the AI Agent
        conversation flow.

        The AI Agent can then invoke the appropriate MCP Server tools,
retrieve
        the required data, format the response, and return it as a string.

        The returned string will be displayed in the chatbot UI.
        """

        # The user_query parameter contains the user input and chat
metadata.
        # Example keys available in user_query:
        #
        # user_query["username"]       -> User name, if provided.
        # user_query["password"]       -> Password, if applicable.
        # user_query["token"]          -> PeopleSoft token. Handle
securely.
        # user_query["basicauthtoken"] -> Basic auth token. Handle
securely.
        # user_query["headers"]        -> Additional request headers.
        # user_query["userprompt"]     -> User-provided prompt/context.
        # user_query["message"]        -> Current chat message.
        # user_query["chatid"]         -> Chat session ID.
        # user_query["authtype"]       -> Authentication type, for example
"pstoken".
        #
        # Avoid logging sensitive values such as password, token, or
basicauthtoken
        # in production environments.

        # Example placeholder for user implementation:
        #
        # response = self.ai_agent.invoke(user_query)
        # return response

        return (
            "CustomAgentOrchestrator default implementation. "
            "Override get_query_result to process user queries."
        )
```

```
    def delete_all_chat_history(self, chatid: str) -> List[str]:
        """
        Delete chat history for the given chat ID.

        Users can override this method if their AI Agent or LLM provider
stores
        chat history externally and requires explicit cleanup.

        The method should return a list of deleted chat history
identifiers,
        if applicable.
        """

        # Default implementation does not perform any chat history cleanup.
        return []
```

Sample User Query Payload Format

Use the following sample `user_query` payload format in the `AgentOrchestrator` class implementation:

```
{
  "user_query": {
    "username": "",
    "password": "",
    "token": "<PSTOKEN_REDACTED>",
    "basicauthtoken": "<BASIC_AUTH_TOKEN_REDACTED>",
    "headers": {},
    "userprompt": "<USER_PROMPT_FROM_UI>",
    "message": "<USER_MESSAGE_FROM_UI>",
    "chatid": "<UNIQUE_CHAT_ID>",
    "authtype": "pstoken"
  }
}
```

Use the following configuration parameters to define `user_query`.

Field Name	Type	Description
`username`	str	Not required for this implementation.
`password`	str	Not required for this implementation.
`token`	str	This field will have `pstoken` value and it is used for PeopleSoft Application Service authentication. This token is generated by the UI and passed to the Agent Orchestrator as part of the `user_query` metadata.

Field Name	Type	Description
`basicauthtoken`	str	Proxy user login details are used for PeopleSoft Application Service authentication. These details are generated by the UI and passed to the Agent Orchestrator as part of the `user_query` metadata.
`headers`	dict	Not required for this implementation.
`userprompt`	str	The prompt data configured in the Prompt table will be available in this field. Users must define and set the prompt through the UI.
`message`	str	The actual user chat message entered in the PeopleSoft Assistant Chat UI will be available in this field.
`chatid`	str	A unique ID generated by the UI to track the conversation.
`authtype`	str	The value is `pstoken`, which is automatically set by the UI.

Related Topics

- Portal Technology: Working with Chat User Interface (Chat UI)