ORACLE
PeopleSoft

PeopleTools 8.60: Security Administration

July 2024

ORACLE

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at https://docs.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customers that have purchased support have access to electronic support through My Oracle Support. For information, visit https://docs.oracle.com/pls/topic/lookup?ctx=acc&id=info or visit https://docs.oracle.com/pls/topic/lookup?ctx=acc&id=trs if you are hearing impaired.

Contents

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 Website.

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, and 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."

To configure the context-sensitive help for your PeopleSoft applications to use a locally installed online help website, see Configuring Context-Sensitive Help Using a Locally Installed Online Help Website.

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 two main types of common help are:

- Application Fundamentals

- Using PeopleSoft Applications

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.

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 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.

Example of a region-specific heading: "(Latin America) Setting Up Depreciation"

Region Identifiers

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

- Asia Pacific
- Europe
- Latin America
- North America

Industry Identifiers

Industries are identified by the industry name or by an abbreviation for that industry. The following industry identifiers 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

PeopleTools 8.60 Home Page

PeopleSoft Search and Insights Home Page

“PeopleTools Product/Feature PeopleBook Index” (Getting Started with PeopleTools)

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

Chapter 1

Getting Started with Security Administration

Security Administration Overview

This section discusses:

- User security.
- Lightweight Directory Access Protocol (LDAP).
- Authentication and single signon.
- Data Encryption.
- Query and definition security.
- PeopleSoft personalizations.

User Security

User security is the core of security administration in PeopleSoft applications. You administer user security using several basic elements.

To establish appropriate user access:

1. Define permission lists.

   Permission lists are the building blocks of user security authorization. A permission list grants a degree of access to a particular combination of PeopleSoft elements, specifying pages, development environments, time periods, administrative tools, personalizations, and so on.

   This level of access should be appropriate to a narrowly defined and limited set of tasks, which can apply to a variety of users with a variety of different roles. These users might have overlapping, but not identical, access requirements.

   You typically define permission lists before you define roles and user profiles. When defining permission lists, however, consider the roles that you will use them with.

   See Understanding Permission Lists

2. Define roles.

   A *role* is a collection of permission lists. You can assign one or more permission lists to a role. The resulting combination of permissions can apply to all users who share those access requirements. However, the same group of users might also have other access requirements that they don't share with each other. You can assign a given permission list to multiple roles.

You typically define roles after first defining their permission lists, and before defining user profiles. You use roles to assign permissions to users dynamically.

See Understanding Roles.

3. Define user profiles.

   A *user profile* is a definition that represents one PeopleSoft user. Each user is unique; the user profile specifies a number of user attributes, including one or more assigned roles. Each role that's assigned to a given user profile adds its permission lists to the total that apply to that user.

   You typically define user profiles after defining their roles. You can assign a given role to multiple user profiles. It's worthwhile to define a set of roles that you're confident can be assigned to user profiles that you'll create in the future.

   See Understanding User Profiles.

LDAP

LDAP is an internet protocol used to access a directory listing. Organizations typically store user profiles in a central repository, or *directory server,* that serves user information for all of the programs that require it. If your existing computer network uses an LDAP V3 compliant directory server, PeopleSoft supports the use of that server for managing user profiles and authenticating users. PeopleSoft enables you to integrate your authentication scheme for PeopleSoft with your existing infrastructure.

You always maintain permission lists and roles using PeopleSoft security. However, you can maintain user profiles in PeopleSoft security or reuse user profiles and roles that are already defined within an LDAP directory server. A directory server enables you to maintain a single, centralized user profile that you can use across all of your PeopleSoft and non-PeopleSoft applications. This approach reduces redundant maintenance of user information stored separately throughout your enterprise, and reduces the possibility of user information getting out of synchronization.

You can configure and extend your Signon PeopleCode to work with any schema implemented in your directory server. You can assign roles to users manually or assign them dynamically. When assigning roles dynamically, you use PeopleCode, LDAP, and PeopleSoft Query rules to assign user profiles to roles programmatically.

See Understanding the PeopleSoft LDAP Solution.

Authentication and Single Signon

PeopleSoft delivers the most common authentication solutions and packages them with your PeopleSoft application. This saves you the trouble of developing your own solutions and saves you time with your security implementation. These prepackaged solutions include PeopleCode that supports basic sign in through HTTP over SSL/TLS (HTTPS), LDAP authentication, and single signon.

Because PeopleSoft applications are designed for internet deployment, many sites must take advantage of the authentication services that exist at the web server level. PeopleSoft takes advantage of HTTPS, SSL/TLS, and digital certificates to secure the transmission of data from the web server to an end user's web browser and also to secure the transmission of data between PeopleSoft servers and third-party servers (for business-to-business processing) over the internet.

PeopleSoft applications support these types of single signon:

- Among PeopleSoft applications.

  A user can signon and be authenticated by one PeopleSoft application server and then, that user can access other PeopleSoft application servers without entering an ID or a password. Although the user is actually accessing different applications and databases, the user navigates seamlessly through the system. Recall that each suite of PeopleSoft applications, such as HCM or CRM, resides in its own database.

- Between PeopleSoft and Oracle applications.

  A user can sign in to either system and freely access the other without having to sign in to the second system.

- Between the desktop and PeopleSoft applications.

  A user can sign in to their computer network and be authenticated by their network credentials and then, that user can freely access all PeopleSoft applications. This is desktop single signon.

See Understanding the Delivered External Authentication Solutions, Understanding Single Signon.

Data Encryption

Data security comprises the following elements:

- Privacy—keeping data hidden from unauthorized parties.

  Privacy is normally implemented with some type of *encryption*. Encryption is the scrambling of information such that no one can read it unless they have a piece of data known as a key.

- Integrity—keeping transmitted data intact.

  Integrity can be accomplished with simple checksums or, better, with more complex cryptographic checksums known as *one-way hashes*, and often with *digital signatures* as well.

- Authentication—verifying the identity of an entity that's transferring data.

  Authentication can be accomplished using passwords, or with digital signatures, which are by far the most popular and most reliable method of authentication.

PeopleSoft Encryption Technology (PET) provides a way for you to use hashes and digital signatures to secure critical PeopleSoft data and communicate securely with other businesses. It enables you to extend and improve cryptographic support for your data in PeopleTools, giving you strong cryptography with the flexibility to change and grow, by incrementally acquiring stronger and more diverse algorithms for encrypting data. PeopleSoft delivers PET with support for the *OpenSSL* and *PGP* encryption libraries.

To implement PET:

1. Load the algorithms of an encryption library into the PET database.
2. Generate accompanying encryption keys, and insert them into the PET keystore.
3. Define a sequence, or *chain*, of algorithms by selecting from all the algorithms in the database.

4. Define an encryption profile, which is an instance of an algorithm chain applicable to a specific encryption task.

5. Write PeopleCode to invoke the encryption profile.

Note: Along with the delivered OpenSSL and PGP encryption libraries, a PeopleSoft database may also contain encryption keys for internal use of the PeopleCode Crypt class. These encryption keys do not need to be modified.

See Understanding PeopleSoft Encryption Technology.

Query and Definition Security

You use PeopleSoft Query to build SQL queries and retrieve information from application tables. For each PeopleSoft Query user, you can specify the records the user is allowed to access when building and running queries. You do this by creating query access groups in PeopleSoft Tree Manager, and then assigning users to those groups with PeopleSoft Query security. PeopleSoft Query security is enforced only when using PeopleSoft Query; it doesn't control runtime page access to table data.

Use Definition Security to govern access to PeopleSoft Application Designer definitions, such as record definitions, field definitions, and page definitions, and to protect particular definitions from being modified by developers.

PeopleSoft Personalizations

PeopleSoft offers a variety of options that enable end users, especially power users, to configure certain aspects of their PeopleSoft environment to produce a more personalized interface. These options improve a user's navigation speed through the system and enable users to select international preferences, such as date and time formats.

You define, group, and categorize personalization options, then use permission lists to control access to them. Users with access to a personalization option can control it through the My Personalizations menu.

See Understanding System Personalizations

Security Administration Integration Points

This section identifies the security integration points using:

- Component interfaces.
- Service operations.
- Application Engine programs.

Component Interfaces

This section describes component interfaces that are delivered with PeopleSoft applications that you can use to manage and administer user profiles and roles.

DELETE_ROLE

The DELETE_ROLE component interface is based on the Delete Role (PURGE_ROLEDEFN) component, and it is used to purge roles. It is keyed by RoleName and has the Get, Find, Save, and Cancel methods. The DELETE_ROLE service operation calls this component interface.

DELETE_USER_PROFILE

The DELETE_USER_PROFILE component interface is based on the Purge Inactive User Profile (PURGE_USR_PROFILE) component, and it is used to remove unused User Profiles. It is keyed by User ID and has the Get, Find, Save, and Cancel methods. The DELETE_USER_PROFILE service operation and the PURGEOLDUSRS Application Engine program call this component interface.

ROLE_MAINT

The ROLE_MAINT component interface is based on the Roles (ROLEMAINT) component. It is keyed by RoleName and has the Cancel, Create, Find, Get, and Save methods.

USERMAINT_SELF

This component interface is based on the My System Profile (USERMAINT_SELF) component. It allows only the current user to access it.

The USERMAINT_SELF component interface is used with the following components:

- Forgot My Password (EMAIL_PSWD)
- Change Password (CHANGE_PASSWORD)
- Change Expired Password (EXPIRE_CHANGE_PSWD)

USER_PROFILE

The USER_PROFILE component interface is based on the User Profiles (USERMAINT) component. It is keyed by User ID.

The USER_PROFILE component interface is used in User Profile Save As (USER_SAVEAS) and with LDAP authentication.

USER_PROFILE_SYNC

The USER_PROFILE_SYNC component interface is based on the User Profiles (USERMAINT) component. It is keyed by User ID and has the Cancel, Get, and Save methods.

The USER_PROFILE_SYNC component interface is used in User Profile Save As (USER_SAVEAS) and with LDAP authentication.

Related Links

“Understanding Component Interfaces” (Component Interfaces)

Service Operations

This section describes service operations that are delivered with PeopleSoft applications that you can use to manage and administer user profiles and roles.

Keep the following in mind when dealing with these security service operations, except the USER_PROFILE_XFR service operation:

- Each service operation has a same-named service definition.
- The service operations are asynchronous one-way.
- A same-named message is defined in each service operation definition.
- At least one handler is defined within each service operation definition, if the node is supposed to consume an inbound service operation.

DELETE_ROLE

This service operation is called from the Delete Role component. It is used to delete a role from subscribing databases. The service operation requires that the DELETE_ROLE component interface be authorized.

DELETE_USER_PROFILE

This service operation is called from the Delete User Profile component. It is used to delete a user profile from subscribing databases. This service operation requires that the DELETE_USER_PROFILE component interface be authorized.

ROLESYNCHEXT_MSG

This service operation is published when a Dynamic Role rule is run. It is called after the DYNROL_PUBL application engine program successfully finishes.

Note: As of release 8.49, the ROLESYNCH_MSG service operation is desupported and replaced with ROLESYNCHEXT_MSG service operation.

ROLE_MAINT

This service operation publishes new roles and updates existing roles in the Roles component.

USER_PROFILE

This service operation publishes user profile messages when adds, updates, and deletes occur through the following:

- User Profiles component (USERMAINT)
- User Profile Save As component
- My System Profile component (USERMAINT_SELF)
- Distributed User Profile component (USERMAINT_DIST)

- USER_PROFILE component interface
- USERMAINT_SELF component interface

User Profile messages may also be published when Password is changed through the Change My Password component (CHANGE_PASSWORD) or Expired Password component (EXPIRE_CHANGE_PSWD) by triggering the USERMAINT_SELF component interface.

USER_PROFILE_XFR

This service operation changes the shape of the inbound USER_PROFILE.VERSION_84 message to an internal shape that you configure based on your needs for partial user profile synchronization.

Related Links

"Configuring Service Operation Definitions" (Integration Broker)

Application Engine Programs

This section describes the Application Engine programs that are designed for use in your security implementation.

DYNROLE_PUBL

The DYNROLE_PUBL Application Engine program is called when Dynamic Role Rules are processed for a single role from the Role component.

You run this program from the Roles page in the Roles component. You can also schedule this program to run as needed through Process Scheduler.

DYNROLE_SYNC

The DYNROLE_SYNC Application Engine program is designed to run in synchronous mode and is primarily used for the Role Maintenance Component Interface.

LDAPSCHEMA

Application Engine Program that puts the LDAP Schema definition into the PeopleSoft database.

You run this program by selecting **PeopleTools** > **Security** > **Directory** > **Cache LDAP Directory Schema**.

LDAPMAP

Application Engine program used to import and export data to and from the LDAP directory into or from a PeopleSoft table. The process is based on an LDAP map.

You run this program by selecting **PeopleTools** > **Security** > **Directory** > **LDAP Authentication Maps**.

PTAUTOLOCKUP

The PTAUTOLOCKUP program selects User Profiles where the **Lock as of** date is past and locks the account by updating the database column for the **Account Locked Out?** field on the User Profiles, General page. The program also publishes the **Account Locked Out?** update to all subscriber nodes.

See Setting General User Profile Attributes.

For example, use the PTAUTOLOCKUP program for an Integration Broker node that is on a PeopleSoft PeopleTools release prior to the introduction of the **Lock as of** date field. In this case, the account will not be locked, even after the **Lock as of** date has been reached, unless you act to update the database column for the **Account Locked Out?** field.

You are responsible for scheduling the process as needed.

See “Scheduling Process Requests” (Process Scheduler) to schedule the program through Process Scheduler.

See “Starting Programs with the Application Engine Process Request Page” (Application Engine) to submit an Application Engine request.

PURGEOLDUSRS

The PURGEOLDUSRS Application Engine program deletes users who have not signed on within a period specified in Password Controls.

You run this program by selecting **PeopleTools** > **Security** > **User Profiles** > **Purge Inactive User Profiles** or by selecting **PeopleTools** > **Security** > **Password Configuration** > **Set Password Controls**, and then clicking the **Schedule** button under Purge Inactive User Profiles. You can also schedule this program to run as needed through Process Scheduler.

USER_SYNC

The USER_SYNC Application Engine program synchronizes user profiles between databases using the USER_PROFILE message. You set up this program on the database that you configured to *send* or publish user profile information. Once you have set up the program, click Run.

To set up this program, create a new request and enter the following information on the Application Engine Request page:

- Program Name: *USER_SYNC*.
- State Record: *AE_USRSYNC_AET*

USR_PRFL_XFR

Sample Application Engine program used to transform outbound USER_PROFILE messages to conform to shapes acceptable to the subscribing nodes. This program transforms USER_PROFILE.VERSION_84 into message shape USER_PROFILE.VERSION_81X.

Related Links

“Application Engine Overview” (Application Engine)

Security Administration Implementation

This section discusses:

- Preparing to use PeopleSoft security.
- Administering security from applications.
- Reviewing and monitoring your security implementation.

Preparing to Use PeopleSoft Security

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

To start administering security, install your PeopleSoft application according to the product documentation.

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

Other Sources of Information

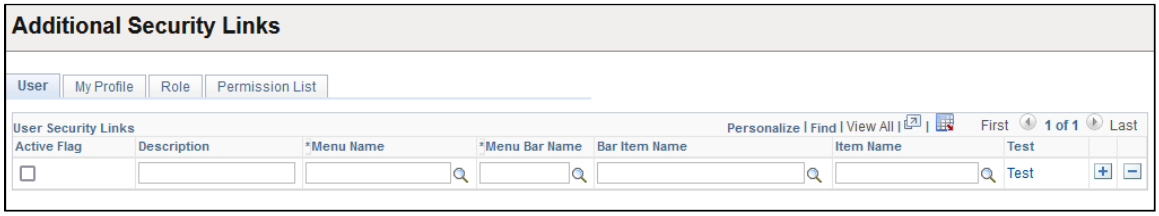
This section provides information to consider before you begin to manage your data. In addition to implementation considerations presented in this section, take advantage of all PeopleSoft sources of information, including the installation guides, release notes, and product documentation.

Administering Security from Applications

If you administer security information outside of the PeopleSoft security interface, for example, using application-specific pages to define application security, then you have the option of modifying the PeopleSoft security pages to include links to those application-specific pages. These links provide administrators a convenient way to access application-specific security pages without having to spend time navigating to them.

You add the extra security links from a browser by selecting **PeopleTools** > **Security** > **Security Objects** > **Additional Security Links**. You can add links to the User Profiles component, My System Profile page, the Role component, or the Permission List (ACCESS_CNTRL_LISTX) component. To add links to a security profile, select the appropriate page in the Security Links (SEC_OTHER_SETTINGS) component and add the link information for the target page. After you save the link information, the link appears on the Links page for the appropriate security profile.

This example illustrates the fields and controls on the Additional Security Links - User page.



Field or Control	*Description*
Active Flag	Enables you to activate and deactivate links. Only those links with the **Active Flag** selected appear for system users.
Description	Add a description of the page that contains the extra security information. This description is the text that appears on the Links page for the security profile.
Menu Name	From the drop-down list, add the menu name. This value is the application in which the page resides, such as Administer HR Security.
Menu Bar Name	From the drop-down list, add the menu bar name, such as Use, Setup, Process, and so on.
Bar Item Name	From the drop-down list, add the bar item name. For example, the bar item name for this page is Security Links.
Item Name	From the drop-down list, add the item name. For example, the item names for this component are User, Role, My Profile, and Permission List.
Test	After you have added all the appropriate information, use this link to test the security link. If it does not work correctly, double-check your selections for the previous options.

To add a Security Link:

1. Select **PeopleTools** > **Security** > **Security Objects** > **Additional Security Links**.
2. Select the security profile type (user, role, or permission list) to which you want to add extra links.
3. If links exist, click the plus sign button to add a new row.
4. Add the appropriate link information (Menu Name, Menu Bar name, and so on).
5. After you enter the appropriate link information, click **Test** to make sure the link points to the correct target.
6. Save your work.

Note: If you need to migrate the security links setup data from one database to another, you can use the following Data Mover scripts: SECOTHER_EXPORT.DMS and SECOTHER_IMPORT.DMS. These scripts reside in the *PS_HOME*\scripts directory.

Reviewing and Monitoring Your Security Implementation

PeopleSoft provides a collection of predefined queries that enable you to review, monitor, and audit system access by user, role, and permission list so that you can detect discrepancies. Use the Review

Security Information page to access these queries (**PeopleTools** > **Security** > **Review Security Information**). The page provides access to these queries:

- User ID queries.
- Role queries.
- Permission list queries.
- PeopleTools objects queries.
- Definition Security queries.
- Access log queries.

To run a query, click the link, enter the appropriate criteria (such as **User ID**), and click **View Results**. Query results typically appear in a new browser window.

Related Links

Running Role and Permission List Alias Queries
Viewing Definition Access by User and Permission List (Windows Client)
Tracking User Sign In and Sign Out Activity

Chapter 2

Understanding PeopleSoft Security

Secure by Default

The PeopleSoft system is delivered “secure by default,” with all default passwords removed from the installation to the extent possible.

The administrator establishes passwords for key system components during the installation, set up and configuration of the database and PeopleSoft system.

The administrator must establish passwords for the following components during system installation, set up and configuration:

System Component	***Description***	***When/Where to Set Password***
Connect ID	Establishes connection to the database.	DPK setup script or database installation and configuration.
Access ID	Access ID to the database.	DPK setup script or database installation and configuration.
User profile (Database)	Defines the access level in the database.	DPK setup script or database installation and configuration.
PTWEBSERVER	Used for initial signon to the PeopleSoft Pure Internet Architecture (PIA) after the web server is started.	DPK setup script or database installation and configuration.
Domain connection	Used for JOLT connections between the web server and application server.	DPK setup script or PIA installation.
Web server	Used for communication with the web server.	DPK setup script or PIA installation.
Integration gateway	Used to access the integration gateway properties file.	DPK setup script or PIA installation.

For those cases where a password cannot be removed from the installation, a password change is required at the time of initial use.

An example of a default password that cannot be removed from the installation is the Java keystore password. Users are prompted to change the password during initial access to the Java keystore and cannot proceed until the default password is changed.

Security Basics

Security is especially critical for core business applications, such as PeopleSoft applications. Typically, you do not want every department in your company to have access to all your applications. Nor do you want everyone within a department to have access to all the functions or all the data of a particular application. Additionally, you may want to restrict who can customize your applications with PeopleTools.

PeopleSoft software provides security features, including components and PeopleTools applications, to ensure that your sensitive application data, such as employee salaries, performance reviews, or home addresses, does not fall into the wrong hands. Most likely, you use other security tools for your network and relational database management system (RDBMS). These tools work together to protect the PeopleSoft system from unauthorized access.

As you implement the PeopleSoft Internet Architecture, you need a robust and scalable means by which you can grant authorization to users efficiently. When you deploy your applications to the internet, the number of potential users of your system increases exponentially. Suddenly, you have customers, vendors, suppliers, employees, and prospects all using the same system.

The PeopleSoft security approach is tailored for the internet. It enables you to easily create and maintain security definitions, and you can perform many maintenance tasks programmatically.

You can apply security to all users, including employees, managers, customers, contractors, and suppliers. You group your users according to roles to give them different degrees of access. For instance, there might be an Employee role, a Manager role, and an Administrator role. Users who belong to a particular role require a specific set of permissions, or authorizations, within your system, so that they can complete their daily tasks.

You must also secure the objects and definitions in your PeopleSoft development environment. Just as you restrict sets of end users from accessing particular pages and components, you also restrict the definitions that your site's developers can access using PeopleSoft Application Designer. A *definition* refers to any of the objects that you create within PeopleSoft Application Designer, such as records, pages, or components. Each object definition may have individual security needs. For example, you may have a large development staff, but perhaps you want only a few developers to have access to specific record definitions.

PeopleSoft Security Definitions

Because deploying your applications to the internet significantly increases the number of potential users your system must accommodate, you need an efficient method of granting authorization to different user types. PeopleSoft security definitions provide a modular means to apply security attributes in a scalable manner.

A security definition refers to a collection of related security attributes that you create using PeopleTools Security. The three main PeopleSoft security definition types are:

- User profiles.
- Roles.
- Permission lists.

Note: A PeopleSoft security definition called an Access Profile also exists, but these are defined at the database level.

User Profiles

User profiles define individual PeopleSoft users.

Each user has an individual user profile, which in turn is linked to one or more roles. You add one or more permission lists, which ultimately control what a user can and cannot access, to each role. A few permission types are assigned directly to the user profile.

Typically, a user profile must be linked to at least one role in order to be a valid profile. The majority of values that make up a user profile are inherited from the linked roles.

Roles

Roles are intermediate objects that link user profiles to permission lists. You can assign multiple roles to a user profile, and you can assign multiple permission lists to a role. Some examples of roles might be Employee, Manager, Customer, Vendor, and Student.

A manager is also an employee and may also be a student. Roles enable you to mix and match access appropriately.

You have two options when assigning roles: assign roles manually or assign them dynamically. When assigning roles dynamically, you use PeopleCode, LDAP, and PeopleSoft Query rules to assign user profiles to roles programmatically.

Permission Lists

Permission lists are groups of authorizations that you assign to roles. Permission lists store sign-in times, page access, PeopleTools access, and so on.

A permission list may contain one or more types of permissions. The fewer types of permissions in a permission list, the more modular and scalable your implementation.

A user profile inherits most of its permissions through roles, but you apply some permission lists, such as process profile or row-level security (data permissions), directly to a user profile.

PeopleSoft Online Security

The PeopleSoft system has many elements, such as batch processes, object definitions, and application data. Use PeopleTools security tools to control access to most of these elements. To secure other elements, you use application-specific interfaces, such as Administer Security.

This section discusses:

- Sign in and time-out security.
- Page and dialog box security.
- Batch environment security.

- Definition security.
- Application data security.
- PeopleSoft Internet Architecture security.
- Data Privacy Masking Framework.

Sign in and Time-out Security

When a user attempts to sign in to PeopleSoft, he or she enters a user ID and a password on the PeopleSoft Signon page. If the ID and password are valid, PeopleSoft connects the user to the application, and the system retrieves the appropriate user profile.

If the user attempts to sign in during an invalid sign in time as defined in the user's security profile, he or she is not allowed to sign in. A sign in time is an adjustable interval during which a user is allowed to sign in to PeopleSoft. For example, if a given sign in time is Monday through Friday from 7 a.m. to 6 p.m. for a set of users, those users cannot access a PeopleSoft application on Saturday or on Friday at 6:05 p.m. If a user is signed in when the sign in period expires, PeopleSoft signs the user out automatically.

After signing in, a user can stay connected as long as the sign in time allows and as long as the browser does not sit idle for longer than the time-out interval. A time-out interval specifies how long the user’s machine can remain idle—no keystrokes, no SQL—before the PeopleSoft system automatically signs the user out of the application.

You specify both the sign in times and time-out interval using PeopleTools Security.

Note: Other time-out intervals, unrelated to security, are controlled by your web server and by PeopleSoft Pure Internet Architecture components.

Page and Dialog Box Security

You can restrict access to PeopleSoft menus. You can set the access rights to the entire menu, such as Administer Workforce or PeopleTools Security, or just a specific item on that menu. Because the only normal way to access a PeopleSoft page is through a menu, if a user has no access to a particular menu or menu item, then you have effectively restricted that user's access to the corresponding page.

You can also restrict access to specific actions or commands on a page. For example, you may want a clerk in your sales office to be able to access contract data but not be able to update the data. In this case, you grant access to the set of pages, but you allow display-only access only. In this case, the clerk cannot update or correct any data. This approach enables users to get their work done while maintaining the security and integrity of your business data.

Batch Environment Security

If a particular user must run batch processes using PeopleSoft Process Scheduler, assign the appropriate process profile to the user profile and create process groups for your processes. A user receives both process group and process profile authorizations through permission lists. A user gets permission to process groups through roles, and they get a process profile through the process profile permission list.

Note: You add the process profile permission list directly to the user profile, not to an intermediary role.

Process Security

Because PeopleSoft applications take advantage of other applications, such as SQR and COBOL, your batch processes should be run in a secure environment.

The three levels of security for batch programs are:

- Each batch program has a run control that you define before you can run the batch program.

  Run controls are set up using PeopleSoft Process Scheduler.

- PeopleSoft Process Scheduler enables you to set up process groups, which are groups of batch processes.

  In PeopleTools Security, you add process groups to a security profile. Users can run processes that belong to the process groups assigned to their security profile.

- In your RDBMS environment, you can restrict offline access to batch processes using the security tools described in your platform manuals.

Reporting Security

PeopleSoft Report Manager uses a logical space on a web server called the Report Repository. PeopleSoft Report Manager enables you to generate and distribute reports over the internet, and it stores the output in the Report Repository. Wherever you decide to situate your repository, make sure that the server is protected from outside access. Ensure that only the PeopleSoft system can access and distribute the generated reports. The Report Repository servlet gets items from the web server and puts them in the browser. With report distribution, you distribute reports and view them according to your role.

PeopleSoft delivers these roles for the specific use in reporting:

- ReportDistAdmin
- ReportSuperUser

Definition Security

Use Definition Security to govern access to database object definitions, such as record definitions, field definitions, and page definitions, and to protect particular object definitions from being modified by certain developers.

Application Data Security

Definition security is a form of data security—you use it to control access to particular rows of data (object definitions) in PeopleTools tables. PeopleSoft software also provides other methods to control the application data that a user is allowed to access in the PeopleSoft system. This task is also known as setting data permissions.

With application data security, you can set data permissions at the following levels:

- Table level (for queries only).
- Row level.

- Field level.

Table-Level Security

You use PeopleSoft Query to build SQL queries and retrieve information from application tables. For each PeopleSoft Query user, you can specify the records the user is allowed to access when building and running queries. You do this by creating query access groups in PeopleSoft Tree Manager and then assigning users to those groups with PeopleSoft Query security. PeopleSoft Query security is enforced only when using PeopleSoft Query; it does not control runtime page access to table data.

Row-Level Security

You can design special types of SQL views—security views—to control access to individual rows of data stored within application database tables. Row-level security enables you to specify the data that a particular user is permitted to access. PeopleSoft applications are delivered with built-in row-level security functions that are tailored to specific applications.

For example, PeopleSoft Human Resources security tables enable you to restrict user access to employee rows of data according to organizational roles. You could also permit users to view and update rows for employees in their departments only. Similarly, in PeopleSoft Financials, you can use security views to determine access to business units and ledgers. You can also use security tables to grant privileges by access group to users who use PeopleSoft Query to access data from the database.

See the documentation for your application for details about implementing row-level security for your applications.

Field Security

Use PeopleCode to restrict access to particular fields or columns within application tables. For example, if you want a certain class of user to be able to access certain pages but not to view a particular field on those pages, such as compensation rate, you can write PeopleCode to hide the field for that user class.

PeopleSoft Internet Architecture Security

PeopleSoft Internet Architecture security is also known as runtime security. Only authorized users can connect to the web and application server, and only authorized application servers can connect to a given database.

PeopleSoft applications use authentication tokens embedded in browser cookies to authorize users and enable single signon throughout the system. To secure links between elements of the system, including browsers, web servers, application servers, and database servers, PeopleSoft applications incorporate a combination of SSL/TLS security and Oracle Tuxedo and Oracle Jolt encryption.

SSL is a protocol developed by Netscape that defines an interface for data encryption between network nodes. TLS, a protocol developed by the Internet Engineering Task Force (IETF), evolved from and is based on SSL.

To establish an SSL/TLS-encrypted connection, the nodes must complete the SSL/TLS handshake. The simplified steps of the SSL/TLS handshake are as follows:

1. Client sends a request to connect.

2. Server responds to the connect request and sends a signed certificate.

3. Client verifies that the certificate signer is in its acceptable certificate authority list.
4. Client generates a session key to be used for encryption and sends it to the server encrypted with the server's public key (from the certificate received in step 2).
5. Server uses a private key to decrypt the client generated session key.

Establishing an SSL/TLS connection requires two certificates: one containing the public key of the server (server certificate or public key certificate) and another to verify the certification authority that issued the server certificate (trusted root certificate). The server needs to be configured to issue the server certificate when a client requests an SSL/TLS connection, and the client needs to be configured with the trusted root certificate of the certificate authority that issued the server certificate.

The nature of those configurations depends on both the protocol being used and the client and server platforms. In most cases you replace HTTP with LDAP. SSL/TLS is a lower level protocol than the application protocol, such as HTTP or LDAP. SSL/TLS works the same regardless of the application protocol.

Note: Establishing SSL/TLS connections with LDAP is not related to web server certificates or certificates used with PeopleSoft integration.

The system uses SSL/TLS encryption in the following locations:

- Between the browser and the web server.
- Between the application server and the integration gateway.
- Between the integration gateway and an external system.

The system uses Oracle Tuxedo and Oracle Jolt encryption in these locations:

- Between the web server and the application server.
- Between the integration gateway and a PeopleSoft system (Oracle Jolt only).

Security between the application server and the database is supplied by RDBMS connectivity.

PeopleSoft Integration Broker and portal products have additional security concerns, which are addressed in the documentation for those products.

Data Privacy Framework

The Data Privacy Framework, delivered by PeopleSoft applications, provides the ability to identify and mask personally identifiable information, or PII, and sensitive data, and is implemented through data masking and filtering.

Data Masking

Data masking allows masking of all or some of the data displayed in certain PeopleSoft Pure Internet Architecture page controls. The SetDisplayMask method will replace each character of the displayed field text value with the chosen mask character.

Several PeopleCode methods related to data masking are available. They are described in the PeopleCode API documentation.

Note: SetDisplayMask works for all page field types with this limitation. You can use SetDisplayMask for long edit text boxes if the field assigned to the long edit box is not a Long Character Data Type.

See “Using Edit Boxes and Long Edit Boxes” (Application Designer Developer’s Guide)

See “SetDisplayMask” (PeopleCode API Reference).

See “UnSetDisplayMask” (PeopleCode API Reference).

See “CopyDisplayMask” (PeopleCode API Reference).

PeopleCode Built-in functions are provided to help decide which fields should be masked.

See “IsRecFieldPII” (PeopleCode Language Reference).

See “IsRecFieldSensitive” (PeopleCode Language Reference).

Note: The PII and sensitive designations for a record field are made through the Data Privacy Framework of the PeopleSoft application that delivers the record field. Therefore, PeopleTools does not provide a way for users to directly modify these designations.

Masking can be applied differently for users by implementing several PeopleCode built-in functions.

See “IsUserInRole” (PeopleCode Language Reference).

See “IsUserInPermissionList” (PeopleCode Language Reference).

See “IsPIIandSensitiveForUser” (PeopleCode Language Reference).

Filtering Drop Downs Based on User Attributes

PeopleSoft provides the ability to specify an override prompt table for any record field that already has a prompt table defined.

At runtime, when the Prompt button is clicked, if there is an override edit table assigned to the Record Field being clicked, that override table will be used to determine the result set instead of the defined prompt table.

The following PeopleCode field object properties will be affected:

- PromptTableName

  If the defined Prompt Table has been overridden then this property will return the Override Prompt Table Name instead of the Prompt Table defined on the Record Field in Application Designer.

- SQLText

  If the defined Prompt Table has been overridden then this Property will return the SQL defined in the Override Prompt Table. When this Property is used to replace the SQL text being used at Runtime it will still work as designed no matter whether there is a configured override or not.

View filtering based on Runtime OPRID is already available via the %OprClause Meta-SQL; however, use of this Meta-SQL is restricted to Dynamic Views. All Prompt Override tables must be Dynamic views and must use %OprClause.

Note: PeopleSoft allows customers to create the Override Prompt Record/view definitions and define the appropriate view SQL by incorporating %OprClause as appropriate. The Override Prompt Table settings and the data will not be comparable or copyable. The customers can propagate their overrides to other database environments.

See “%OPRCLAUSE” (PeopleCode Language Reference).

Related Links

“Using Query Administration” (Query)
“Using RTF Templates” (BI Publisher for PeopleSoft)
“Masking of Data in Search Results” (Search Technology)
“Data Masking in Pivot Grid” (Pivot Grid)
“Masking Data in Simplified Analytics” (Pivot Grid)

PeopleSoft Authorization IDs

The PeopleSoft system uses various authorization IDs and passwords to control user access. You use PeopleTools Security to assign two of these IDs: the user ID and the symbolic ID.

This section discusses:

- User IDs.
- Connect ID.
- Access IDs.
- Symbolic IDs.
- Administrator access.

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

Related Links

PeopleSoft Sign In

User IDs

A PeopleSoft user ID is the ID you enter at the PeopleSoft sign in page. You assign each PeopleSoft user a user ID and password. The combination of these two items grants users online access to the PeopleSoft application. The system can also use a user ID stored within an LDAP directory server.

The user ID is the key that the application uses to identify the user profile definition.

Connect ID

The connect ID performs the initial connection to the database.

Note: PeopleSoft no longer creates users at the database level.

A connect ID is a valid user ID that, when used during sign in, takes the place of PeopleSoft user IDs. Using a connect ID means you do not have to create a new database user for every PeopleSoft user that you add to the system.

Note: A connect ID is required for a direct connection (two-tier connection) to the database. Application servers and two-tier Microsoft Windows clients require a connect ID. You specify the connect ID for an application server in the Signon section of the PSADMIN utility. For Microsoft Windows clients, you specify the connect ID on the Startup tab of PeopleSoft Configuration Manager. You can create a connect ID by running the ConnectSQL and GrantSQL scripts.

Note: When performing a database compare or copy, both databases must have the same connect ID.

Warning! Without a connect ID specified, the system assumes the workstation is accessing PeopleSoft through an application server. The option to override the database type is disabled.

Access IDs

When you create any user ID, you must assign it an access profile, which specifies an access ID and password.

The PeopleSoft access ID is the RDBMS ID with which PeopleSoft applications are ultimately connected to your database after the PeopleSoft system connects using the connect ID and validates the user ID and password. An access ID typically has all the RDBMS privileges necessary to access and manipulate data for an entire PeopleSoft application. The access ID should have Select, Update, and Delete access.

Users do not know their corresponding access IDs. They just sign in with their user IDs and passwords. Behind the scenes, the system signs them into the database using the access ID.

If users try to access the database directly with a query tool using their user or connect IDs, they have limited access. User and connect IDs only have access to the few PeopleSoft tables used during sign in, and that access is Select-level only. Furthermore, PeopleSoft encrypts the sensitive data that resides in those tables.

Note: Access profiles are used when an application server connects to the database, when a Microsoft Windows workstation connects directly to the database, and when a batch job connects directly to the database. Access profiles are not used when end users access applications through PeopleSoft Pure Internet Architecture. During a PeopleSoft Pure Internet Architecture transaction, the application server maintains a persistent connection to the database, and the end users leverage the access ID that the application server domain used to sign in to the database.

Note: PeopleSoft suggests that you only use one access ID for your system. Some RDBMS do not permit more than one database table owner. If you create more than one access ID, it may require further steps to ensure that this ID has the correct rights to all PeopleSoft system tables.

Symbolic IDs

PeopleSoft encrypts the access ID when it is stored in the PeopleTools security tables. Consequently, an encrypted value cannot be readily referenced or accessed. So when the access ID, which is stored in

PSACCESSPROFILE, must be retrieved or referenced, the query selects the appropriate access ID by using the symbolic ID as a search key.

The symbolic ID acts as an intermediary entity between the user ID and the access ID. All the user IDs are associated with a symbolic ID, which in turn is associated with an access ID. If you change the access ID, you need to update only the reference of the access ID to the symbolic ID in the PSACCESSPROFILE table. You do not need to update every user profile in the PSOPRDEFN table.

Administrator Access

As an administrator, you must customize your own user definition. PeopleSoft delivers at least one full-access user ID with each delivered database. Your first task should be to sign in with this ID and personalize it for your needs or to create a new, full-access ID.

Note: PeopleSoft-delivered IDs are documented in your installation manual.

When you install PeopleSoft, you are prompted for an RDBMS system administrator ID and password. This information is used to automatically create a default access profile. If you will be using more than one access profile, set up the others before creating any new PeopleSoft security definitions. Most sites only use one access profile.

The number of database-level IDs you create is up to your site requirements. However, in most cases, having fewer database-level IDs reduces maintenance issues.

For example, if you implement pure LDAP authentication, at a minimum you need two database-level IDs—your access ID and your connect ID. With this scenario, in PeopleSoft you need to maintain only a symbolic ID to reference the access ID and maintain a user ID that the application server uses during sign in. With this minimal approach, each user who needs a two-tier connection, to run an upgrade, for example, could use the same user ID that the application server uses.

PeopleSoft Sign In

This section discusses:

- PeopleSoft sign in process.
- Directory server integration.
- Authentication and Signon PeopleCode.
- Single signon.

PeopleSoft Sign In Process

The most common direct sign in to the PeopleSoft database is the application server sign in.

These are the basic steps that are taken when the application server signs in to the database:

1. Initial connection.

The application server starts and uses the connect ID and user ID specified in its configuration file (PSAPPSRV.CFG) to perform the initial connection to the database.

2. The server performs a SQL Select statement on the PeopleTools security tables.

   After verifying the connect ID, the application server performs a Select statement on PeopleTools security tables, such as PSOPRDEFN, PSACCESSPROFILE, and PSSTATUS. Using these tables, the application server authenticates the user and gathers such items as the user ID and password, symbolic ID, access ID, and access password. After the application server has the required information, it disconnects.

3. The server reconnects using the access ID.

   When the system verifies that the access ID is valid, the application server begins the persistent connection to the database that all PeopleSoft Pure Internet Architecture and Microsoft Windows three-tier clients use to access the database. Typically, the users signing in using a Microsoft Windows workstation are developers using PeopleSoft Application Designer.

Note: A Microsoft Windows workstation attempting a two-tier connection uses the same process as the application server.

PeopleSoft recommends that all connectivity be made through either a three-tier Microsoft Windows client or through the browser. A two-tier connection is not necessary other than for the application server, PeopleSoft Process Scheduler, or for a user who will be running upgrades or PeopleSoft Data Mover scripts.

Signon PeopleCode does not run during a two-tier connection, so maintaining two-tier users in a directory server is not supported.

Directory Server Integration

PeopleSoft recognizes that your site uses software produced by numerous vendors, and each different product requires security authorizations for users. Most of these products adhere to the model that includes user profiles and roles (or groups) to which users belong. PeopleSoft enables you to integrate your authentication scheme for the PeopleSoft system with your existing infrastructure. You can reuse user profiles and roles that are already defined within an LDAP directory server.

Organizations typically store user profiles in a central repository that serves user information for all of the programs that require it. The central repository is typically an LDAP directory server.

A directory server enables you to maintain a single, centralized user profile that you can use across all of your PeopleSoft and non-PeopleSoft applications. This approach reduces redundant maintenance of user information stored separately throughout your enterprise, and it reduces the possibility of user information getting out of synchronization.

You always maintain permission lists and roles by using PeopleTools Security. However, you can maintain user profiles in PeopleTools Security or with an external directory server.

Authentication and Signon PeopleCode

You can store PeopleSoft passwords in the PSOPRDEFN PeopleTools table. You can also store and maintain user passwords and the rest of the user profile data in an LDAP directory server. PeopleSoft

applications retrieve the information stored in an external directory server using a combination of the User Profiles component interface and Signon PeopleCode.

If you decide to reuse existing user profiles stored in a directory server, you don't need to perform dual maintenance on the two copies of the user data—one copy in the LDAP server and one copy in PSOPRDEFN. PeopleSoft applications ensure that the user information stays synchronized. If you configure LDAP authentication, you maintain your user profiles in LDAP and not in PeopleTools Security.

Signon PeopleCode copies the most recent user profile data from a directory server to the local database whenever a user signs in. PeopleSoft applications reference the user information stored in the PeopleSoft database rather than making a call to the directory server each time the system requires user profile information. Signon PeopleCode ensures the local database has a copy of the most current user profile based on the information in the directory. Each time the user signs in, Signon PeopleCode checks to see if the row in the user profile cache needs to be updated.

The sign in process occurs as follows:

1. The user enters a user ID and password on the sign in page.
2. PeopleTools attempts to authenticate the user against the PSOPRDEFN table.
3. Signon PeopleCode runs.

   The default Signon PeopleCode program updates the user profile based on the current data stored in the directory server.

You can use Signon PeopleCode and business interlinks to synchronize the local copy of the user profile with any data source at sign in time; the program that ships with PeopleTools is designed to synchronize the user profile with an LDAP directory server only. Because the sign in program is PeopleCode, you can modify it, incorporating any of the PeopleSoft integration technologies that PeopleCode supports.

To edit the Signon PeopleCode program, you open the LDAP function library record and use the PeopleCode editor to customize the PeopleCode programs. Developers who modify the Signon PeopleCode program need to have a good understanding of PeopleCode and the integration features it offers.

Note: Only users who sign in through PeopleSoft Pure Internet Architecture or three-tier Microsoft Windows clients take advantage of Signon PeopleCode.

Single Signon

PeopleSoft Pure Internet Architecture uses browser cookies for seamless single signon across all PeopleSoft nodes. A node refers to a database and the application servers connected to it. For example, a user can complete a PeopleSoft Human Resources transaction, and then click a link for a PeopleSoft Financials transaction without reentering a password. Single signon is especially important to the PeopleSoft Interaction Hub, which aggregates content from several different applications and data sources into a single, integrated display.

Implementation Options

By using our integration technologies, you can configure PeopleSoft security to work with numerous schemes.

This section discusses:

- Authentication options.
- Role assignment options.
- Cross-system synchronization options.

Authentication Options

Consider how you plan to authorize users as they sign in to your PeopleSoft system. Do you want to store and maintain the PeopleSoft user passwords within a PeopleSoft database, or do you plan to take advantage of existing user profiles in an external directory server?

PeopleSoft-Based Authentication

This option is, generally, the way PeopleSoft customers have authorized users in previous releases. PeopleSoft user passwords are stored and maintained solely within PeopleSoft. Although this method does not require a large amount of storage, it does add administration issues, mainly because PeopleSoft passwords are yet another password users need to remember.

With this option there are only two database-level IDs, the access ID and the connect ID. The passwords reside in the PSOPRDEFN along with the other user information.

Directory-Based Authentication

You can also use a central repository for user information in a directory server that uses the LDAP protocol.

The advantage of this option is that a user has one user ID and password that allows access to numerous software systems.

Role Assignment Options

Consider how you plan to assign authorizations to your users. Recall that users inherit permissions through the roles to which they are assigned. When you plan your authorization assignment, you are really planning how you intend to assign roles to users. You can assign roles to users in two ways: the static approach and the dynamic approach.

Static

Using the static approach, you assign users to roles manually. Static role assignment is not scalable to the thousands of users that are likely to use your system when you deploy applications to the internet.

The static approach requires an administrator to maintain each user's set of roles. For that reason, Oracle recommends that you explore and implement the dynamic role assignment.

Dynamic

Using dynamic role assignment, the system assigns roles based on business rules. You can manually run the rule, but typically, you run the rules from a scheduled batch process.

Suppose an employee changes jobs and becomes a manager in a new department. When you run your dynamic rule, the system removes the roles associated with the employee's previous position and then adds the appropriate roles required for the new position. In addition, you can have the rule publish a message to other nodes, such as a PeopleSoft Financials node, which might subscribe to changes in the PeopleSoft Human Resources database.

You can use PeopleSoft Query, LDAP, or PeopleCode to define dynamic role assignment. If necessary, you can use a combined approach with the rules for assigning roles. For example, you can have one role rule based on LDAP, another based on a query, and so on. You can also have multiple rule types for one role. For example, a Manager role could be derived partially from an LDAP rule and partially from a PeopleSoft Query rule. As the following list describes, where the information that drives your role assignments is stored determines the types of role rules you use:

- If the membership data for your roles resides in your PeopleSoft database, use PeopleSoft Query to construct your role rules.

  One query could be MANAGER, another EMPLOYEE, and so on. When the rule runs, the system assigns your employee users to the EMPLOYEE role and the manager employees to the MANAGER role based on the results returned from the query.

- If you already have LDAP directory server groups organized by region, department, position, and so on, base your rules on the existing LDAP structure.

  Based on the directory setup and hierarchy, your rule assigns PeopleSoft users to the appropriate roles. Your PeopleSoft application uses your existing LDAP configuration. You should use this role rule type in conjunction with LDAP authentication.

- If you have user information in other third-party systems, such as legacy mainframe applications or UNIX account groups, use PeopleCode.

  You can take advantage of the multiplicity of integration technologies that PeopleCode supports, such as business interlinks and component interfaces. The business interlinks retrieve the data from the external system and write it to the role assignment tables in the PeopleSoft database.

Cross-System Synchronization Options

If you have multiple PeopleSoft applications, consider how to keep user information synchronized. Synchronization is especially important for the portal deployment, where users are likely to move from one system to another seamlessly. For instance, after completing a transaction in PeopleSoft Human Resources, a user may click a link that takes her directly to PeopleSoft Financials.

If you are using dynamic role assignment, the dynamic role batch program, by default, publishes a message that indicates a particular change. You need to make sure that nodes that require such information changes are configured to subscribe to the message that publishes the changed data. For example, suppose PeopleSoft Financials needs a list of managers for a particular transaction. Because the manager information resides in PeopleSoft Human Resources, PeopleSoft Human Resources publishes any changed information to PeopleSoft Financials to keep the data synchronized.

PeopleSoft security also publishes a message when a user profile changes (if the corresponding Service Operation version is active), which is most useful if you are not using LDAP to store user information. If you store user information in the PeopleSoft system, the message makes sure that password changes are replicated across multiple databases. If you store your user information in a central LDAP server, then the passwords, and so on, are already—in a sense—synchronized.

You can upgrade permission lists and roles using the PeopleSoft Application Designer upgrade features. For user information, PeopleSoft Data Mover scripts migrate user profiles between systems for upgrades or bulk loads.

Chapter 3

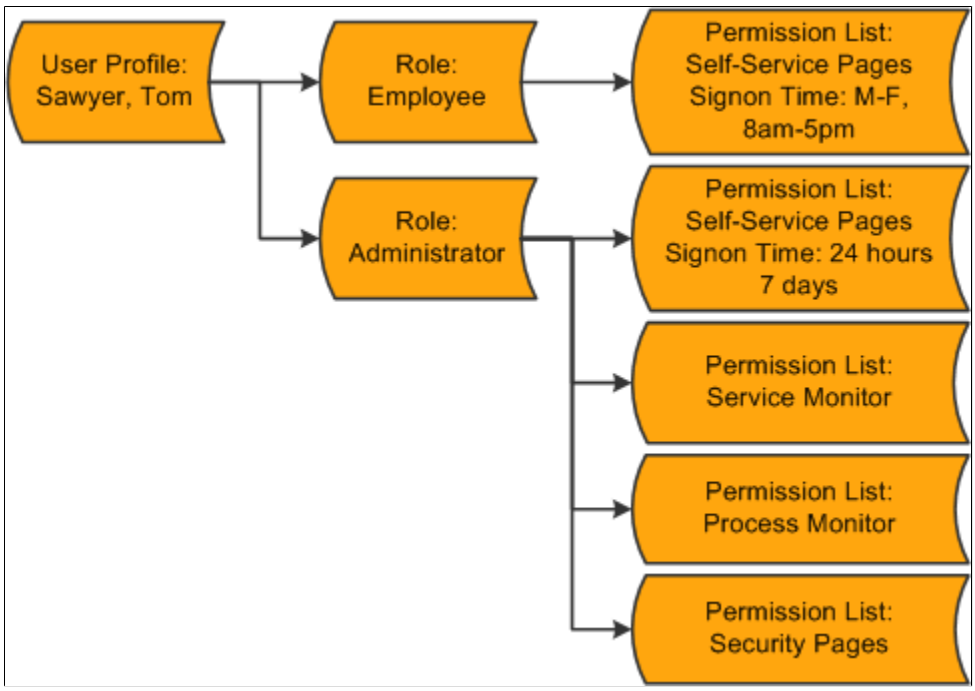
Setting Up Permission Lists

Understanding Permission Lists

Permission lists are the building blocks of user security authorizations. You typically create permission lists before you create user profiles and roles. When defining permission lists, however, consider the roles and user profiles with which you will use them. Recall that roles are intermediary objects between permission lists and users. You use roles to assign permissions to users dynamically.

Permission lists may contain any number of permissions, such as sign in times, page permissions, web services permissions, and so on. Permission lists are more flexible and scalable when they contain fewer permissions.

The following diagram illustrates how permission lists are assigned to roles, which are then assigned to user profiles. A role may contain numerous permissions, and a user profile may have numerous roles assigned to it. A user inherits all permissions assigned to each role to which the user belongs. User access is determined by the combination of all assigned roles.



The diagram represents the security authorizations of Tom Sawyer. Mr. Sawyer inherits the five permission lists that are assigned to the two roles that are assigned to his user profile. In this example, he has access to the employee self-service pages, the service monitor, PeopleSoft Process Monitor, and PeopleTools Security. If Tom were to become a manager, then the permission lists assigned to the Manager role would be added to his profile.

Theoretically, you could create a permission list tailored for every role, and that permission list could contain a permission for every category, from General to Web Libraries. However, permission lists like this do not scale to encompass roles that might be similar but not exactly alike. For a similar role, you would have to create a new role from the beginning. This kind of approach is not efficient for larger, more complicated implementations.

Alternatively, you can use a more modular, or mix-and-match, approach whereby you create numerous, generic permission lists that you can add to and remove from role definitions. Suppose you have three 8-hour shifts at your site. Using the modular approach, you could create three different versions of sign in permissions: one for 6 a.m. to 2 p.m., one for 2 p.m. to 10 p.m., and another for 10 p.m. to 6 a.m. Then, depending on the shift for a particular role, you can easily apply or remove the appropriate permission as needed without affecting any other permissions.

Although how you decide to implement Permission Lists depends on your site's security scheme and your security administrator, the modular approach provides increased scalability. As a general rule, your permission lists should be assigned to roles so that the common user has between 10 to 20 lists. This range represents the best balance of performance and flexibility. If you have too many permission lists, you may notice performance degradation, and if you have too few permission lists, you may sacrifice flexibility.

Managing Permission Lists

This section discusses how to:

- Create new permission lists.
- Copy permission lists.
- Delete permission lists.
- View related content references.

Creating New Permission Lists

To create a new permission list:

1. Select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists**.
2. On the search page, click **Add a New Value.**
3. In the **Permission List** edit box, enter the name of the permission list to create.

   Note: Permission list names have a 30-character limit. PeopleSoft HCM requires certain naming conventions for permission lists, but PeopleTools does not enforce these application-specific requirements. Therefore, when creating permission lists, keep in mind that PeopleSoft HCM requires primary permission lists to start with *PP* and data permission lists to start with *DP*.
4. From the pages in the Permission List component, select the appropriate permissions.
5. Save your permission list.

Copying Permission Lists

To copy a permission list:

1. Select **PeopleTools** > **Security** > **Permissions and Roles** > **Copy Permission Lists**.
2. On the search page, locate and select the permission list that you want to copy (clone).

   The Permission List Save As page appears.
3. On the Permission List Save As page, enter a new name in the **Save As:** edit box for the permission list that you want to copy.
4. Click Save.

Note: When copying a permission list, you also copy the access specified for content references by the original permission list. When deleting a permission list, you also remove access to the content references associated with that permission list.

Deleting Permission Lists

To delete a permission list:

1. Select **PeopleTools** > **Security** > **Permissions and Roles** > **Delete Permission Lists**.
2. On the search page, locate and select the permission list that you want to delete.

   The Delete Permission List page appears.
3. Click **Delete Permission List.**
4. Click **OK** to confirm the deletion, or click **Cancel** to end without deleting.

Note: This action deletes content reference permissions and all references to the permission list (even where referenced in application data).

Viewing Related Content References

This section discusses:

- Viewing content references.
- Synchronizing content references.

Viewing Content References

Select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** > **Pages** to access the Pages page, and then click the **Edit Components** link to access the Component Permissions page.

When you set component permissions and web library permissions, use the **View Content References** link to view the content references pointing to a given component or script. PeopleTools automatically propagates changes to permission lists to the content references.

When you click the link, the Content References page appears, showing the following:

- Name of the portal.
- Name of the content reference.
- The label.
- Whether or not it is accessible.
- The path.

Synchronizing Permission Lists and Content References

Use the PORTAL_CSS application engine program to synchronize permission lists with content references for the portal. By default, the system synchronizes changes in permission lists with content references; however, after an upgrade or any time when you want to make sure, you can run the PORTAL_CSS program. A process definition of the same name also exists.

Related Links

"Administering Content References" (Portal Technology)

Defining Permissions

This section discusses how to:

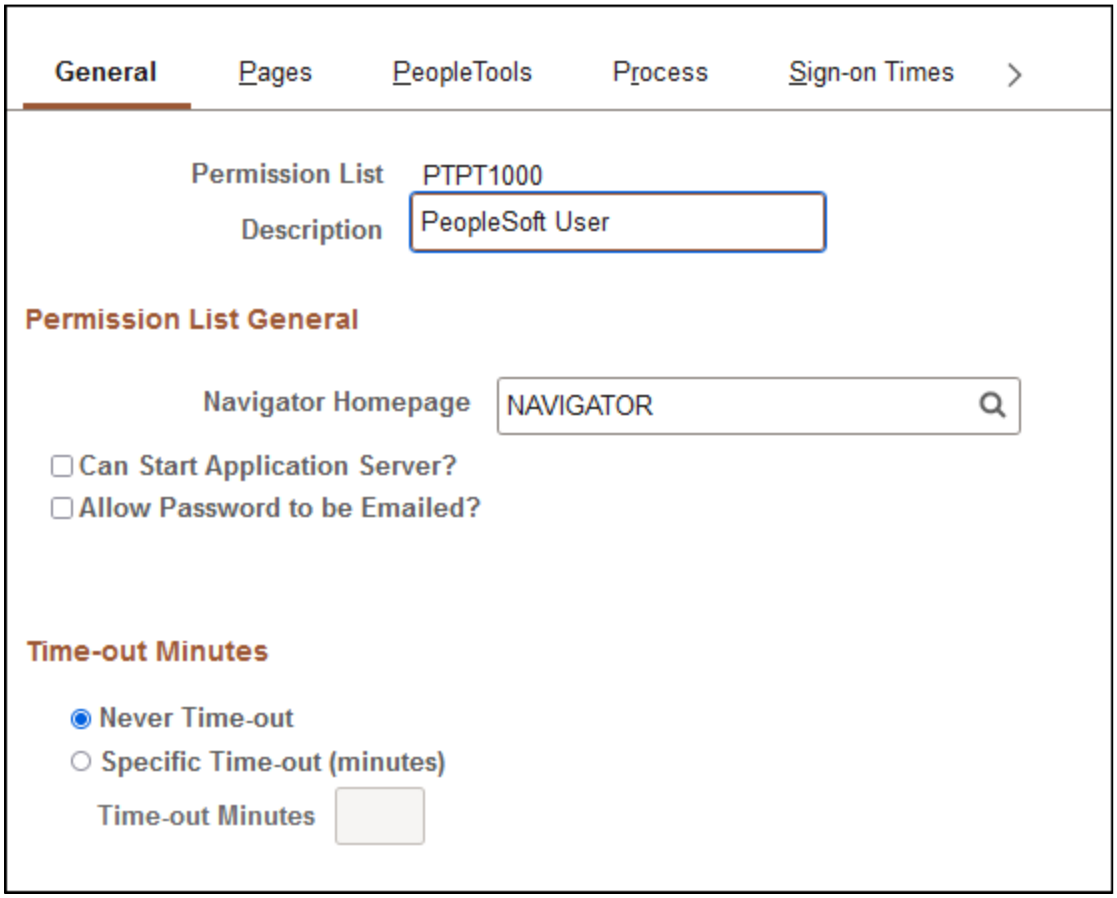
- Set general permissions.
- Set page permissions.
- Set PeopleTools permissions.
- Set process permissions.
- Set sign in time permissions.
- Set component interface permissions.
- Set web library permissions.
- Set web services permissions.
- Set Application Services permissions.
- Set personalization permissions.
- Set query permissions.
- Set mass change permissions.
- Display additional links.
- View when a permission list was last updated.

- Set data migration permissions.
- Assign search group permissions.
- Work with definition security permissions.
- Add permission lists to ACM templates.
- Run permission list queries.

Setting General Permissions

Access the Permission Lists - General page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the **General** tab).

This example illustrates the fields and controls on the Permission Lists - General page.



Field or Control	*Description*
Navigator Homepage	Select a graphic representation of a business process that is displayed by PeopleSoft Navigator. For each security profile definition, you can specify a map to be displayed on startup. If this is the user profile's PeopleSoft Navigator homepage permission list, the system is passed this value at runtime.

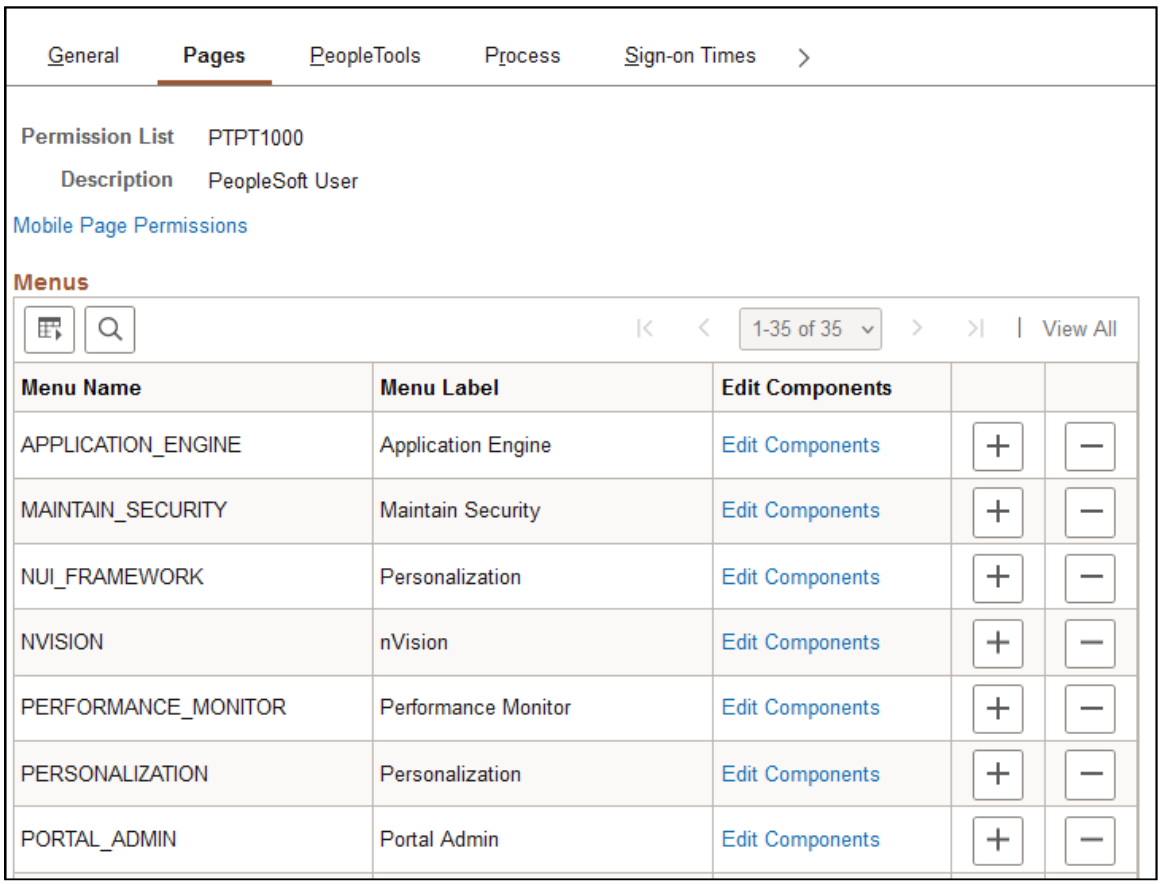
Field or Control	*Description*
Can Start Application Server?	Select to enable user profiles with this permission list to start PeopleSoft application servers. **Note:** This setting also applies to starting PeopleSoft Process Scheduler servers. Typically, you will create a user profile that is dedicated to starting application servers. When you define an application server domain, one of the parameters you specify in PSADMIN is the PeopleSoft user ID (and password) for that profile, which must be associated with at least one permission list that has this option enabled. The user ID and password are stored in the Startup section of the PSAPPSRV.CFG file, which Oracle Tuxedo reads when the application server is started. In many installations, an application server starts with an automated process. A user profile with this property enabled should not be used by an actual user who signs in to the application server and starts it by submitting the appropriate commands. **Note:** Password controls do not apply when a password is used for two-tier activities like starting application servers. They apply only when the password is used to sign in over three-tier connections. **Important!** For a given user profile, the password controls that you set for account lockout (maximum logon attempts) and age (expiration) apply to three-tier and web sign in only; they do not apply if the user profile is used for two-tier activities like starting an application server or process scheduler. However, make sure that you do not use the same user profile for both types of activities. When you use it for both three-tier and web sign in, the profile becomes subject to the account lockout and age controls, which prevents it from completing the two-tier activities.
Allow Password to be Emailed?	Select to enable users to receive forgotten passwords through email. At some sites, the security administrator may not want passwords appearing unencrypted in any email. You implement this feature by permission list. None can use it, some can use it, or all can use it, depending upon your implementation. Users who do not have the proper authority receive an error message if they attempt to have a new password emailed to them.

Field or Control	*Description*
Never Time-Out and **Specific Time-out (minutes)**	Select the number of minutes of inactivity allowed at a terminal before the system automatically signs the user out of the PeopleSoft online system. Inactivity means no mouse clicks, keystrokes, import, file print, or SQL activity. The default time-out minutes setting is Never Time-out. **Note:** Time-out limits are also controlled at the web server and application server levels. If you select Never Time-out, an inactive user is never automatically signed out. Otherwise, select Specific Time-out (minutes) and enter the appropriate value in minutes. A custom time-out interval: • Must be a positive integer. • Cannot contain edit characters, such as commas or a $. • Must be a SMALLINT in the valid range allowed for this field (0-32767). Entering a value of zero (0) is equivalent to selecting Never Time-out. In order for the Never Time-Out or Specific Time-out (minutes) settings to work for a user, every permission list for that user profile must have a non-zero value set for Specific Time-out (minutes). The time-out that will apply to the user profile is the highest value in all the permission lists. Any user profile with the PeopleSoft Administrator Role will ignore the time-out settings on any other permission list. **Note:** Because timeout limits are also controlled at the web server level, you will need to change the web server timeout values also. See “Configuring Portal Security” (Portal Technology) and “PIA Timeouts” (System and Server Administration).

Setting Page Permissions

Access the Permission Lists - Pages page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the Pages tab).

This example illustrates the fields and controls on the Permission Lists - Pages page.



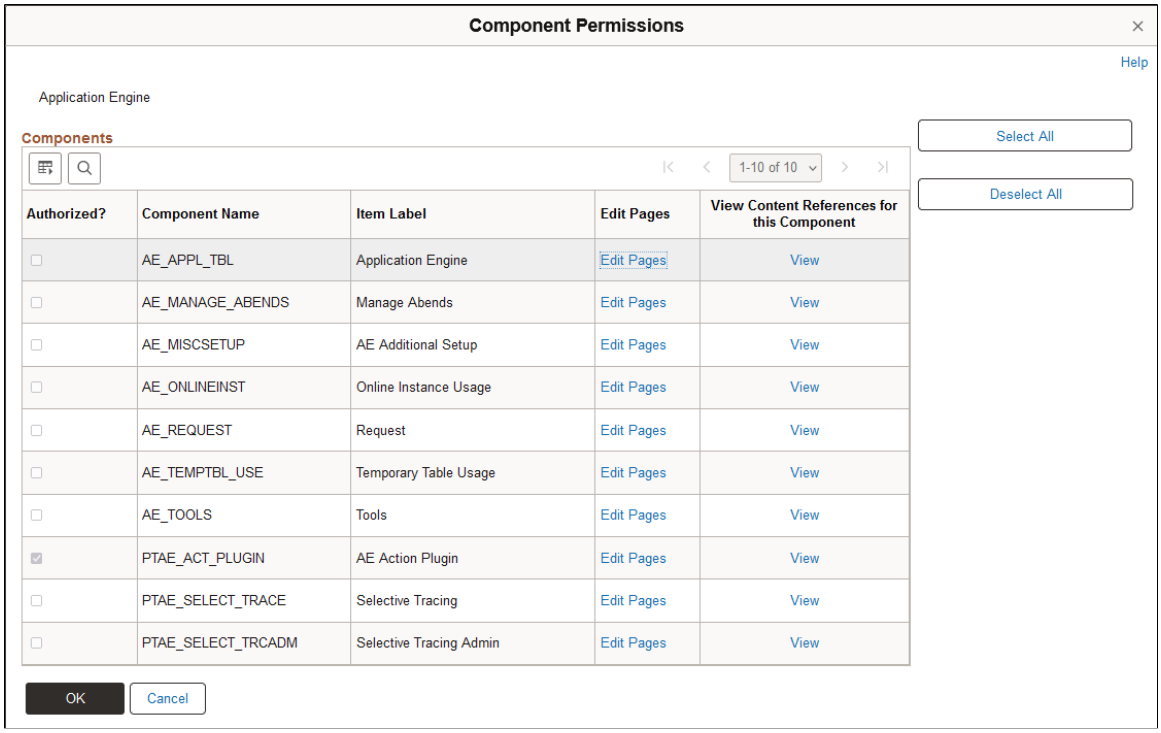
This table describes the fields on the Pages page.

Field or Control	*Description*
Mobile Page Permissions	Click to grant access to mobile application pages. **Important!** PeopleSoft Mobile Agent is a desupported product. These features exist for backward compatibility only.
Menu Name	Displays all menu names in the database. Add new rows to add more menu names. The name reflects the definition name in PeopleSoft Application Designer.
Menu Label	Displays the menu label associated with the PeopleSoft Application Designer menu name.
Edit Components	Click to grant access to specific pages.

Page permissions refer to the pages to which a user has access. Pages are contained within components, which are ultimately contained within a menu name. To grant access to a particular page, determine the component it is in and the menu name the component falls under. This enables you to drill down to the appropriate page.

When you click the **Edit Components** link, the Component Permissions page appears:

This example illustrates the fields and controls on the Component Permissions page.

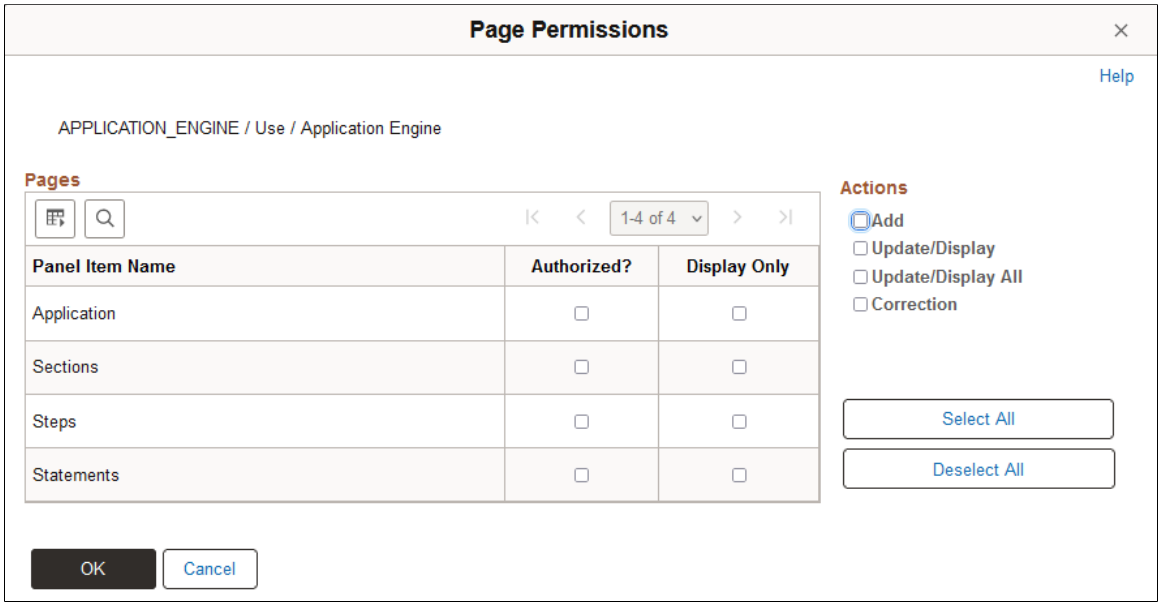


This table describes the fields on the Component Permissions page:

Field or Control	***Description***
Authorized	This field indicates whether at least one page in the component is authorized for current the permission list. This field is display-only.
Component Name	This field indicates the component where the pages reside. This field is display-only.
Item Label	This field indicates the item label on the menu definition where the component resides. This field is display-only.
Edit Pages	Click this link to grant access to individual pages and the appropriate actions.
View Content References for this Component	Click this link to access the content reference.

When you click the **Edit Pages** link, the Page Permissions page appears:

This example illustrates the fields and controls on the Page Permissions page.



This table describes the fields on the Page Permissions page:

Field or Control	*Description*
Panel Item Name	This is the name of the panel item as entered in the component in Application Designer. This field is display-only.
Authorized?	Select this check box to authorize user access to the page.
Display Only	Select this check box to authorize view only user access to the page. No fields are active when this check box is selected.
Actions	Select from the following check boxes: **Add:** The user can create new high-level key information through the search page. **Update/Display:** The user can view the current row. The user can view, insert, and update future rows. **Update/Display All:** The user can view the history and current rows. The user can view, insert, and update future rows. **Correction:** The user can view, insert, and update history, current, and future rows. **Note:** Only actions that are selected in the component definition in Application Designer are enabled.

Note: To find the name of a menu, component, or page, you can press **Ctrl+J** or **Ctrl+Alt+J** while accessing the page with the browser (the keyboard shortcut differs depending on the browser you use), or use the Find Definition References feature in PeopleSoft Application Designer.

Granting access to PeopleTools and PeopleSoft applications requires serious consideration. For each role, carefully consider what the members of that role must access to complete their jobs and to what degree they need access. Then make the appropriate permission lists.

After you add a menu name, you grant access to its components and pages on an item-by-item basis. In PeopleSoft applications, menu items represent components. If a component consists of more than one page, then selecting the menu item opens another layer with more items—individual pages. For example, if you added the UTILITIES menu name to a permission list, you could then grant access to the Utilities, Use menu items but not to the Utilities, Process menu items. Alternatively, you could grant access to only a few of the Use menu items or make some items display only.

You grant access permission to two categories of components:

- All PeopleSoft applications.
- Page-driven PeopleTools.

Note: With PeopleTools programs, the process of editing menu items varies. With page-based PeopleTools, such as PeopleSoft Process Scheduler, you can grant access to menu items just as you can for PeopleSoft applications. However, the other PeopleTools programs do not allow you to grant item-by-item access; you can either access all the menus and menu items or you cannot. PeopleSoft Application Designer is an exception; you can restrict access to it at the definition level.

Granting Access to Components and Pages

The following procedure describes how to set access permissions to your PeopleSoft applications and your page-driven PeopleTools. You begin at the component level and drill down to the page level, making the appropriate selections as you go.

Note: The same procedure applies to both PeopleSoft applications and page-driven PeopleTools.

To add access to PeopleSoft components and pages:

1. Locate the menu name of the component to which you want to add access.
2. Click **Edit Components.**

   The Components page appears.
3. Locate the component to which you want to grant access.

   By default, when adding a new permission list, no components are authorized.
4. Click the **Edit Pages** button associated with each component to which you want to grant access.

   The Page Permissions page appears. You specify the actions that a user can complete on this page. You can select from these options for each page that appears in the Page column:

   - **Authorized?**

Select to enable a user to access the page. Decide the degree to which a user is authorized on a page by selecting **Display Only** or one or more of the available options in the **Actions** group.

- **Display Only.**

  Select to enable the user to view the information provided by the page but not to alter any data.

- **Actions.**

  Specify how users can alter information on a page, such as Add, Update/Display, and Correction. The available options depend upon the options selected when the page was initially developed in PeopleSoft Application Designer.

  To grant access to all pages and all actions for each page, click **Select All.**

5. When you have finished making the appropriate selections, click **OK** on the Page Permissions page, and then again on the Component Permissions page.

   Repeat each step for each menu name.

Note: After you delete access to a component or iScript, you must clear the browser cache or wait for 20 minutes (default time) for the deletion to appear in the menu.

Granting Access to Mobile Pages

To add access to mobile pages:

Important! PeopleSoft Mobile Agent is a desupported product. These features exist for backward compatibility only.

---

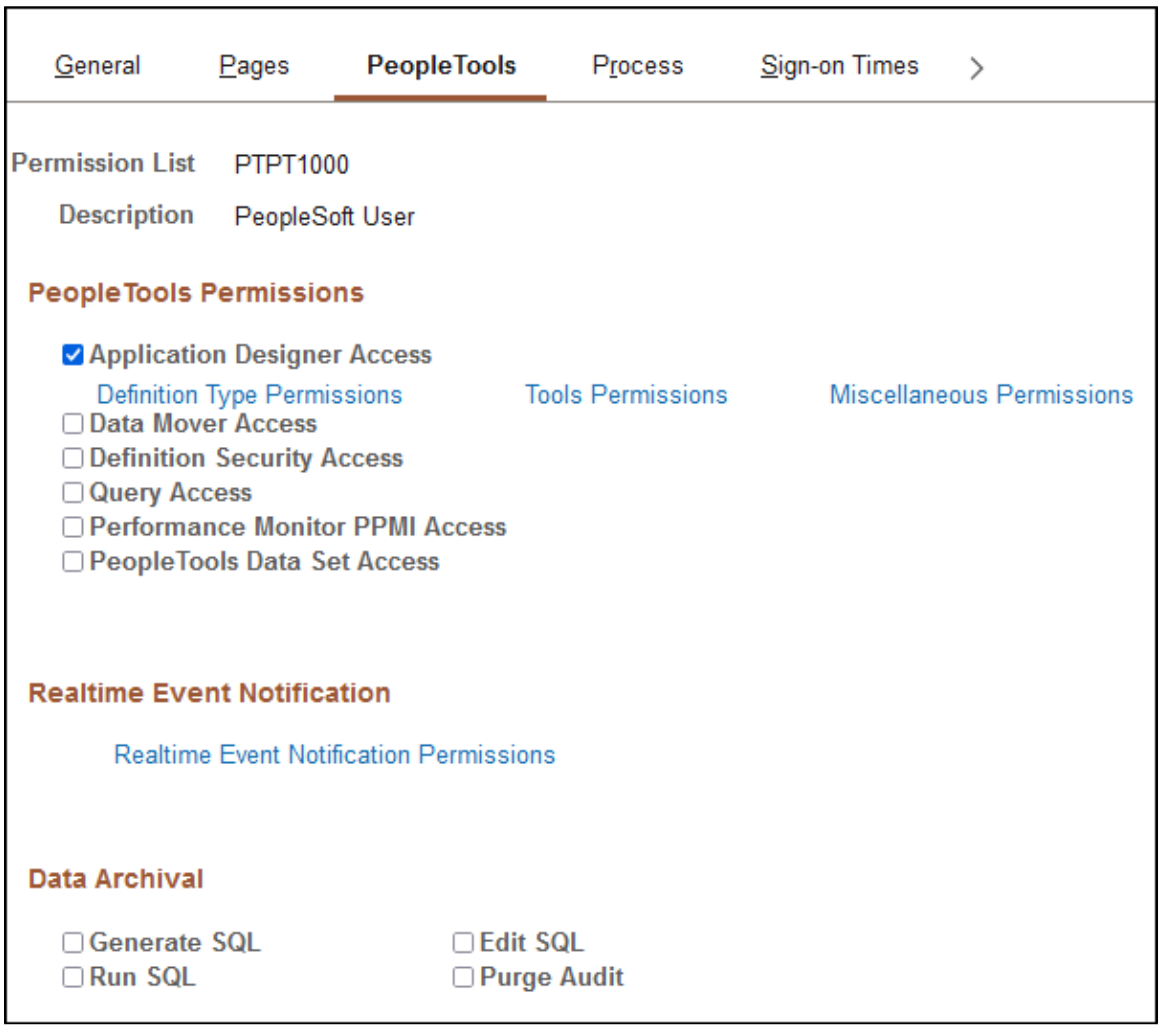
1. Select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists**, and select the Pages page.
2. Click the **Mobile Page Permissions** link.

   The Mobile Page Permissions page appears.
3. To add a new mobile page to the permission list, click the plus sign.
4. For the Mobile Page Name edit box, click the search button.
5. Search for and select the mobile page for which you need to grant access.
6. Click **OK.**
7. Save the permission list.

Setting PeopleTools Permissions

Access the Permission Lists - PeopleTools page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the **PeopleTools** tab).

This example illustrates the fields and controls on the Permission Lists - PeopleTools page.



The PeopleTools Permissions section of this page applies to standalone PeopleTools applications. They are not PeopleSoft Pure Internet Architecture-based, but are Microsoft Windows programs that were not developed using PeopleSoft Application Designer. They include:

- PeopleSoft Application Designer.
- PeopleSoft Data Mover.
- PeopleSoft Definition Security.
- PeopleSoft Query (Microsoft Windows interface, not the browser interface).

The **Performance Monitor PPMI Access** check box does not control access to an application; rather, it enables PeopleSoft Performance Monitor data collators to insert performance data into the database, which enables you to view the data.

To grant access to these PeopleTools features, select the check box next to the appropriate item.

With PeopleSoft Application Designer, the procedure for applying permissions is slightly more complex, because security for PeopleSoft Application Designer also controls what object definition types can be accessed and what degree of modifications can be made. The **Definition Type Permissions, Tools**

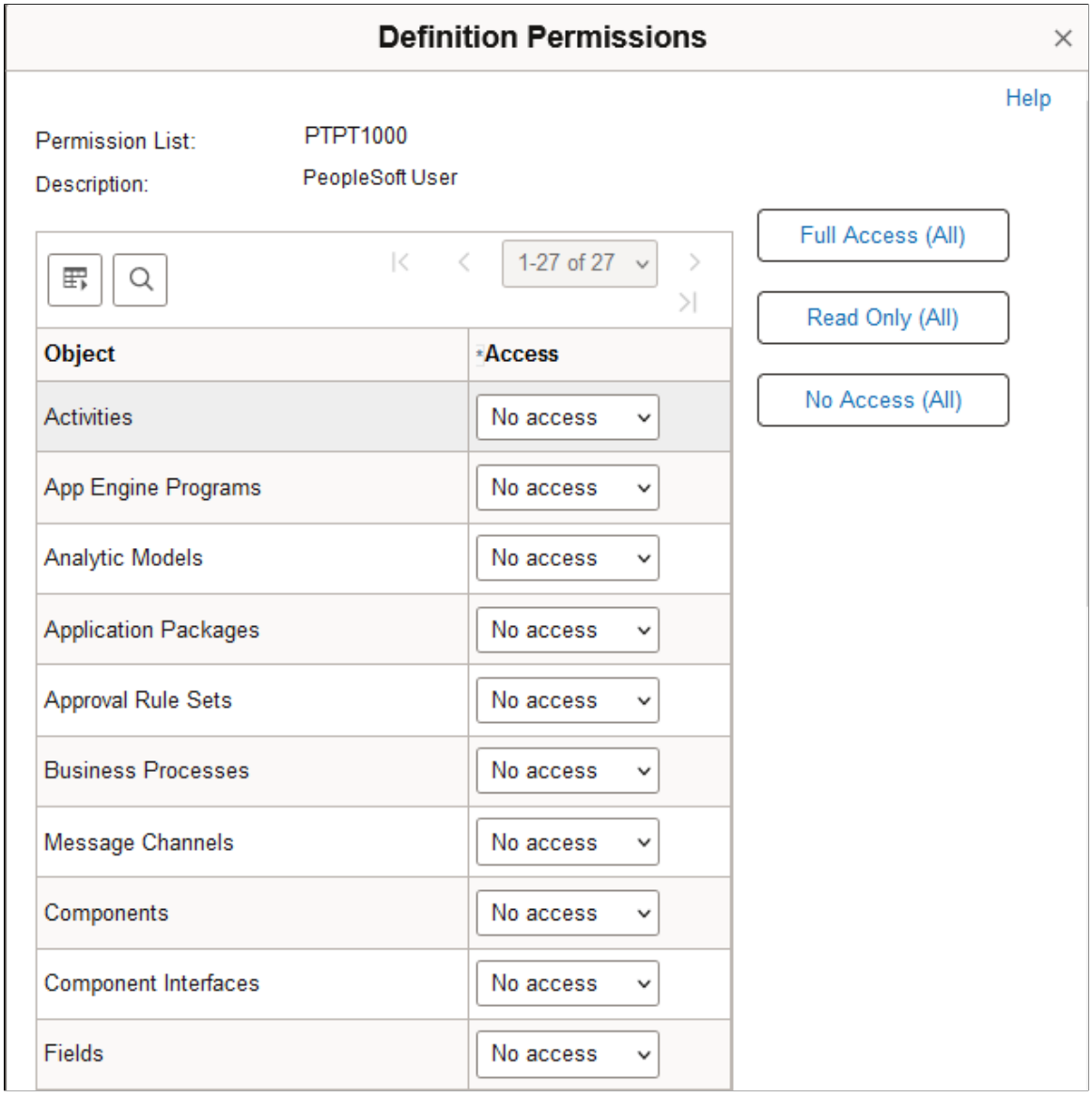
Permissions, and **Miscellaneous Permissions** links enable you to provide more detail to PeopleSoft Application Designer access permissions.

Definition Type Permissions

Use the Definition Permissions page to control access to definition types in PeopleSoft Application Designer. If you want to control access to definition types at runtime, you must use the Definition Security tab to set the access permissions. For more information on setting runtime access permissions, see Working with Definition Security Permissions.

Access the Definition Permissions page (click the **Definition Type Permissions** link on the Permission Lists - PeopleTools page).

This example illustrates the fields and controls on the Definition Permissions page.

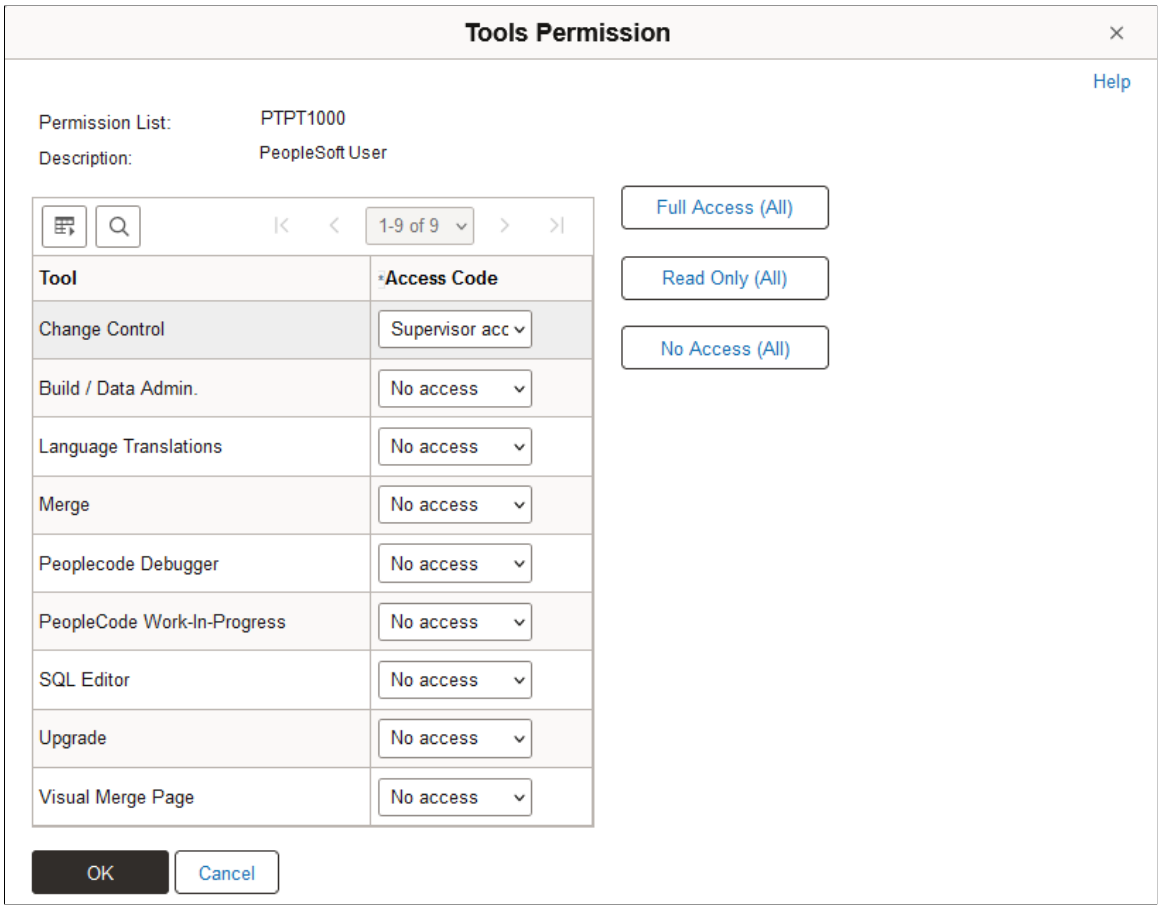


Field or Control	*Description*
Access Code	Select the appropriate access level. Options are: *Full access:* Definitions of the specified type can be modified. For records, this setting allows access to the Build dialog box. *No access:* No definitions of the specified type can be opened. *Read-only access:* Definitions of the specified type can be opened and viewed, but not modified. *Update translates only:* This level applies only to fields. This setting allows a user to modify only Translate table values. *Data admin only:* This level applies only to records. It allows a user to modify only those record attributes found in the Tools, Data Administration menu (tablespaces, indexes, and record DDL). **Note:** Projects cannot be set to *No access* because Application Designer requires access to a project to launch successfully.
Full Access (All), Read Only (All), and **No Access (All)**	Click to set all definition types in the list to the same access level.

Tools Permissions

Access the Tools Permission page (click the **Tools Permissions** link on the Permission Lists - PeopleTools page).

This example illustrates the fields and controls on the Tools Permission page.



In addition to securing definitions, PeopleSoft Application Designer security also involves a collection of tools, such as Build and the PeopleCode Debugger, to which developers need access.

The tools within PeopleSoft Application Designer include:

- Change Control (select **Tools** > **Change Control**).
- Build/Data Admin (select **Build** > **Project and Tools** > **Data Administration**).
- Language Translations (select **Tools** > **Translations**).
- PeopleCode Debugger (select **Debug** > **PeopleCode Debugger Mode**).
- SQL Editor (the PeopleSoft Application Designer utility for adding SQL objects and statements to applications and application engine programs).
- Upgrade (select **Tools** > **Upgrade**).

  This tool includes Copy Project, Compare and Report, and so on.

You can set the access level individually for the Tools Permissions page options or your can use the **Full Access (All)**, **Read Only (All)**, or **No Access (All)** buttons to set across-the-board settings. Remember that every button affects every access level for the tools.

Field or Control	*Description*
Change Control	The change control access levels are valid only when change control is enabled. You enable change control locking using PeopleSoft Application Designer. Select from: • *Restricted access:* Restricts users from locking or unlocking objects. When change control locking is enabled, users with restricted access can only view PeopleSoft Application Designer definitions; they cannot create, modify, or delete them. **Note:** With locking enabled, this setting overrides any *Full Access* settings on the Object Permissions page or Miscellaneous Permissions page. • *Developer access:* The user can lock any unlocked objects and unlock any objects that he or she has locked. • *Supervisor access:* The user can unlock any locked objects, regardless of who locked them.

Field or Control	*Description*
Build/Data Admin	Control access to the Build and Tools, Data Administration menu items. Select from: • *No access:* The user cannot access the Build menu items or the Tools, Data Administration menu items. **Note:** This setting is not available if you have set records access to No Access or to Data Admin only. • *Build scripts only:* A user with this access level can use the Build dialog box options, but the Execute SQL now and Execute and build script options are disabled. The Tools, Data Administration menu items are not available. **Note:** This setting is not available if you have set records access to No Access. • *Build Online:* With this access level, a user can use all Build dialog options, but the Tools, Data Administration menu items are not available **Note:** This setting is not available if you have set records access to No Access. • *Full data admin access:* A user with this access level can use all the Build dialog options and access the Tools, Data Administration menu items. **Note:** This setting is not available if you have set records access to No Access or Read-only.
Language Translations	Set only two levels of access, *No access* and *Full access*. Enable this set of menu options for people involved in translating or globalizing your applications.
PeopleCode Debugger	Restrict access to the PeopleCode Debugger.
SQL Editor	Restrict developers from modifying the SQL in your applications.

Field or Control	*Description*
Upgrade	Select *No access* to make all the Upgrade menu items on the Tools menu unavailable. Developers can still access the Upgrade view and modify upgrade settings in the project definition, but they cannot run any the upgrade processes. With *Read-only access*, users can run compare reports against the database, but they cannot copy objects into the database.

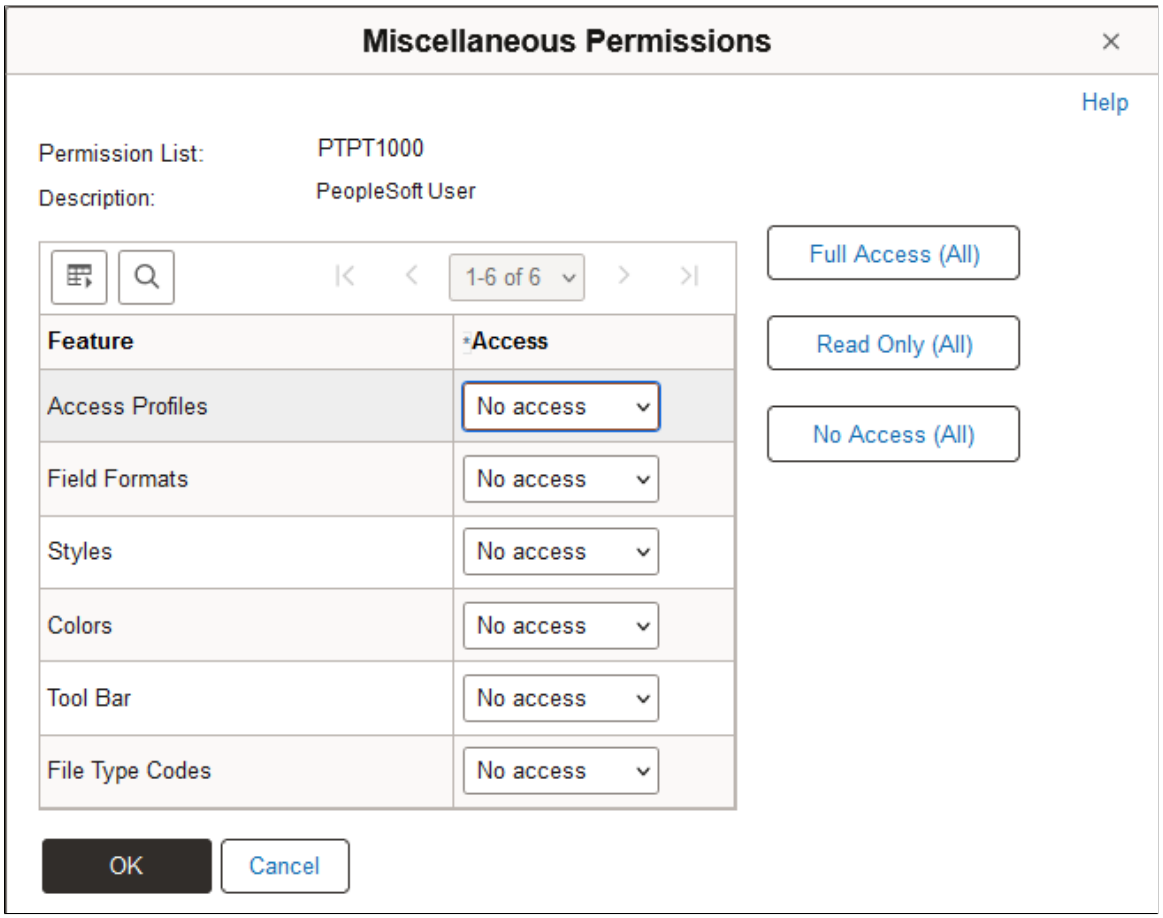
The following table shows the relationship between the permissions that are set up within the source and the target databases, which you should consider in upgrade situations:

Source DB	*Target DB*	*Compare?*	*Copy?*	*Export?*	*Import?*
No access	No access	No	No	No	No
No access	Read-only access	No	No	No	No
No access	Full access	No	No	No	No
Read-only access	No access	No	No	Yes	No
Read-only access	Read-only access	Yes	No	Yes	No
Read-only access	Full access	Yes	Yes	Yes	No
Full access	No access	No	No	Yes	Yes
Full access	Read-only access	Yes	No	Yes	Yes
Full access	Full access	Yes	Yes	Yes	Yes

Miscellaneous Permissions

Access the Miscellaneous Permissions page (select the **Miscellaneous Permissions** link on the Permission Lists - PeopleTools page).

This example illustrates the fields and controls on the Miscellaneous Permissions page.



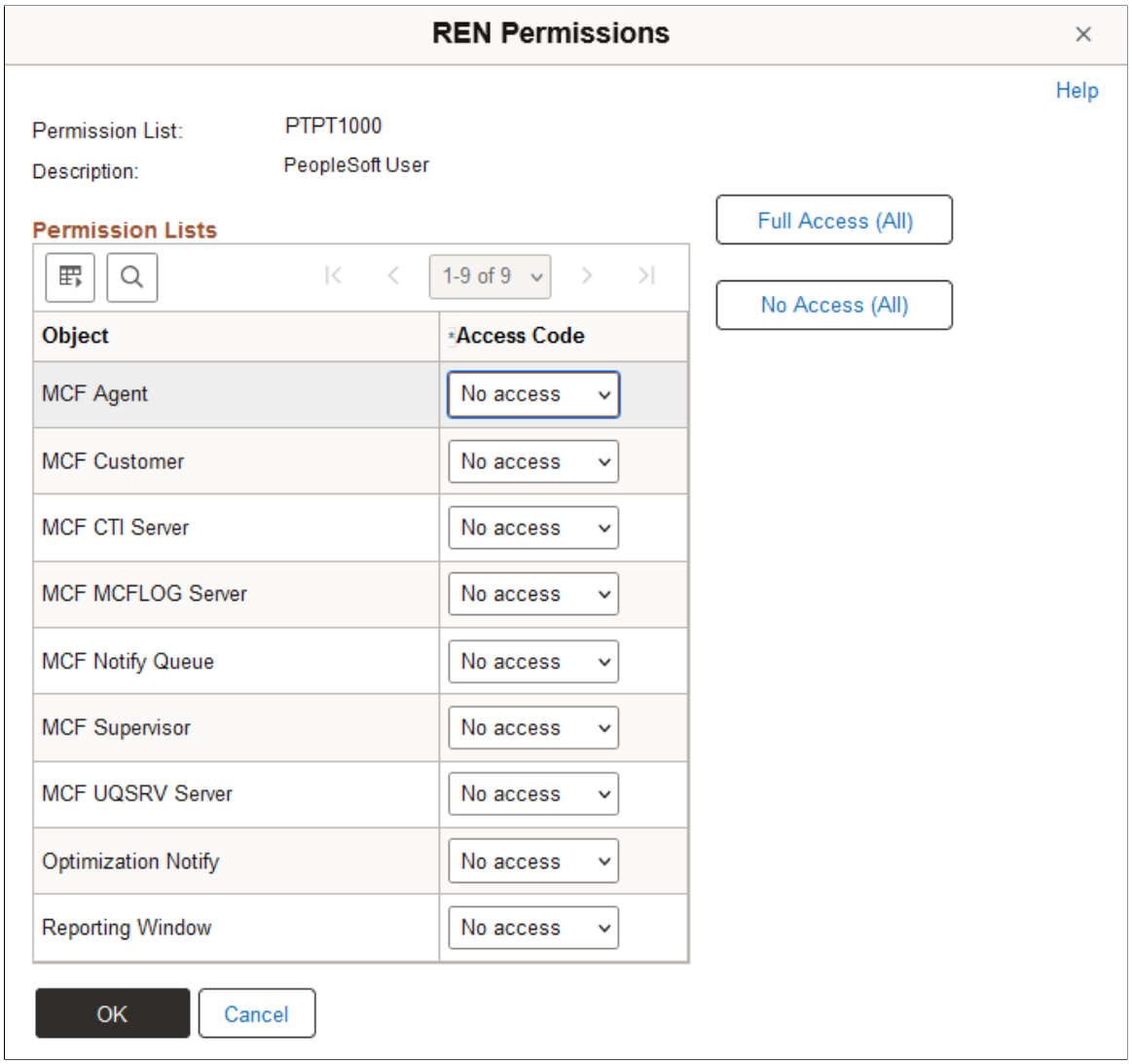
Set access levels for the Miscellaneous Definitions items that appear in the PeopleSoft Application Designer Tools menu, including Access Profiles, Color, Field Format, Style, and Tool Bar.

Each of the miscellaneous definitions can be set for *No access*, *Read-only access*, or *Full access*. You can select the **Full Access (All)**, **Read Only (All)**, or **No Access (All)** buttons to grant the same permissions to each item.

Real-time Event Notification Permissions

Access the REN Permissions page (click the **Realtime Event Notification Permissions** link on the Permission Lists - PeopleTools page).

This example illustrates the fields and controls on the REN Permissions page.



The REN Permissions page enables you to control REN server access. Before you grant any permissions to these actions, read the PeopleSoft MultiChannel Framework documentation.

See “Configuring REN Server Security” (MultiChannel Framework).

Data Archival

PeopleSoft Data Archive Manager is a page-driven PeopleTools application that you use to archive your application data as part of regular database maintenance. The security options in this group relate specifically to actions a system administrator would make while using PeopleSoft Data Archive Manager. The actions that a system administrator can perform within PeopleSoft Data Archive Manager are controlled by permission lists. Before you grant any permissions to these actions, read the PeopleSoft Data Archive Manager documentation.

Setting Process Permissions

Access the Permission Lists - Process page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the **Process** tab).

Just as you define permissions for the pages a user can access, you also must specify the batch (and online) processes that users can invoke through PeopleSoft Process Scheduler. Typically, process groups are arranged by department or task. For example, the batch programs used by your payroll department probably all belong to the PAYROLL process group, or a similarly named group.

When you create a process permission list, you add the appropriate process groups so that a user belonging to a particular role can invoke the proper batch programs to complete their business transactions. You do this using the Process Group Permission page.

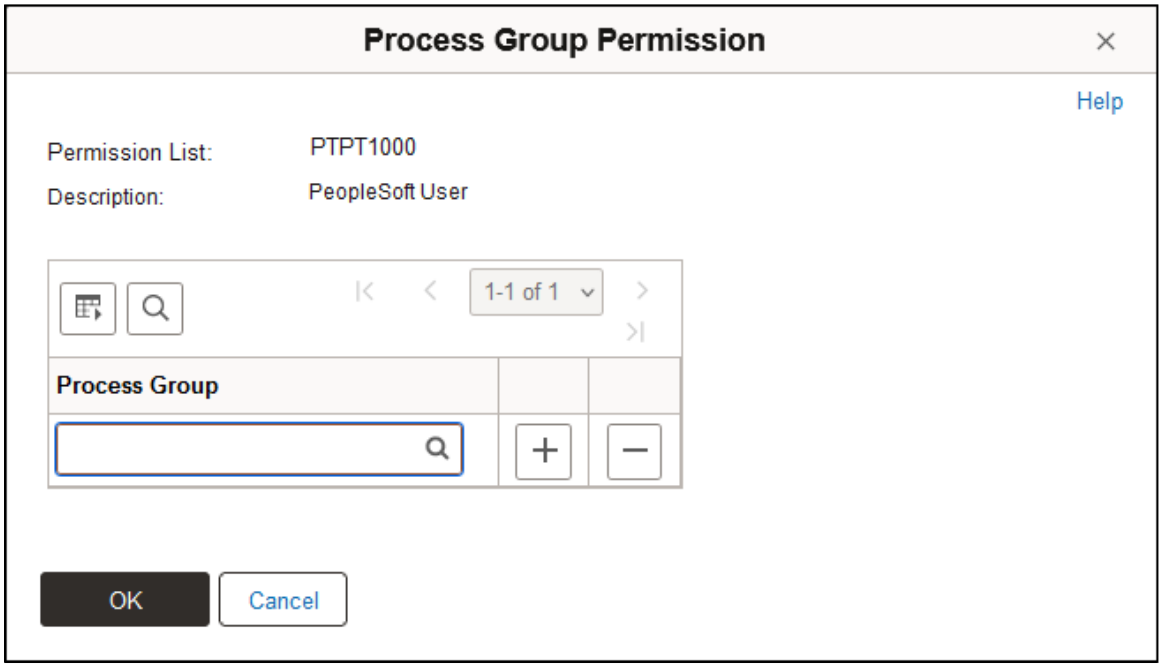
You use the Process Profile Permission page to specify when a user or role can modify certain PeopleSoft Process Scheduler settings.

Note: You grant Process Profile permissions directly to the user profile and Process Group permissions through permission lists.

Process Group Permissions

Access the Process Groups page (click the **Process Group Permissions** link on the Permission Lists - Process page).

This example illustrates the fields and controls on the Process Group Permission page.



This page lists the process groups associated with a permission list. Process groups are collections of process definitions that you create using PeopleSoft Process Scheduler.

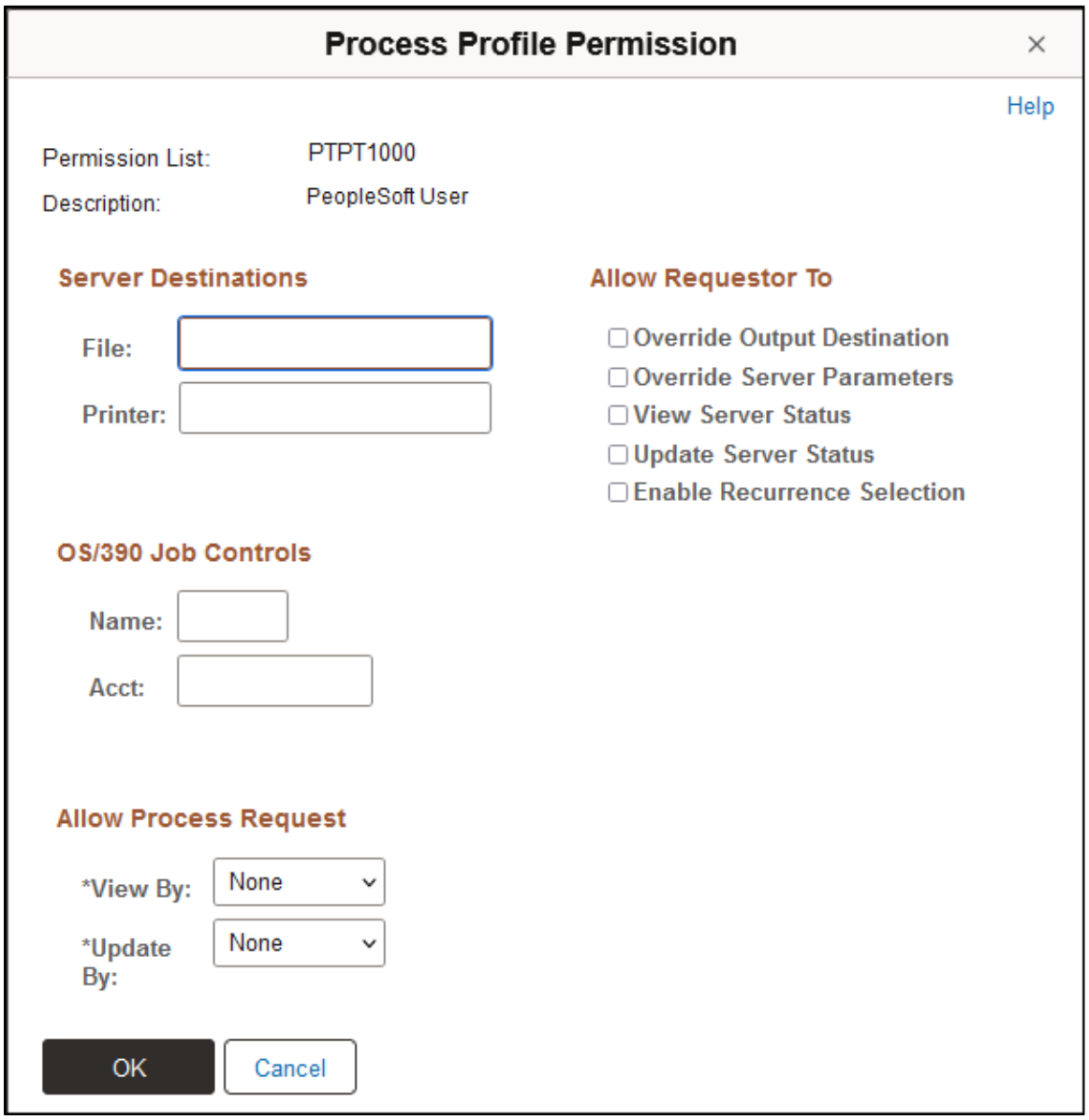
Typically, you group process definitions according to work groups within your organization, and typically that work group has a particular role associated with it. Regardless of how you organize process definitions, you must assign process groups to a permission list.

Users can run only the processes that belong to process groups assigned to their roles. For example, you may have a set of process definitions that relate to your Human Resources department and another set for your Manufacturing department.

Process Profile Permissions

Access the Process Profile Permission page (select the **Process Profile Permissions** link on the Permission Lists - Process page).

This example illustrates the fields and controls on the Process Profile Permission page.



Field or Control	*Description*
Server Destinations	You can specify output variables when running processes or jobs on a server. You have the following options: • **File:** If the output is going to a file, then specify the directory to which the file should be written. %%OutputDirectory%% is a meta-variable that resolves to the output directory that you specified in PSADMIN (or PSPRCS.CFG) for the Process Scheduler Server Agent. • **Printer:** Specify the network or local printer to which the hard-copy output should be sent. You must explicitly specify the printer; no meta-variables are available for this value.
OS/390 Job Controls	**Note:** This functionality is no longer supported.

Field or Control	*Description*
Allow Process Request	These options apply to using PeopleSoft Process Monitor. You can restrict which users are permitted to view or update a given process based on the user who launched (and owns) the process. You can specify restrictions as follows: • **View by:** Specify who can view processes that are launched by users who have this permission list assigned as their process profile permission list on the User Profile - General page. Select from the following options: • *Owner:* For a process launched by a user who has this process profile permission list assigned, only the user who launched the process can view it. • *All:* All users can view processes that are launched by a user who has this process profile permission list assigned. • *None:* No one can view processes that are launched by a user who has this process profile permission list assigned. • **Update By:** Specify who can update the status of processes that are launched by users who have this permission list assigned as their process profile permission list on the User Profile - General page. For example, you decide whether users can restart or cancel a request. **Note:** Updates are made using the PeopleSoft Process Monitor Process Detail page in the Update Process component. Select from the following options: • *Owner:* For a process launched by a user who has this process profile permission list assigned, only the user who launched the process can update it. For example, nobody else can restart a request that this user submitted. However, this user might still be able to update another user's processes.

Field or Control	*Description*
	• *All:* All users can update processes that are launched by a user who has this process profile permission list assigned. • *None:* No one can update processes that are launched by a user who has this process profile permission list assigned. **Note:** Be careful as you grant update authority to submitted processes. An inexperienced user can easily disrupt batch processing by deleting or holding processes, especially when restarting processes. If a program is not coded for a restart, then users should not be able to restart it. Restarting a program that is not properly coded to acknowledge the previous program run can threaten data integrity. Remember, the process profile permissions are based on the profile of the user who is submitting the process, not the user viewing the process monitor.

The **Allow Requestor To** options apply to using PeopleSoft Process Monitor and PeopleSoft Process Scheduler Request pages. These options enable you to restrict the authority that a user has while monitoring scheduled processes.

Field or Control	*Description*
Override Output Destination	Select to allow a user to change the value in the Output Destination column on the Process Scheduler Request page.
Override Server Parameters	Select to enable users to select the server name and modify the run date/time group on the Process Scheduler Request page.
View Server Status	Select to enable users to access the Server List page in PeopleSoft Process Monitor.
Update Server Status	Select to allow a user to suspend, restart, or bring down a server using the Server Detail page from the server list in PeopleSoft Process Monitor.
Enable Recurrence Selection	Select to enable a run recurrence value for processes and jobs scheduled to run on the server.

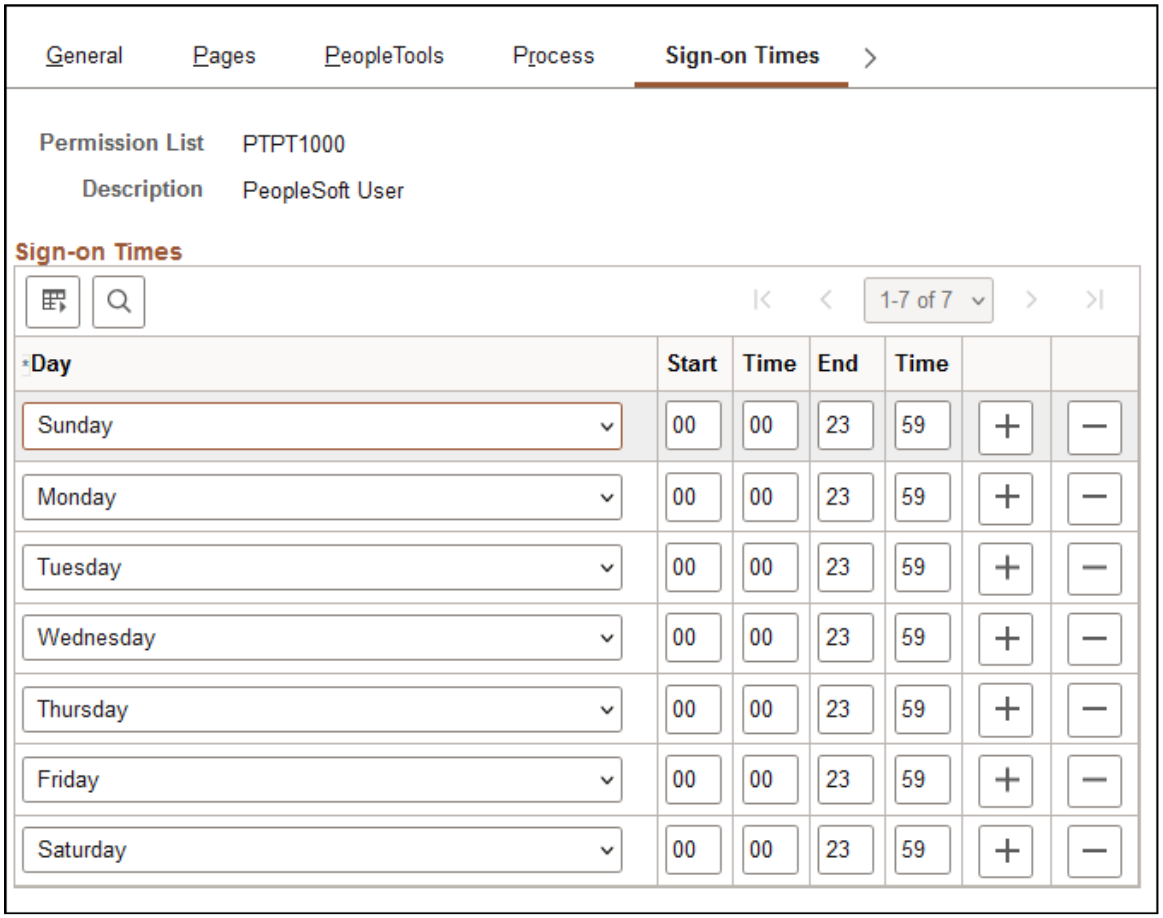
Related Links

"Setting Up PeopleSoft Process Scheduler Privileges and Profiles" (Process Scheduler)
"Setting Process Definition Options" (Process Scheduler)

Setting Sign-on Time Permissions

Access the Permission Lists - Sign-on Times page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the Sign-on Times tab).

This example illustrates the fields and controls on the Permission Lists - Sign-on Times page.



Pick a day and set a sign-on duration.

Sign-on times use the 24-hour clock and run through the end time value. For example, a user with an end time of 16:30 can use the system until 4:31 p.m.

To create a sign-on time that spans multiple days, use adjoining sign-on times. For example, to create a sign-on time running from 8 p.m. Tuesday to 6 a.m. Wednesday, you need a Tuesday start time of 20:00 and end time of 23:59. Then you need to add a Wednesday sign-on time with a start time of 00:00 and an end time of 05:59.

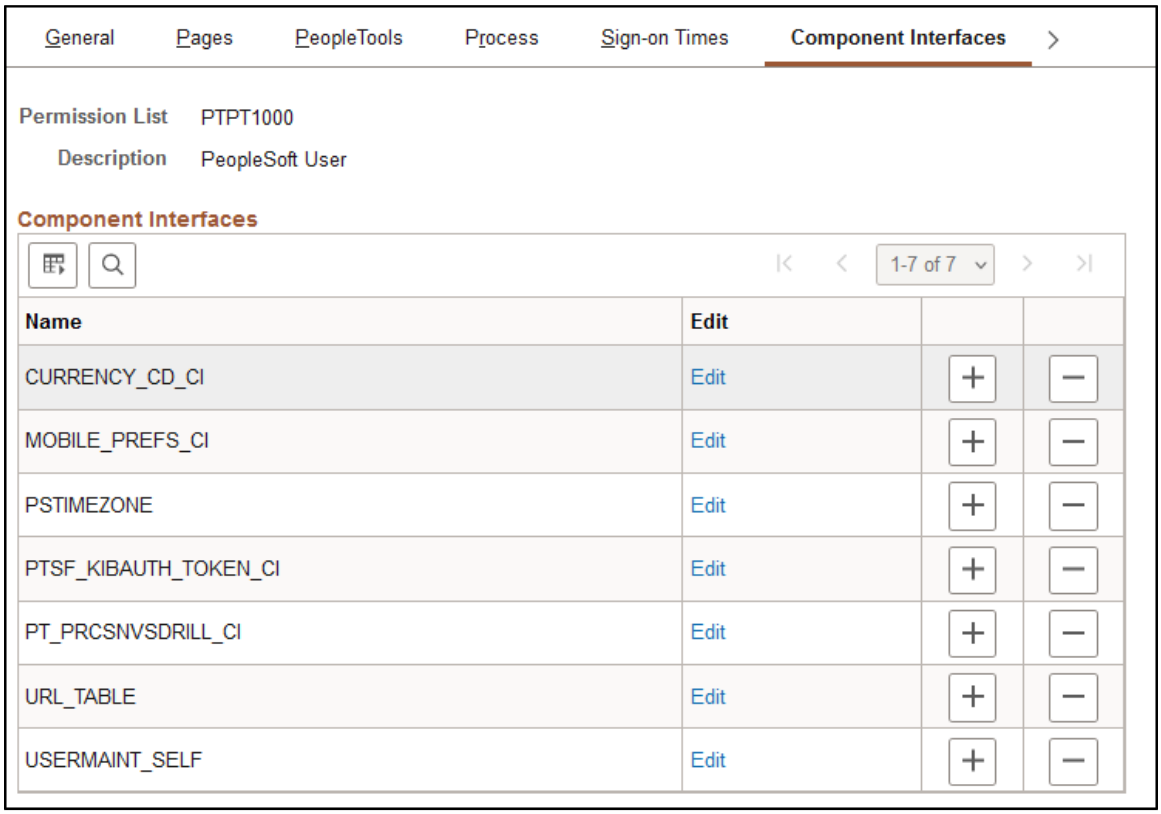
By default, all start times are 00:00 and end times are 23:59, and all days are listed. Delete days and change the times to restrict access.

A single day can have more than one sign-on period as long as the periods do not overlap. If a single day has multiple non-overlapping sign-on periods, then that day appears once for each period.

Setting Component Interface Permissions

Access the Permission Lists - Component Interfaces page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the **Component Interfaces** tab).

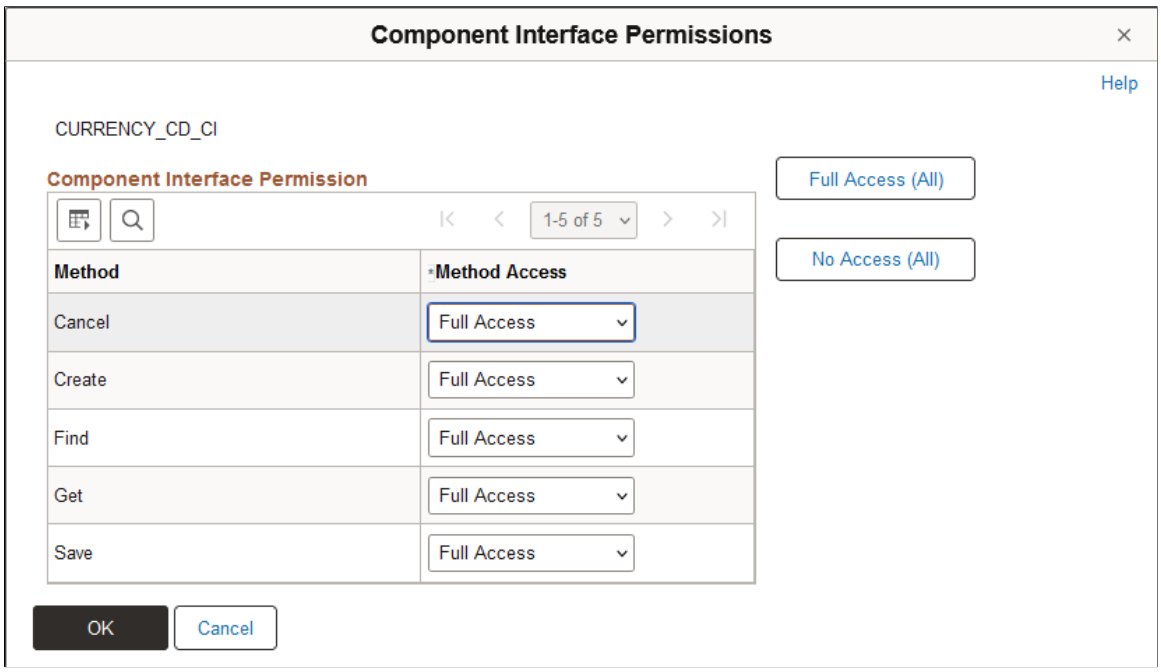
This example illustrates the fields and controls on the Permission Lists - Component Interfaces page.



Field or Control	*Description*
Name	Shows the name of the component interface.
Edit	Click to access the Component Interface Permissions page and grant access to a particular component interface method.

Click the **Edit** button to authorize individual methods in each component interface:

This example illustrates the fields and controls on the Component Interface Permissions page.



Field or Control	*Description*
Method	Displays each method created within the component interface.
Method Access	Select from these two types of authorization: *Full Access:* The method is authorized. *No Access:* The method is not authorized.
Full Access (All)	Grants full access to all scripts listed on the page.
No Access (All)	Denies access to all scripts listed on the page.

You grant access to component interfaces similarly to adding page access. Add a new row to insert a component interface into the definition list. You must also grant access to the component interface methods.

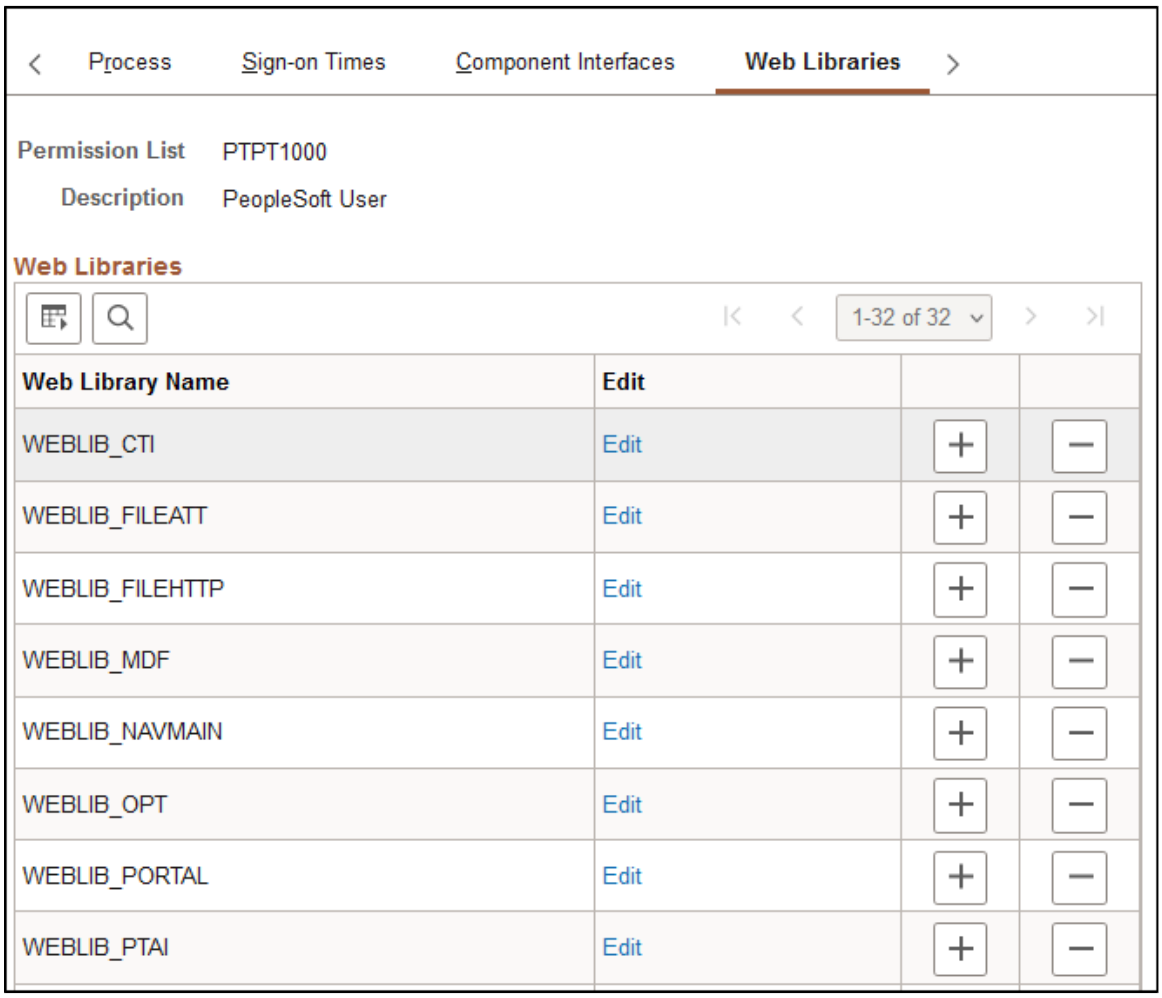
After adding a new permission to a component, you must delete the web server cache for users to access the component through the portal. To delete the web server cache, reboot the web server.

Note: If more than one JVM services the web server, then rebooting the web server only purges the in-memory cache. No procedure exists to specify which JVM receives the request. For this reason, you must reboot all JVMs that service the web server.

Setting Web Library Permissions

Access the Permission Lists - Web Libraries page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the **Web Libraries** tab).

This example illustrates the fields and controls on the Permission Lists - Web Libraries page.



A web library is a derived/work record whose name starts with *WEBLIB_*. All PeopleSoft iScripts are embedded in records of this type. An iScript is a specialized PeopleCode function that generates dynamic web content.

Administrators should make sure that users have the proper access to web libraries. For example, the default navigation system for application users is implemented using a web library. If users do not have the proper authorization to the web library and its associated scripts, then they will not have proper access to the system. If users are not authorized for a particular web library or script, then they cannot invoke it.

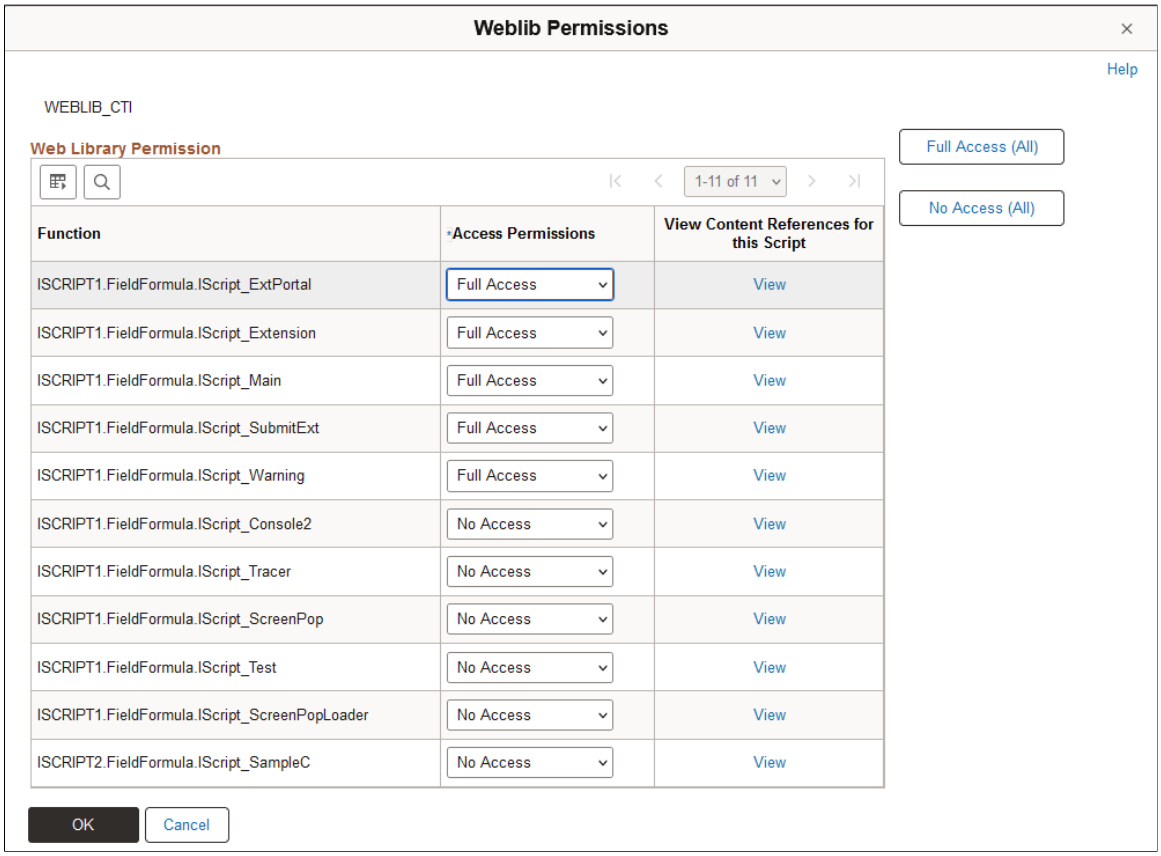
After you add a web library, you set the access for each script function individually. Invoking an iScript requires the assembly of a URL. Developers assemble the URL using PeopleCode.

Field or Control	***Description***
Web Library Name	Displays the web libraries added to the permission list.

Field or Control	*Description*
Edit	Click to set access to web library functions. Select from these access rights for each function: *Full Access:* Select this value to authorize the script. *No Access:* Select this value to deny access to the script.

Click the **Edit** button to authorize each script in the web library:

This example illustrates the fields and controls on the Weblib Permissions page.



Field or Control	*Description*
Function	Displays each script stored in the web library.
Access Permissions	Click to set access to web library functions. Select from these access rights for each function: *Full Access:* Select this value to authorize the script. *No Access:* Select this value to deny access to the script.
Full Access (All)	Grants full access to all scripts listed on the page.

Field or Control	*Description*
No Access (All)	Denies access to all scripts listed on the page.
View	Click to launch the content reference associated with the iScript.

Note: You must grant access to at least one script in the web library, otherwise the system removes the web library from the permission list when you leave the component—even if you save the component.

Setting Web Services Permissions

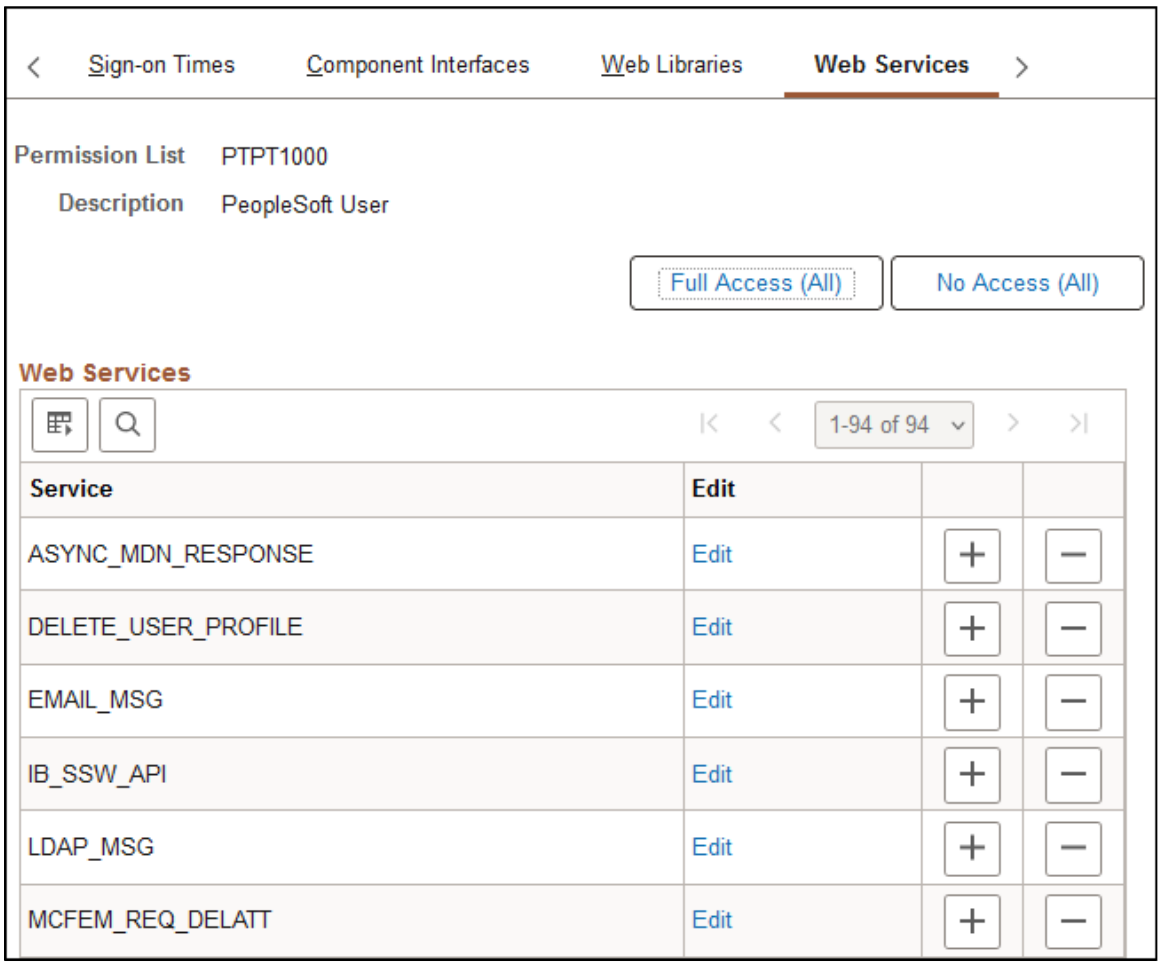
The web services offered by the PeopleSoft Integration Broker can be secured at the user ID level through the use of the web services permissions you specify. This applies to external web service requests only, not internal web service requests. Internal requests are those submitted from within your PeopleSoft system by a PeopleSoft user of one of your deployed PeopleSoft applications. External requests are those received from third party systems, such as other applications in your organization or other systems outside your organization sending requests over the internet.

If the user ID and password contained in the web service request has the appropriate permissions, the user can invoke the web service. If the submitted user ID and password fails authentication, the user has no permission to invoke the service. If only a User ID is provided, the PeopleSoft system attempts to verify if the user ID is a valid PeopleSoft user. If the verification fails, the system checks if the request is from a trusted node, and then uses the external user ID and password associated with the node from which the request was generated. If the request is not from a trusted node, the system checks the user ID associated with the ANONYMOUS node. How PeopleSoft Integration Broker handles authenticating web service request permissions is discussed in detail in the product documentation for PeopleTools: Integration Broker.

See “Implementing Web Services Security” (Integration Broker Administration).

Access the Permission Lists - Web Services page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the Web Services tab).

This example illustrates the fields and controls on the Permission Lists - Web Services page.



Add the web services to which a permission list should have access. Add and remove web services to and from the list using the standard plus and minus buttons.

Note: Web service requests contain user IDs. For the web service to be invoked, the submitted user ID must be valid in the PeopleSoft system. For example, the user account cannot be locked, the request must be submitted during the user ID's valid sign-on times, and the user ID must have permission to invoke the web service operation.

Web Services

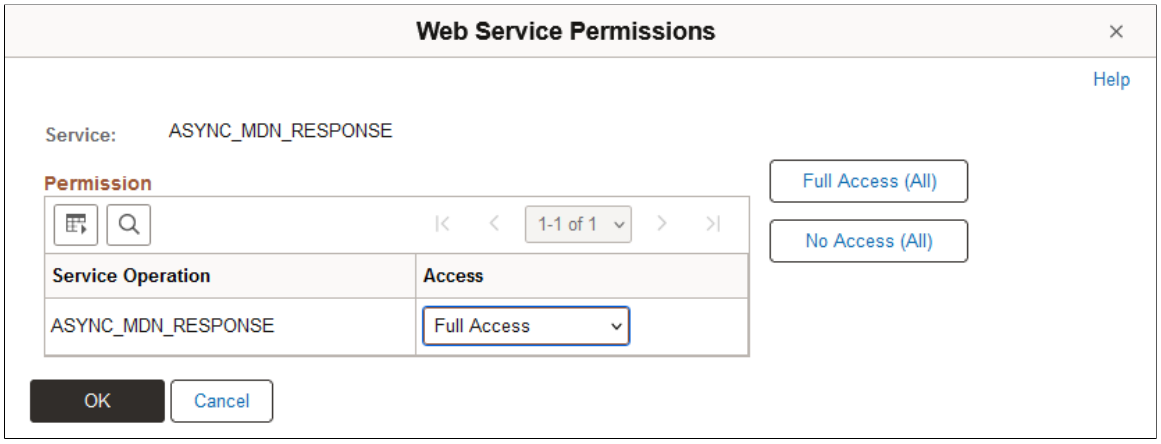
Field or Control	*Description*
Service	Displays the name of the web service defined in the PeopleSoft system.
Edit	Click to launch the Web Service Permissions page.
Full Access (All)	Click to grant full access to all services listed on the page.

Field or Control	*Description*
No Access (All)	Click to set all services listed on the page to *No Access*.

Web Service Permissions

Access the Web Service Permissions page (click the **Edit** link on the Web Services page).

This example illustrates the fields and controls on the Web Service Permissions page.



Field or Control	*Description*
Service Operation	Each operation performed by the web service appears in the **Service Operation** list.
Access	Grant access to the operation by selecting *Full Access*. Deny access by selecting *No Access*. **Note:** By default, the system sets the value to *No Access*. Make sure to modify the access values to reflect the desired level.

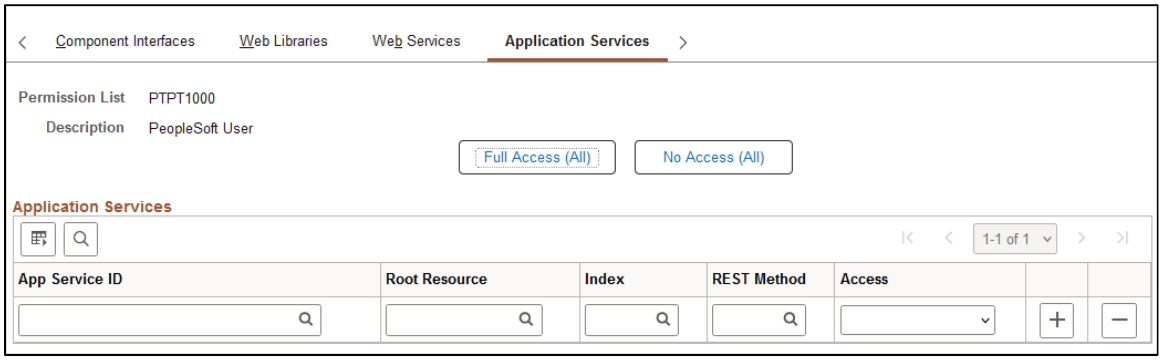
Related Links

"Setting Permissions to Service Operations" (Integration Broker)
"Understanding the PeopleSoft WS-Security for WSRP" (Portal Technology)

Setting Application Services Permissions

Access the Application Services page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the Application Services tab).

This example illustrates the fields and controls on the Permission Lists - Application Services page.



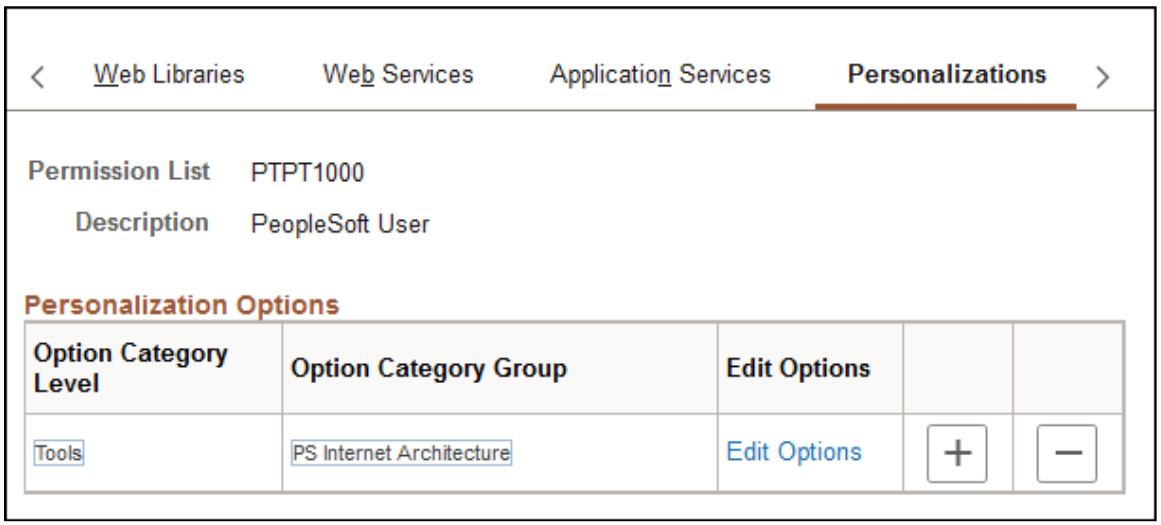
Field or Control	*Description*
Full Access (All)	Click to grant *Full access* to all services listed on the page.
No Access (All)	Click to set all services listed on the page to *No Access*.
App Service ID	Select the ID for an Application Service defined in the PeopleSoft system. See “Searching for Application Service” (Integration Broker).
Root Resource	Select the root resource for the Application Service.
Index	Select the index number for the Application Service.
REST Method	Select the HTTP REST method used by the Application Service.
Access	Grant access to the operation by selecting *Full Access*. Deny access by selecting *No Access*.

See the information on managing Application Services in the *Integration Broker* product documentation.To perform a bulk update for Application Service permissions, see “Setting Application Services Security” (Integration Broker).

Setting Personalization Permissions

Access the Permission Lists - Personalizations page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the Personalizations tab).

This example illustrates the fields and controls on the Permission Lists - Personalizations page.



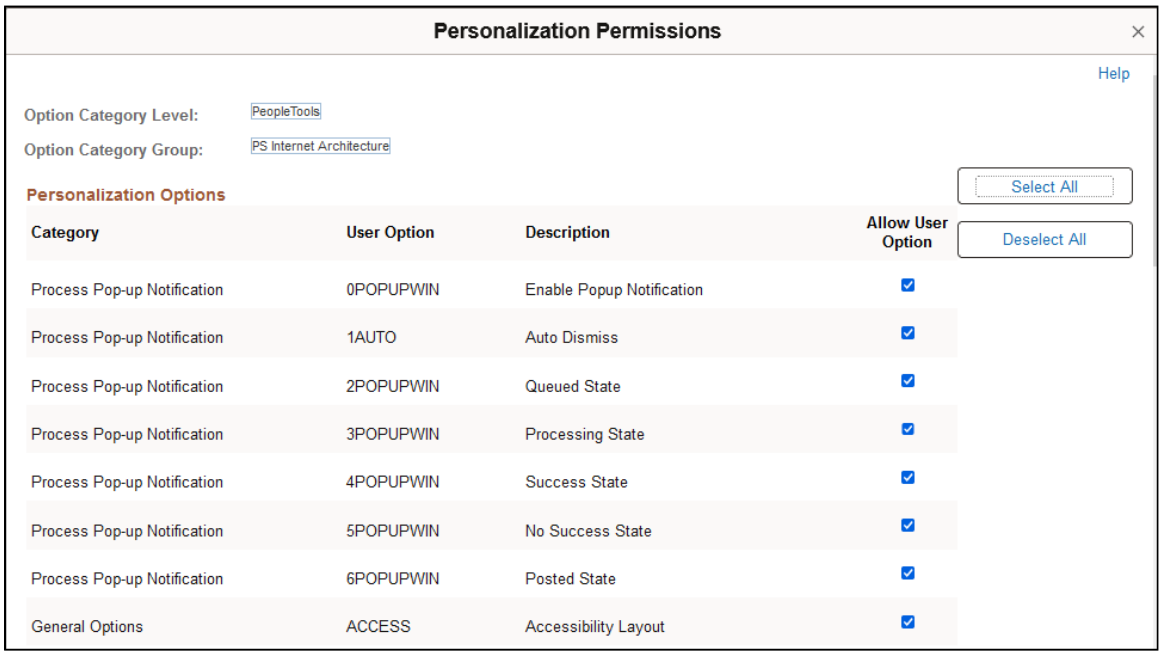
Note: Only those personalization options that accept customization are available for your users to modify.

Field or Control	*Description*
Option Category Level	Displays the high-level grouping of personalizations.
Option Category Group	Shows the further categorizations of personalization options within the category level.
Edit Options	Click to access the Personalization Permissions page and enable specific personalization options for an option category group.

Personalization Permissions

When you click the **Edit Options** link, the Personalization Permissions page appears.

This example illustrates the fields and controls on the Personalization Permissions page.



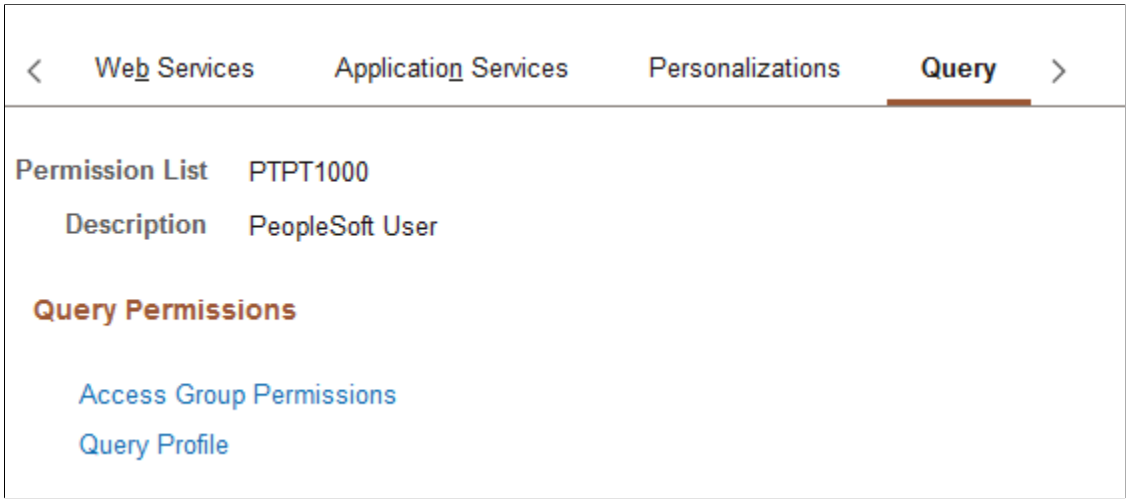
Field or Control	*Description*
Category	Categorizes and encompasses a set of options for the end user. This field is display-only.
User Option	Displays the code associated with the user option, which is the code that the system (PeopleCode) recognizes at runtime. This field is display-only.
Description	Displays the description of the user option. This field is display-only.
Allow User Option	Select this check box to enable the user option.
Select All	Click this button to select the **Allow User Option** check box for each row in the grid.
Deselect All	Click this button to clear the **Allow User Option** check box for every row in the grid.

See Understanding System Personalizations

Setting Query Permissions

Access the Permission Lists - Query page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the Query tab).

This example illustrates the fields and controls on the Permission Lists - Query page.

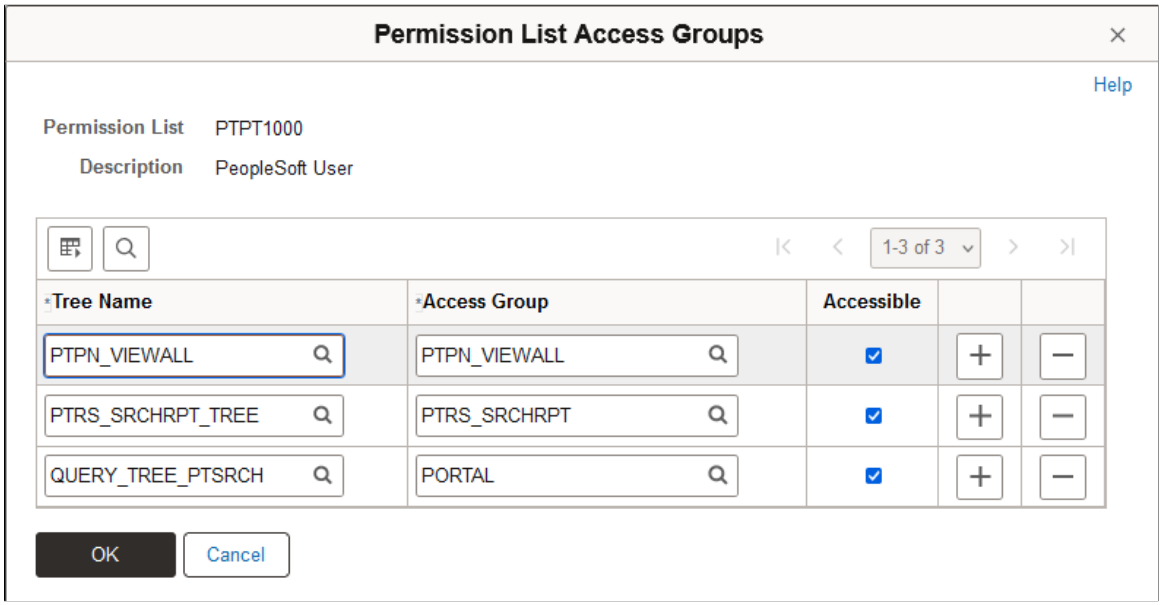


The Query page has links to the Permission List Access Groups page, where you can define the records to which the user can have access in PeopleSoft Query, and the Query Profile page, where you can define the query operations that the user can perform.

Defining Access Groups

Access the Permission List Access Groups page (click the **Access Group Permissions** link on the Permission Lists - Query page).

This example illustrates the fields and controls on the Permission List Access Groups page.



Access groups are nodes in a query tree, which you build with PeopleSoft Query Manager. After you build a query tree, you give users access to one or more of its access groups. Then, they can generate queries on any tables in the access groups accessible to them.

When you open Query Manager, it displays either an access group structure or an alphabetical list of records to which you have access. Access groups enable you to logically organize the record components

to control security access within PeopleSoft Query. This listing is not a physical representation of your database.

You can generate queries on and retrieve information only from the tables whose record definitions are within these access groups. If, for example, you were querying an order table and wanted to display data from a related table (like the customer name rather than the customer code), you must have both tables—the order table and the customer prompt table—in your access groups.

To create new queries, or even to run existing ones, users must have access rights to the record components used in the queries. After you build your query trees, you must grant users access to them. You can grant and restrict access to entire query trees or portions of them through the Access Groups page.

To add an access group to a permission list:

1. Open the permission list and select Query, Access Groups Permissions.
2. Select a tree name.
3. Select the highest access group that the user can access.

   The system displays access groups in the selected query tree only.

   The access group that you select should be the highest-level tree group to which this permission list needs access. The **Accessible** check box is selected by default. For example, users in the ALLPANLS permission list have access to all record components in the EIS_ACCESS_GRP and all access groups below it in the QUERY_TREE_EIS query tree—in other words, to all record components in the tree.
4. (Optional) Deselect the **Accessible** check box.

   To grant access to most of the record components in a high-level access group but restrict access to one of the lower-level groups, you can add a new row for the lower-level access group and deselect the **Accessible** check box. Users can then access all record components within the higher-level group except for those you explicitly made inaccessible.

   Note: Because it hinders system performance, do not deselect the **Accessible** check box for lower-level access groups. To restrict access to record components on a particular branch of a tree, consider creating a new tree for those definitions. Attempting to expand an access group that is not accessible causes all access groups below that access group to be loaded into memory.

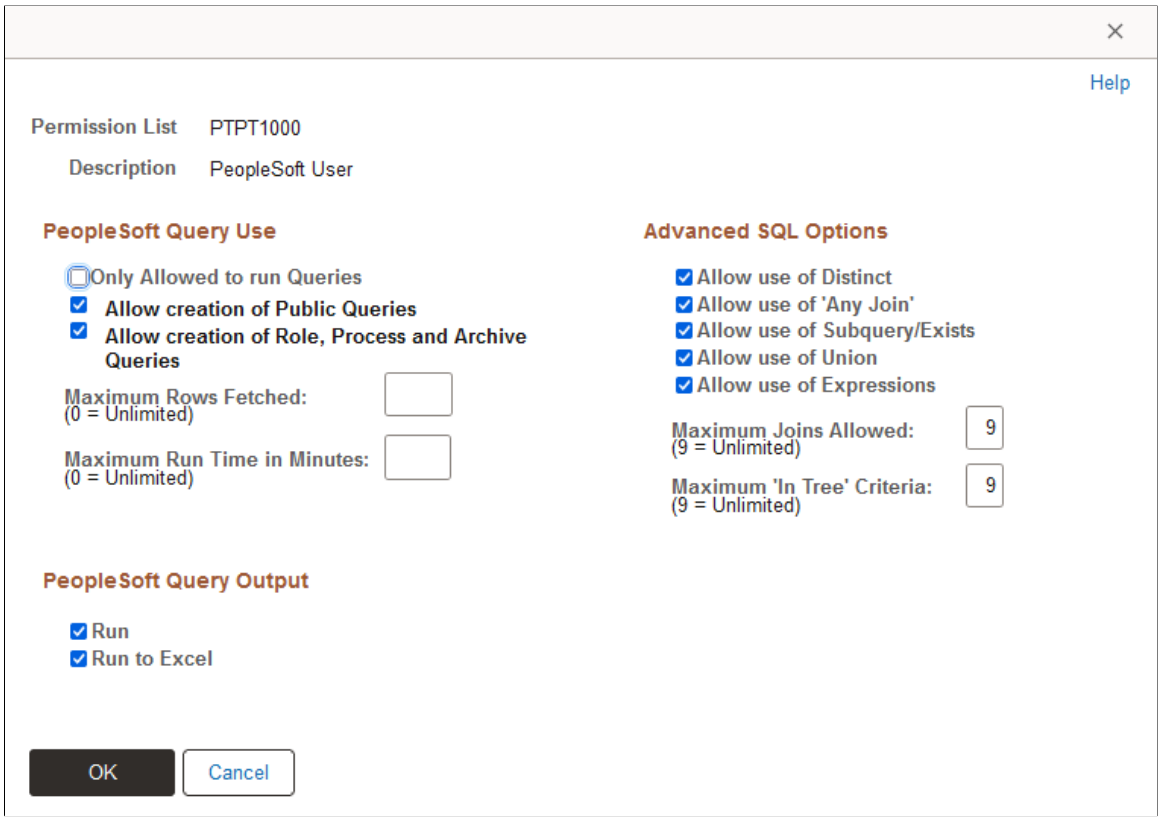
   ---
5. Save your changes.

   Note: When the system loads an access group into memory for the first time, you will likely experience a small delay. This delay is the result of a physical database read for each record component that is associated with that access group. For this reason, do not group a large number of record components into a single access group.

Defining Query Profiles

Access the Query Profile page (click the **Query Profile** link on the Permission Lists - Query page).

This example illustrates the fields and controls on the Query Profile page.



Query profiles specify available query operations. You can give users the right to run queries but not create them, or to create regular queries but not workflow queries, and you can restrict the SQL operations that users can perform. You control these options through the query profile.

Each permission list has its own query profile, and the combination of all permission lists that are assigned to a role determine the total query access for the role. User profiles inherit query access only through the roles that you assign to them.

Note: The first level of security is access to PeopleSoft Query itself. Not every user needs to create queries. You grant access to the Windows client of PeopleSoft Query by selecting the **Query Access** check box on the PeopleTools page of a permission list. You grant access to Query Manager by including the QUERY_MANAGER menu and its related components on the Pages page of a permission list.

You select at least one of the options in the **PeopleSoft Query Use** section of this page to give users query access.

Field or Control	*Description*
PeopleSoft Query Use	Select from: • **Only Allowed to run Queries:** Select to prevent users from being able to create queries and restrict them from running PeopleSoft Query. The values of the remaining options in this group are irrelevant if you select this option. **Note:** If you select this option, it only applies to the current permission list. If a user has permission to create public queries through another permission list, then that user can run *and* create queries against the cumulative set of tables specified through all access groups. For example, assume permission list X has **Only Allowed to run Queries** selected and is limited to tables A, B, and C. Also assume that permission list Y has **Allow creation of Public Queries** selected and is limited to tables B, C, and D. If a user ID has both permission list X and Y associated with it through roles, then that user can create Public Queries with tables A, B, C, and D. • **Allow creation of Public Queries:** Select to enable users to create public queries. • **Allow creation of Workflow Queries:** Select to enable users to create workflow queries in addition to private queries. A workflow query is used in PeopleSoft Workflow, either as a database agent query or a role query. These queries can circumvent security restrictions; the system does not check access group rights while running the query. To make sure that users cannot bypass system security, deselect this check box. • **Maximum Rows Fetched:** Enter a number to restrict the number of rows retrieved by a query. Some queries can return many data rows. For performance or time considerations, you may want users to view only some of those rows rather than all of them. • **Maximum Run time in Minutes**

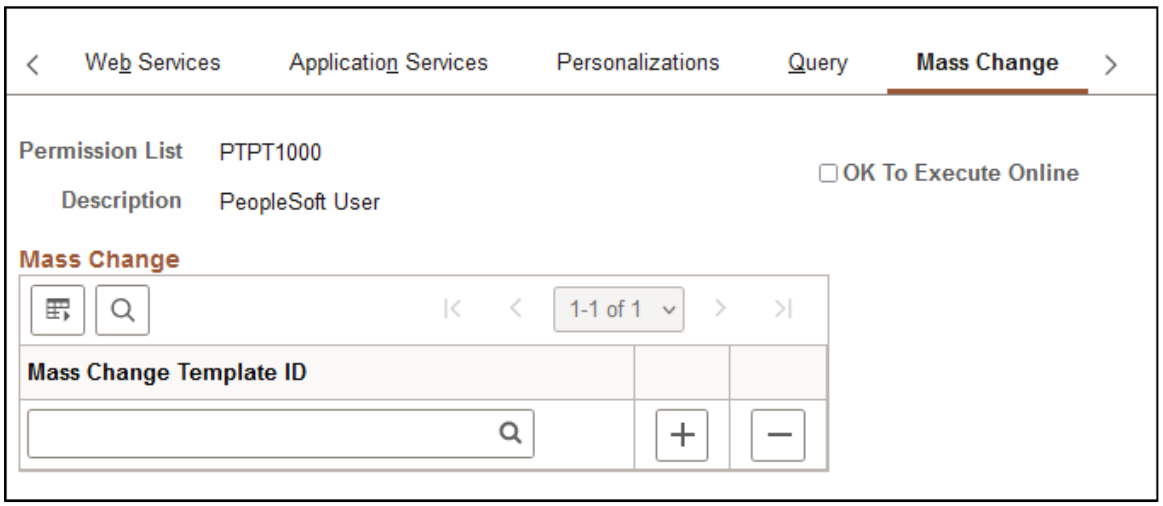
Field or Control	*Description*
PeopleSoft Query Output	Select at least one of these values: • **Run:** PeopleSoft Query displays the query results in a view-only grid control. This option is useful as users are refining their queries. • **Run to Excel:** PeopleSoft Query passes the query results to Microsoft Excel, where users can analyze the results further. **Note:** If using PeopleSoft Query in the Microsoft Windows environment, you grant runtime access through PeopleSoft Navigator by selecting at least one of the PeopleSoft Query output options.
Advanced SQL Options	Restrict less experienced users from generating complex queries, as such queries can affect system performance. Select one or more of these options: • Allow use of Distinct • Allow use of 'Any Join' • Allow use of Subquery/Exists • Allow use of Union • Allow use of Expressions • Maximum Joins Allowed • Maximum 'In Tree Criteria'

Setting Mass Change Permissions

Important! Mass Change is a desupported product. If you used Mass Change in previous PeopleSoft releases, it is strongly recommended that you use Application Engine instead. For more information on PeopleSoft Application Engine, see Application Engine

Access the Permission List - Mass Change page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists**and click the **Mass Change** tab).

This example illustrates the fields and controls on the Permission List - Mass Change page.



Mass change operator security controls:

- What mass change templates a user can access to create new definitions.
- Whether a user can run mass change definitions online.
- What mass change definitions a user can open, view, or run.

  These definitions must also be based on a template with the same PeopleSoft owner as the user.

Note: Users inherit mass change authorizations through their primary permission lists, not through roles.

Before you can use a new template to create definitions, you must have permission to access it.

To modify mass change template permissions:

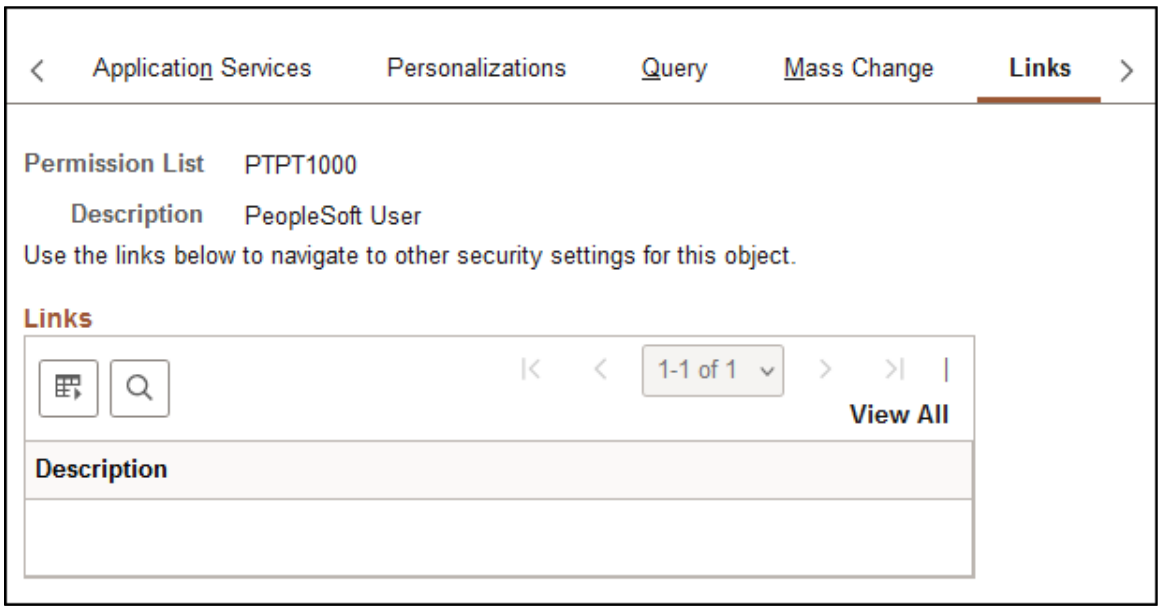
1. Add or remove templates from the **Mass Change Template ID** list.
2. Select or deselect **OK To Execute Online**, as needed.

   When you have enabled the **OK To Execute Online** option, users with the given primary permission list can run mass change definitions after saving any modifications to the Mass Change Definitions pages.
3. Save your work.

Displaying Additional Links

Access the Permission List - Links page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the **Links** tab).

This example illustrates the fields and controls on the Permission Lists - Links page.



Use this page to access links to other pages within your PeopleSoft system. For example, perhaps a PeopleSoft application requires a specific security setting to be associated with a permission list. If this application-specific setting appears on a page not in PeopleTools Security, add a link to the application page so that anyone updating the permission list can easily navigate to it.

Note: The Links page is read-only. You create the inventory of links to pages that exist outside of PeopleTools Security by using the Security Links component (**PeopleTools** > **Security** > **Security Objects** > **Additional Security Links**).

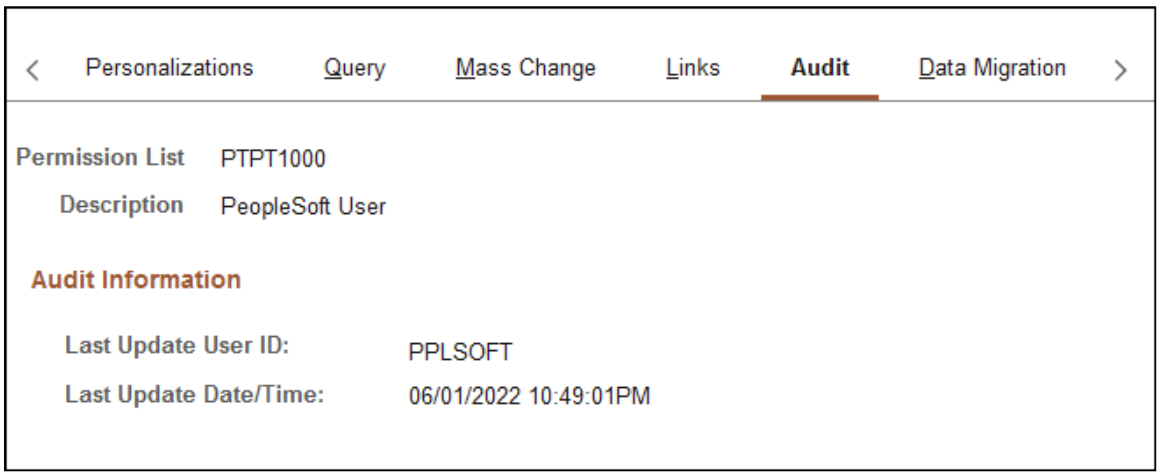
Related Links

Administering Security from Applications

Viewing When a Permission List Was Last Updated

Access the Permission List - Audit page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the **Audit** tab).

This example illustrates the fields and controls on the Permission Lists - Audit page.



Use the Permission Lists - Audit page to view when a permission list was last updated and by whom. You can also view who has made changes to security tables by using the Database Level Auditing feature.

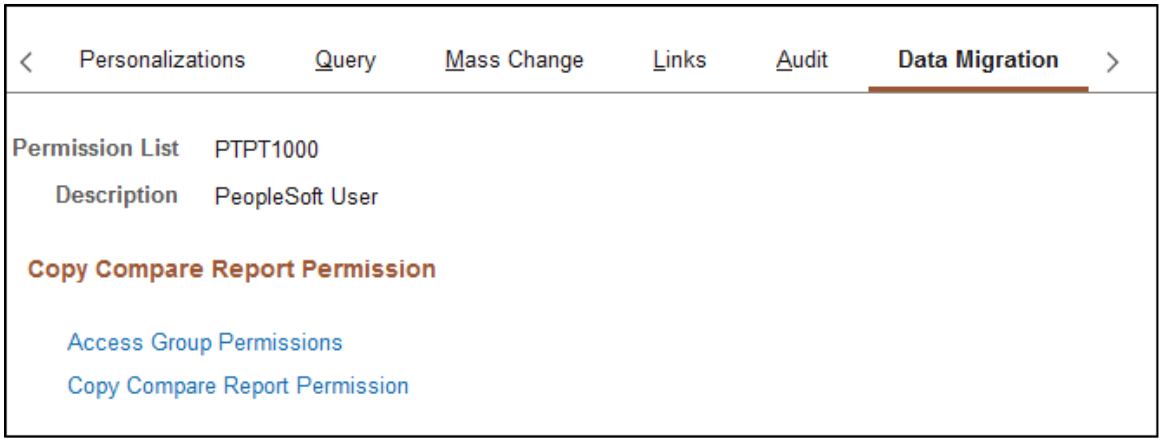
Related Links

"Understanding Database Level Auditing" (Data Management)

Setting Data Migration Permissions

Access the Permission List - Data Migration page (**PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the **Data Migration** tab.

This example illustrates the fields and controls on the Permission Lists - Data Migration page.



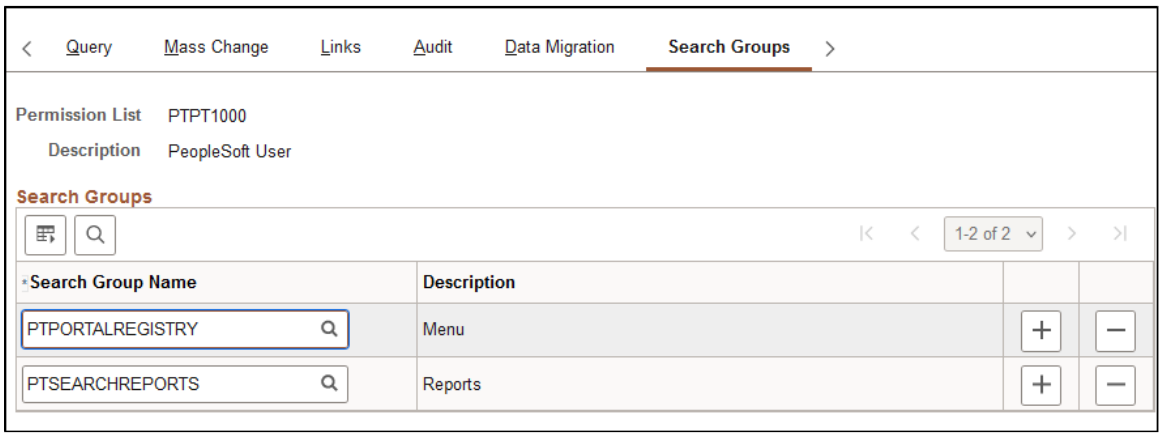
The Data Migration page has links to the Access Group Permissions page, where you can define the records to which the user can have access in the Data Migration Workbench, and the Copy Compare Permissions page, where you can define the copy and compare operations that the user can perform.

For more information about these settings, see "Setting Data Migration Permissions" (Lifecycle Management Guide).

Assigning Search Group Permissions

Access the Search Groups page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the **Search Groups** tab).

This example illustrates the fields and controls on the Permission Lists – Search Groups page.



Use the Permission Lists – Search Groups page to assign search groups to permission lists.

To assign search groups to a permission list:

1. Click the **Search Group Name** lookup button to search for and select a search group to add to the permission list.
2. To specify additional search groups, click the Add button and select a search group for each row that you insert.
3. Click the Save button.

Related Links

"Setting Up Role-Based Search Group Access" (Search Technology)

Working with Definition Security Permissions

Use the Permission Lists – Definition Security page to:

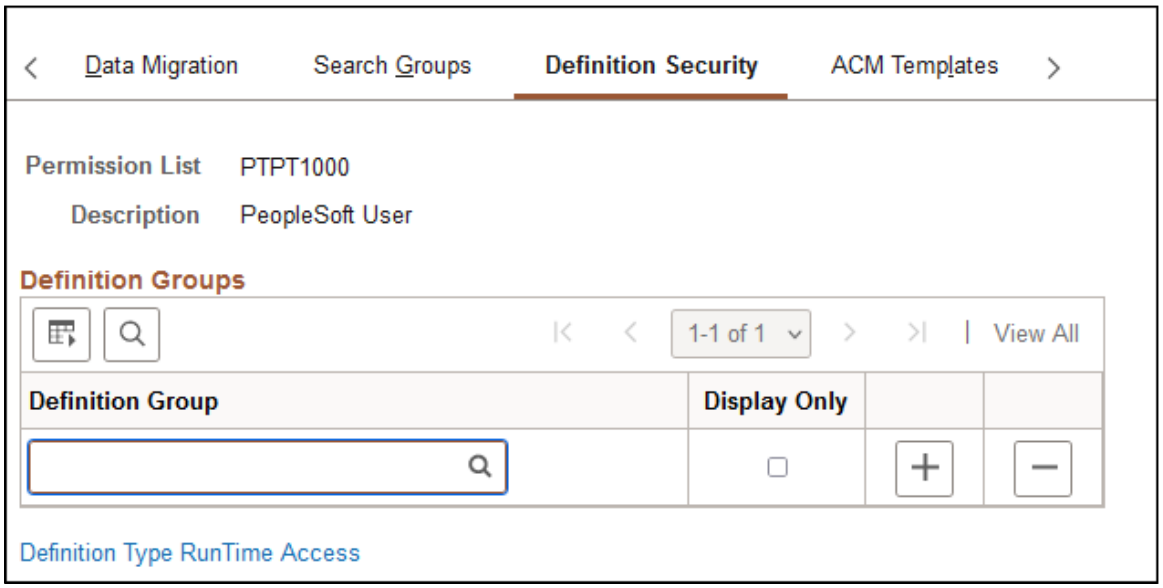
- Add definition groups assigned to a permission list.
- Access the Definition Permissions page to define access to definition types.

Adding Definition Groups to Permission Lists

Use the Definition Security page to assign definition groups to a permission list.

Access the Definition Security page (**PeopleTools** > **Security** > **Permissions and Roles** > **Permissions**, and click the Definition Security tab).

This example illustrates the fields and controls on the Permission Lists – Definition Security page.



Field or Control	*Description*
Permission List	Displays the name of the permission list.
Description	Displays the description of the permission list.
Definition Group	Click the **Lookup** button to search for a definition group to assign to the permission list.
Display Only	Select the box to allow read-only access to the definition group for users belonging to the permission list.
Definition Type RunTime Access	Click the link to set definition type runtime access, which is described later in this topic.

Note that the Group Content Summary – Group Permissions page features similar functionality and enables you to assign permission lists to definition groups.

See Using the Group Permissions Page

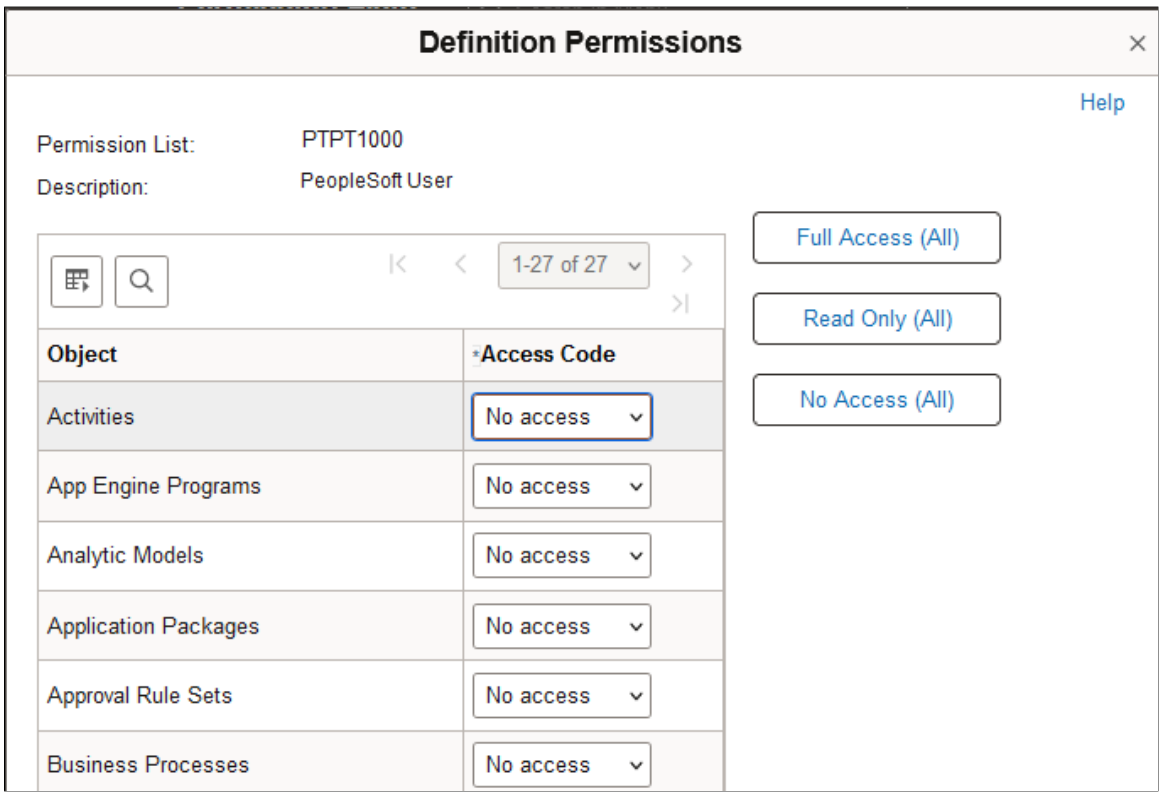
Defining Definition Type RunTime Access

Use the Definition Permission page to define runtime access to definition types for a permission list.

Runtime access is needed so that applications designed to modify or alter managed objects (objects created in Application Designer) can use the GetDefinitionAccess() built-in function to restrict certain end-users from that functionality. With runtime access, applications can be designed to allow modifying Application Designer objects without allowing access to Application Designer.

Access the Definition Permissions page (click the **Definition Type RunTime Access** link on the Permission Lists - Definition Security page).

This example illustrates the fields and controls on the Definition Permissions page.



Grant runtime access to the definitions that developers create using PeopleSoft Application Designer. Each type of definition that you create with PeopleSoft Application Designer appears in the definition permissions list.

Note: On this page, you add permissions to a definition type, such as Application Engine programs. You grant runtime access to *specific* definitions, such as PeopleSoft Payroll Application Engine programs, using Definition Security.

Field or Control	*Description*
Access Code	Select the appropriate access level. Options are: *Full Access:* Definitions of the specified type can be modified. For records, this setting allows access to the Build dialog box. *No Access:* No definitions of the specified type can be opened. *Read-Only:* Definitions of the specified type can be opened and viewed, but not modified. *Update translates only:* This level applies only to fields. This setting allows a user to modify only Translate table values. *Data admin only:* This level applies only to records. It allows a user to modify only those record attributes found in the Tools, Data Administration menu (tablespaces, indexes, and record DDL). **Note:** Projects cannot be set to *No access* because Application Designer requires access to a project to launch successfully.
Full Access (All), Read Only (All), and **No Access (All)**	Click to set all definition types in the list to the same access level.

Note: If change control locking is enabled, the Change Control access setting on the Tools Permissions page can override object types settings.

See “Building and Maintaining Data” (Application Designer Developer’s Guide), Understanding Definition Security (Windows Client).

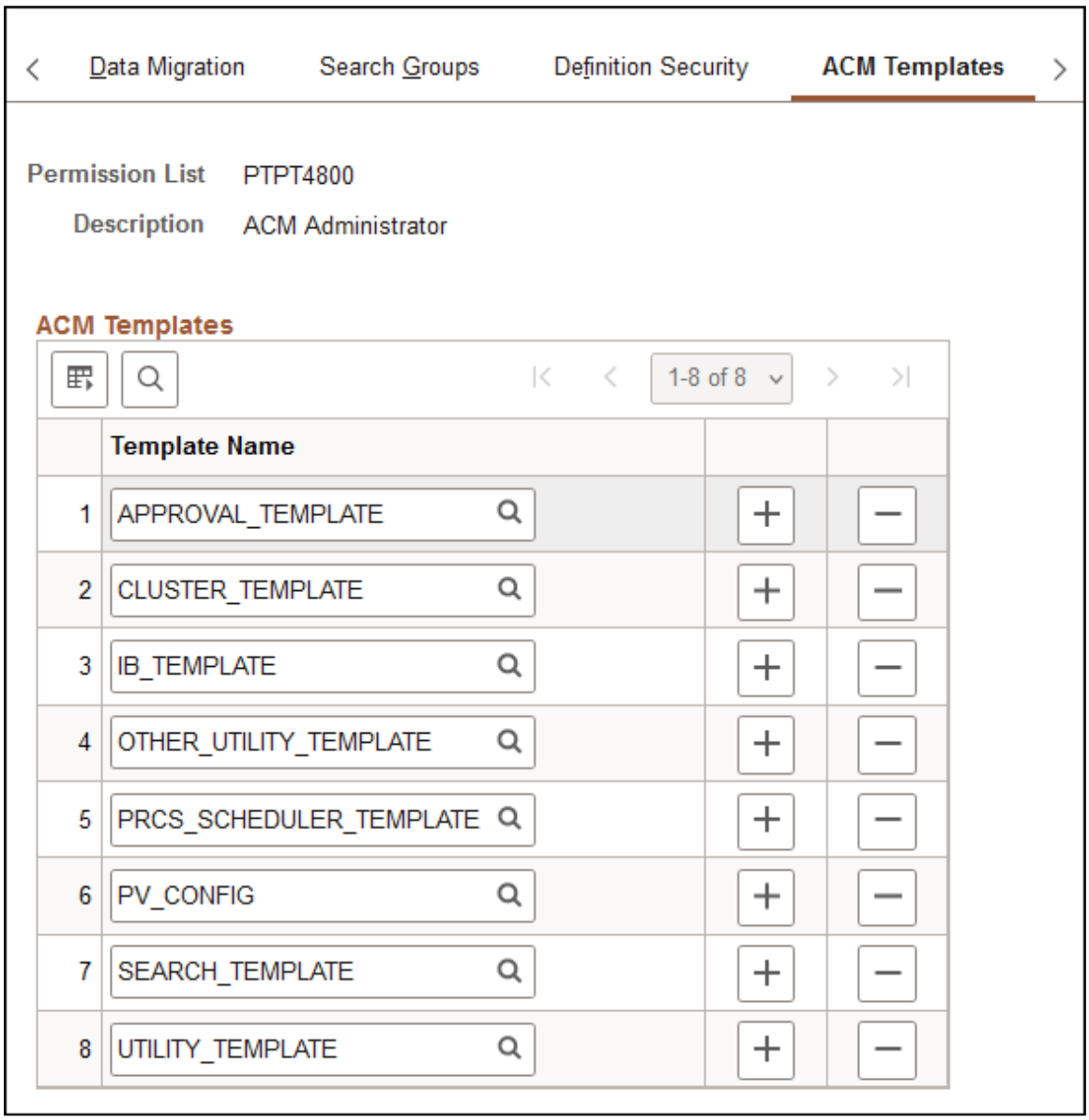
Adding Permissions Lists to ACM Templates

Use the Permission Lists – ACM Templates page (PTACM_ACCESS) to define access to automated configuration management (ACM) templates to the PTPT4800 permission list.

Note: You can add access to ACM templates only to the PTPT4800 permission list.

To access the page select **PeopleTools** > **Security** > **Permission and Roles** > **Permission Lists** and click the ACM Templates tab.

This example illustrates the fields and controls on the Permission Lists – ACM Templates page.



Field or Control	*Description*
Permission List	Displays the name of the permission list.
Description	Displays the description of the permission list.
Template Name	Click the **Lookup** button to search for an ACM template to assign to the permission list.

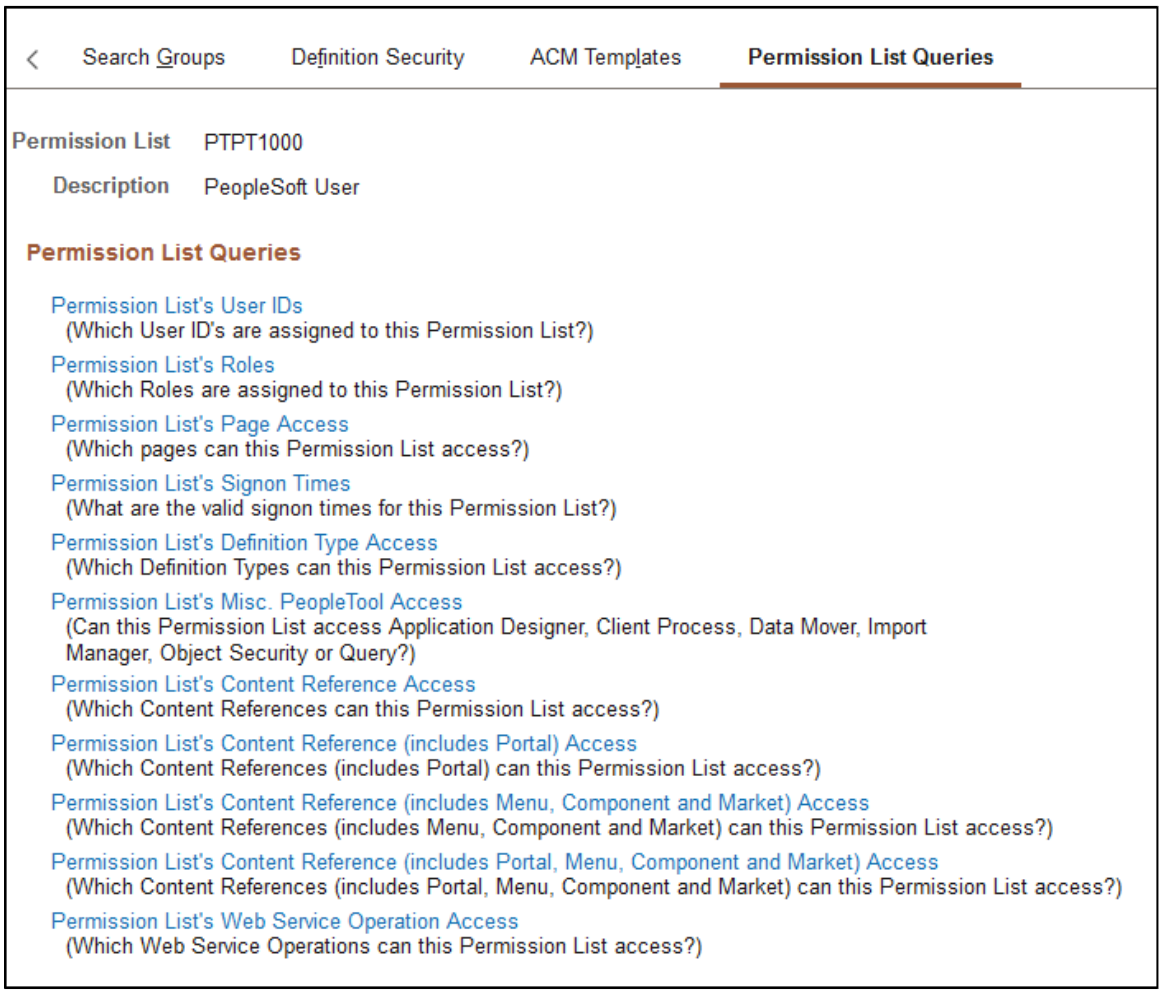
Related Links

“Working with Custom Templates” (Automated Configuration Management)

Running Permission List Queries

Access the Permission List Queries page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the Permission List Queries tab).

This example illustrates the fields and controls on the Permission List - Permission List Queries page.



Permission list queries provide detailed information regarding a permission, such as the user IDs and roles that are associated with a permission list. The available queries are documented on the page.

To run permission list queries:

1. Click the link associated with the query that you want to run.

   A new browser window opens.

2. View the information the query returns or click a download results link.

   Note: The size of the file appears in parentheses beside the download options.

   For downloading, you have the following options:

   - Microsoft Excel spreadsheet.

Downloads the query results as a Microsoft Excel spreadsheet (.xls) file.

- CSV text file.

  Downloads the query results as a Comma-Separated Value (.csv) file.

- XML file.

  Downloads the query results as an Extensible Markup Language (.xml) file.

Chapter 4

Setting Up Roles

Understanding Roles

Roles are an intermediate object that exist between permission lists and user profiles. Roles aggregate permission lists so that you can arrange permissions into meaningful collections.

Note: In previous releases, roles were associated with PeopleSoft Workflow. PeopleTools has expanded role definitions so that they are also a part of the security architecture. There is only one type of role definition, and you maintain it within Security.

Users inherit most of their permissions from the roles assigned to the user profile. However, you assign the following permission lists directly to a user profile:

- Data permissions.

  These are assigned through a primary permissions list or a row security permissions list.

- PeopleSoft Navigator homepage permissions.

- Process profile permissions.

When you assign roles to profiles manually, through the Security pages, these users are static role members.

Other users may obtain membership in a role programmatically. You can run a batch process that uses predefined role rules and assigns roles to user profiles according to these rules. Users who become members of a particular role programmatically are dynamic role members.

Use dynamic role assignment to make your security system scale to large user populations. If you have thousands of users and need to make every change to a user profile manually, the security administrator becomes a bottleneck. If you implement dynamic roles, you reduce administrative tasks.

Managing Roles

This section discusses how to:

- Copy roles.

- Delete roles.

- Remove users from roles.

Copying Roles

To copy a role:

1. Select **PeopleTools** > **Security** > **Permissions and Roles** > **Copy Roles**.
2. On the search page, locate and select the role that you want to copy (clone).

   The Role Save As page appears.
3. On the Role Save As page, enter a new name in the **as:** edit box.
4. Click **Save.**

Deleting Roles

To delete a role:

1. Select **PeopleTools** > **Security** > **Permissions and Roles** > **Delete Roles**.
2. On the search page, locate and select the role that to delete.

   The Delete Permission List page appears.
3. Click **Delete Permission List.**
4. Click **OK** to confirm the deletion, or click **Cancel** to cancel the deletion.

Note: If you attempt to delete a role definition that is currently in use by one or more static or dynamic role users, you must confirm deletion of the role definition. When you confirm, you remove all references to the role.

Removing Users From Roles

To delete the users who are assigned dynamically, use the NO_USERS query to locate the users. You invoke this query using the query rule with dynamic roles.

Related Links

Displaying Dynamic Role Members

Defining Role Options

This section discusses how to:

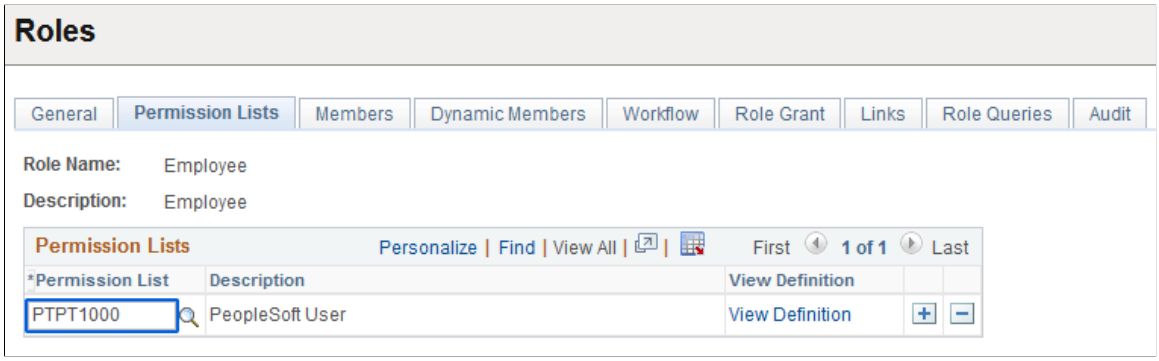
- Assign permissions to roles.
- Display static role members.
- Display dynamic role members.
- Set user routing options.

- Decentralize role administration.
- Display additional links for user profiles.
- Run role queries.
- View when a role was last updated.

Assigning Permissions to Roles

Access the Roles - Permission Lists page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Roles** and click the **Permission Lists** tab).

This example illustrates the fields and controls on the Roles - Permission Lists page.



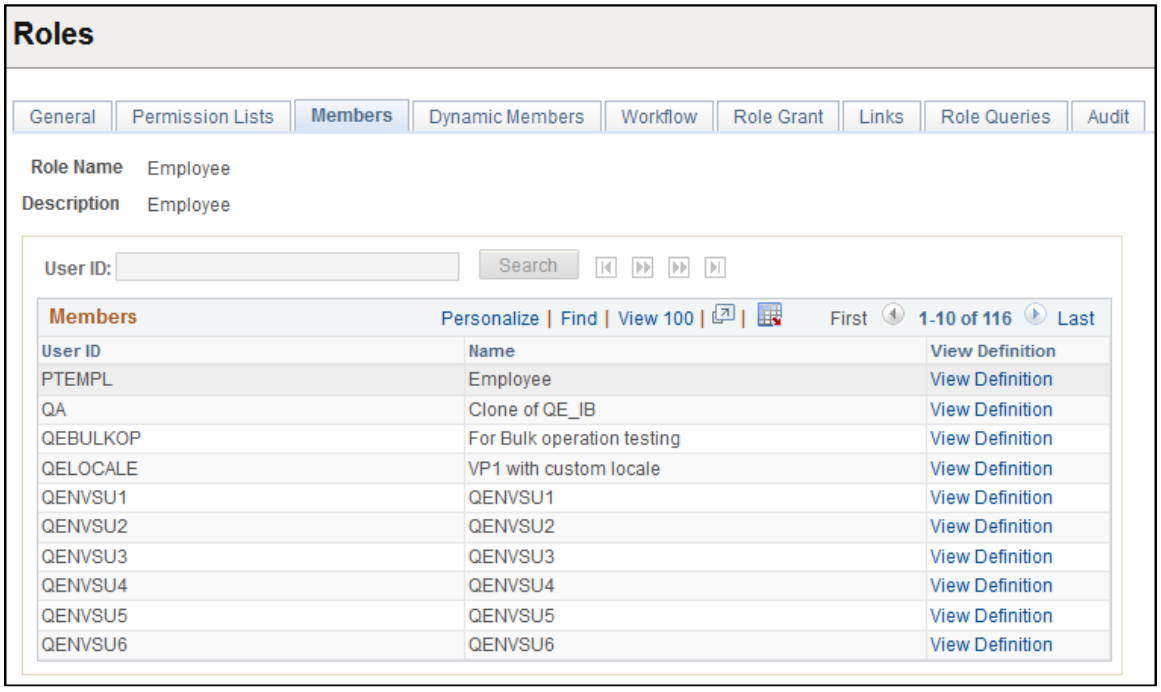
To add new permission lists to a role, add more rows. Remember that a user's access is determined by the sum of all the permission lists applied to each role to which the user belongs. For instance, suppose you add permission list X and permission list Y to a role. Permission list X has a sign-on time of 8 a.m. to 5 p.m. and permission list Y has a sign-on time of 1 p.m. to 9 p.m. In this scenario, the users assigned to this role can sign in to the system from 8 a.m. to 9 p.m. Always be aware of the contents of each permission list before adding it to a role.

Field or Control	*Description*
View Definition	Click to open the permission list definition, where you can view the options in the permission to ascertain whether it is suitable for a particular role.

Displaying Static Role Members

Access the Members page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Roles** and click the **Members** tab).

This example illustrates the fields and controls on the Roles - Members page.



If your database contains more than 1000 role members, this page initially retrieves only the first 1000. You can view the other chunks of 1000 members one chunk at a time, either by searching for a user ID or by using the navigation buttons above the **Members** grid. The navigation buttons enable you to display the first chunk, the previous chunk, the next chunk, or the last chunk.

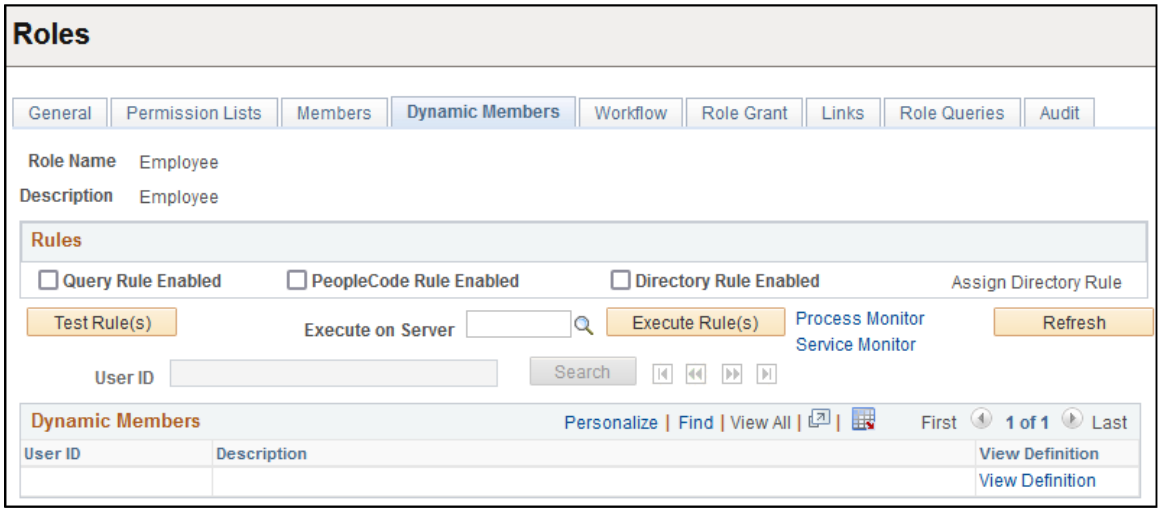
Note: The navigation buttons, the User ID field, and the Search button are available for use when there are more than 1000 role members in the grid.

Field or Control	*Description*
User ID	Enter part or all of a role member user ID for which to search.
Search	Click to search through the role members for the first chunk of rows that contains the user ID you entered.
View Definition	Click to view the user ID of the role member to ensure that you selected the appropriate definition for inclusion in the role.

Displaying Dynamic Role Members

Access the Roles - Dynamic Members page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Roles** and click the **Dynamic Members** tab).

This example illustrates the fields and controls on the Roles - Dynamic Members page.



Use this page to set the rule to invoke to assign roles. A dynamic role rule is defined or coded in PeopleSoft Query, PeopleCode, or your Lightweight Directory Access Protocol (LDAP) directory. A rule can use a combination of PeopleSoft Query and PeopleCode, or PeopleSoft Query and LDAP. For the rule to successfully assign a role to the appropriate users, you must select the rule type you have in place for a particular role and then specify the object that contains the rule you coded.

Note: You must define your role rules before you apply the options on this page. If you change the name of the rule, add a new rule, and so on, save all changes before you run the rule.

If your database contains more than 1000 dynamic role members, this page initially retrieves only the first 1000. You can view the other chunks of 1000 dynamic members one chunk at a time, either by searching for a user ID or by using the navigation buttons above the **Dynamic Members** grid. The navigation buttons enable you to display the first chunk, the previous chunk, the next chunk, or the last chunk.

Note: The navigation buttons, the User ID field, and the Search button are available for use when there are more than 1000 role members in the grid.

Field or Control	*Description*
User ID	Enter part or all of a role member user ID for which to search.
Search	Click to search through the role members for the first chunk of rows that contains the user ID you entered.
View Definition	Click to view the user ID of the role member to ensure that you have selected the appropriate definition for inclusion in the role.
Query Rule Enabled	Select if you defined your rule with PeopleSoft Query. The **Query Rule** group box appears below the **Rules** group box. Use the **Query** drop-down list box to select the query that contains your role rule.

Field or Control	*Description*
PeopleCode Rule Enabled	Select if your rule is a PeopleCode program. The **PeopleCode Rule** group box appears. Specify the record, field, event, and function associated with your PeopleCode role rule.
Directory Rule Enabled	Select if your role rule is based on information in your directory server. With a directory-based rule, you must assign directory groups. The **PeopleCode Rule** group box appears because directory rules are implemented using the DynRoleMembers PeopleCode program. This program uses the Directory business interlink to retrieve user and group information from the directory. To view the program, open the FUNCLIB_LDAP record in PeopleSoft Application Designer. Click **Assign Directory Groups** to select a particular directory group that exists in your LDAP server hierarchy. For example, if your directory server is grouped by geographic region, then your rule could assign a new self-service role to all users in the North America group. Use the **Directory Group** drop-down list box to select the appropriate directory group value. The values are derived from the LDAP data that you import using the Directory Group Import process.
Execute on Server	Select the appropriate PeopleSoft Process Scheduler server to run the rule.
Execute Rule(s)	The Execute Dynamic Role Rules button on this page launches the DYNROLE_PUBL application engine program which executes the rule(s). The application engine program runs through PeopleSoft Process Scheduler. After the DYNROLE_PUBL application engine program runs the rule, it publishes a message, ROLESYNCHEXT_MSG, that contains a list of users in the role. **Note:** The successful completion of the dynamic roles program does not ensure that your roles were updated; the associated message must also be published successfully. Use the Process Monitor link on the page to view the status of the application engine program. Use the Service Monitor link on the page to view the status of the message publication. You can also execute dynamic role rules for all roles and users. See Executing Dynamic Role Rules.
Refresh	After you run a rule, click to repopulate the grid with updated information.
Process Monitor	Click to view the status of the DYNROLE_PUBL application engine program in the Process Scheduler Monitor.

Field or Control	*Description*
Service Monitor	After the DYNROLE_PUBL application engine program runs, it publishes a message, ROLESYNCHEXT_MSG, that contains a list of users in the role. Click the link access the Service Operations Monitor and to view the publication status of the ROLESYNCHEXT_MSG message.

Note: To clear all dynamic users from the role, run the delivered NO_USERS query.

Query Rule Example

This section describes the process of creating a PeopleSoft Query rule that assigns dynamic role membership. This example should also help to illustrate similar techniques that you would use for a PeopleCode or LDAP rule.

Note: This example assumes a working knowledge of PeopleSoft Query.

In this example, you need to find all users who currently have job code KC012 (Human Resource Analyst) and add them to the appropriate role.

To create this rule:

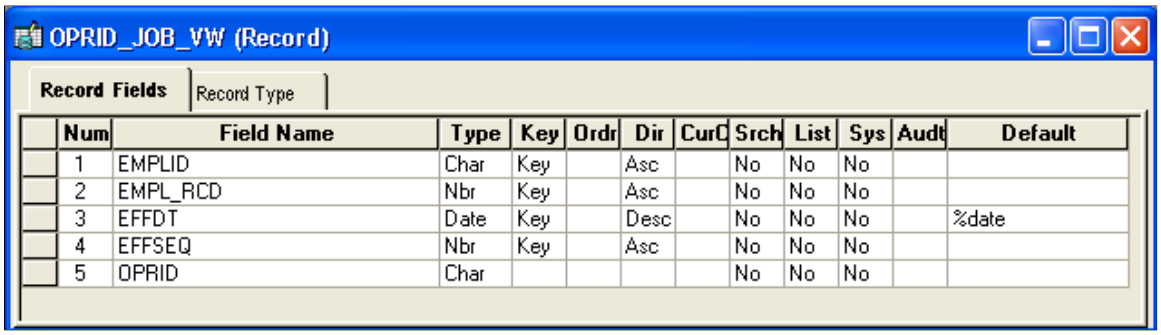
1. Create a view.
2. Create the query.
3. Run the dynamic rule.

Note: The Dynamic Role functionality is not designed to resolve bind variables. When you select a query with a bind variable as a dynamic role rule, the system issues an error. Do not use queries with bind variables as a query rule for dynamic roles. Many of the delivered queries are intended to be used with PeopleSoft Workflow, and many of them contain bind variables. These queries are not designed to work as role rules, but you can modify them to do so.

Note: To create a role query based on PSOPRALIAS and avoid issues with row-level security, use PSOPRALIAS_VW instead. You must manually synchronize this view with PSOPRALIAS.

Note: If the query returns duplicate user IDs, dynamic roles will fail on the insert into PSROLEUSER and may have mixed results. You should add a DISTINCT clause to your query role rule to return unique IDs, especially when your query involves thousands of user IDs.

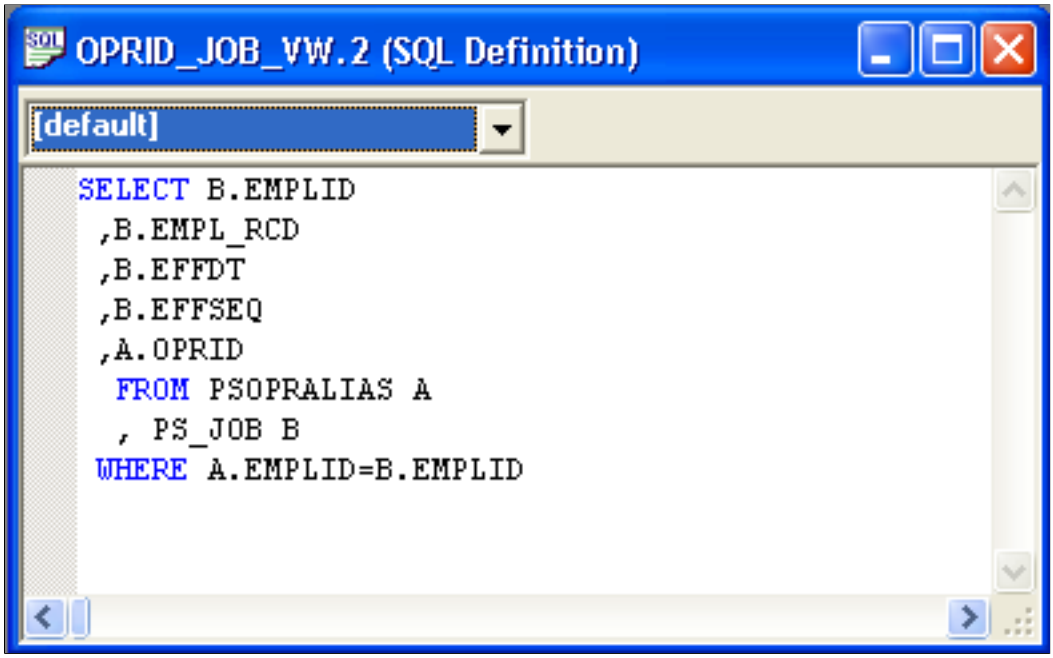
This example shows a possible view definition for the example role rule:



Num	Field Name	Type	Key	Ordr	Dir	CurC	Srch	List	Sys	Audt	Default
1	EMPLID	Char	Key		Asc		No	No	No		
2	EMPL_RCD	Nbr	Key		Asc		No	No	No		
3	EFFDT	Date	Key		Desc		No	No	No		%date
4	EFFSEQ	Nbr	Key		Asc		No	No	No		
5	OPRID	Char					No	No	No		

Review the associated SQL definition.

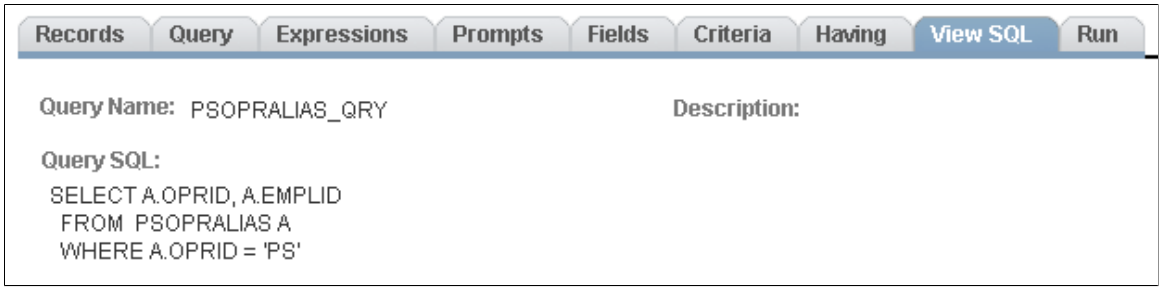
This example illustrates the associated SQL definition.



Note: The OPRID field must not be a key in this view because PeopleTools appends AND OPRID = “<CURRENT_USER_ID>” in PeopleSoft Query Manager. This action occurs if you use the record OPRALIAS directly in the query.

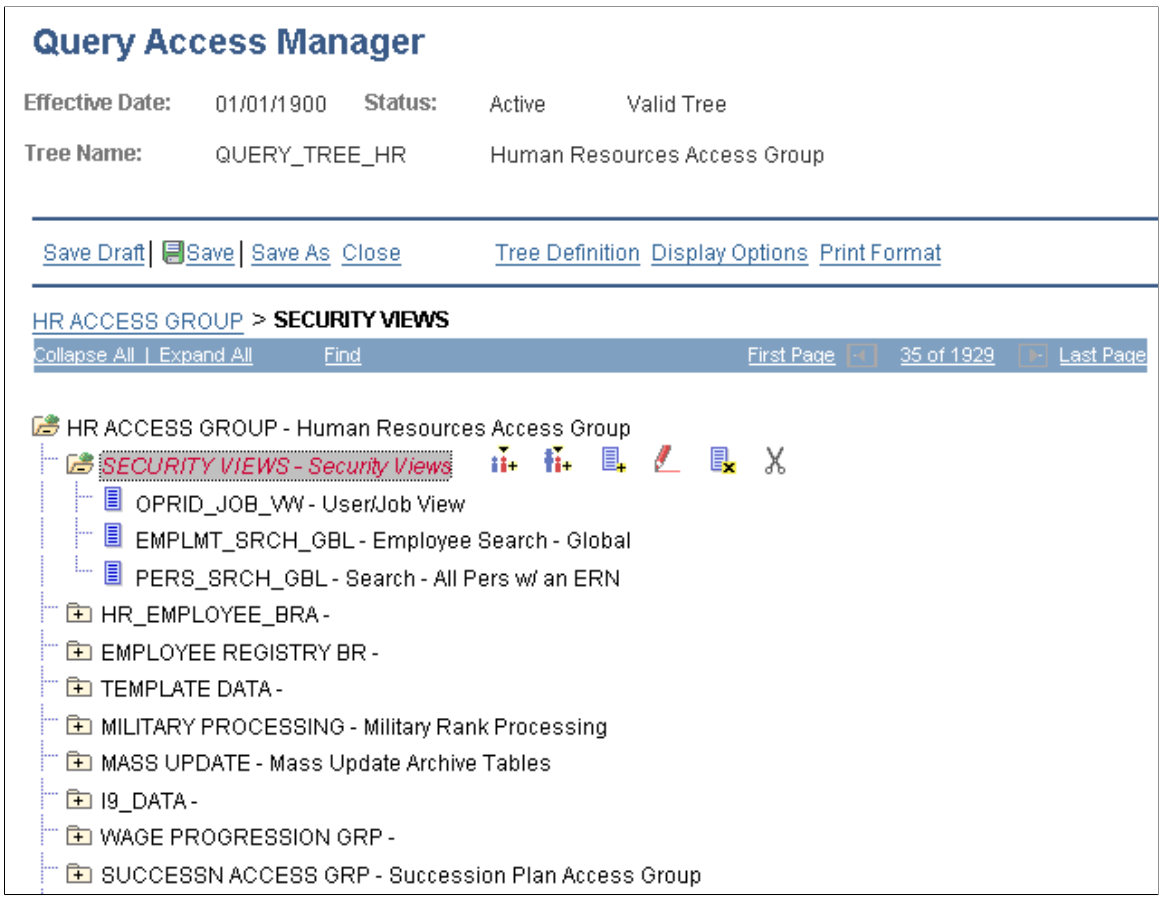
Review the Query view SQL on the View SQL page.

This example illustrates the Query view SQL.



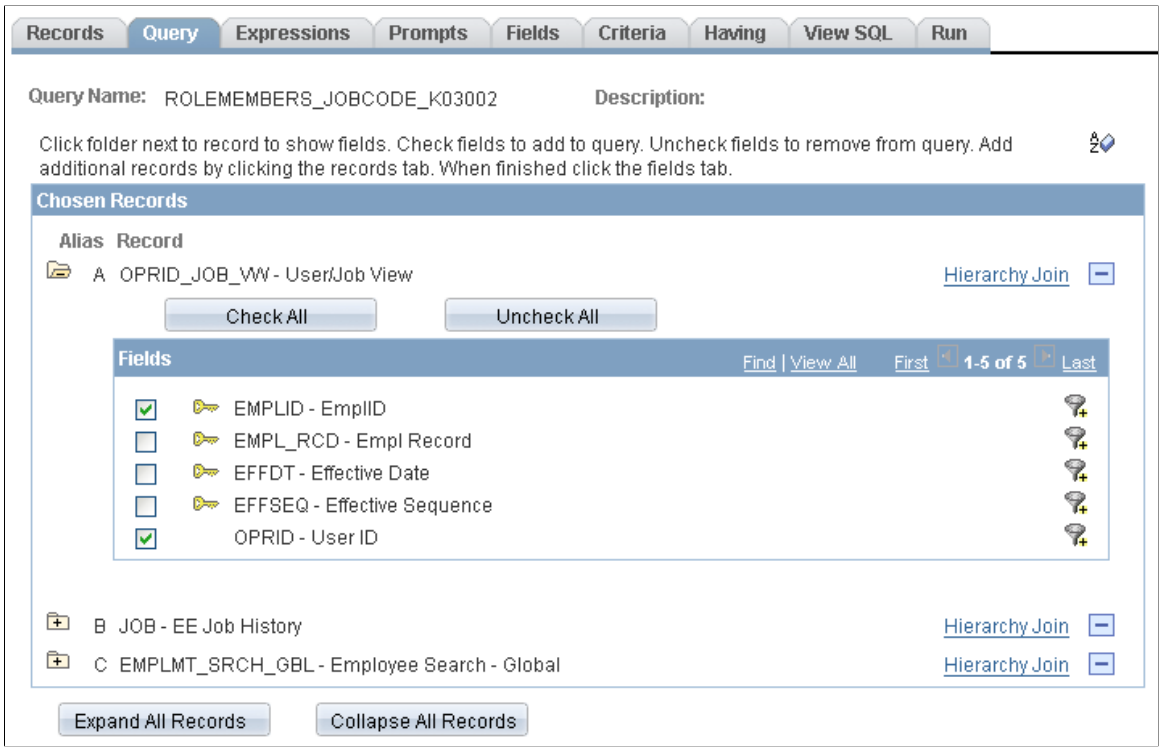
After you create the view, add it to the appropriate query tree. In this case, you add the new view to the QUERY_TREE_HR:

This example illustrates adding the view to a query tree.



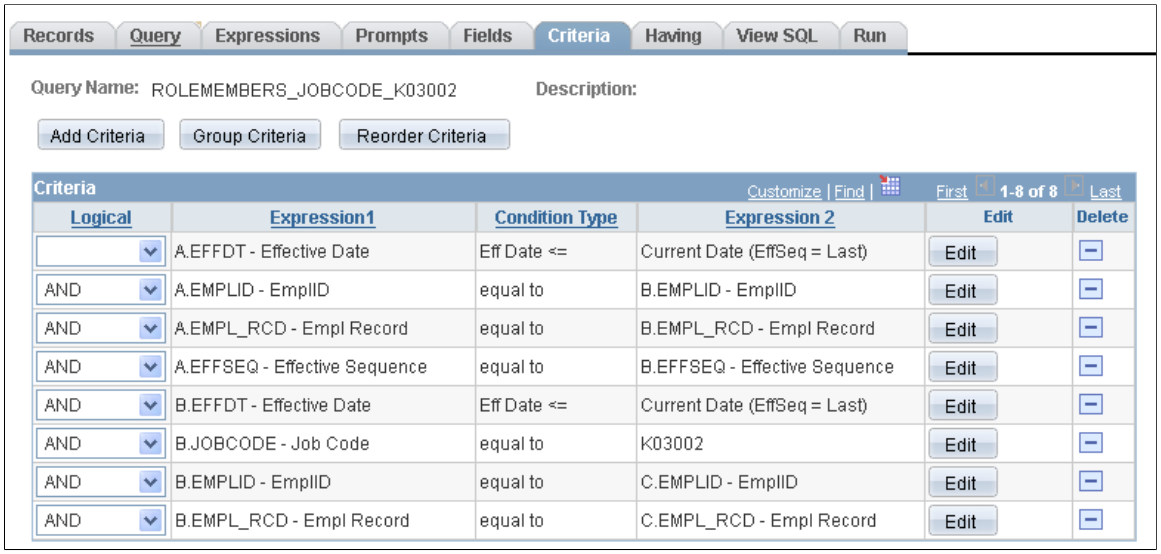
After you create the view, you create a query. In this example, the properties assigned to the query enable it to assign a role to users who currently have the job code K03002, Human Resource Analyst.

This example shows the query properties.



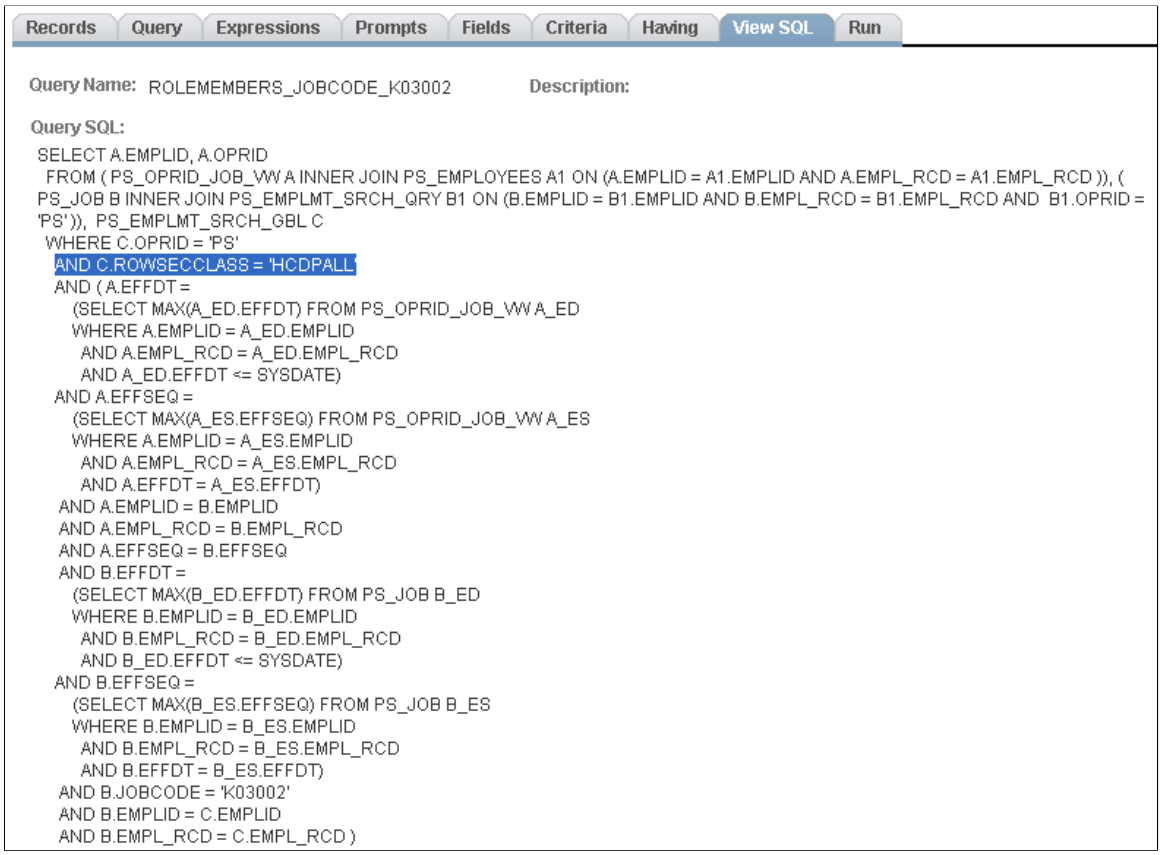
Review the query criteria.

This example illustrates a sample Criteria page.



Review the SQL for the query.

This example illustrates the query SQL.



Because the view does not have OPRID as a key, the resulting SQL does not contain the extra line `AND B.OPRID = PS.`

Note: When you save a query used for a dynamic role query, you should specify that it is a role query.

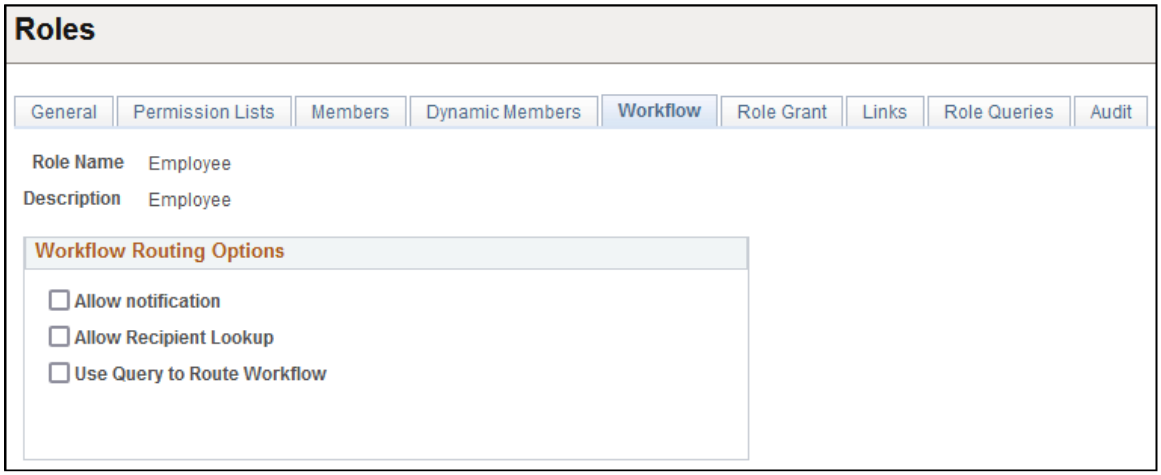
With the view and the query created, you then set up the query rule on the Roles - Dynamic Members page. Select **Query Rule Enabled** and select the query in the **Query** field.

After enabling the query rule, test the rule to make sure the system assigns the appropriate roles to the appropriate users. To populate the role membership table, click **Execute Rule.**

Setting User Routing Options

Access the Roles - Workflow page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Roles** and click the **Workflow** tab).

This example illustrates the fields and controls on the Roles - Workflow page.



Field or Control	*Description*
Allow notification	Select to enable PeopleSoft Workflow notification. Users can notify others of data on a PeopleSoft page through email or worklists. When components are designed, developers can enable the Notify toolbar on the Component Properties dialog box in PeopleSoft Application Designer. If this option is set for a particular component, then this check box enables security administrators to enable the Notify feature per role.
Allow Recipient Lookup	Select to enable role users to browse the database for the email addresses of other users in the PeopleSoft system, such as vendors, customers, employees, sales leads, and so on. This check box is available only if the **Allow notification** check box is selected.
Use Query to Route Workflow and **Query Name**	Select to determine workflow routings by a workflow query. This value depends on your workflow scheme. If this option is selected, the **Query Name** field appears, where you specify the query to use.

Decentralizing Role Administration

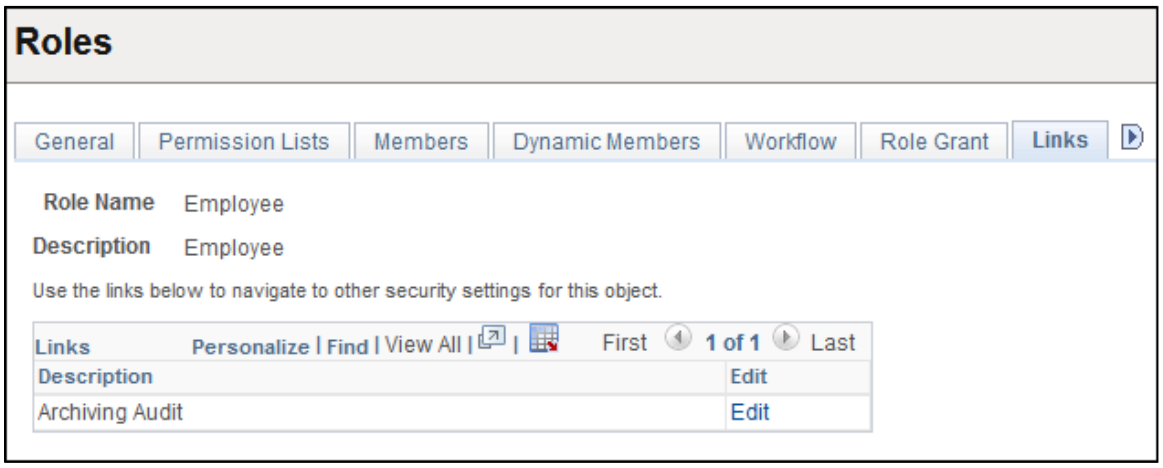
You use the Roles – Role Grant page to assign limited security administration capability to specified users. You designate them as *remote security administrators* by defining roles that they can grant to other users. Because the settings on this page are part of the implementation of *distributed user profiles*, the page is documented along with the Distributed User Profiles component.

See Implementing Distributed User Profiles.

Displaying Additional Links

Access the Roles - Links page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Roles** and click the Links tab).

This example illustrates the fields and controls on the Roles - Links page.



Use this page to access links to other pages within your PeopleSoft system. For example, perhaps a PeopleSoft application requires a specific security setting to be associated with a role. If this application-specific setting appears on a page not in PeopleTools Security, add a link to the application page so that anyone updating the role can easily navigate to the page.

Note: The Links page is read-only. You create the inventory of links to pages that exist outside of PeopleTools Security by using the Security Links component.

If you have added any links for roles in the Security Links component, they appear on the Links page.

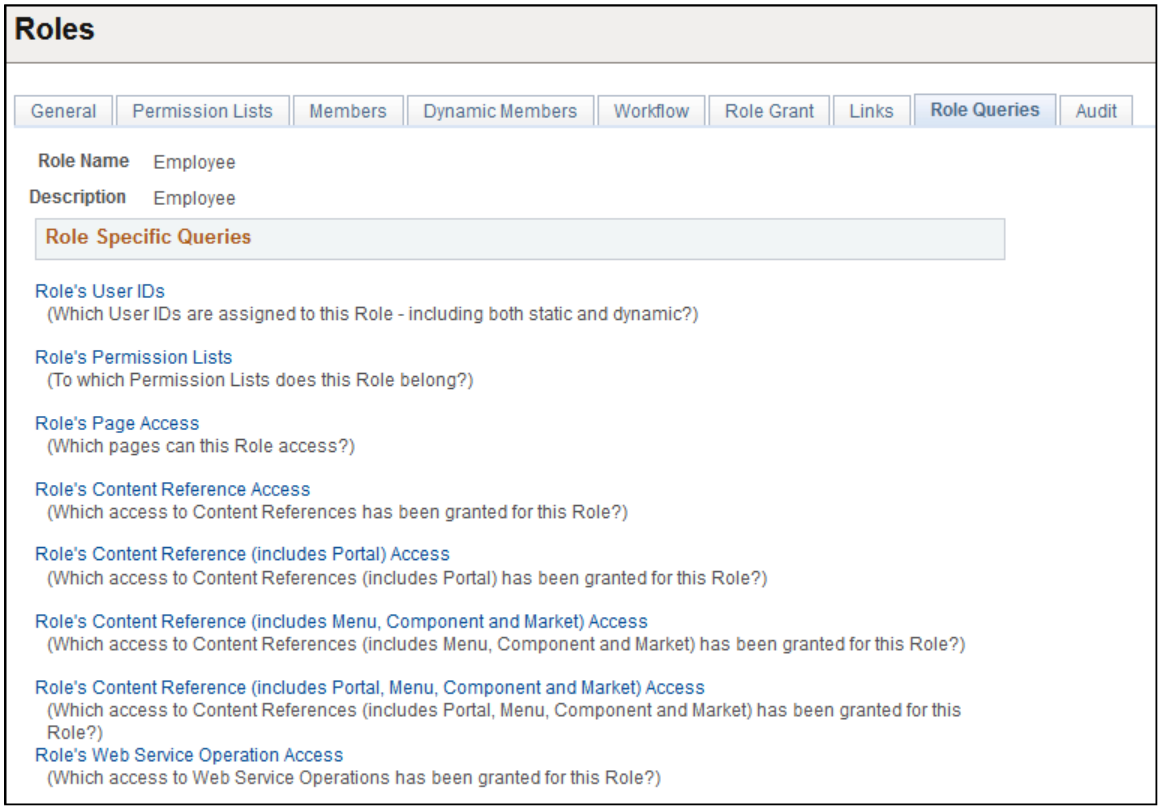
Related Links

Administering Security from Applications

Running Role Queries

Access the Roles -Role Queries page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Roles** and click the **Role Queries** tab).

This example illustrates the fields and controls on the Roles - Role Queries page.



Use role queries to provide detailed information about a role, such as the user IDs and permission lists associated with the role. The available queries are documented on the Roles - Role Queries page.

To run a role query:

1. Click the link associated with the query that you want to run.

   This action invokes a new browser window.

2. View the information the query returns or click a download results link.

   Note: The size of the file appears in parentheses next to the download options.

   The download options are:

   - Microsoft Excel spreadsheet

     Downloads the query results as a Microsoft Excel spreadsheet (.xls) file.

   - CSV text file

     Downloads the query results as a comma-separated values (.csv) file.

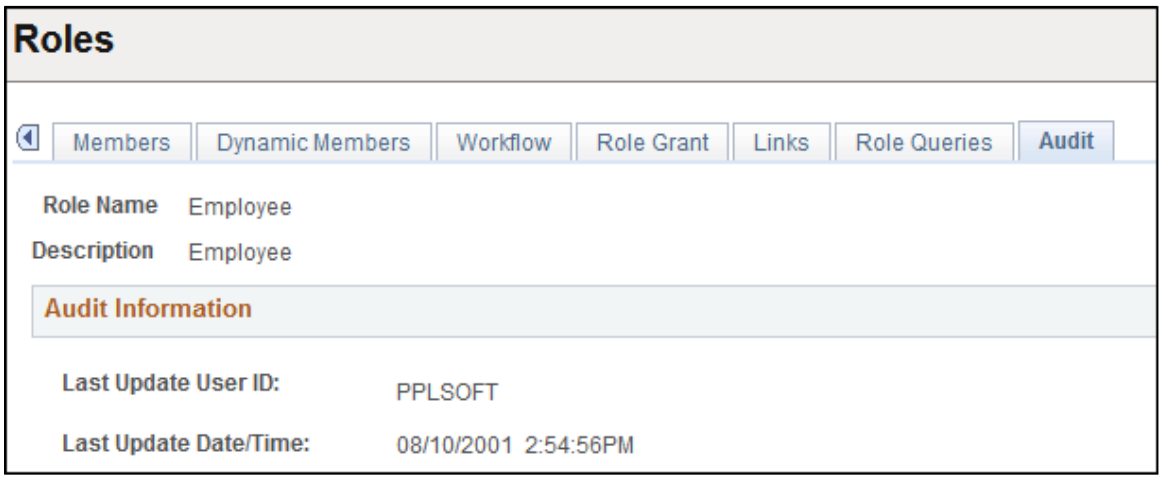
   - XML file

     Downloads the query results as a xml (.xml) file.

Viewing When a Role Was Last Updated

Access the Roles - Audit page (select **PeopleTools** > **Security** > **Permissions and Roles** > **Roles** and click the **Audit** tab).

This example illustrates the fields and controls on the Roles - Audit page.



View when a role was last updated and by whom. You can also view who has made changes to security tables by using the Database Level Auditing feature.

Related Links

“Understanding Database Level Auditing” (Data Management)

Creating a NEWUSER Role

When a new user enters the system and you have implemented dynamic role rules, the user does not belong to any roles until your role rules execute. When you enter a new user into the system, the user has access only to the public pages you authorize for the NEWUSER role. When the dynamic role rules execute, the new user becomes a member of the roles that apply based on the user's employee position.

Note: The NEWUSER role is not a PeopleSoft-delivered role. You can name the role to suit your requirements.

To implement a NEWUSER role:

1. Create your NEWUSER role.
2. Add permission lists to the role so that members of this role have access to the pages that are appropriate for *all* users within the system, like My Profile and any other areas that are not a threat to your system security.
3. Apply the appropriate roles.

   If you use dynamic role assignment, then wait until the batch program runs; if you use static role assignment, then you must wait until an administrator manually applies the appropriate roles.

If the role rules run only one once in a 24-hour period, new employees may not have access to the system until the next day. If the rules run more frequently, they may have access within a couple of hours. If a new user cannot wait until the next run of the dynamic role rule, you can use one of the following options:

- Add required pages to one of the permission lists used by the NEWUSER role.
- Reduce the time between the dynamic rule executions.

  Note: Reducing the execution interval of the dynamic rules may affect performance, depending on how the rules are implemented.

- Add a Signon PeopleCode script that detects that the user needs access to a certain role.

  To do this, run a query against LDAP, the database, or the location where the information resides. Use the User Profile component interface to add the appropriate roles to the user, according to the query results.

Executing Dynamic Role Rules

This section discusses how to:

- Execute dynamic role rules for a role.
- Execute dynamic role rules for all roles assigned to a user profile.
- Execute dynamic role rules for all roles and user profiles.

Understanding Executing Dynamic Role Rules

You can execute dynamic role rules in the three modes. You can execute dynamic role rules by:

- Role.
- All roles for a user profile.
- All roles and user profiles.

Roles rules are executed by the DYNROLE_PUBL application engine program that runs through PeopleSoft Process Scheduler. After the program runs, it publishes a message, ROLESYNCHEXT_MSG, that contains a list of users and roles for the rule. The application engine program does not update any tables; the message (subscription PeopleCode) performs the actual database updates.

Note: The successful completion of the dynamic roles program does not ensure that the roles were updated; the associated message must also be published successfully.

Each page that you can use to execute dynamic role rules features a link to the Process Scheduler Monitor where you can monitor the status of application engine program processing. In addition, each page features a link to the Service Operations Monitor where you can view details of the ROLESYNCHEXT_MSG message publication of users and roles for the rule.

Executing Dynamic Role Rules for a Role

To execute a dynamic role rule for a single role use the Roles - Dynamic Members page (ROLE_DYNMEMBER). To access the Roles - Dynamic Members page, select **PeopleTools** > **Security** > **Permissions and Roles** > **Roles** and click the **Dynamic Members** tab.

Click the **Execute Rule(s)** button on the page to execute the role rule(s). The **Execute Rule(s)** button launches the DYNROLE_PUBL application engine program which executes the rule(s).

See Displaying Dynamic Role Members.

Executing Dynamic Role Rules for All Roles Assigned to a User Profile

To execute dynamic role rules for all roles assigned to a user profile use the User Profile - Roles page (USER_ROLES). To access the User Profile - Roles page, select **PeopleTools** > **Security** > **User Profiles** and click the **Roles** tab.

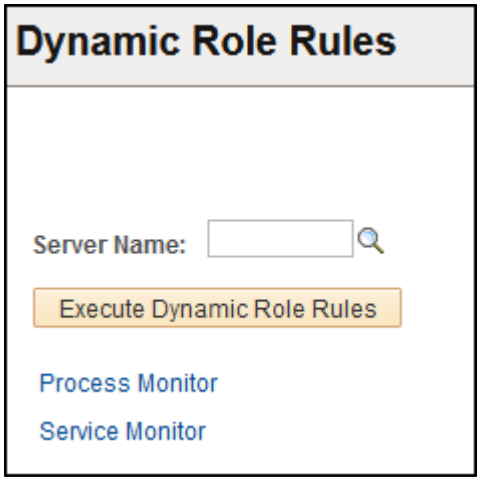
Click the **Execute Rule(s)** button on the page to execute the role rule(s). The Execute Rule(s) button launches the DYNROLE_PUBL application engine program which executes the rule(s).

See Setting Roles.

Executing Dynamic Role Rules for All Roles and Users Profiles

To execute a dynamic role rule for all roles and user profiles use the Dynamic Role Rules page (ROLEDYNLAUNCH). To access the page select **PeopleTools** > **Security** > **Permissions and Roles** > **Run Dynamic Role Rules**. The following example shows the Dynamic Role Rules page:

This example illustrates the fields and controls on the Dynamic Role Rules page.



The Dynamic Role Rules page features the following page controls:

Field or Control	***Description***
Server Name	Enter the name of the process scheduler server to run the rule (s).

Field or Control	*Description*
Execute Dynamic Role Rules	Click to launch the DYNROLE_PUBL application engine program which executes the rule(s). The application engine program runs through PeopleSoft Process Scheduler. After the DYNROLE_PUBL application engine program runs the rule, it publishes a message, ROLESYNCHEXT_MSG, that contains a list of users in the role. **Note:** The successful completion of the dynamic roles program does not ensure that your roles were updated; the associated message must also be published successfully.
Process Monitor	Click to view the status of the DYNROLE_PUBL application engine program in the Process Scheduler Monitor.
Service Monitor	Click the link to check the status of the publication of the ROLESYNCHEXT_MSG message in the Service Operations Monitor.

Related Links

“Viewing the Status of Processes” (Process Scheduler)

“Monitoring Asynchronous Service Operation Transactions” (Integration Broker Service Operations Monitor)

Defining the PeopleSoft Administrator Role

The PeopleSoft Administrator role gives full access to all folders and content reference definitions in the portal registry — that is, a user with the PeopleSoft Administrator role can navigate to all menu folders and menu items in the portal registry even if the portal registry security settings do not specifically include this user or role. In addition, the PeopleSoft Administrator role gives full access to menus and pages in the PSAUTHITEM table.

The PeopleSoft Administrator role cannot be viewed, edited, modified, or cloned because it is not defined as other roles are defined. The PeopleSoft Administrator role is hard-coded into every application. You will not find this role if you search for it in the roles component.

Note: The PeopleSoft Administrator role does *not* have access to data. Data security is granted through the primary and row-level permission lists assigned directly to a user profile.

Chapter 5

Using Role and Permission List Aliases

Using Role and Permission List Aliases

This topic discusses how to:

- Work with role and permission list aliases.
- Identify hard-coded roles and permission lists.
- Use pages to define and manage role and permission list aliases.
- Enable alias options.
- Define permission list aliases.
- Run role and permission list alias queries.

Understanding Role and Permission List Aliases

IsUserInRole and IsUserInPermissionList PeopleCode functions are used by PeopleSott developers to control discrete functionality or access that is not controlled by normal role and permission list settings. These PeopleCode functions use hard-coded references to PeopleSoft roles and permissions lists and therefore make it hard for customers to adopt their own roles and permission lists when they want to use their own naming conventions. If they adopt a new name for a PeopleSoft-delivered role or permission list and there are any IsUserInRole and IsUserInPermissionList PeopleCode references to roles and permissions, the customer may want to find and modify all those PeopleCode references.

As a workaround, security administrators can create aliases for hard-coded roles and permission lists. At runtime, IsUserInRole or IsUserInPermissionList PeopleCode runs against the aliases and returns a value of *TRUE*.

Key points about role and permission list aliases:

- Role and permission list aliases are not given the security access of the role or permission list on which they are defined. Only the name reference uses the alias.
- System variable %Role will not contain any roles that are only associated via an alias. Likewise, system variable %PermissionLists will not contain any permission lists that are only associated via an alias.
- Aliases defined for roles and permission lists are not included with the original object when copying Application Designer projects using Project Copy.

Example: Role Alias

Role alias example:

- Role A is assigned to User A.
- There is functionality controlled by IsUserInRole(“Role B").

  This will return *FALSE* for User A.
- If Role B is set as an alias for Role A, IsUserInRole(“Role B”) will return TRUE for User A, even though Role B is not actually assigned to User A.

The IsUserInRole built-in function takes an arbitrary-length list of strings representing the names of roles and determine whether the current user belongs to any role in an array of roles. The syntax is as follows:

```
IsUserInRole(rolename1 [, rolename2]. . .)
```

In this example before the alias assignment, only Role A is included in the array, as follows:

```
IsUserInRole (Role A)
```

When Role B is defined as an alias of Role A, Role B is included in the array of roles:

```
IsUserInRole (Role A, Role B)
```

Example: Permission List Alias

Permission list alias example:

- Permission List A is assigned to User A (through a role).
- There is functionality controlled by IsUserInPermissionList (“Permission List B”)

  This will return *FALSE* for User A.
- If Permission List B is set as an alias for Permission List A, IsUserInPermissionList("Permission List B") will return *TRUE* for User A, even though Permission List B is not actually assigned to User A (through a role).

The IsUserInPermissionList built-in function takes an arbitrary-length list of strings representing the names of permission lists and determines and determine whether the current user belongs to any of the permission lists. The syntax is as follows:

```
IsUserInPermissionList(PermissionList1 [, PermissionList2]. . .)
```

In this example before the alias assignment, only Permission List A is included in the array list, as follows:

```
IsUserInPermissionList(PermissionList A)
```

When Permission List B is defined as an alias of Permission List A, Permission List B is included in the array:

```
IsUserInPermissionList(PermissionList A, PermissionList B)
```

Identifying Hard-Coded Roles and Permission Lists

To identify hard-coded roles and permission lists, use either of these options to scan for instances of IsUserInRole or IsUserInPermissionList:

- PeopleCode trace.

Use the Show Each option on the Trace PeopleCode page. To access the page select in PIA select **PeopleTools** > **Utilities** > **Debug** > **Set PeopleCode Trace Options.**

Note that the trace can considerably slow system performance and it should not be performed in a production environment.

- Change Impact Analyzer.

  Change Impact Analyzer allows you to search for references in the application that match a string that you specify. Once you connect to a database you can use the Find In page to find references in PeopleCode, SQL, and HTML objects. You select your scope (Search Criteria) and add one or more strings for which you want to find references in the Find What text box. For each string you specify you click on Add to List button. Once you have completed building your list of strings, you select the Run Find In button to perform the search.

  To access the Find In page, in Change Impact Analyzer select **Tools** > **Find In.**

Related Links

"Configuring PeopleCode Trace" (System and Server Administration)
"Find In Feature" (Change Impact Analyzer)

Pages Used to Define and Manage Role and Permission List Aliases

This table describes the pages used to define and manage role and permission list aliases.

Page	*Object ID*	*Description*	*Navigation*
Alias Options	PTSECALIASOPTIONS	Use this page to enable role and permission list alias functionality. You must enable the Role Alias option for the Role Aliases page to appear in the Roles component. Likewise, you must enable the Permission List Alias option for the Permission List Alias page to appear in the Permission List component. **Important!** You must reboot the application server for changes you make on the page to take effect.	**PeopleTools** > **Security** > **Security Objects** > **Role and Permission Aliases**
Role Aliases	PTROLENAMEALIAS	Use this page to define aliases for a role.	**PeopleTools** > **Permissions and Roles** > **Roles** > **Role Aliases**

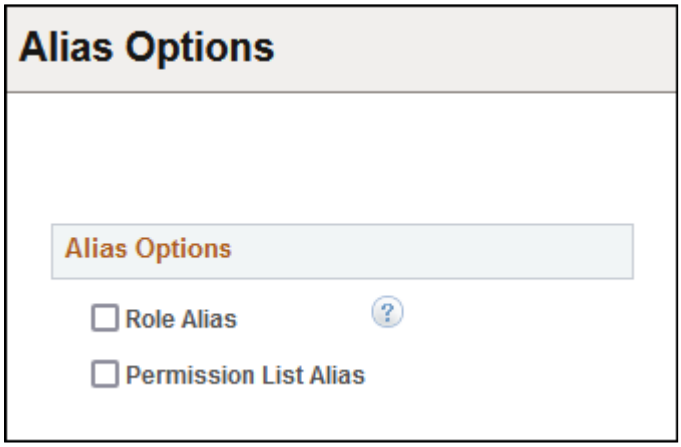
Page	Object ID	Description	Navigation
Permission Aliases	PTCLASSIDALIAS	Use this page to define aliases for a permission list.	**PeopleTools** > **Permissions and Roles** > **Permission Lists** > **Permission List Aliases**
Role Security Queries	MAINT_SEC_ROLE_QRY	Use this page to run a query that returns a list of all role aliases defined in the database.	**PeopleTools** > **Security** > **Review Security Information** > **Role Queries**
Permission List Report Queries	MAINT_SEC_PLIST_QR	Use this page to run a query that returns a list of all permission list aliases defined in the database.	**PeopleTools** > **Security** > **Review Security Information** > **Permission List Queries**

Enabling Alias Options

For the pages used to define role and permission list aliases to appear, the Roles - Role Aliases and Permission Lists - Permission List Aliases pages respectively, you must enable the alias options on the Alias Options page (PTSECALIASOPTIONS).

To access the Alias Options page, select **PeopleTools** > **Security** > **Security Objects** > **Role and Permission Aliases.**

This example illustrates the fields and controls on the Alias Options page.



Important! You must reboot the application server for changes you make on the page to take effect.

Field or Control	Description
Role Alias	Select this option to enable the alias functionality for roles. When you enable this option the Roles - Role Aliases page appears in the Roles component.

Field or Control	*Description*
Permission List Alias	Select this option to enable the alias functionality for permission lists. When you enable this option the Permissions List - Permission Aliases page appears in the Permissions List component.

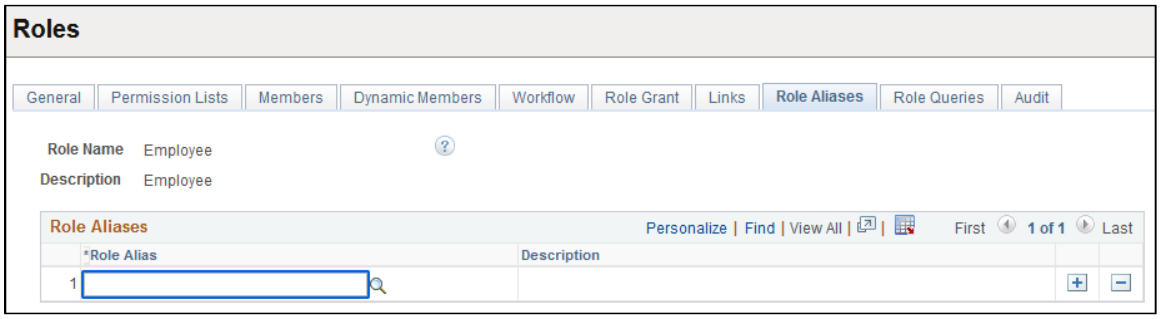
Defining Role Aliases

Use the Roles - Role Aliases page to define role aliases. To access the page select **PeopleTools** > **Permissions and Roles** > **Roles** > **Role Aliases.**

For the page to be accessible, you must enable the Role Alias option on the Alias Options page as described earlier in this topic.

To assign a role alias it must be an existing role in the database.

This example illustrates the fields and controls on the Roles – Role Aliases page.



To assign a role alias to a role, in the Role Aliases grid, enter or search for a role to add.

Click the Add Row button to add additional aliases.

The system does not assign role aliases any permissions or access of the base role on which you define it. (In the previous example, *Employee* is the base role.) Role aliases are only used in conjunction with the IsUserInRole built-in function. When IsUserInRole PeopleCode runs, the system will include the base role and any role aliases you define in the Role Aliases grid in the string array and return a value of *TRUE*.

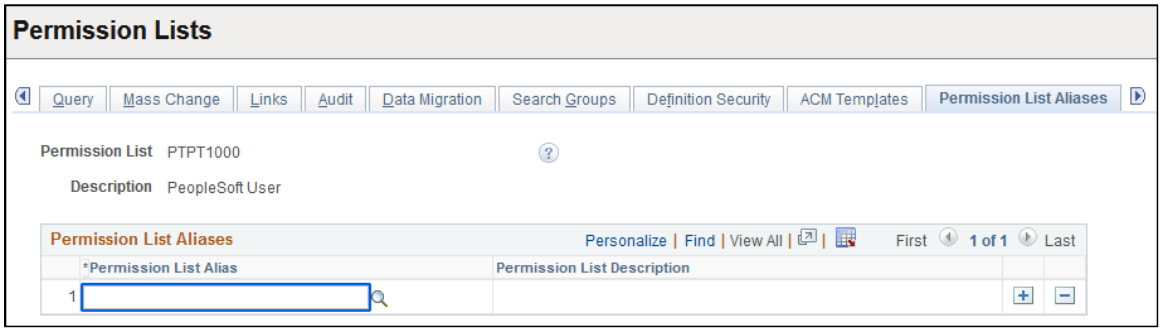
Defining Permission List Aliases

Use the Permission Lists - Permission List Aliases page to define permission list aliases. To access the page select **PeopleTools** > **Permissions and Roles** > **Permission Lists** > **Permission List Aliases.**

For the page to be accessible, you must enable the Permission List Alias option on the Alias Options page as described earlier in this topic.

To assign a permission list alias it must be an existing permission list in the database.

This example illustrates the fields and controls on the Permission Lists – Permission List Aliases page.



To assign a permission list alias to a permission list, in the Permission List Aliases grid, enter or search for a permission list to add.

Click the Add Row button to add additional aliases.

The system does not assign permission list aliases any permissions or access of the base permission list on which you define it. (In the previous example, *PeopleSoft User* is the base permission list.) Permission list aliases are only used in conjunction with the IsUserInPermissionList built-in function. When IsUserInPermissionList PeopleCode runs, the system includes the base permission list and any aliases you define in the Permission List Aliases grid in the string array and return a value of *TRUE*.

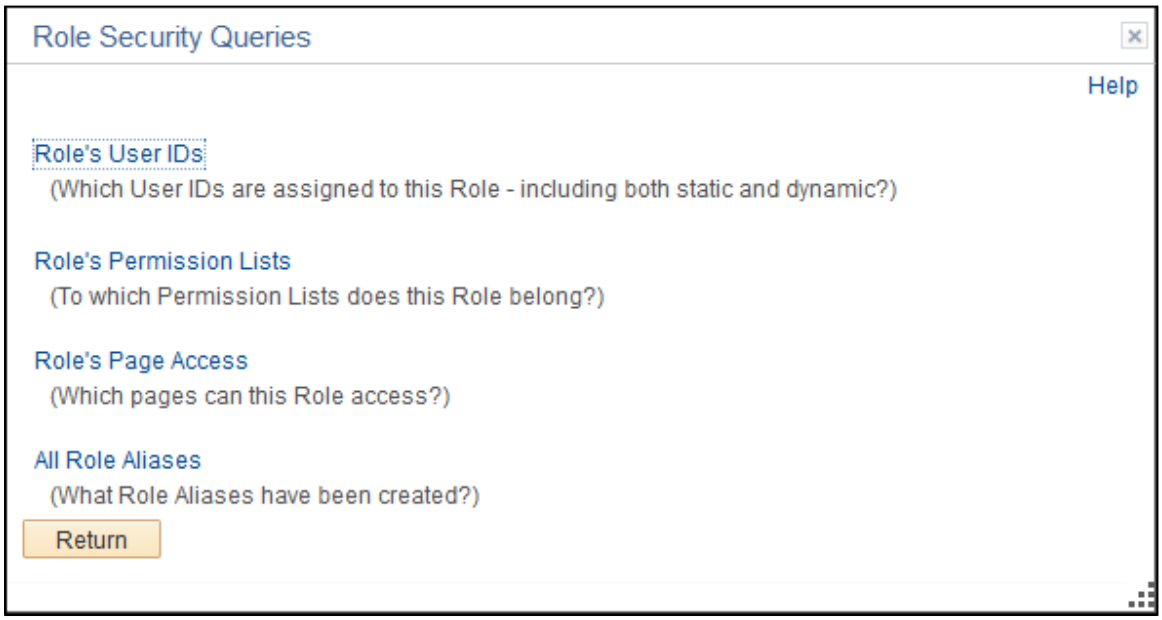
Running Role and Permission List Alias Queries

PeopleSoft provides queries that generate reports of the role and permission list aliases defined in the database:

Running the Role Alias Query

Use the Role Security Queries page (MAINT_SEC_ROLE_QRY) to run a query on role aliases defined in the database. To access the page select **PeopleTools** > **Security** > **Review Security Information** > **Role Queries.**

This example illustrates the Role Security Queries page.

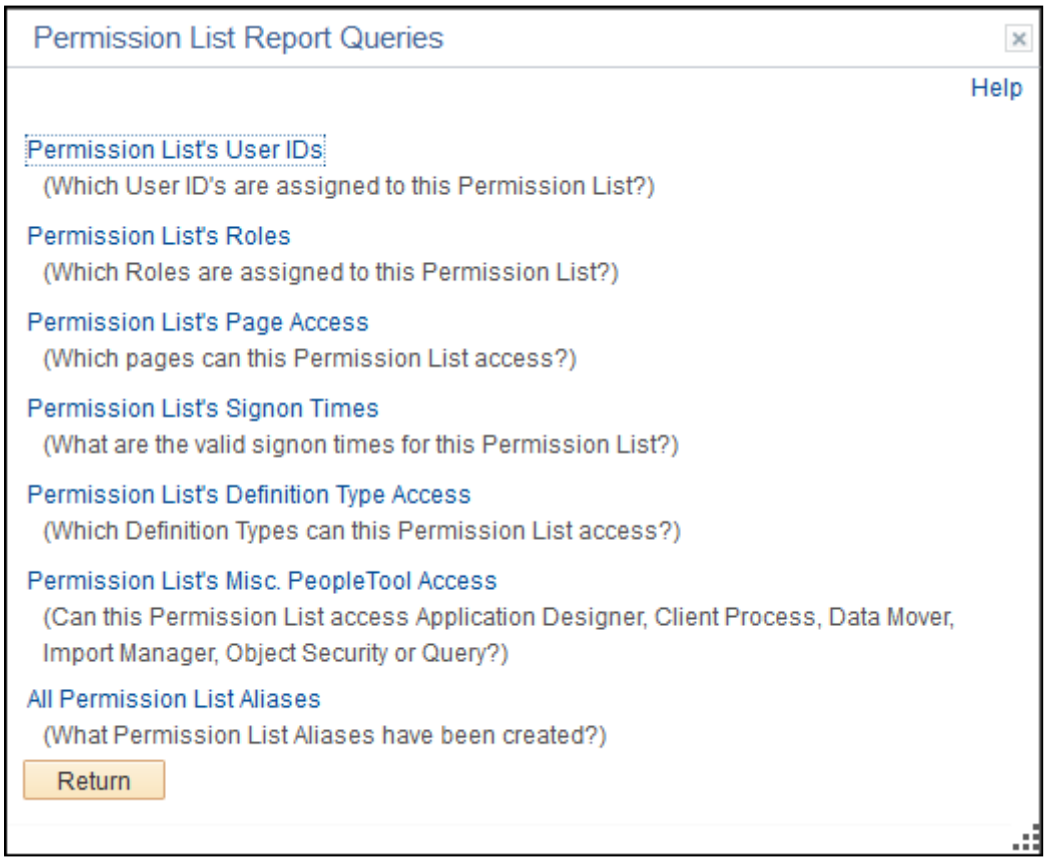


To view the role aliases defined in the database, click the **All Role Aliases** link. The system runs the delivered PT_SEC_ROLE_ALIASES query, generates a report of all defined role aliases in the database, and displays the results in a new browser window.

Running the Permission List Alias Query

Use the Permission List Report Queries page (MAINT_SEC_PLIST_QR) to run a query on permission list aliases defined in the database. To access the page select **PeopleTools** > **Security** > **Review Security Information** > **Permission List Queries.**

This example illustrates the Permission List Report Queries page,



To view the permission list aliases defined in the database, click the All Permission List Aliases link. The system runs the delivered PT_SEC_PLIST_ALIASES query, generates a report of all defined permission list aliases in the database, and displays the results in a new browser window.

Chapter 6

Administering User Profiles

Understanding User Profiles

User profiles define individual PeopleSoft users. You define user profiles and then link them to one or more roles. Typically, a user profile must be linked to at least one role to be a usable profile. The majority of values that make up a user profile are inherited from the linked roles.

Note: A user profile may have no roles; for example, a user who is not allowed access to the PeopleSoft application. You still want workflow-generated email sent to the user.

You define user profiles by entering the appropriate values on the user profile pages. The user profile contains values that are specific to a user, such as a user password, an email address, an employee ID, and so on.

The user ID and description appear at the top of each page to help you recall which user profile you are viewing or modifying as you move through the pages.

Setting Up Access Profiles

This section provides an overview of access profiles and discusses how to:

- Use the Access Profiles dialog box.
- Set access profile properties.
- Work with access profiles.

Understanding Access Profiles

Every user profile must be assigned to an access profile, by way of a Symbolic ID. The Access ID consists of a relational database management system (RDBMS) ID and a password. Access profiles provide the necessary IDs and passwords for the database logon operations that occur in the background. Access IDs are used:

- When an application server initializes and connects to a PeopleSoft database.
- When a developer or power user signs in to the PeopleSoft database directly (two-tier).
- When batch programs connect to the database.

Users signing in to the system through PeopleSoft Pure Internet Architecture take advantage of the Access ID that the application server used for connecting to the database.

Access profiles enable you to minimize the number of users who need to know system administrator passwords. In fact, only one person needs to know these passwords. That person can create the required access profiles—by providing the necessary passwords when prompted—and all other security administrators can assign users to the predefined access profiles. The Access ID and password are encrypted in the database in the PSACCESSPROFILE table.

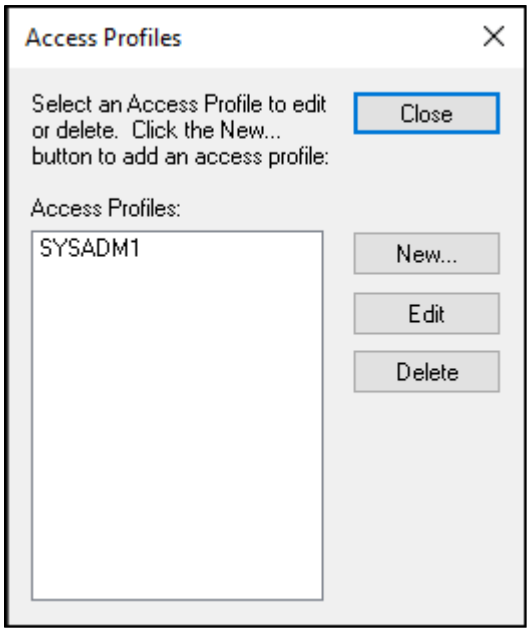
Before you begin creating your user profiles, roles, and permission lists, you need to set up your access profiles in the database. Ultimately, the access profile is the profile that your users use to connect to your PeopleSoft database. Without being associated with an access profile, users cannot sign in, even with a test ID. This association is by way of the symbolic ID, which is a proxy ID for the Access ID and Access password.

The ID that you use must be defined at the RDBMS level as a valid RDBMS ID. You do not use PeopleSoft or PeopleTools software to create an RDBMS ID; create it using the utilities and procedures defined by your RDBMS platform. After you create the RDBMS ID, use the PeopleTools access profiles utility to link your RDBMS ID to the access profile. This profile is created when you first install your database.

Using the Access Profiles Dialog Box

Access the Access Profiles dialog box in Application Designer (Tools, Miscellaneous Definitions, Access Profiles).

This example illustrates the fields and controls on the Access Profiles dialog box.



Field or Control	*Description*
Close	Click to exit this dialog box.
New	Click to create a new access profile definition.

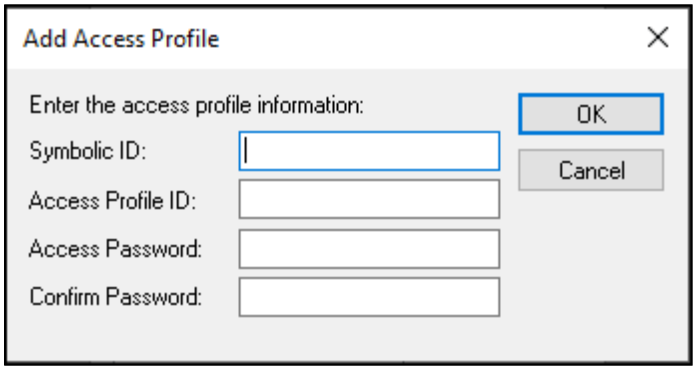
Field or Control	*Description*
Edit	Click to edit an access profile definition.
Delete	Click to delete an access profile definition.

Setting Access Profile Properties

When you create or modify an access profile using the Access Profiles dialog, you need to understand the properties that comprise an access profile. After reading this section, you will be familiar with these properties.

Access the Add Access Profile dialog box (click the **New** button in the Access Profiles dialog box).

This example illustrates the fields and controls on the Add Access Profile dialog box.



Field or Control	*Description*
Symbolic ID	Enter the Symbolic ID that is used to retrieve the encrypted access ID and access ID password from the PSACCESSPROFILE table. For your initial installation, set it equal to the database name.

Field or Control	*Description*
Access Profile ID	Enter the Access Profile ID, which must be a valid RDBMS ID with system administrator privileges and must match the associated RDBMS ID. The system assumes that the RDBMS ID that you enter is the same as the Access Profile ID. The Access Profile ID must be a different logon ID than the User ID. Logic within PeopleTools ensures that if Access ID = User ID, then PeopleTools does not log off and log on again, nor does the system issue a SET CURRENT SQLID = 'owner ID'. **Note:** In DB2 terminology, Access ID is a primary ID and Owner ID is a secondary Auth ID. If the Access ID does not equal the Owner ID, then secondary authorization security exists in DB2 to issue a SET CURRENT SQLID command. DB2 will qualify tables (required) with the Owner ID provided by SET CURRENT SQLID statements issued by the PeopleSoft software. If the Access ID equals the Owner ID, then the secondary authorization exits are not required. DB2 will qualify the table name with the Access ID.
Access Password	Enter the password associated with your RDBMS ID/Access Profile ID, which is the password that the Access ID uses to sign in to the database.

Working with Access Profiles

This section discusses how to create a new Access Profile definition, change an Access Profile password, and delete an Access Profile in the PeopleSoft system.

To create a new Access Profile definition:

1. In PeopleSoft Application Designer, select **Tools** > **Miscellaneous Definitions** > **Access Profiles.**

   The Access Profiles dialog box appears.

2. Click **New.**

   The Add Access Profile dialog box appears.

   This dialog box prompts you for the Symbolic ID, name, and password of the new access profile.

3. Enter a Symbolic ID.

   The Symbolic ID is used as the key to retrieve the encrypted access ID and access ID password from the PSACCESSPROFILE table.

4. Enter an Access Profile ID.

   This ID must be a valid RDBMS ID with system administrator privileges.

5. Enter and confirm a password.

The access password is the password string for the RDBMS ID/Access Profile ID. The **Confirm Password** field is required, and its value must match that of the **Access Password** field.

6. Click **OK.**

Note: You should use only one Access ID for your system. Some RDBMSs do not permit more than one database table owner. If you create more than one Access ID, additional steps may be required to ensure that this ID has the correct rights to *all* PeopleSoft system tables.

To change an Access Profile password:

1. In Application Designer, select **Tools** > **Miscellaneous Definitions** > **Access Profiles.**

   The Access Profiles dialog box appears.

2. In the Access Profiles: list, highlight the profile that you want to modify, and click Edit.

   The Change Access Profile dialog box appears.

   This dialog box prompts you for the old password, the new password, and then a confirmation of the new password for the access profile.

3. Enter and confirm the new password.

   The access password is the password string for the ID. The **Confirm Password** field is required, and its value must match that of the **Access Password** field.

4. Click **OK.**

To delete an Access Profile:

1. Select **Tools** > **Miscellaneous Definitions** > **Access Profiles.**

   The Access Profiles dialog box appears.

2. Highlight the access profile that you want to remove, and click Delete.

   You are prompted to confirm the deletion.

   Click Yes at the prompt dialog box if you want to delete the selected access profile.

Important! Make sure you don't delete the *only* available Access ID or you will not be able to log on to PeopleSoft software in any capacity.

Setting Up User Profile Types

This section provides an overview of user profile types and discusses how to define user profile types.

Understanding User Profile Types

When deploying your applications to the internet, you potentially can generate thousands of different user profiles. In some situations, you may need to aggregate your user profiles by category. For example, ID

types enable you to use employee ID numbers that begin at 1 as well as customer ID numbers that begin at 1.

User profile types also provide a way to link user profiles with data stored in application-specific records. PeopleSoft applications primarily need this link for self-service transactions. For example, you want employees to see only their own benefits, or you want customers to view and pay only their own bills. Customer ID, Employee ID, and so on are the keys for the application data. User profile types enable the system to find the correct ID based on the user profile. The system needs the value because personal data and vendor contact data may have the same key field. Because personal data and vendor contact data resides in different records, no edit exists that will prevent the two records from having the same key.

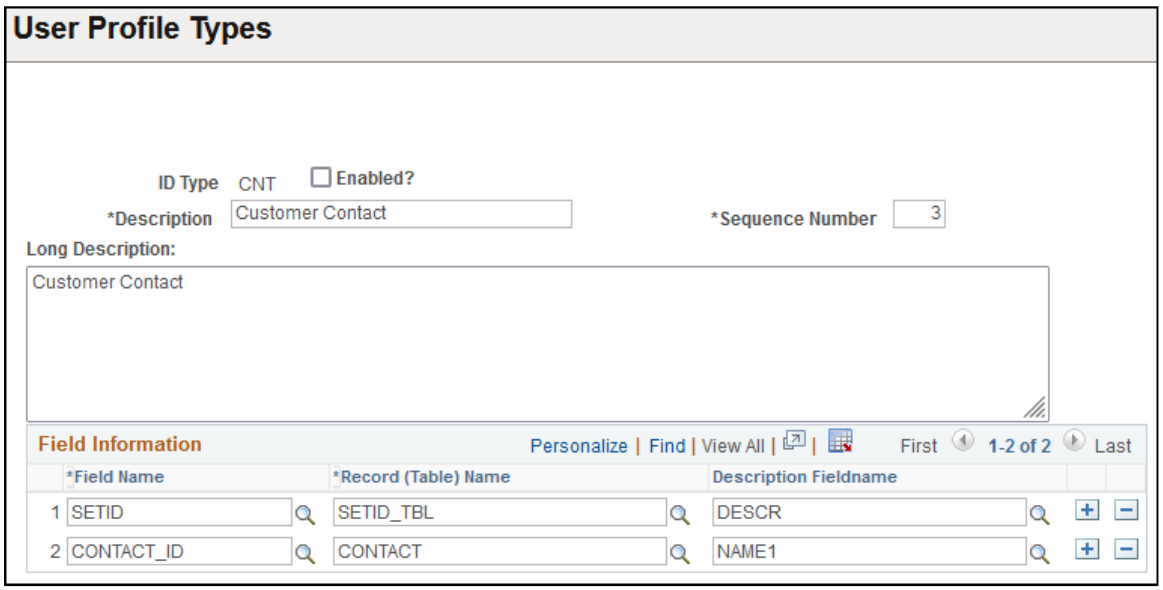
This table lists the profile types that PeopleSoft delivers:

ID Type	*Description*
BID	Bidder
CNT	Customer Contact
CST	Customer
EJA	External Job Applicant
EMP	Employee
NON	None
ORG	Organization ID
PER	Person (CRM)
VND	Vendor
PTN	Partner

Defining User Profile Types

Access the User Profile Types page (select **PeopleTools** > **Security** > **Security Objects** > **User Profile Types**).

This example illustrates the fields and controls on the User Profile Types page.



Field or Control	*Description*
ID Type	Displays the abbreviated form of the profile type name.
Description	Enter a name for the profile type that is no more than 30 characters. This value appears on the ID page in the User Profiles component.
Enabled?	Select this check box to enable a profile type. When selected, you can assign the profile type to user profiles. When deselected, the profile type does not appear in the **Profile Type** drop-down list box on the User Profile - ID page. **Note:** Do not enable the ID type until the fields and tables in the Field Information section are defined and built using Application Designer.

Field or Control	*Description*
Sequence Number	The **SetUserDescr( )** function uses this value. After you assign one or more ID types on the User Profiles - ID page, click the **Set Description** link and the **SetUserDescr ( )** function automatically retrieves the value of the record field that you reference in the **Edit Table** and **Description Fieldname** fields on the User Profile Types page. If you assign multiple ID types, the sequence number determines which user profile type to use. The function looks to the user profile type with the lowest sequence number and checks for the presence of a value in the description field. If no value exists, the function moves to the next higher sequence number. For example, if you assign a user both the Employee (seq no 1) and Customer Contact (seq no 3) ID types, then the function first looks to the Employee user profile type and retrieves the value in the PERSONAL_DATA.NAME field. If the PERSONAL_DATA.NAME field contains no value, the function looks to the Customer Contact ID type and retrieves the value from the CONTACT.NAME1 field. **Note:** For user types that list multiple fields, the system uses the **Description Fieldname** of the last field in the field list. For example, the Customer Contact user profile type lists two fields: SETID and CONTACT_ID. The set user description function uses the **Description Fieldname** CONTACT.NAME1 corresponding to the last field, CONTACT_ID.
Long Description	Enter details about a profile type. The maximum length of this field is 250 characters.
Field Information	The fields that you select enable the User Profiles component to prompt for an ID value when you select a type on the ID page. For example, if the user selects the *Employee* ID type from the User Profiles - ID page, the system must know the table that contains the valid ID values to display to the user when the user clicks the prompt button. The Edit Table column specifies the record, and the Field Name column specifies the field. You can specify multiple fields if the ID has multiple keys, as is the case of the Customer user profile type where the keys for customer information are SETID and CUST_ID.

Working With User Profiles

This section discusses how to:

- Create a new user profile.
- Copy a user profile.
- Delete a user profile.
- Bypass tables during the Delete User Profile process.

- Configure user profiles for forgotten user ID emails.
- Create email text for forgotten user IDs.
- Set up a web site for forgotten user ID emails.
- Request emails for forgotten user IDs.

Creating a New User Profile

To create a new user profile:

1. Select **PeopleTools** > **Security** > **User Profiles** > **User Profiles** to access the User Profiles search page.
2. Click **Add a New Value.**
3. On the Add a New Value page, enter the new user ID in the **User ID** field and click **Add.**

   The user ID can contain up to 30 characters. The name that you specify cannot contain white space or any of the following characters:

   ```
   ; : & , < > \ / " [ ] ( ) '
   ```

   Also, you cannot create a user ID named *PPLSOFT*; this user ID is reserved for use within PeopleTools.
4. Specify the appropriate values from the pages in the User Profiles component (USERMAINT), and click **Save.**

Copying a User Profile

To copy a user profile:

1. Select **PeopleTools** > **Security** > **User Profiles** > **Copy User Profiles** to access the User Profiles search page.
2. Select the user ID that you want to copy.
3. On the User Profile Save As page, enter the new user ID, a description, and the password that the new user ID should use to sign in to the system.

Note: If **Copy ID Type Information** is not selected, the system does not save the EMPLID value to the PSOPRDEFN table.

Note: The only personalization data that is copied from My Preferences is what is found on the General Settings page. Personalization settings on other pages of My Preferences is not copied.

Deleting a User Profile

To delete a user profile:

1. Select **PeopleTools** > **Security** > **User Profiles** > **Delete User Profiles** to access the Delete User Profile page.
2. If the search page appears, select the user profile to delete (make sure that the *correct* user profile is selected.)
3. Click **Delete User Profile** to remove information related to this particular user profile in every PeopleTools and application data table in which the OPRID field is a key field.

   Note: Query the PS_TBLSELECTION_VW view to list the tables in which the OPRID field is a key field.

   To prevent user information in a specific table from being deleted, you can designate tables that the delete user process bypasses.

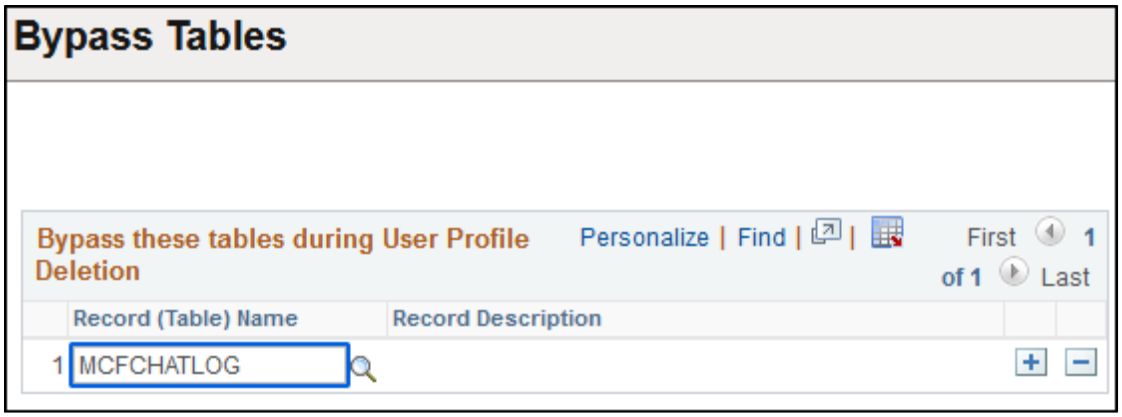
Related Links

Component Interfaces

Bypassing Tables During the Delete User Profile Process

Access the Bypass Tables page (select **PeopleTools** > **Security** > **Security Objects** > **Profile Delete: Tables to Skip**).

This example illustrates the fields and controls on the Bypass Tables page.



When you delete a user profile and its related information, you might not want to delete tables that contain rows of user profile data. For instances such as these, you can specify the tables for the delete process to skip.

To bypass tables during the Delete User Profile process:

1. Click the prompt button to select the record name to skip.

   Note: The prompt displays only records that contain the OPRID field as a key field. The view behind this prompt is the PS_TBLSELECTION_VW.

2. Insert additional rows for other table names, as necessary.

3. Click the **Save** button.

Related Links

Preserving Historical User Profile Data

Specifying User Profile Attributes

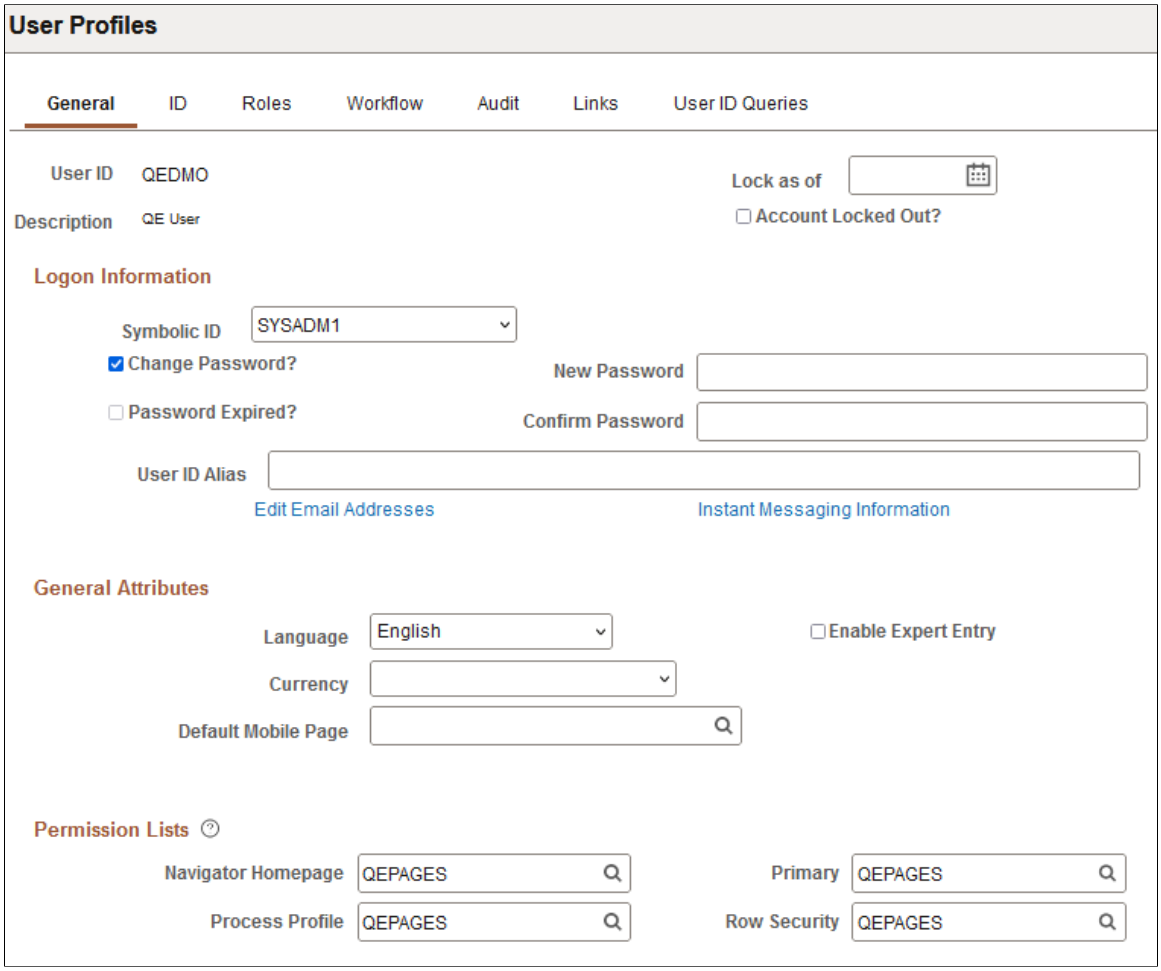
This section discusses how to:

- Set general user profile attributes.
- Set ID type and attribute value.
- Set roles.
- Specify workflow settings.
- View when a user profile was last updated.
- Display additional links.
- Run user ID queries.

Setting General User Profile Attributes

Access the User Profiles - General page (select **PeopleTools** > **Security** > **User Profiles** > **User Profiles** and click the **General** tab).

This example illustrates the fields and controls on the User Profiles - General page.



Field or Control	*Description*
Account Locked Out?	Select this check box to deactivate a user profile for any reason. The user cannot sign in until you have deselected this option. **Note:** The system automatically selects this check box if you are using password controls and the user exceeds the maximum number of failed logon attempts. The administrator needs to manually open the user profile and deselect this check box to reinstate the user. See Setting Password Controls.

Field or Control	*Description*
Lock as of	Specify a date to lock the account. Enter a date or click the calendar icon to select a date. The user will not be able to sign in after this date. If you set a date in the future, when the date is reached or past the system makes the **Lock as of** field unavailable for entry, and selects the **Account Locked Out?** check box. If you set a date in the past, the system automatically makes the **Lock as of** field unavailable for entry, and selects the **Account Locked Out?** check box. If the Security Administrator unlocks a locked account and the date specified here is past, the system removes the date and makes the field available for entry. If the Security Administrator unlocks a locked account and the date specified here is in the future, the field will remain available and the date unchanged.

Logon Information

Field or Control	*Description*
Symbolic ID	Enter a value to retrieve the appropriate encrypted access ID and access password. This value determines which access ID and password are used to log the user onto the database after the system validates the user ID. The access ID is required only when a user needs to connect directly to the database (in two-tier), and when users submit jobs using Process Scheduler. The access ID is not required with the portal or if you use a Lightweight Directory Access Protocol (LDAP) directory server to manage user IDs. With PeopleSoft Pure Internet Architecture, the application server maintains the connection to the database, so the application server must submit an access ID.
Change Password?	When you select this option, the Password and Confirm Password fields appear.
Password and **Confirm Password**	Enter the password string that the user must supply when signing in. The value in the **Confirm Password** field must match that in the **Password** field. The maximum password length is 64 characters. **Note:** These values are required to sign in to the system, but you can save the profile without populating these fields.

Field or Control	*Description*
Password Expired?	If you are using PeopleSoft password controls, this option enables you to force users to change their passwords in the following situations: • The first time that a user signs in to PeopleSoft software. • The next time that a user signs in. • The first time that a user signs in after the system has emailed the user a randomly generated password. **Note:** To use this option, you must enable the **Password Expires in 'x' Days** PeopleSoft password control. See Setting Password Controls. When a user's password has expired, the **Password Expired** check box becomes enabled and selected. By deselecting the check box and saving the change, you can renew the password, although we do not recommend this practice. **Note:** The password expiration applies only to signing into the system through the PeopleSoft Pure Internet Architecture (PIA). When you log in to PeopleTools utilities such as Application Designer, Application Engine, or Data Mover, the password expiration control does not apply. For example, if you try to use an expired password to sign in to PIA, you will see an error message, but you can use the same password to sign into Application Designer.
User ID Alias	Enter a fully qualified email ID (email address) as a user ID alias. For example, tom.x.sawyer@oracle.com could be the user ID used to sign in to the system. The maximum character length is 70.

Field or Control	*Description*
Edit Email Addresses	If a user is part of the workflow system or you have other systems that generate email for users, click this link to enter an email address for a user. You can enter multiple email addresses for a user, but you must select one as the primary email address. The system allows only one email address per type. For example, you cannot enter two home email addresses. The Email Addresses interface has the following controls: • Primary Email Account: If you enter multiple email accounts, you must select one as the primary account. • Email Type: Select from Blackberry, Business, Home, Other, or Work. The Blackberry email type is used with the Workflow/ RIM technology. • Email Address: Enter the email address in this field.
Instant Messaging Information	The instant messaging feature is no longer available. This link will be removed in a future release.

General Attributes

Field or Control	*Description*
Language	Select a value. The language code on the User Profile page has a limited use. For example, when a user runs a batch job, the system needs to know in which language to generate the reports for the user who submitted the job. In PeopleSoft Pure Internet Architecture, the user's language preference is based on the selection that the user makes on the signon page. For Microsoft Windows workstations, the user's language preference is derived from the Display tab in PeopleSoft Configuration Manager. For the Microsoft Windows environment, the value specified as language code in the user profile acts as a default in case the language code is not specified in PeopleSoft Configuration Manager.

Field or Control	*Description*
Currency	If the user works with international currencies, select a currency code to reflect the native or base currency. Values will appear in the currency with which the user is familiar.
Default Mobile Page	Select the mobile homepage that should appear after users sign in to their mobile device. **Important!** PeopleSoft Mobile Agent is a desupported product. These features exist for backward compatibility only.
Enable Expert Entry	Select to specify that some users, such as expert or power users, can defer all processing of the data that they enter. This selection enables users to reduce the number of trips to the server for data processing, regardless of how the developer set field deferred or interactive processing. You enable this option in a component in Application Designer, and you specify which users have this option using the **Enable Expert Entry** check box. Deselect this check box to prevent a user from specifying deferred processing.
Allow Switch User	Select this option to designate users who can change identities in a PeopleSoft system. This feature applies only when accessing PeopleSoft applications using a browser; it has no effect on two-tier or three-tier connections. The default for this feature is hidden. You display this check box by changing the Enable Switch User options on the PeopleTools Options page. See “Using Administration Utilities” (System and Server Administration)

Permission Lists

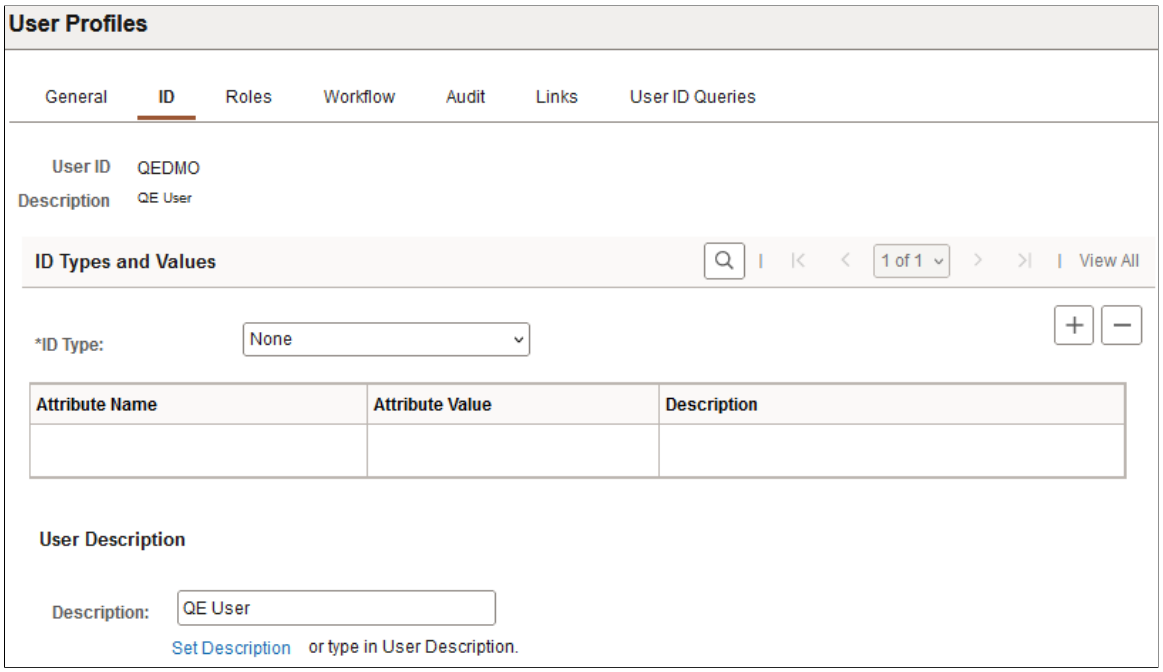
Field or Control	*Description*
Navigator Homepage	Enter a value associated with PeopleSoft Workflow.

Field or Control	*Description*
Process Profile	Displays a value that contains the permissions that a user requires for running batch processes through PeopleSoft Process Scheduler. For example, the process profile is where users are authorized to view output, update run locations, restart processes, and so on. **Note:** Only the process profile comes from this permission list, not the list process groups.
Primary and **Row Security**	Displays which data permissions to grant a user by examining the primary permission list and row security permission list. Which one is used varies by application and data entity (employee, customer, vendor, business unit, and so on). Consult your application documentation for more details. The system also determines mass change (if needed), and definition security permissions from the primary permission list.

Setting ID Type and Attribute Value

Access the User Profiles - ID page (select **PeopleTools** > **Security** > **User Profiles** > **User Profiles** and click the **ID** tab).

This example illustrates the fields and controls on the User Profiles - ID page.



ID Types and Values

Field or Control	*Description*
ID Type and **Attribute Value**	Select the ID type and attribute value. Separating user profiles by ID type enables you to have multiple categories of user profiles with ID numbers all within a range of 1–1000, for example, and it also enables you to grant data permission by entity (customer, employee, and so on). When users sign in to your benefits or payroll deductions application, they see only information that applies to them. A user profile is a set of data about an entity—a user—who interacts with the system. The human resources (HCM) system, which keeps track of your employee data, is designed to focus more on your employee user types. On the other hand, your financials system is designed to keep track of customer and supplier user types. ID types enable you to link user types with the records that are most relevant when a user interacts with the system. In the **Attribute Value** field, select the value associated with the attribute name. For example, the value could reflect the employee number, a customer number, or vendor number.

User Description

The User Description section enables you to help identify the user.

Field or Control	*Description*
Description	Add a description, such as the name of an individual or an organization, for the user profile.
Set Description	Click this link to populate the field with a description from the database.

Note: Before you assign a user type to a user, you must create user types.

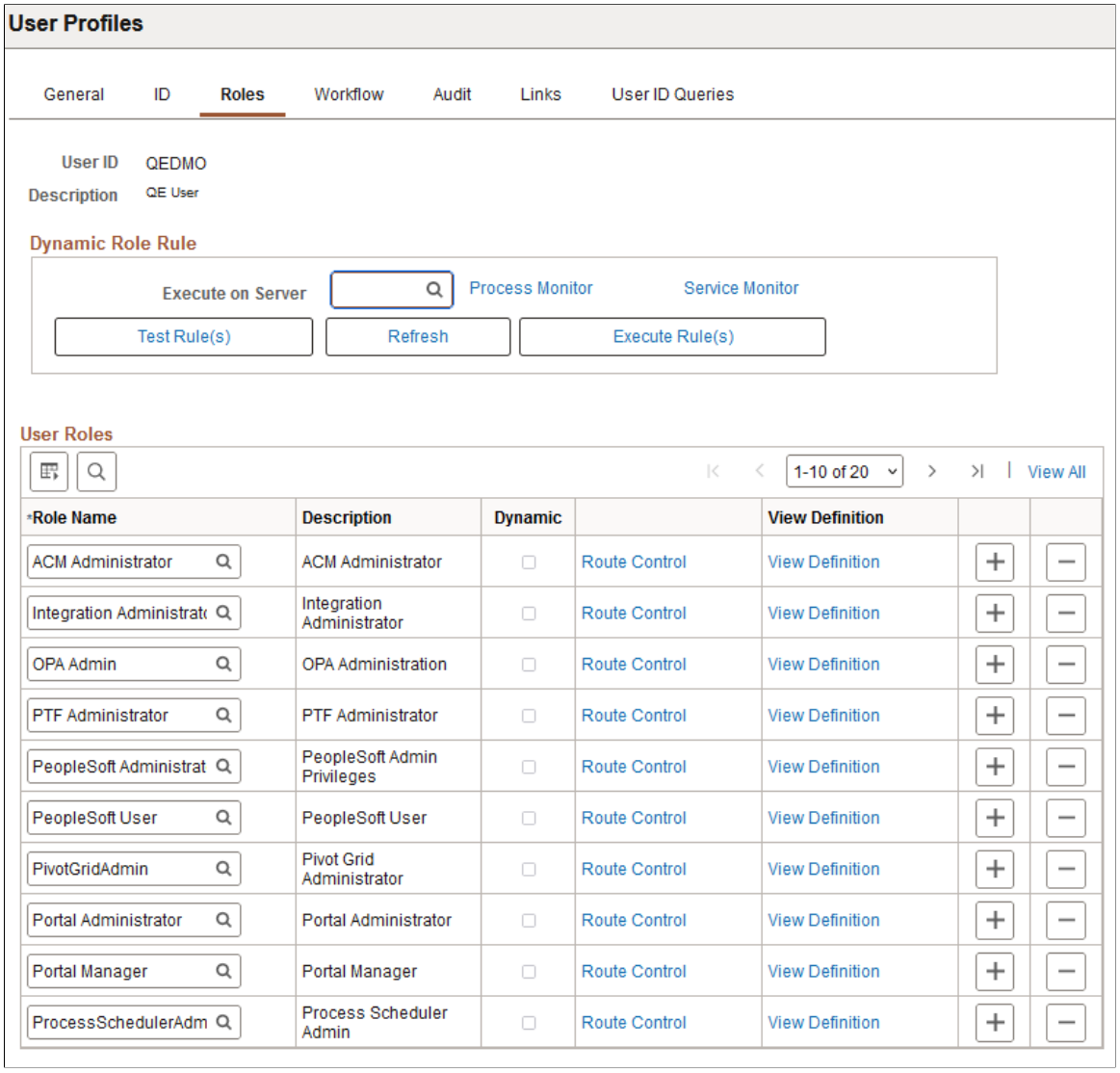
Related Links

Setting Up User Profile Types

Setting Roles

Access the User Profiles - Roles page (select **PeopleTools** > **Security** > **User Profiles** > **User Profiles** and click the **Roles** tab).

This example illustrates the fields and controls on the User Profiles - Roles page



Note: You cannot overwrite roles in the Roles grid. To add a role click the Add Row button (+). To delete a role click the Delete row button (-).

Field or Control	*Description*
Role Name	Displays the name of the role added to the user profile.
Description	Displays a description of the role added to the user profile.
Dynamic	Selected if the system assigned a particular role dynamically.

Field or Control	*Description*
Route Control	Specify a route control profile for each role assigned to a user. For example, suppose that you have a role named EXPENSE_REP. If you want a particular expense representative to handle all of the expense reports submitted by people whose last names begin with *A*, you could assign the user a specific route control profile to send the user reports submitted by individuals with last names beginning with *A*.
View Definition	Click to view the role definition associated with this user profile.

See “Understanding Route Control Development” (Workflow Technology).

See Defining the PeopleSoft Administrator Role.

Dynamic Role Rule

Use these options to test and manually carry out business rules for dynamically updating roles and assigning them to user profiles. You design your role rules using Query Manager, PeopleCode, or LDAP directory rules.

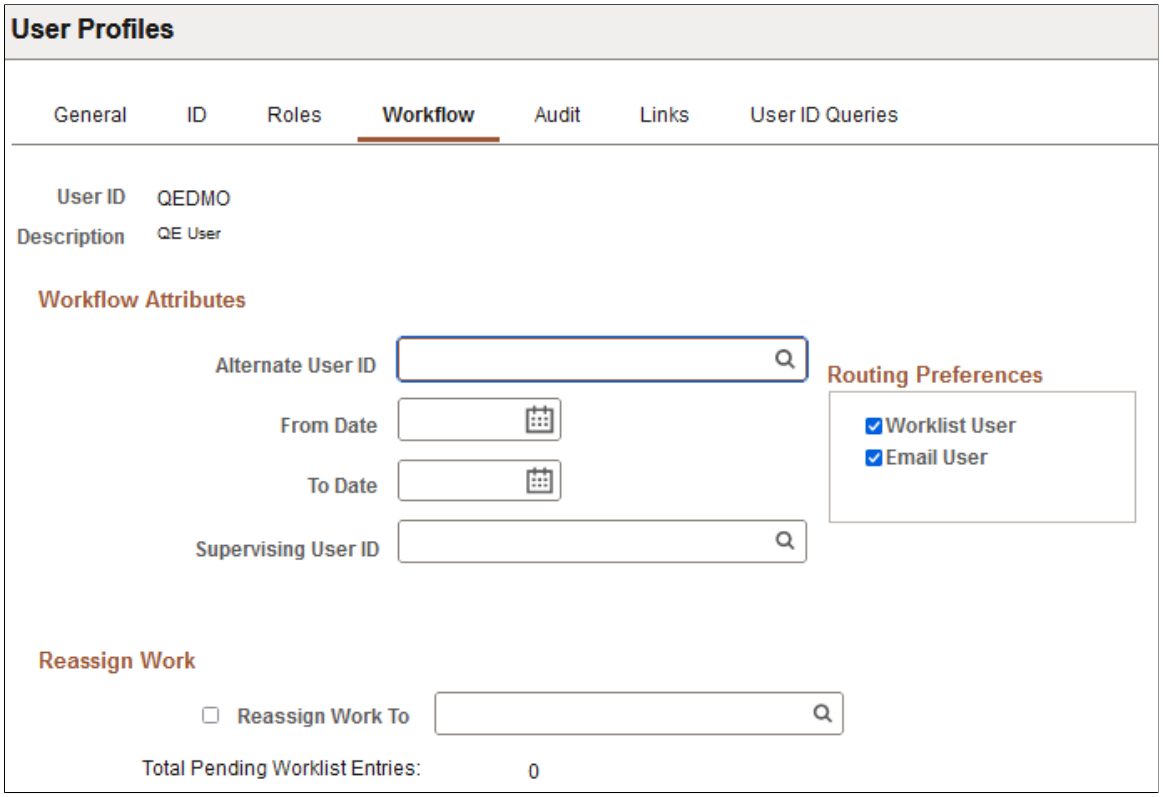
Field or Control	*Description*
Execute on Server	Select the Process Scheduler server that should run your role rule.
Test Rule(s)	Click to test the rules and verify if they will produce the desired results for a particular user. None of the roles are actually assigned, but the system provides you a report as to what roles will be assigned when you run the rule. **Note:** The Test Rules button returns total number of members that are assigned both statically and dynamically. Therefore, the count of dynamic members assigned via the Execute Rule may not match the count that the Test Rules button returned.

Field or Control	*Description*
Execute Rule(s)	Click to launch the DYNROLE_PUBL application engine program, which executes the rule(s). The application engine program runs through PeopleSoft Process Scheduler. After the DYNROLE_PUBL application engine program runs the rule, it publishes a message, ROLESYNCHEXT_MSG, that contains a list of users in the role. **Note:** The successful completion of the dynamic roles program does not ensure that your roles were updated; the associated message must also be published successfully. You can also execute dynamic role rules for all roles and users. See Executing Dynamic Role Rules.
Process Monitor	Select to view the status of the DYNROLE_PUBL application engine program in the Process Scheduler Monitor.
Service Monitor	Select to check the status of the publication of the ROLESYNCHEXT_MSG message in the Service Operations Monitor. See Executing Dynamic Role Rules.

Specifying Workflow Settings

Access the User Profiles - Workflow page (select **PeopleTools** > **Security** > **User Profiles** > **User Profiles** and click the **Workflow** tab).

his example illustrates the fields and controls on the User Profiles - Workflow page.



Workflow Attributes

Field or Control	*Description*
Alternate User ID	Select an alternate role user to receive routings sent to this role user. Use this option when the role user is temporarily out (for example, on vacation or on leave). If the field contains a role user name, the system automatically forwards new work items for whoever is assigned as the current role user to the alternate role user. **Note:** The system forwards *new* work items to the alternate role user. It does not reassign items already in the user's worklist. **Note:** When applying an alternate user ID in your workflow settings, make note of the fact that the system only sends workflow routings to the immediate alternate user ID. The system does not send routings down multiple levels of alternate user IDs. For example, assume user A specifies user B as the alternate user ID while user A is out of the office. Also assume that user B is out of the office at a time during user A's absence, and user B specifies user C as an alternate user ID for this time. In this case, the system does not send workflow routings originally intended for user A to user C. **Note:** The Alternate User ID routing functionality works only with role-based applications, such as Virtual Approver (VA) Workflow in PeopleTools, and Enterprise Component Approval Framework. In VA Workflow, the route is to roles, not specific users. And where the Enterprise Component Approval Framework worklist uses roles, the Alternate User ID routing functionality works. The workflow field mapping must be mapped to a role or a role query in order for alternate user to work.
From Date and **To Date**	Enter the date on which the current role user is going to begin and return from a temporary vacancy. This field specifies the time period that the alternate user ID is used.

Field or Control	*Description*
Supervising User ID	Select the user ID of the user's supervisor from this drop-down list box. The system uses this value when it needs to forward information to the user's supervisor. The system uses the JOB record to determine the user's supervisor. **Note:** If you are using PeopleSoft Human Capital Management (PeopleSoft HCM) applications, this field should not appear. If it does, you must set your workflow system defaults.
Routing Preferences	Specify the routing types that this role user can receive. The **Routing Preferences** box shows the two places where the system can deliver work items: to a worklist or to an email mailbox. If the user does not have access to one or both of these places, deselect the check box. For example, if this person is not a PeopleSoft user, deselect **Worklist User.**

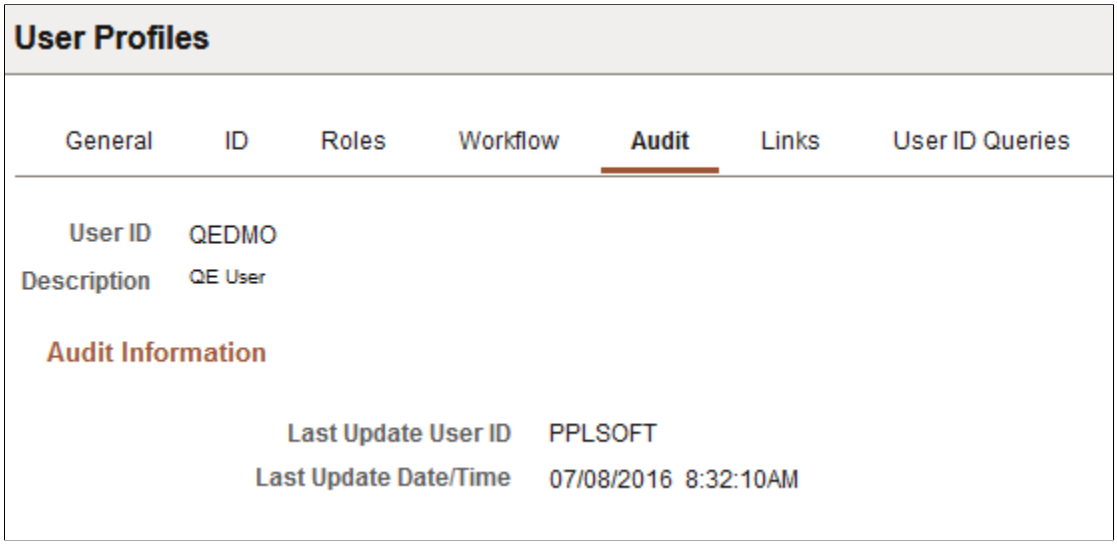
Reassign Work

Field or Control	*Description*
Reassign Work To	Use to reassign pending work for this role user if positions change or a user is temporarily out, such as on leave or on vacation. If this user has work items waiting (as shown by the Total Pending Worklist Entries in your Workflow interface), select this check box and select the user to whom work items should be forwarded from the drop-down list box. When you save the page, the system reassigns existing worklist entries to the specified user. **Note:** If you don't reassign pending work items, they remain unprocessed.
Total Pending Worklist Entries	Displays the number of worklist items that require a user's attention.

Viewing When a User Profile Was Last Updated

Access the User Profiles - Audit page (select **PeopleTools** > **Security** > **User Profiles** > **User Profiles** and click the **Audit** tab).

This example illustrates the fields and controls on the User Profiles - Audit page.



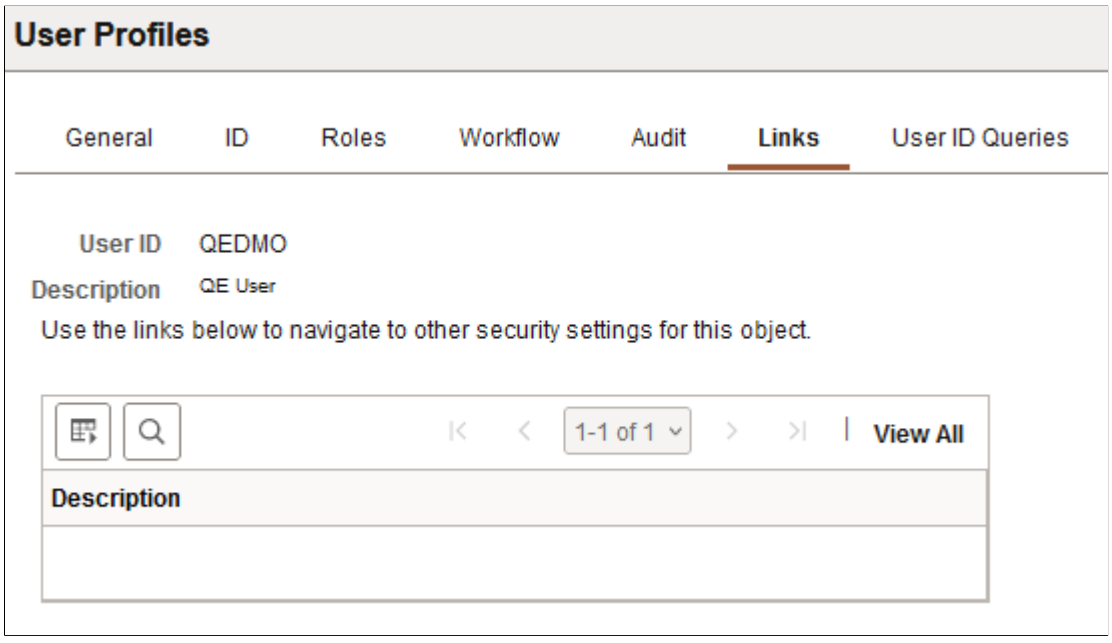
The User Profiles - Audit page is a display-only page that lists:

- When a profile was last updated (date and time).
- Who updated the profile (User ID).

Displaying Additional Links

Access the User Profiles - Links page (select **PeopleTools** > **Security** > **User Profiles** > **User Profiles** and click the Links tab).

This example illustrates the fields and controls on the User Profiles - Links page.



Use this page to access links to other pages within your PeopleSoft system. For example, perhaps a PeopleSoft application requires a specific security setting to be associated with a user profile. If this

application-specific setting appears on a page not in PeopleTools Security, add a link to the application page so that anyone updating the user profile can easily navigate to the page.

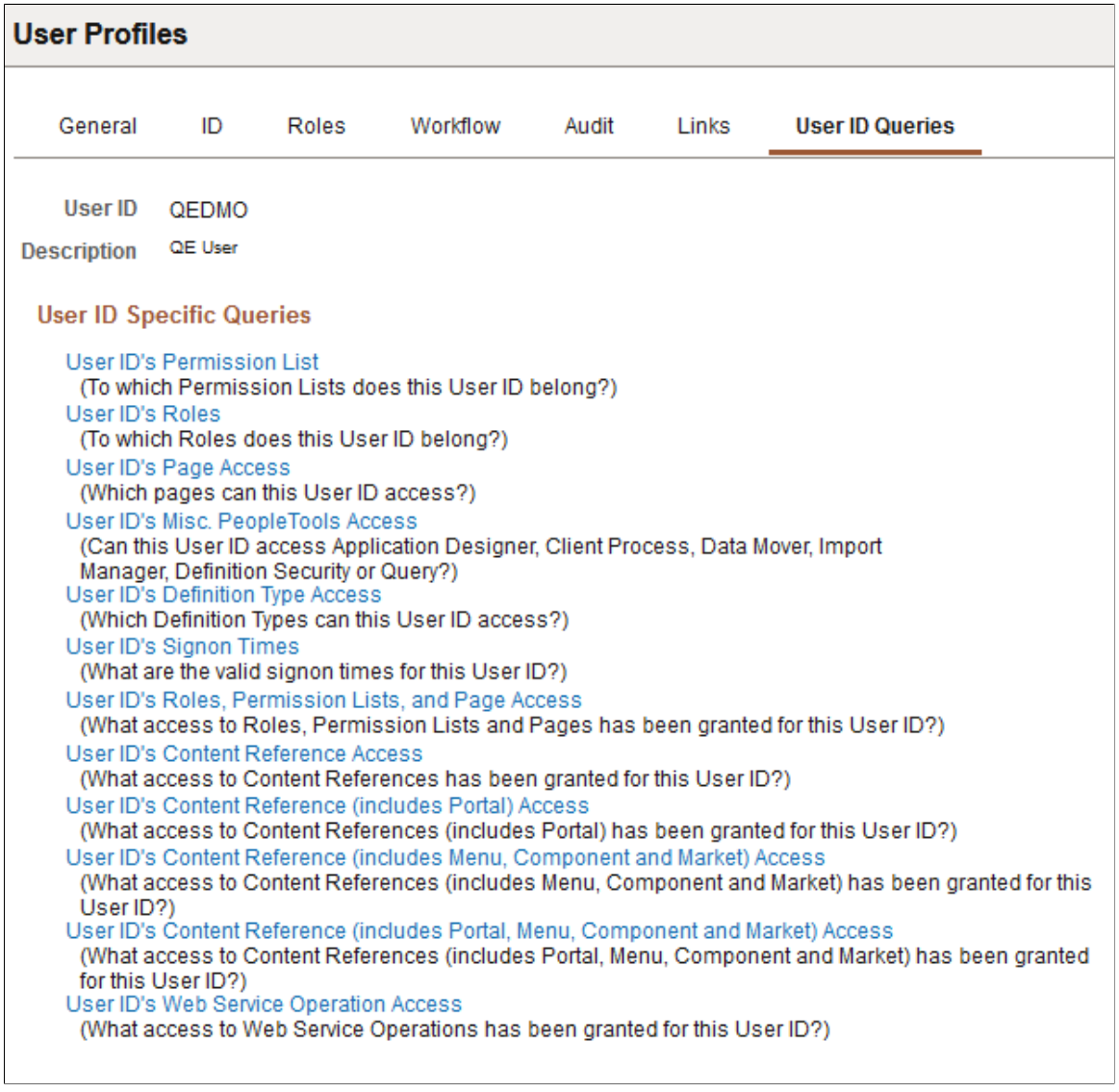
Note: The Links page is read-only. You create the inventory of links to pages that exist outside of PeopleTools Security by using the Security Links component.

If you added links for user profiles in the Security Links component, they appear on the Links page.

Running User ID Queries

Access the User Profiles - User ID Queries page (select **PeopleTools** > **Security** > **User Profiles** > **User Profiles** and click the **User ID Queries** tab).

This example illustrates the fields and controls on the User Profiles - User ID Queries page.



User ID queries enable you to run queries that provide detailed information about a user profile, such as the permission lists and roles associated with the user profile. The available queries are documented on the page.

To run a user ID query:

1. Click the link associated with the query that you want to run.

   This action invokes a new browser window.

2. View the information that the query returns to the new browser window or select a download option.

   For downloading, you have the following options:

   - Excel Spreadsheet: Downloads the query results as an Excel spreadsheet (.xls) file.
   - CSV Text File (comma-separated values text file): Downloads the query results as a CSV (.csv) file.
   - XML file: Downloads the query results as an xml (.xml) file.

Working With Passwords

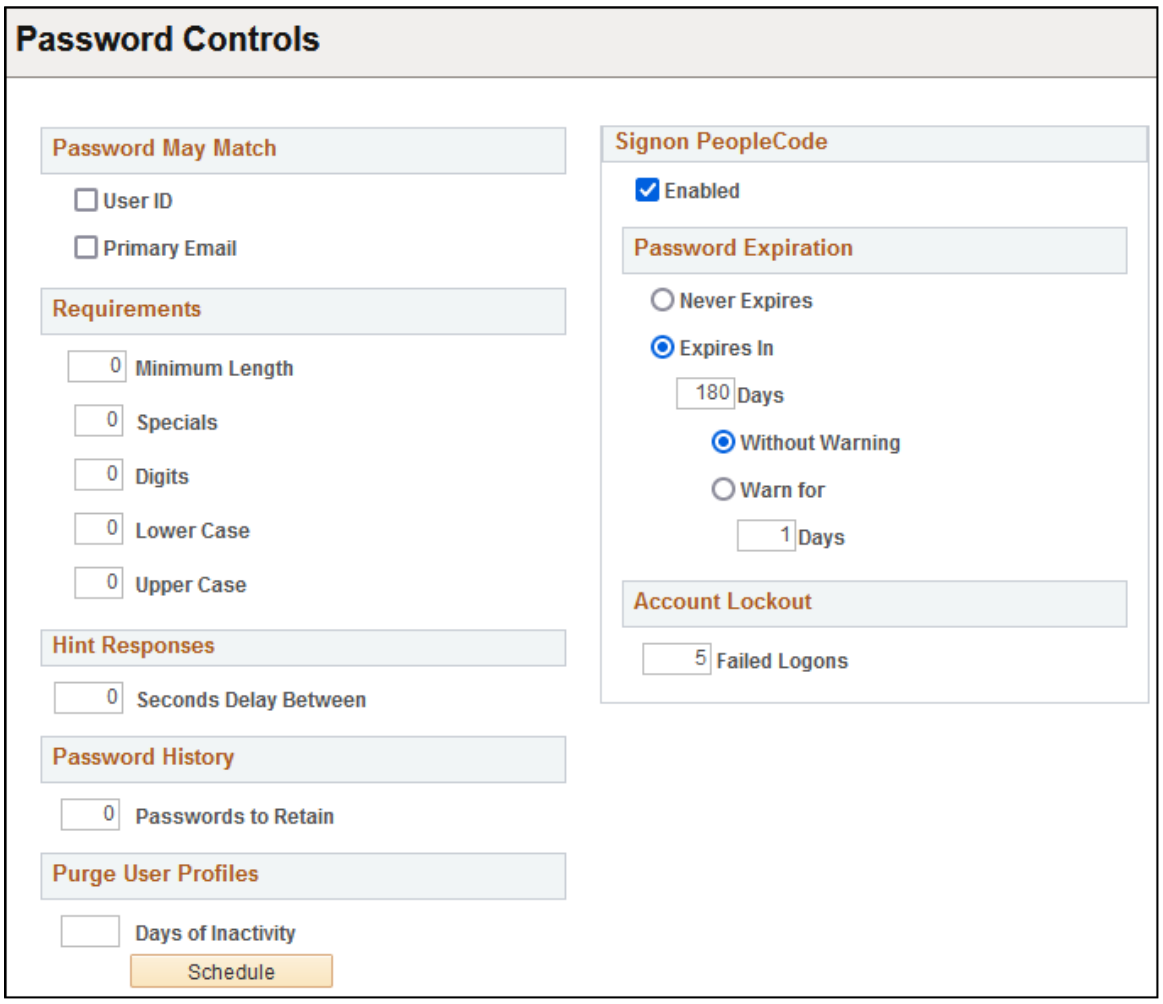
This section discusses how to:

- Set password controls.
- Change passwords.
- Create hints for forgotten passwords.
- Delete hints for forgotten passwords.
- Define answers for forgotten password hints.
- Create email text for forgotten passwords.
- Create email text for incorrect hint responses.
- Set up the site for forgotten passwords.
- Request new passwords.

Setting Password Controls

Access the Password Controls page (**PeopleTools** > **Security** > **Password Configuration** > **Set Password Controls**).

This example illustrates the fields and controls on the Password Controls page.



You use the Password Controls page to set any password restrictions, such as duration or minimum password length, that you want to impose on your end users. These options apply when you are maintaining your user profiles within PeopleSoft databases.

Important! PeopleTools delivers the Password Controls page with a number of default field values. When you perform a standard database installation the default values are set. The default values are not automatically set during an upgrade.

The following tables described the fields on the Password Controls page, including any default field values delivered.

Password May Match

Field or Control	*Description*
User ID	Select to enable users to use their own user ID as a password. By default the control is not selected and users cannot use their user ID as a password.

Field or Control	*Description*
Primary Email	Select to enable users to use the email address that is associated with their user profile (as designated by the **Primary Email Account** check box on the Email Address page) as a password. By default the control is not selected and users cannot use their email address as a password.

Note: Clearing these controls helps you prevent hackers from guessing passwords based on a list of employee names.

Requirements

Use these fields to specify the number and types of characters that passwords *must* include. Passwords can include up to 64 characters.

Field or Control	*Description*
Minimum Length	Enter the value that determines the *fewest number of characters* that a user must enter when creating his password. The default value is *8* characters. If the minimum length is set to *0,* then the PeopleSoft password controls do not enforce a minimum length on the password; however, the password cannot be blank. When you create a new user or a user changes a password, the system checks this value. If it is not zero, then the system tests the password to ensure it meets length requirements and if it does not, an error message appears.
Specials	Enter the required number of special characters that the password must include. All special characters are allowed, but spaces are not allowed in the password. The default value is *0.*
Digits	Enter the required number of integers, such as *1* or *2,* that the password must include. The default value is *0.*

Field or Control	*Description*
Lower Case	Enter the required number of minuscule letters (such as "q" or "i") that the password must include. The default value is *0*.
Upper Case	Enter the required number of majuscule letters (such as "Q" or "I") that the password must include. The default value is *0*.

By default, leading, intermediate, and trailing white spaces are not supported in PeopleSoft passwords. If your security policy requires that you allow intermediate white spaces, you must comment out the following USERMAINT.GBL.PSOPRDEFN.SaveEdit Component PeopleCode:

```
&find = Find(" ", PSOPRDEFN.OPRID); If &find > 0 Then Error MsgGet(48,
14, "Message not found."); End-If;
```

Warning! When these statements are commented out, users can include intermediate white spaces in passwords. Although you can use the preceding PeopleCode modification as a workaround, it is *strongly recommended* that you not do so. This modification can cause unexpected behaviors that are problematic for batch processes, upgrades, application server configuration files, and two-tier applications, such as PeopleSoft Application Designer, Data Mover, Application Engine.

Hint Responses

Field or Control	*Description*
Seconds Delay Between	The setting controls the length of time to wait between processing consecutive hint responses regardless if the response is correct. The default value is *0*.

Password History

Field or Control	*Description*
Passwords to Retain	Enter the number of user passwords to retain in the password history table (PSPSWDHISTORY). The default value is *0.* If the user attempts to reuse a password that is stored in the password history table, the application issues an error and prompts the user to enter a different password. When the number of retained passwords for a user surpasses the number indicated in the **Passwords to Retain** field, the system deletes the oldest password and then stores the current password as the newest password.

Note: If the password history table contains values and you change the **Passwords to Retain** field value to *0,* the system deletes the password history for all users.

Purge User Profiles

Field or Control	*Description*
Days of Inactivity	Enter the maximum number of days that a user can go without accessing the application, after which the system marks the profile as inactive. By default the field is blank. After you set the value and save the page, click the **Schedule** button to access and automate the PURGEOLDUSRS Application Engine program that performs the delete process. If you maintain user profiles in a directory server, a row is added to the PSOPRDEFN table for the system to access while the user interacts with the system. However, when the user is deleted from the directory server, you must manually delete the row in PSOPRDEFN associated with the deleted user profile.

Signon PeopleCode

Field or Control	*Description*
Enabled	Select the box to enable the PeopleSoft Password Expiration and Account Lockout fields. By default this option is *Enabled.* You must restart the application server whenever you change this setting. You can extend or customize the controls by modifying the PeopleCode.

Field or Control	*Description*

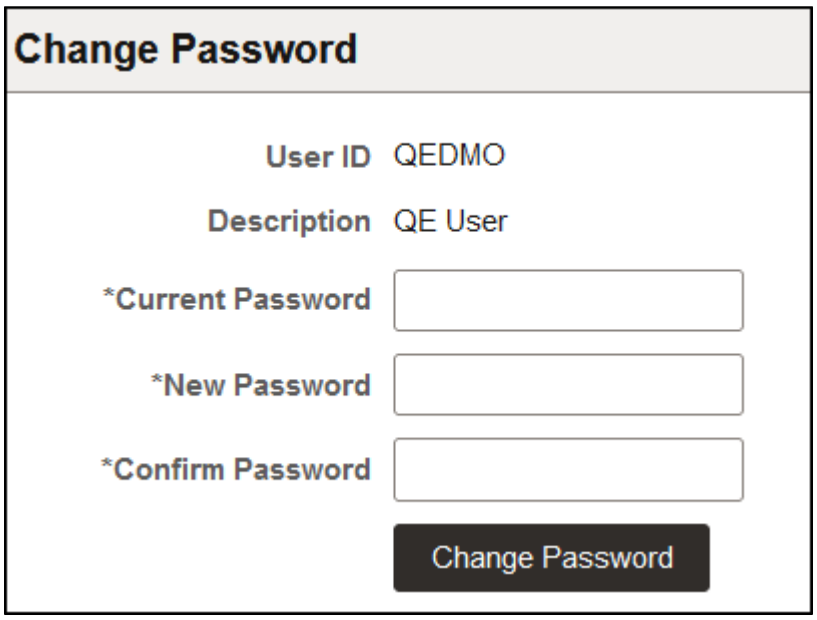
Password Expiration	Use the controls in this section to manage password expiration options:
	• **Never Expires:** Select to disable password expiration options for all users.
	• **Expires In:** (Default) Select to set password expiration options for all users.
	  • **Days:** You must enter a value between *1* and *365* in the **Days** field to specify the number of days that a password is valid.
	  The default value is *180* days.
	  Users signing on after a password expires must change their password to sign in.
	  You must select a warning option.
	  • Without Warning: (Default) Select to disable notification of impending password expiration.
	  • Warn For: Select to enable notification of impending password expiration.
	  The value that you enter in the **Days** field determines when the system begins notifying users of impending password expiration.
	  The default value is *5*.
	PeopleSoft delivers a default permission list named PSWDEXPR (Password Expired). When a user's password expires, the system automatically removes all of the user's roles and permission lists, and temporarily assigns them the PSWDEXPR permission list only.
	A user whose password has expired can access only items in the PSWDEXPR permission list, which typically grants access to only the Change Password component (CHANGE_PASSWORD). For the duration of the session, as in until the user changes the password, the user is restricted solely to the PSWDEXPR permission list.
	Note: The actual user profile stored in the database is not changed in any way when the password expires. You do not need to redefine the profile. When the password is changed, the system restores the user profile's previous roles and permission lists.

Field or Control	*Description*
	Note: The password expiration applies only to signing into the system through the PeopleSoft Pure Internet Architecture (PIA). When you log in to PeopleTools utilities such as Application Designer, Application Engine, or Data Mover, the password expiration control does not apply. For example, if you try to use an expired password to sign in to PIA, you will see an error message, but you can use the same password to sign into Application Designer.
Account Lockout	**Failed Logons:** Enter the maximum number of failed sign in attempts to allow before the system disables the user profile. The default value is *5*. For example, if you set the **Failed Logons** value to 3, and a user fails three sign in attempts, she is automatically locked out of the system. Even if she correctly enter a user ID and password on the fourth attempt, she is not permitted to sign in. This feature reduces the risk of any intruders using brute force to break into your system. After an account is locked out, a system administrator must open the user profile and deselect the **Account Locked** check box manually.

Changing Passwords

Access the Change My Password page (select **Change My Password** from the NavBar menu). The PeopleSoft system enables users to change their passwords as needed.

This example illustrates the fields and controls on the Change Password page.



To change a PeopleSoft password:

1. From the homepage, click **Change My Password**.
2. On the Change Password page, enter the current password in the **Current Password** field.
3. In the **New Password** field, enter a new password.
4. Confirm the new password by entering it again in the **Confirm Password** field.
5. Click **Change Password**.

Note: For troubleshooting, the administrator may check the entered values and so on through the PeopleCode that supports the page.

Implementing Forgotten Password Emails

Set up hints and email text to allow end users who forget their passwords to request new, randomly generated passwords.

This setup assumes that the system is configured to send emails to end users. To allow users to request new passwords, the security administrator fulfills these requirements:

1. Configures the requirements for the replacements passwords.

   For example, specify the length and allowed characters. See Setting Password Controls.

2. Specifies an email address on the user profile.

   On the User Profile - General page, select **Edit Email Addresses** and add a valid email address. See Setting General User Profile Attributes.

3. Allows emails on one of the end user's permission lists.

On the Permission Lists - General page, select the option **Allow Password to be Emailed**. If this setting is not selected, the user is not allowed to receive the new password through email. If the user is allowed to receive new passwords through email, the user can request a new password. See Setting General Permissions.

4. Creates security questions (hints) that the end user must answer to continue with the email request.

   See Creating Hints for Forgotten Passwords.

5. Composes text for the email to send to end users who provide a valid user ID and answer the security question correctly.

   See Creating Email Text for Forgotten Passwords.

6. Composes text for the email to send to end users who do not answer the security question correctly.

   See Creating Email Text for Incorrect Hint Responses.

7. Sets up a web site for the end user request a replacement password.

   See Setting Up the Site for Forgotten Passwords.

When the prerequisite setup is complete, the end user who needs a new password:

1. Chooses a security question and supplies an answer.

   The Change or Set Up Forgotten Password Help page is where users select the security question and enter their answer into the system. See Defining Answers for Forgotten Password Hints.

2. Accesses the forgotten password page and enters their user ID.

3. Answers the security question.

   See Requesting New Passwords

Creating Hints for Forgotten Passwords

Use the Forgot My Password Hint page to define questions for users to answer as a means to authenticate themselves if they forget their password.

The security administrator sets up multiple questions, but users can only select one question to answer.

To access the Forgot My Password Hint page (PSPSWDHINT) select **PeopleTools** > **Security** > **Password Configuration** > **Forgotten Password Hints**.

This example illustrates the fields and controls on the Forgot My Password Hint page.

With these hints set up, users can access the Forgot My Password page. If the user answers the question correctly, a new password is sent through the email system.

To create a forgotten password hint:

1. Click Add a New Value.
2. On the Add a New Value page, enter a three-character ID in the **Password Hint ID** field.
3. Click Add.
4. Select the **Active** check box.
5. In the **Question** field, enter the question to use as a password hint.
6. Click the **Save** button.

Deleting Hints for Forgotten Passwords

To delete a password hint:

1. Select **PeopleTools** > **Security** > **User Profiles** > **Delete Forgotten Password Hint.**
2. Enter the specific code for the hint or perform a search for it.
3. On the Delete Forgot My Password Hint page, select the appropriate hint.
4. Click Delete.

Creating Email Text for Forgotten Passwords

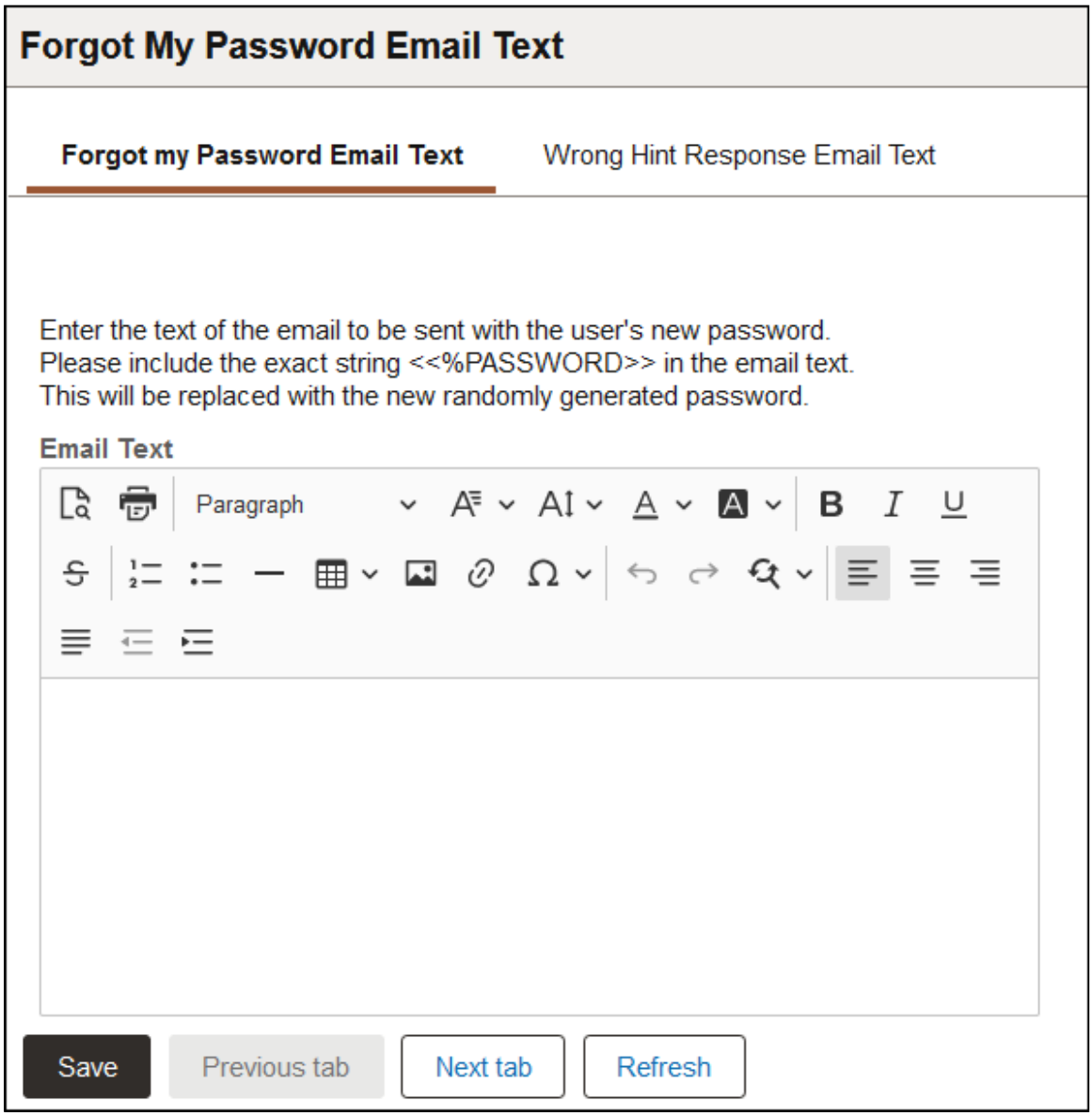
Before the system emails a new, randomly generated password to a user, you want to make sure they are who they claim to be. The Forgotten Password feature enables you to pose a standard question to users requesting a new password to verify the user's authenticity. If the user enters the appropriate response, then the system automatically emails a new password.

When a user has forgotten a PeopleSoft password, the system sends the user a new password within an email message. You can have numerous password hints, but typically, you send all new passwords

using the same email message template. Because of this, PeopleSoft provides a separate page just for composing the standard email text that you use for your template.

To access the Forgot My Password Email Text page select **PeopleTools** > **Security** > **Password Configuration** > **Forgot My Password Email Text** and click the Forgot My Password Email Text tab.

This example illustrates the fields and controls on the Forgot My Password Email Text page.



For information on the rich text editor interface, see “Working With Rich Text Editor Fields” (Applications User’s Guide).

Add the following text string in the **Email Text** field:

```
<<%PASSWORD>>
```

The system inserts the new password here. The *%PASSWORD* variable resolves to the generated value.

Note: You might instruct the user to change the password to something easier to remember after they sign in to the system with the randomly generated password. Only users who have the **Allow Password to be Emailed** option enabled on the Permission List - General page can receive a new password using this feature.

For example:

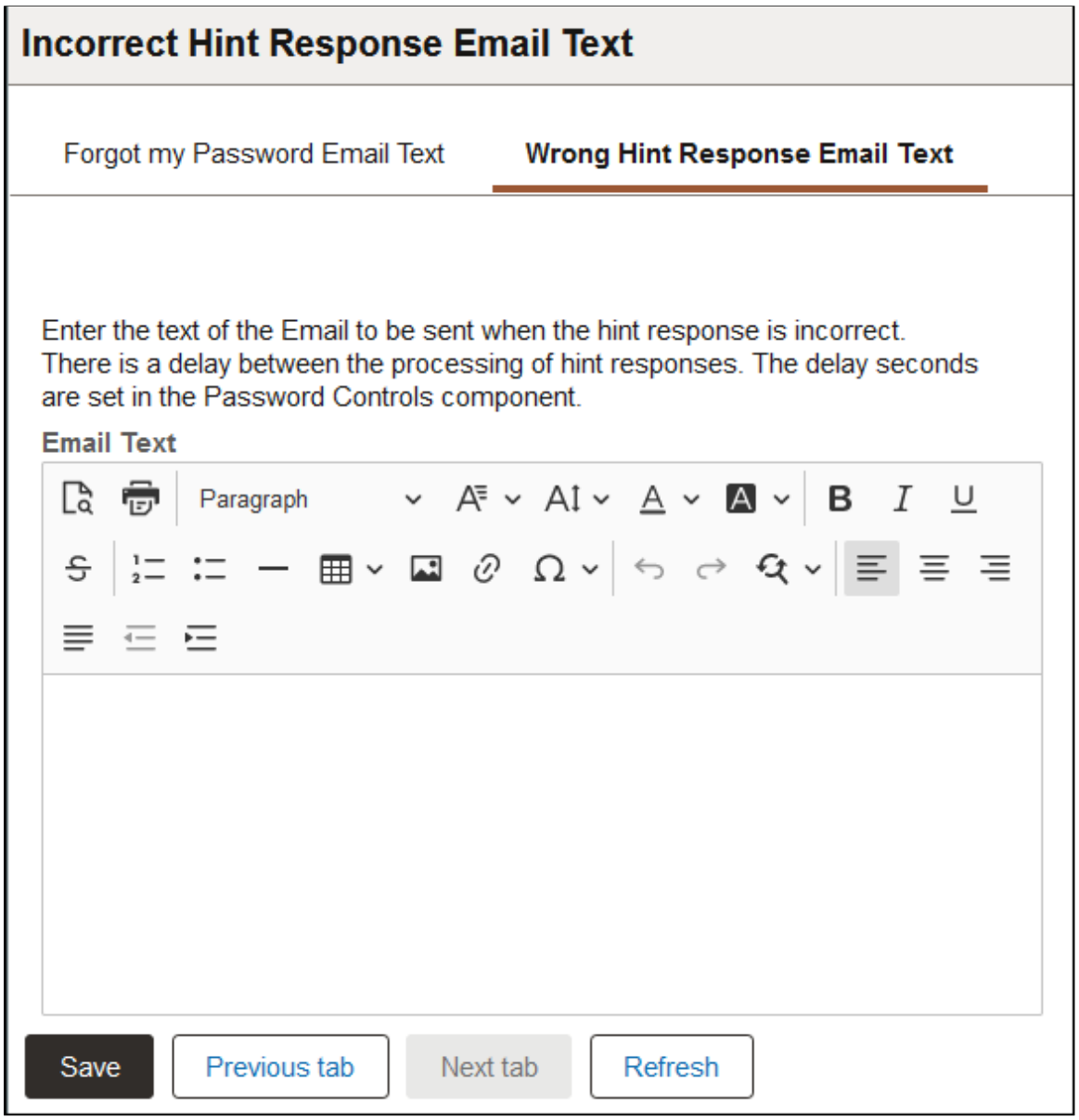
```
Your new password is <<%PASSWORD>>.

To change this system-generated password, from the Main Menu click the Change Passw⇒

ord link.
```

Creating Email Text for Incorrect Hint Responses

If a user provides an incorrect response to a password hint question, the system can automatically send an email notification to the user that indicates that they provided an incorrect response.

Use the Incorrect Hint Response Email Text page (EMAILHINTFAIL) to compose a generic message that the system sends to users if they enter an incorrect response to a password hint. To access the page select **PeopleTools** > **Security** > **Password Configuration** > **Forgotten Password Email Text** and click the Wrong Hint Response Email Text tab.

This example illustrates the fields and controls on the Incorrect Hint Response Email Text page.



Enter any message that suits your business requirements. Keep in mind that the same message is sent to all users who provide an incorrect password hint response.

You can change the delay between the processing of hint responses on the Password Controls page in the Seconds Delay Between field. See Setting Password Controls.

For information on the rich text editor interface, see “Working With Rich Text Editor Fields” (Applications User’s Guide).

Setting Up the Site for Forgotten Passwords

PeopleSoft recommends that the security administrator sets up a site specifically designed for users who have forgotten their passwords. This site would require no password to enter, but it would provide access only to forgotten password pages.

To set up a forgotten password site:

1. Set up a separate PeopleSoft Pure Internet Architecture site on your web server.
2. Set up a direct connection to the site, such as a link to it.
3. In the web profile, enable public access and specify a public user ID and password for automatic authentication.

   This *direct* user should have limited access, for example, only to the Email New Password component. Users go directly to it, and a new password is emailed.
4. Place a link to the forgotten password site within the public portion of the PeopleSoft portal or on another public web site.
5. Notify your user community of the link.

Note: The URL for the site should have this format: http://<*webserver*>/psp/<*sitename*>/<*portalname*>/<*localnodename*>/c/MAINTAIN_SECURITY.EMAIL_PSWD.GBL?

Defining Answers for Forgotten Password Hints

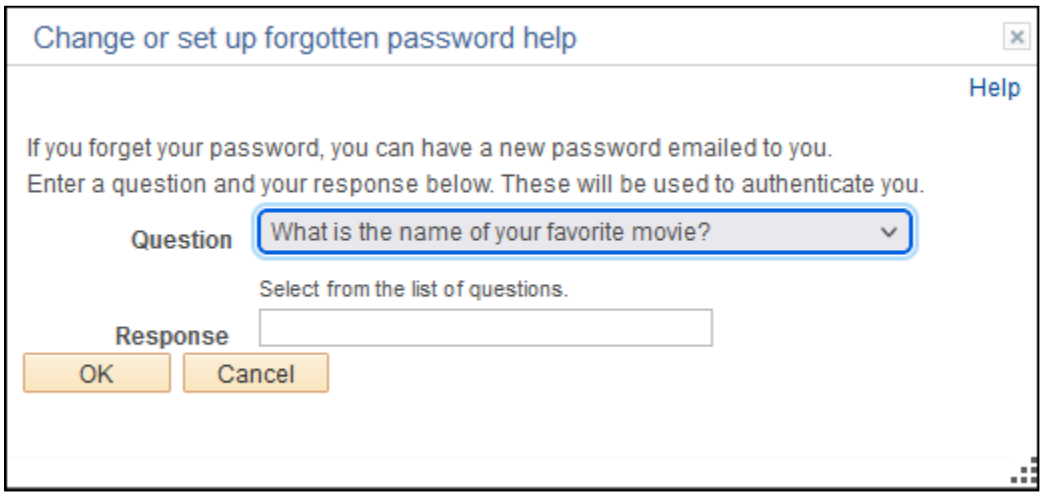
End users can use the Change or Set Up Forgotten Password Help page (USER_PSWDHINT) to define an answer to a predefined password hint question set up by the system administrator.

If you forget your password, the system will present you with a security question. When you provide the answer, the system emails you instructions to reset your password.

Select **My System Profile** from the NavBar menu and click the link Change or set up forgotten password help.

See “Setting Up Your System Profile” (Applications User’s Guide).

This example illustrates the fields and controls on the Change or set up forgotten password help page.



Field or Control	*Description*
Question	This field contains the security question set up by the administrator.
Response	Enter the answer to the question.

Requesting New Passwords

This section describes how an end user requests new passwords.

Prerequisites for Requesting New Passwords

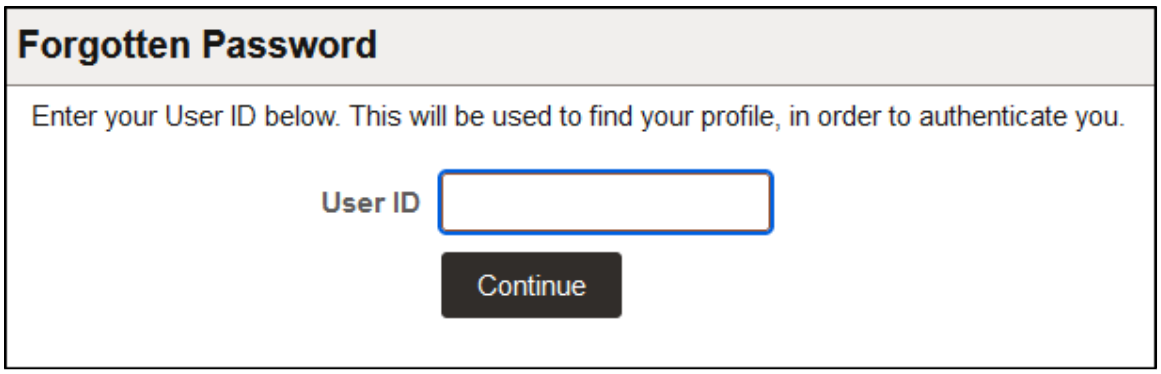
Before the system can email the user a new password, the security administration must complete the requirements in Implementing Forgotten Password Emails.

Specifying the User to Validate

Use the Forgot My Password page to specify the ID of the user to validate

To access the Forgot My Password page, click the **Forgotten Password** link on the PeopleSoft signon page or use a link as provided by the security administrator.

This example illustrates the fields and controls on the Forgotten Password page.



To specify the user to validate:

1. In the **User ID** field enter the user name to validate.
2. Click the **Continue** button.

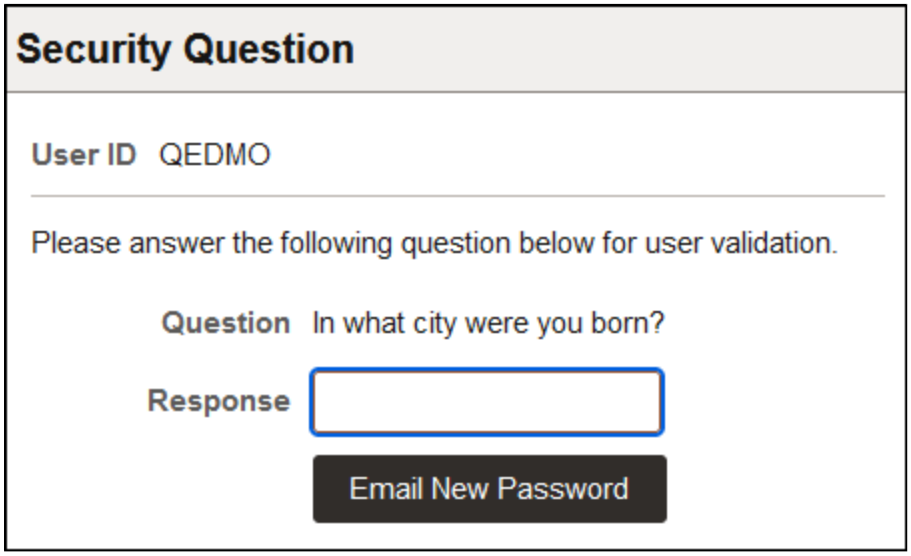
For security purposes no indication is provided if a user enters a correct user ID or an incorrect user ID. If an incorrect user ID is entered, a user is able to proceed in the process, but the password reset will not be successful.

At the end of the procedure the system displays a message advising users to contact their security administrator or system administrator if the password reset is not successful, and users who inadvertently entered an incorrect user ID may contact their administrator for assistance.

Entering Password Hint Responses

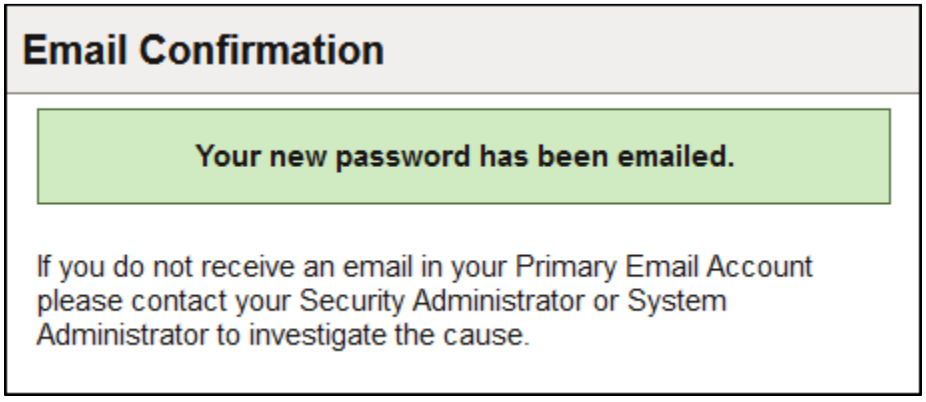
After you enter the user ID to validate on the Forgot My Password page, you are presented with a question to answer.

This example illustrates the fields and controls on the Security Question page.



After a user enters a response to the password question and clicks the **Email New Password** button, the system displays a confirmation that the password has been emailed to the primary email address defined for the user.

This example illustrates the Email Confirmation page.



In the interest of security, the system does not provide feedback if a correct response is entered for the password question or if an incorrect response is entered.

If the user enters a valid user ID in the previous step and enters the correct response to the password question, a new password is emailed to the primary email account as defined in the user profile, provided that the administrator has satisfied the prerequisites described previously in this section.

If the security administrator has configured the Incorrect Hint Response Email Text message as described previously in this topic, at the end of the procedure the system sends an email to the address defined in the user profile providing information and instructions as determined by the administrator.

If the user did not enter a valid user ID in the previous step, he or she is able to enter a response to the password hint. However, no new password generation is performed.

To enter a password hint response:

1. In the **Response** field enter the answer to the question.
2. Click the **Email New Password** button.

   The Password Emailed page appears.

Implementing Distributed User Profiles

This section provides an overview of distributed user profiles and discusses how to:

- Define user profile access for remote security administrators.
- Define remote security administrator role grant capability.
- Administer distributed user profiles.

Understanding Distributed User Profiles

As your user population increases in size, it can become impractical for one person to centrally administer all of your system's user profiles. You can distribute some or all user profile administration tasks by enabling selected users to use the Distributed User Profiles component (USERMAINT_DIST) to control the granting of selected roles to other users.

The pages in the Distributed User Profiles component are identical to the corresponding pages in the User Profiles component, except that its User Roles page does not include links for editing the assigned roles. You can restrict who can use the component, which users they can administer, and what roles they can grant, based on the roles to which they themselves belong. For example, you might specify that users in the Line Manager role can grant the Shipping Clerk role to other users. The effect of this is to designate line managers as *remote security administrators* who can administer the user profiles of shipping clerks. In addition to granting and managing roles, a remote security administrator can administer all parts of a user profile, including passwords, email addresses, and workflow.

Important! Distributing user profile administration might affect regulatory compliance (for example, Sarbanes Oxley). You are responsible for determining and accounting for any effect of using this feature.

To implement distributed user profiles:

1. Use permission lists and roles to configure security to give selected remote security administrators access to the Distributed User Profiles component.

   Note: The PIA navigation path to this component is **PeopleTools** > **Security** > **User Profiles** > **Distributed User Profiles**.

2. Use the Set Distributed User Profile Search Record page to define which user profiles can be administered with the Distributed User Profiles component.

See Defining User Profile Access for Remote Security Administrators.

3. Use the Role Grant page in the Roles component (ROLEMAINT) to specify which roles your remote security administrators can grant with the Distributed User Profiles component.

   See Defining Remote Security Administrator Role Grant Capability.

Defining User Profile Access for Remote Security Administrators

To define user profile access:

1. Define a search record that returns only the user IDs that you want remote security administrators to be able to administer.

   Note: Initially, PSOPRDEFN_SRCH is the default search record for this purpose. You can accept the default and skip this step, but that action enables access to every user profile in your system. We encourage you to define a more restrictive search record.

2. In a browser, select **PeopleTools** > **Security** > **User Profiles** > **Distributed User Setup** to access the Set Distributed User Profile Search Record page.

3. In the **New Search Record** field, select the search record that you defined in Step 1, and then save.

   When remote security administrators access the Distributed User Profiles component, this search record enforces row-level security to restrict the set of user IDs that they can select and administer.

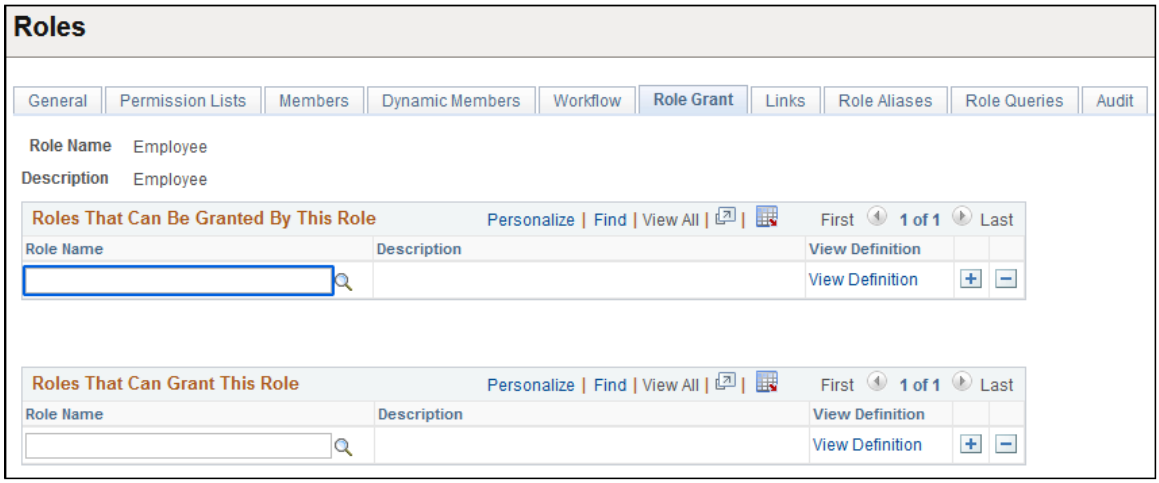
Related Links

"Using Search Records" (Application Designer Developer's Guide)

Defining Remote Security Administrator Role Grant Capability

In a browser, select **PeopleTools** > **Security** > **Permissions and Roles** > **Roles** > **Role Grant** to access the Roles - Role Grant page.

This example illustrates the fields and controls on the Roles - Role Grant page.



You use this page to specify which roles can be granted using the Distributed User Profiles component and which users can grant them. This page is part of a role definition; you can configure this role to be a remote security administrator, a role that a remote security administrator can grant to users, or both.

Field or Control	*Description*
Roles That Can Be Granted By This Role	By specifying one or more roles in this grid, you effectively designate users who belong to roles, and who have access to the Distributed User Profiles component, as remote security administrators. Add rows to enable this role to grant as many roles as appropriate. For example, you might want users who belong to the *Shipping Manager* role to be able to grant the *Shipping Clerk (Temporary)* role and the *Packing Clerk (Temporary)* role to other users. **Note:** This grid is complementary to the **Roles That Can Grant This Role** grid, and it propagates its values accordingly. Using the example given, on the Role Grant page for the *Shipping Clerk (Temporary)* role and the *Packing Clerk (Temporary)* role, the **Roles That Can Grant This Role** grid now specifies *Shipping Manager*.
Roles That Can Grant This Role	By specifying one or more roles in this grid, you effectively designate users who belong to roles. and who have access to the Distributed User Profiles component,as remote security administrators, able to grant roles to users. Add more rows to enable additional roles to grant this role. For example, you might want users who belong to the *Security Administrator* role to be able to grant the *Shipping Manager* role to other users. **Note:** This grid is complementary to the **Roles That Can Be Granted By This Role** grid, and it propagates its values accordingly. Using the example given, on the Role Grant page for the *Security Administrator* role, the **Roles That Can Be Granted By This Role** grid now specifies *Shipping Manager*.
View Definition	Click to view the associated role definition and ensure that you have selected the appropriate role to grant or to serve as a remote security administrator.

Administering Distributed User Profiles

In a browser, select **PeopleTools** > **Security** > **User Profiles** > **Distributed User Profiles** to access the Distributed User Profiles component.

Remote security administrators can fully edit the user profiles that they access through the Distributed User Profiles component, including granting roles.

The users who remote security administrators can administer are determined by the search record you specified on the Set Distributed User Profile Search Record page.

The roles that a given remote security administrator can grant are determined by the selections that you made on the Roles - Role Grant page.

Related Links

Specifying User Profile Attributes

Transferring Users Between Databases

You occasionally need to copy security information from one database to another. Typically, you do this as part of an upgrade or to transfer security information from your production environment to your development or testing environment. PeopleTools provides a set of Data Mover (DMS) scripts designed to export and import user profile security information. The provided scripts transfer user profile data from a source to a target database using these tables:

- PSOPRDEFN
- PSOPRALIAS
- PSROLEUSER
- PSUSERATTR
- PSUSEREMAIL
- PSUSERPRSNLOPTN
- ROLEXLATOPR
- PS_RTE_CNTL_RUSER

Note: Use Application Designer upgrade feature to upgrade both roles and permission lists.

One script exports User Profile data from the source database. The source database refers to the database that contains the User Profiles that you want to migrate. The target database refers to the database to which you are copying the user information.

After exporting the security information from the source database, you then run the import script against the target database. The target database refers to the database to which you want to transfer the security data. The scripts involved in transferring security information from one database to another are:

- USEREXPORT.DMS.

  This script exports User Profiles from the source database and stores them in a Data Mover DAT file. The output file is named USEREXPORT.DAT.

- USERIMPORT.DMS.

  This script reads the file created by USEREXPORT.DMS and copies the User Profile data into the target database.

You will find this set of scripts in the <*PS_HOME*>/scripts folder.

Considerations

Before running scripts to export and import your security information, you should consider these topics:

- Duplicate Rows

  If the target database already contains a row of data with identical keys to a row transferred by the import script, then the duplicate row will not be transferred to the target. The scripts make no attempt to merge the duplicate row; the row is not transferred.

  To ensure that you do not have data rows with duplicate keys, ensure that a User Profile in the source database does not exist in the target database with the same name.

  You should not have data rows with duplicate keys in your source and target databases when you begin the copy, as unexpected results may occur that will compromise database integrity.

- Release Levels

  Because the PeopleTools table structures change between major releases (8.59 to 8.60, for example), you cannot transfer users between databases that run different versions of PeopleTools. Before starting the migration process, upgrade your source and target databases so the release levels match.

Running the Scripts

Complete the following procedure to run the user transfer scripts:

1. Using Data Mover, sign in to the source database and run USEREXPORT.DMS for user definitions.

   You can edit this script to specify the location and file name of the output file and the log file.

2. Using Data Mover, sign in to the target database and run USERIMPORT.DMS for user definitions.

   You can edit the script to specify the location and file name of the input file and the log file. The name and location of the input file must match the output file you specified in Step 1.

3. After copying user and role definitions, run the PeopleTools audits, including DDDAUDIT and SYSAUDIT, to check the consistency of your database.

Tracking User Sign In and Sign Out Activity

This topic describes tracking user sign in and sign out activity.

Understanding the PSACCESSLOG Table

The PSACCESSLOG table includes the following fields. You can view the data in the table from the Access Log Queries page.

Note: PSACCESSLOG is a non-authoritative general view of the users of a PeopleSoft system. It should not be used as a critical part of PeopleSoft's security infrastructure. Oracle offers dedicated user management tools that do a better job of detailed user tracking.

PSACCESSLOG Record Field	*Query Result*	*Description*
OPRID	User	The PeopleSoft user ID.

PSACCESSLOG Record Field	*Query Result*	*Description*
LOGIPADDRESS	Client IP	The remote client's address at the time of sign on. The address is either an IP address or a host name as resolved by the PIA server's DNS server.
LOGINDTTM	Log In	The date and time that the client logged in, given in the database server's local time.
LOGOUTDTTM	Log Out	The date and time that the client logged out, given in the database server's local time. When the client logs in this field is initialized to LOGINDTTM.
NA	Duration (Min)	The time that the user is logged in to PIA, in minutes.
PT_SIGNON_TYPE	Signon Type	The valid values are: • *1* (one) - This indicates a user signing on to PIA. • *0* (zero) - This is another, non-PIA signon.

PSACCESSLOG Record Field	*Query Result*	*Description*
PT_SIGNOUT_REASON	Signout Reason	The reason the client signed out, if known. The valid values are: • - (hyphen, Not Set) The signon has not yet occurred. When the client signs on this field is initialized to this value. • *A* - user abandoned The user abandoned their PIA browser windows and the web server eventually closed the user session. • *E* - browser expired The browser window timed out due to inactivity and sent the web server notice of the expiration event. • *L* - user logout The user actively signed out of PIA. • *R* - user relogin While the user was signed on, a new login from the same browser session arrived for the same or another user. • *O* - other The signout type was not one of the other defined reasons.
PT_TRACING_ID	TRID	The Tracing ID assigned to the user's session.

Understanding Abandoned Sessions

Abandoned sessions are identified in PSACCESSLOG when the PT_SIGNOUT_REASON record field is *A*. An abandoned session can occur when:

- A user closes the browser’s last PIA page without signing out.
- A user closes the entire browser, which has signed-in PIA pages.
- A user puts a new URL in the current PIA page and navigates away to another web site.

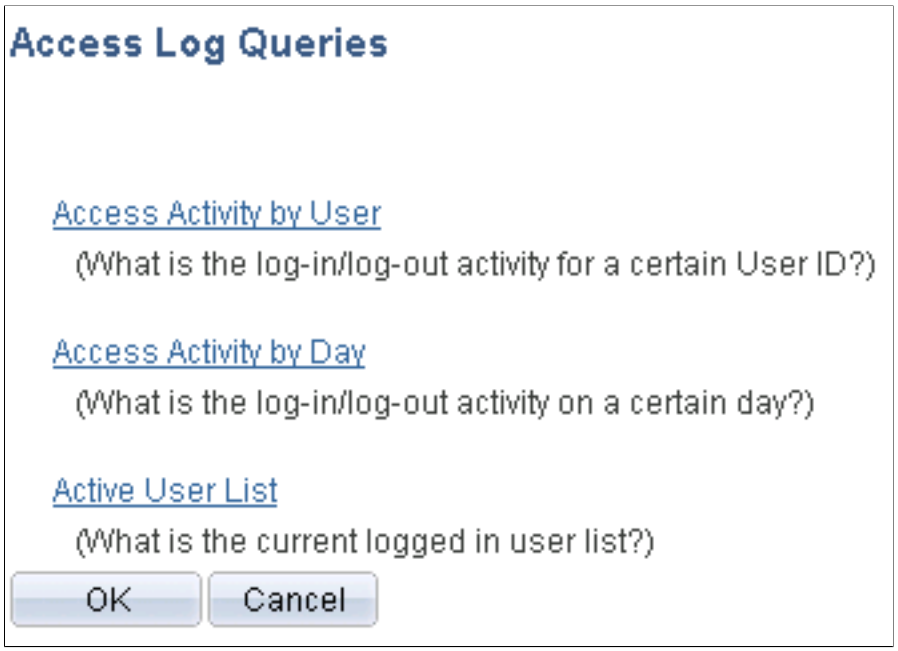
Recording of abandoned sessions requires the web server to retain its list of active sessions. If the web server crashes then all session information is lost. Users who are logged on when the web server crashes will appear as signed into the system and not as signed out due to an abandoned session.

Using Access Log Queries

PeopleSoft provides several queries to track user sign-in and sign-out activity based on the data in the PSACCESSLOG table.

Access the Access Log Queries page (select **PeopleTools** > **Security** > **Review Security Information** and click the **Access Log Queries** link on the Review Security Information page).

This example illustrates the fields and controls on the Access Log Queries page.



Select one of the following queries:

- Access Activity by User

  View a single user's sign-in and sign-out activity (public query PT_SEC_ACCESSLOG_USER). Choose the User ID and click View Results. This log includes the following items:

  - Client IP
  - Log In
  - Log Out
  - Signon Type
  - Signout Reason
  - TRID

- Access Activity by Day

View one or more days of all user sign-in and sign-out activity (public query PT_SEC_ACCESSLOG_DAY). Choose from and through dates, and click View Results. This log includes the following items:

- User
- Client IP
- Log In
- Log Out
- Signon Type
- Signout Reason
- TRID

- Active User List

  View the users who are currently signed in to the application in the browser (public query PT_SEC_CURUSER_LIST). This log includes the following items:

  - User
  - Client IP
  - Log In
  - Duration in minutes
  - Signon Type
  - TRID

These logs are generated using data from the PSACCESSLOG table. If you are not interested in monitoring access activity, you can delete the PSACCESSLOG table. Deleting this table has no negative effect.

Note: If you delete the PSACCESSLOG table and then decide that you would like to track user sign-in and sign-out activity, you must recreate the table. Use Application Designer to open the PSACCESSLOG record definition and create the table.

Tracking User Sign-In Attempts

This topic discusses how to:

- Enable user sign-in attempt tracking.
- View results in the PSPTLOGINAUDIT table.

Understanding Tracking User Sign-In Attempts

For every sign on attempt to the PeopleSoft Pure Internet Architecture (PIA) the following information can be logged:

- Information about the last successful login, including timestamp and authentication type.
- Information about the last failed attempt, including timestamp, failed logic count and authentication type.

The information is logged in and can be queried from the PSPTLOGINAUDIT table.

Understanding Abandoned Sessions and the PSPTLOGINAUDIT Table

The information captured in the PSPTLOGINAUDIT table does not account for abandoned PIA sessions.

An abandoned session can occur when:

- A user's PIA session times out.
- A user puts a new URL in the current PIA page and navigates to another web site.

If a user does not sign out of the system properly, he or she appears as still signed into the system.

Enabling User Sign-In Attempt Tracking

To track user sign-in attempts:

1. Access the application server configuration file.
2. Locate the Security section.
3. Set the **Enable Login Audit** option equal to one of the following values:
    - *Y.* (Default.) Enable login audits.
    - *N.* Disable login audits.
4. Save the file.

You do not need to reboot the application server when you define or change this parameter.

Viewing Results in the PSPTLOGINAUDIT Table

When the **Enable Login Audit** parameter is enabled, the following information is stored in the PSPTLOGINAUDIT table:

Please note the following:

- The system captures only information about sign-in attempts by operator IDs in the PSOPRDEFN table.
- The table does not record application server or process scheduler boot activity.
- The table does not capture two-tier sign on.

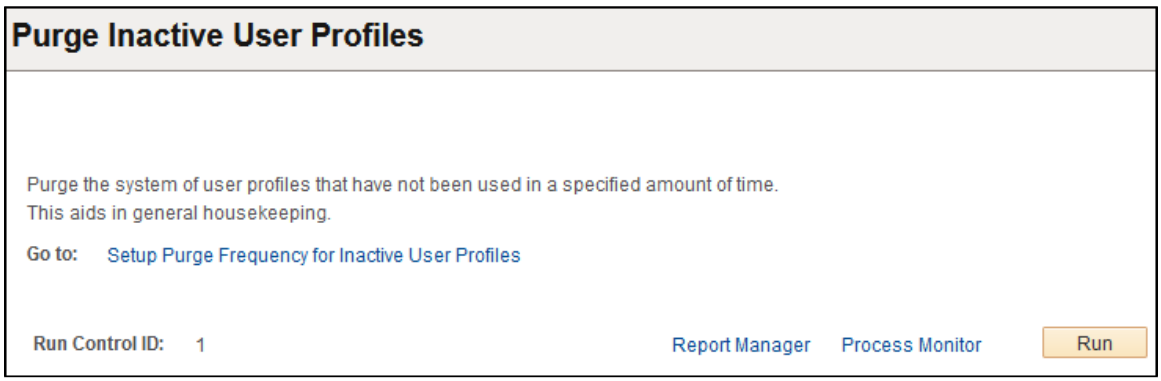
- The table records the first application server connection from the web server.

Field or Control	*Description*
PT_AUTH_TYPE	Displays the type of authentication used during the sign-in attempt. The valid values are: • *0.* Authentication token. • *1.* Database authentication. • *2.* Signon PeopleCode authentication.
OPRID	The user profile ID from the PSOPRDEFN table of the person who attempted to sign into the system.
PTSIGNONID	User ID used when the sign-in attempt was made. This may be different than the OPRID when LDAP user authentication or user ID aliases are in use.
PT_SIGNON_STATUS	Displays the status of the login attempt for a user. The valid values are: • *0.* Success. • *1.* Failure.
FAILEDLOGINS	Number of failed logins since the last successful login attempt. Note the following: • The value reflects the value recorded in PSOPRDEFN. FAILEDLOGINS. • The value will always be *0* (zero) on both successful and unsuccessful attempts, once a successful login occurs. • A non-zero value reflects that the most recent login attempt was a failure and the number of consecutive failures that occurred.
LASTSIGNONDTTM	Time at which the last sign on occurred.

Purging Inactive User Profiles

Access the Purge Inactive User Profiles page (select **PeopleTools** > **Security** > **User Profiles** > **Purge Inactive User Profiles**).

This example illustrates the fields and controls on the Purge Inactive User Profiles page.



Note: Before accessing this page, you must enter a run control ID.

See “Understanding Run Control IDs” (Process Scheduler).

This page enables you to access, run, and schedule the PURGEOLDUSRS Application Engine program. The PURGEOLDUSRS program deletes user profiles having an inactive status that exceeds the period specified in the **Purge Inactive User Profiles** section on the Password Controls page.

The **Setup Purge Frequency for Inactive User Profiles** link takes you to the Password Controls page, where you can enter a period (in days) under Purge Inactive User Profiles.

The Purge Inactive Users page is similar to the Delete User Profile page in that it invokes the process that removes all references to the user in any PeopleTools or application data table in which the OPRID field is a key. Before deleting user profiles, archive historical data according to local, state, and federal laws. Be sure to list historical and archival tables on the Tables to Skip page.

Related Links

Working With Passwords
Bypassing Tables During the Delete User Profile Process
Component Interfaces

Preserving Historical User Profile Data

Although you probably do not want to keep the permissions or sign-on access information for every user who has ever existed in the system, you generally do need to retain certain historical user profile data from your system. For example, local, state, and federal laws might demand that you retain certain employee history information. As another example, you might audit changes that users make to vital company data in the event you need to check that information a few months later if you discover some interesting financial allocations.

Use Data Archive Manager to archive and restore user profile data.

See “History Tables” (Data Management).

Important! Remember that deleting and purging user profile data deletes *every* row of data associated with a particular user profile from *every* table in which the OPRID field is a key field, including archived tables if they remain in your production database.

To preserve user profile information in a table for which the OPRID field is a key field, use the Bypass Tables page .

See Bypassing Tables During the Delete User Profile Process.

Chapter 7

Working with User Profiles Across Multiple PeopleSoft Databases

Understanding User Profile Synchronization

For implementations that use multiple PeopleSoft databases, you commonly have the same user in more than one database. Typically in production environments, you want the user profile information of the same user to be synchronized among databases. For example, if a user modifies her password or other user profile information in one database, you prefer that the system automatically synchronize the changes across the enterprise rather than have the user or an administrator manually replicate changes in multiple databases.

User profile synchronization involves setting up each PeopleSoft database in the enterprise to send and receive user profile updates through the Integration Broker. When you enter new profiles or modify and delete existing profiles on any publishing database and save, PeopleCode publishes a user profile service operation—which contains a user profile message—and routes the message to all subscribing nodes according to your specifications. The subscribing databases then update the user profile data with data from the publishing database.

Note: User profiles contain sensitive information. Design and implement user profile synchronization across different nodes with special care. As delivered, user synchronization behavior may not be acceptable in all cases.

Components Used to Update User Profiles

You can use these online components to make changes to user profile data:

- User Profiles (USERMAINT)
- Distributed User Profiles (USERMAINT_DIST)
- My System Profile (USERMAINT_SELF)
- Change My Password (CHANGE_PASSWORD)
- Expired Password (EXPIRE_CHANGE_PSWD)
- Forgot My Password (EMAIL_PSWD)

Administrators use the first two online components. The My System Profile component is a self-service component, which can be used to modify a limited set of data about a user. The Change My Password, Expired Password, and Forgot My Password components are used to change only the user password. Generally, the Forgot My Password component is configured as a public site that is separate from the PeopleSoft application. You can also modify user profile data through batch processes.

Types of User Profile Synchronization

PeopleSoft applications have two types of user profile synchronization:

- Default user profile synchronization.
- Configurable user profile synchronization.

The publishing processes for default and configurable user profile synchronization use different PeopleCode programs. PeopleSoft applications are delivered with the PeopleCode programs for both types of user profile synchronization. You select the appropriate PeopleCode by using the Security PeopleCode Options page. This page eliminates the need to access Application Designer to select the PeopleCode for the corresponding type of user profile synchronization.

Note: You should select the user profile synchronization type at the time of your implementation, after which you should restrict access to the Security PeopleCode Options page.

User Profile ID Types

Each user profile features an ID type (and related attributes) that are defined in the ID Types and Values section of the User Profile – ID page (PeopleTools, Security, User Profiles, User Profiles and click the ID tab).

To perform user profile synchronization, the ID type for a user profile must be the same on the source system and target system. That is, the system cannot map user profiles with different ID types.

Important! The system cannot map user profiles with different ID types.

If a different ID type is required for a user profile on one of the systems, after performing user profile synchronization, consider using On Request PeopleCode to convert the ID type to the required type.

Implementing Default User Profile Synchronization

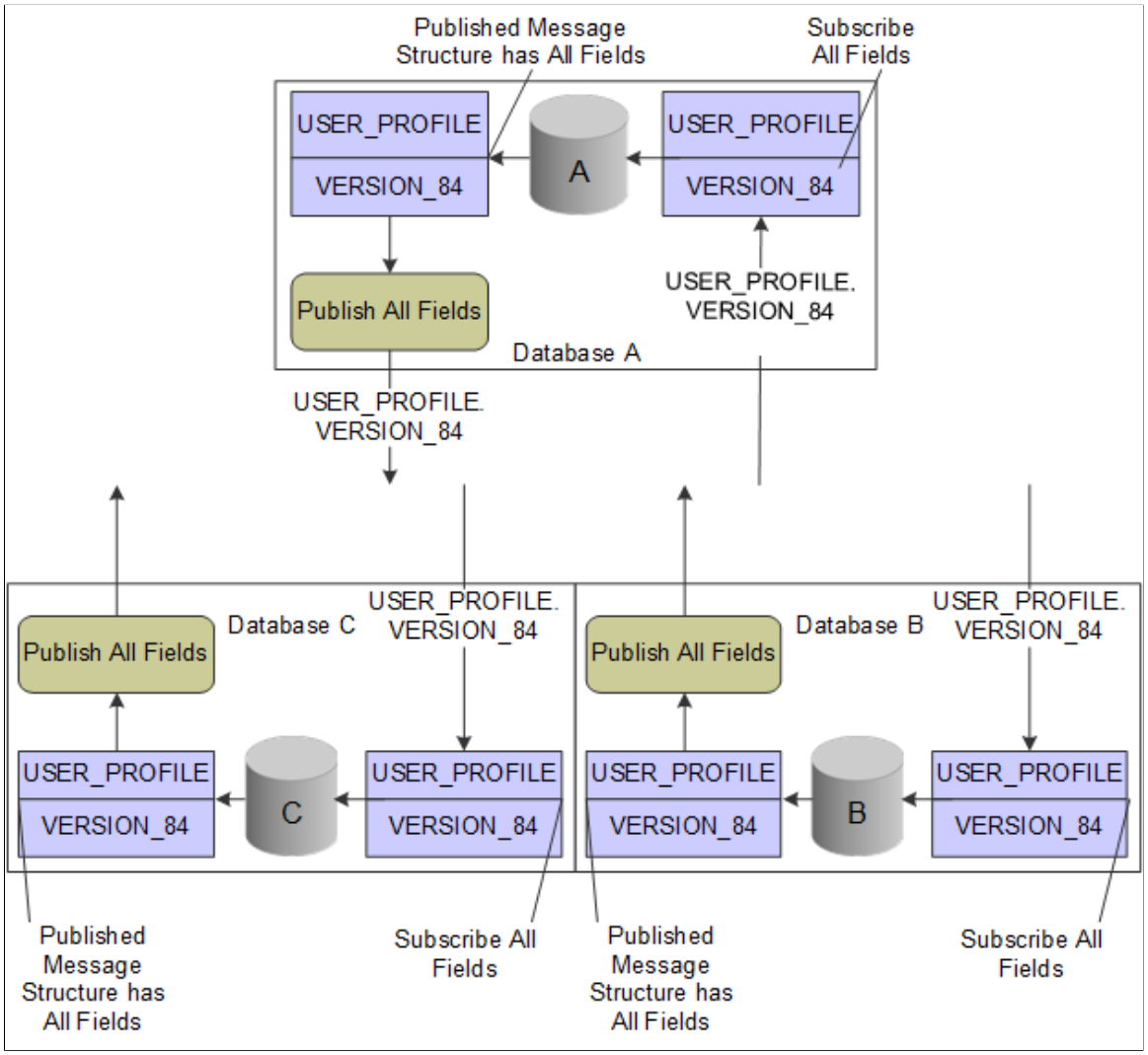
This section provides an overview of default user profile synchronization and discusses how to set up a default user profile synchronization.

Understanding Default User Profile Synchronization

When you implement default user profile synchronization among databases, other than the default user profile synchronization exceptions mentioned below, the subscribing databases have no control over the data that they receive and process.

All participating databases use the USER_PROFILE service operation and the USER_PROFILE.VERSION_84 message during the publish and the subscribe processes.

This diagram shows the service operations and messages, and the way in which user profile data is published by and subscribed to by three PeopleSoft systems that are using default user profile synchronization.



Default User Profile Synchronization Designed Exclusions

Adding and deleting user profiles on the publishing node cause corresponding changes on the subscribing nodes. Modifying user profiles on the publishing node causes corresponding changes on the subscribing nodes with these exceptions:

- Changes to the primary email account are ignored if a primary email exists in the subscribing node.
- Changes to a user ID type are ignored if the user ID type is not valid on the subscribing node. Instead, the subscribing node inserts an ID type of *None* if the subscribing node does not have a row for *None* already.
- In general, changes that produce invalid field values in the subscribing node are ignored by the subscribing node.

Setting Up Default User Profile Synchronization

To set up standard user profile synchronization, perform these tasks:

1. Turn on the Pub/Sub servers.

   See “Using the Quick-Configure Menu” (System and Server Administration).

2. Define the local gateway URL for the integration broker.

   See “Defining Integration Gateways” (Integration Broker Administration).

3. In each participating database, activate the domain in integration broker.

   See “Activating Pub/Sub Server Domains” (Integration Broker Administration).

4. In each participating database, create and configure the remote nodes.

   See “Adding Node Definitions” (Integration Broker Administration).

5. In each participating database, configure single signon by setting up each subscribing database as a trusted node.

   See Defining Nodes for PeopleSoft-Only Single Signon.

6. In each participating database, define the gateway properties; include all PeopleSoft nodes.

   See “Setting Oracle Jolt Connection String Properties” (Integration Broker Administration).

7. In each participating database, activate the USER_PROFILE service operation.

   Note: The default setting is *Enabled*.

   See “Configuring Service Operation Definitions” (Integration Broker).

8. In each participating database, configure and activate routings for the USER_PROFILE service operation.

   - In each subscribing database, select the **Generate Any-to-Local** check box to create the necessary *inbound* routings; or create point-to-point *inbound* routings.

     See “Configuring Routing Definitions” (Integration Broker).

   - In each publishing database, you must create *outbound* routings to each subscribing node. For example, if you are in a CRM database publishing to an HCM and a FIN database, you must create two outbound routings.

     See “Configuring Routing Definitions” (Integration Broker).

9. For each subscribing database, grant permission list security for the USER_PROFILE service operations.

   See Setting Web Services Permissions.

Implementing Configurable User Profile Synchronization

This section provides an overview of configurable user profile synchronization and discusses how to:

- Enable Security PeopleCode options.
- Set up configurable user profile synchronization.

Understanding Configurable User Profile Synchronization

When you implement configurable user profile synchronization among databases, you can select, or configure, the fields containing data for which you want to subscribe.

All participating databases use the USER_PROFILE service operation and the USER_PROFILE.VERSION_84 message to publish user profile information.

All participating databases use the USER_PROFILE_XFR service operation and the USER_PROFILE.VERSION_XFR message to subscribe to the incoming data. You configure the USER_PROFILE_XFR inbound routing with a USER_PROFILE.VERSION_84 external alias. This alias enables the subscribing databases to receive the inbound USER_PROFILE.VERSION_84 message and transform it based on your field configuration.

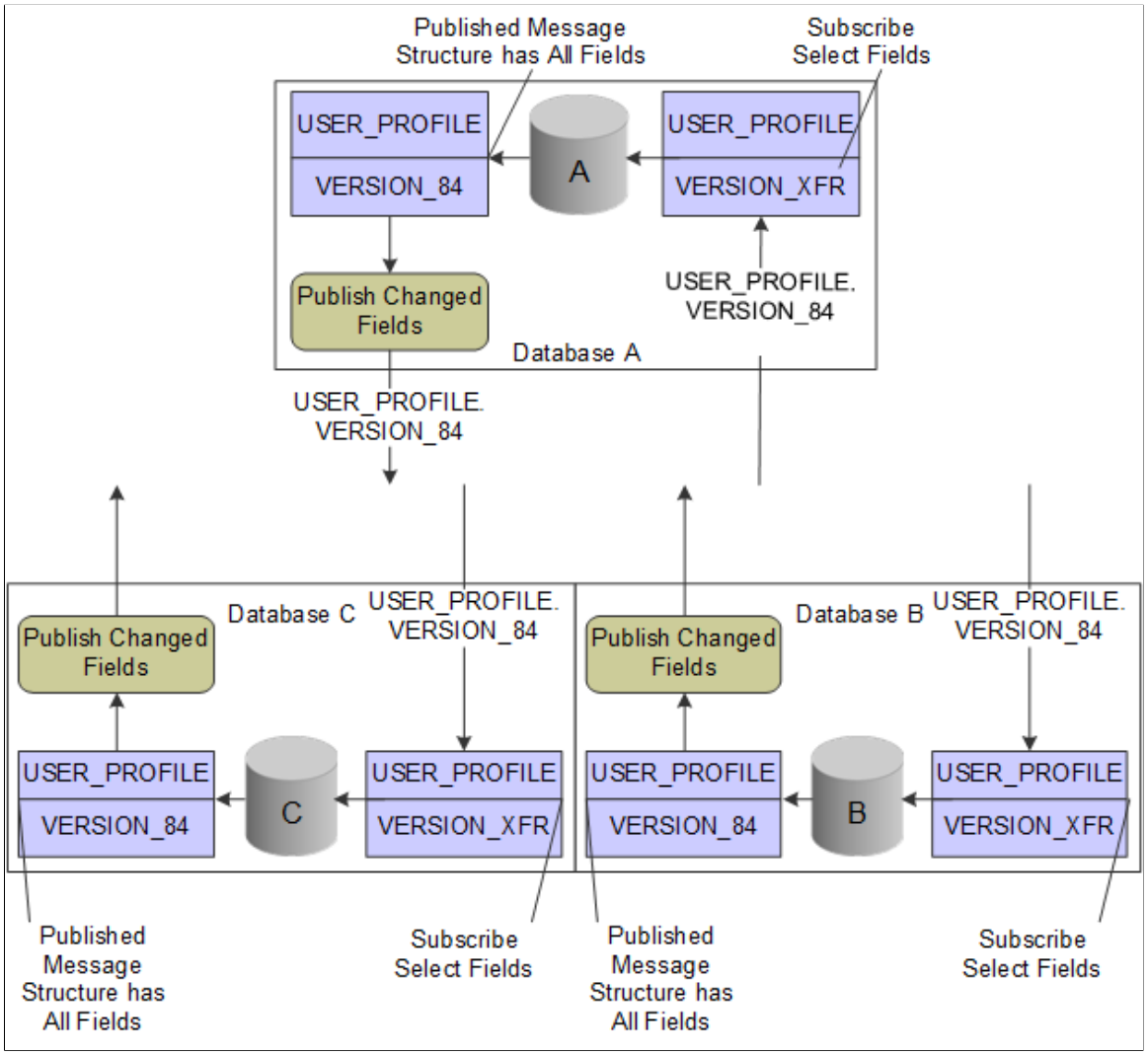
The USER_PROFILE.VERSION_XFR message definition excludes only the following record.fields by default:

- PSOPRDEFN.OPRCLASS
- PSOPRDEFN.ROWSECCLASS
- PSOPRDEFN.SYMBOLICID
- PSOPRDEFN.PRCSPRFLCLS
- PSOPRDEFN.DEFAULTNAVHP

The subscription PeopleCode for the USER_PROFILE_XFR service operation will fail if any expected records are missing or out of order. It will also fail if certain record.fields are not in the USER_PROFILE.VERSION_XFR message. The following is a list of the required record.fields for the USER_PROFILE.VERSION_XFR message to function:

- PSOPRDEFN.OPRID
- PSOPRALIAS.OPRALIASTYPE
- PSROLEUSER_VW.ROLENAME
- RTE_CNTL_USERVW.ROLENAME
- RTE_CNTL_USERVW.RTE_CNTL_PROFILE
- PSUSEREMAIL.EMAILTYPE
- PSUSEREMAIL.EMAILID

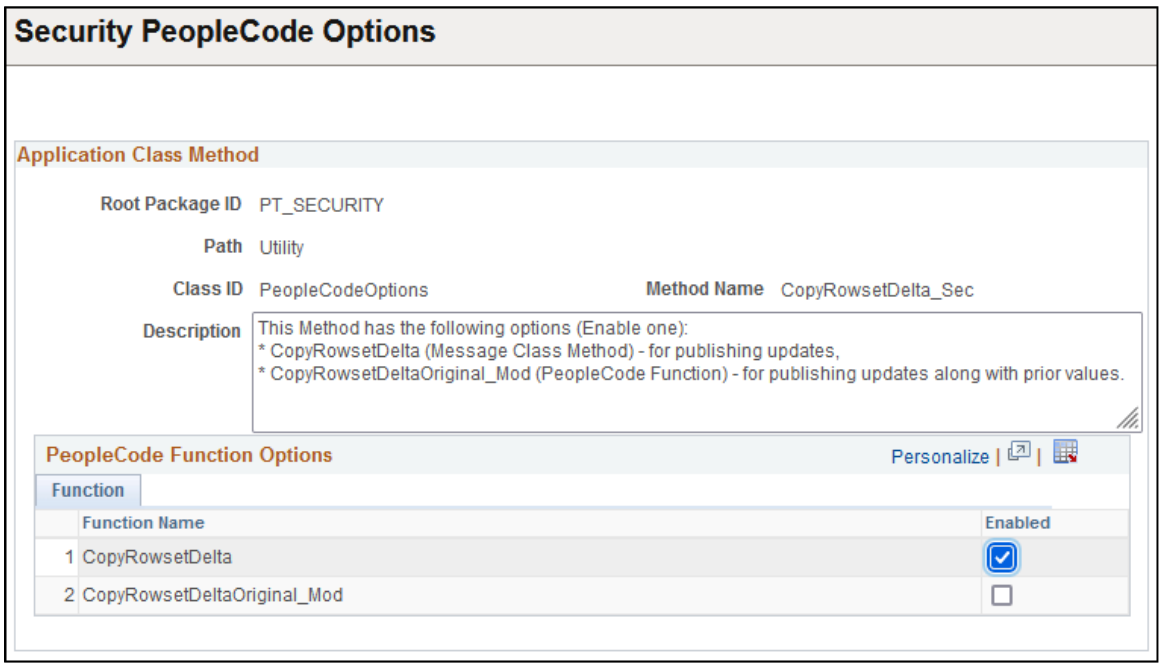
This diagram shows the service operations and messages, and the way in which user profile data is published by and subscribed to by three PeopleSoft systems that use configurable user profile synchronization.



Enabling Security PeopleCode Options

Access the Security PeopleCode Options page (**PeopleTools** > **Security** > **Security Objects** > **Security PeopleCode Options**).

This example illustrates the fields and controls on the Security PeopleCode Options page.



Field or Control	***Description***
Application Class Method	Application classes, at their base level, are PeopleCode programs. In addition, application classes provide more structure. Using the Application Packages, you have a clear definition of each class, as well as its listed properties and methods, which makes it easier for you to create a complex program that uses many functions. This group box displays information about the PT_SECURITY application package.
PeopleCode Function Options	This group box displays the available PeopleCode functions for the CopyRowsetDelta_Sec method, which you use to implement user profile synchronization.
Function Name	Select from these two functions: • CopyRowsetDelta Select this function to implement standard user profile synchronization. • CopyRowsetDeltaOriginal_Mod Select this function to implement configurable user profile synchronization.
Enabled	Select a check box to choose the type of user profile synchronization. You can enable only one option in the list of functions.

Setting Up Configurable User Profile Synchronization

To set up configurable user profile synchronization:

1. Turn on the Pub/Sub servers.

   See “Using the Quick-Configure Menu” (System and Server Administration).

2. In each publishing database, access the Security PeopleCode Options page and enable the **CopyRowsetDeltaOriginal_Mod** PeopleCode function.

   See Enabling Security PeopleCode Options.

3. Define the local gateway URL for Integration Broker.

   See “Defining Integration Gateways” (Integration Broker Administration).

4. In each participating database, activate the domain in Integration Broker.

   See “Activating Pub/Sub Server Domains” (Integration Broker Administration).

5. In each participating database, create and configure the remote nodes.

   See “Configuring Nodes” (Integration Broker Administration).

6. In each participating database, configure single signon by setting up each subscribing database as a trusted node.

   See Defining Nodes for PeopleSoft-Only Single Signon.

7. In each participating database, define the gateway properties; include all PeopleSoft nodes.

   See “Setting Oracle Jolt Connection String Properties” (Integration Broker Administration).

8. In each participating database, deactivate the *inbound* generated USER_PROFILE.VERSION_84 routing definition.

   See “Activating and Inactivating Routing Definitions in the Routing Component” (Integration Broker).

   Note: This step is necessary only if you implemented standard user profile synchronization and are switching to configurable user profile synchronization.

9. In each participating database, configure and activate the routings for the USER_PROFILE service operation.

   In each publishing database, you must create *outbound* routings to each subscribing node. For example, if you are in a CRM database publishing to an HCM and a FIN database, you must create two outbound routings.

   See “Activating and Inactivating Routing Definitions in the Routing Component” (Integration Broker).

10. In each participating database, activate the USER_PROFILE_XFR service operation.

    See “Configuring Service Operation Definitions” (Integration Broker).

11. In each participating database, configure the routings for the USER_PROFILE.VERSION_XFR service operation.

    - In each subscribing database, select the **Generate Any-to-Local** check box to create the necessary *inbound* routings or create point-to-point *inbound* routings.
    - In each subscribing database, change the external alias on the Parameters page to *USER_PROFILE.VERSION_84*.

      See “Configuring Routing Definitions” (Integration Broker).

12. In each subscribing database, grant permission list security for the USER_PROFILE_XFR service operations.

    See Setting Web Services Permissions.

13. In each subscribing database, configure the USER_PROFILE.VERSION_XFR message definition:

    - Expand the User Profile message records.
    - Select the fields that you want the *subscribing* database to update.
    - Clear the fields that you want the *subscribing* database to ignore.

14. Save the message.

Securing User Profile Synchronization

During service operation configuration consider implementing the following to secure user profile synchronization:

Consider using any or all of the following to secure the service operations used in user profile synchronization:

Field or Control	***Description***
Point-to Point Routings	Consider using point-to-point routings for user profile synchronization. In a point-to-point routing you explicitly define the sending and receiving nodes in a transaction on the Routings – Routing Definitions page. See “Understanding Routing Definitions” (Integration Broker)
Inbound Security Verification	You can set a required level of security for inbound service operations, including that they be digitally signed, encrypted, sent using SSL, or a combinations of these. See “Validating Security on Inbound Integrations” (Integration Broker Administration)

Field or Control	*Description*
User Authentication	When integrating with other PeopleSoft systems, user authentication determines the user ID to set on outbound integrations. The receiving system extracts this information and uses the user ID to validate against the permission list to which a service operation is assigned. If the user ID is assigned to the permission list, the sender can invoke the service operation. See “Managing User Authentication” (Integration Broker Administration)

Transferring Users Between Databases

Sometimes you might want to transfer all user information from a source database to a target database, for example, during the upgrade process or when moving users from the production environment to a development or a testing environment. PeopleSoft applications provide Data Mover (DMS) scripts that export and import user profile security information. These scripts transfer user profile data from a source to a target database. The scripts use these tables:

- PSOPRDEFN
- PSOPRALIAS
- PSROLEUSER
- PSUSERATTR
- PSUSEREMAIL
- PSUSERPRSNLOPTN
- ROLEXLATOPR
- PS_RTE_CNTL_RUSER

Note: Use the Application Designer upgrade feature to upgrade both roles and permission lists.

One script exports User Profile data from the source database. The source database refers to the database that contains the User Profiles that you want to migrate. The target database refers to the database to which you are copying the user information.

After exporting the security information from the source database, you then run the import script against the target database. The target database refers to the database to which you want to transfer the security data. The scripts involved in transferring security information from one database to another are:

- USEREXPORT.DMS.

  This script exports User Profiles from the source database and stores them in a Data Mover DAT file. The output file is named USEREXPORT.DAT.

- USERIMPORT.DMS.

  This script reads the file created by USEREXPORT.DMS and copies the User Profile data into the target database.

You will find this set of scripts in the <*PS_HOME*>/scripts directory.

Note: Using Data Mover to transfer user profiles from one database to another does *not* trigger user profile synchronization.

Considerations

Before running scripts to export and import your security information, you should consider these topics:

- Duplicate Rows

  If the target database already contains a row of data with identical keys to a row transferred by the import script, the duplicate row *will not* be transferred to the target. The scripts make no attempt to merge the duplicate row; the row is not transferred.

  To ensure that you do not have data rows with duplicate keys, you must ensure that the source database does not contain a User Profile with the same name as in the target database.

  You should not have data rows with duplicate keys in your source and target database when you begin the copy, as this can lead to unexpected results that compromise database integrity.

- Release Levels

  Because the PeopleTools table structures change between major releases (8.59 to 8.60,for example), you cannot transfer users between databases that run different versions of PeopleTools. Before starting the migration process, upgrade your source and target databases so the release levels match.

Running the Scripts

Complete the following procedure to run the user transfer scripts.

1. Using Data Mover, sign in to the source database and run USEREXPORT.DMS for user definitions.

   You can edit this script to specify the location and file name of the output file and the log file.

2. Using Data Mover, sign in to the target database and run USERIMPORT.DMS for user definitions.

   You can edit the script to specify the location and file name of the input file and the log file. The name and location of the input file must match the output file you specified in Step 1.

3. After copying user and role definitions, run the PeopleTools audits, including DDDAUDIT and SYSAUDIT, to check the consistency of your database.

Chapter 8

Employing LDAP Directory Services

Understanding the PeopleSoft LDAP Solution

Three PeopleSoft-delivered technologies enable you to:

- Authenticate against an LDAP V3 compliant directory server.
- Reuse your existing user profiles stored within LDAP.

The three technologies are:

- Directory Business Interlink, which exposes the LDAP to PeopleCode.

  The system uses it for all communication with the LDAP server process running on a directory server.

- User Profile Component Interface, which exposes the User Profiles component to PeopleCode.

  The system uses it to programmatically manage a local cache of user profiles.

- Signon PeopleCode, which runs when a user signs in to the system—similar to the login scripting of most network systems.

  Signon PeopleCode uses the Directory Business Interlink and the User Profile Component Interface to verify directory-based credentials and programmatically create a local User Profiles cache.

The combination of these three technologies provides a flexible way to configure PeopleSoft for integration with your directory server. No set schema is required in the directory. Instead, you can configure and extend the Signon PeopleCode to work with any schema implemented in your directory server.

The topics in this documentation describe setting up the LDAP integration technology on your site. The tasks assume that an LDAP V3 compliant directory service is already installed, and that you intend to import LDAP group values and apply them to PeopleSoft roles.

Note: PeopleTools uses JNDI libraries only. JNDI requires no added installation as it is part of the standard PeopleTools installation. This documentation assumes you have a working knowledge of LDAP-enabled directory servers.

Note: When you enable LDAP authentication, the password column on the PSOPRDEFN record is no longer used. Directory-level users are not authenticated against the PSOPRDEFN table; they are authenticated by Signon PeopleCode. Because Signon PeopleCode only runs on the application server, LDAP authentication requires an application server. That is, LDAP authentication does not work for a two-tier signon.

Configuring LDAP Connection Parameters

PeopleSoft features a set of parameters that you can set that impact the connection between the application server and the LDAP directory server.

You can access and set the parameters in the LDAP section of the psappsrv.cfg file.

Should the Referrals be Followed in the Directory Server?

Use this parameter to indicate how to handle referrals in the configured directory servers.

Note that this property affects both referral error responses and continuation references.

Set the **Follow Referrals** parameter equal to one of the following values:

- *Y.* (Default.) Follow referrals.
- *N.* Do not follow referrals.

LDAP Connection Time Out in Seconds

The value of this parameter determines the time limit in seconds for the PeopleTools LDAP client to make a connection with the directory server.

Set the **Connection TimeOut** parameter equal to a value in seconds. The default is *30* (seconds).

Enable/Disable Connection Pooling

Use this parameter to enable or disable connection pooling.

Set the **Connection Pooling** parameter equal to one of the following values:

- *Y.* Enable connection pooling.
- *N.* (Default.) Disable connection pooling.

LDAP Connection Pooling Time Out

When connection pooling is enabled, the value of this parameter determines the time limit in seconds to hold the connections in the connection pool without being closed and removed from the pool.

To use this parameter set the **Connection Pooling TimeOut** parameter equal to a value from *0* to *999999* (seconds).

The default value is *300* seconds.

Should LDAP Pick Up Any Changes from SSL Certificates?

This parameter is reserved for future use.

LDAP Logging Detail Level

Use the **LDAP LogFence** parameter to set the log level. The valid values are:

- *0.* Suppress logging.
- *1.* (Default.) Errors and exceptions.
- *2.* Transaction status.
- *3.* Detailed tracing.

Configuring the LDAP Directory

This section provides an overview of LDAP directory configuration and discusses how to:

- Specify network information for LDAP.
- Specify additional connect DNs.
- Install selected PeopleSoft-specific schema extensions.
- Test connectivity.

Understanding LDAP Directory Configuration

The Configure Directory component (PSDSSETUP) contains four pages that you use for specifying connection information and testing directory server connections.

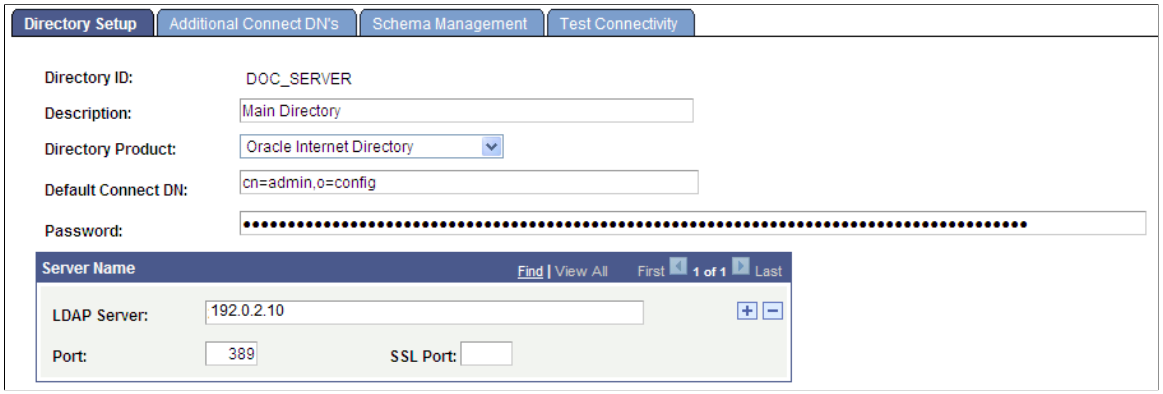
To enable your PeopleSoft system to successfully connect to your directory server, you must enter the appropriate connection information. This information includes the server name (DNS or IP address) and the listening port number. You also must enter the user distinguished name (User DN) and associated password.

The PeopleSoft application server uses the User DN and password to connect to the LDAP server to retrieve user profile information about the specific user signing in to the system. The User DN must reflect a user with the appropriate LDAP browse rights.

Specifying Network Information for LDAP

Access the Configure Directory - Directory Setup page (select **PeopleTools** > **Security** > **Directory** > **Configure LDAP Directories** and click the **Directory Setup** tab).

This example illustrates the fields and controls on the Configure Directory - Directory Setup page.



Field or Control	***Description***
Directory ID	Displays the directory connection that you are creating. The directory ID that you enter can identify a specific LDAP server or a collection of LDAP servers, depending on how many servers you add in the Server Name section.
Description	Enter a description of the directory connection.
Directory Product	Select your directory product from the list of options.
Default Connect DN (default connect distinguished name) DNs connecting to LDAP servers	Displays the default connect DN associated with the directory ID that you entered or selected on the initial search page. The connect DN is the ID that you can use to connect to the directory server. You can enter an alternative connect DN.
Password	Enter the password associated with the directory-based account that appears in the **Default Connect DN** field. The maximum password length is 64 characters. **Note:** The password is stored in encrypted form in the database; not even individuals with administration access to the database can view the password.
Server Name	Add LDAP directory servers to a connection list. You can add multiple servers for failover purposes using the plus button. All servers you add must participate in the same directory service.
LDAP Server	Identify a specific LDAP server. You can use the DNS name or you can use IP address dotted notation. For example, either of the following formats is acceptable: ldap12.yourcompany.com or 192.201.185.90.

Field or Control	*Description*
Port	Enter the port number on which the LDAP server is configured to receive search requests. The standard LDAP port is 389. If you do not specify the correct port, PeopleSoft Directory Interface cannot exchange data with your LDAP server.
SSL Port	If you are implementing SSL, enter the SSL port on the LDAP server.

Specifying Additional Connect DNs

Access the Additional Connect DN's page (select **PeopleTools** > **Security** > **Directory** > **Configure LDAP Directories** and click the **Additional Connect DN's** tab).

This example illustrates the fields and controls on the Configure Directory - Additional Connect DN's page.

The PeopleSoft application server uses the user DN and password specified on this page to connect to the LDAP server to retrieve user profile information about the specific user signing in to the system. The user DN must reflect a user with the appropriate LDAP browse rights.

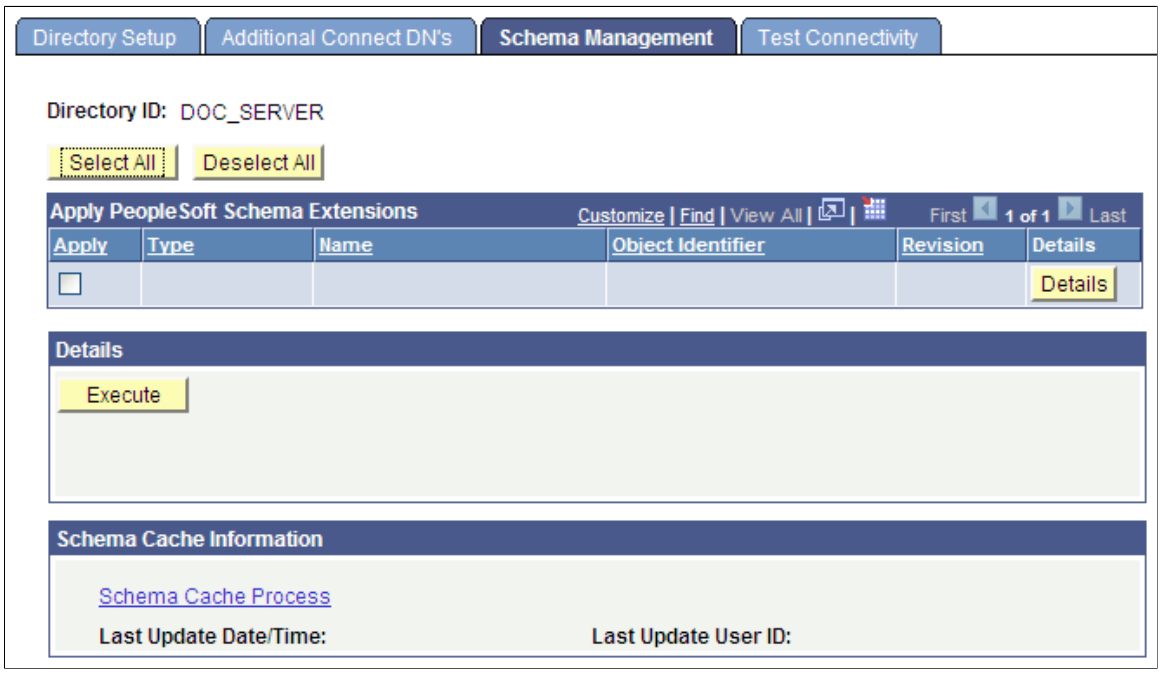
Note: You will not see any available schema extensions unless you have installed the PeopleSoft Directory Interface.

Field or Control	*Description*
User DN	Add any DNs that you need in addition to the default connect DN that you entered on the Directory Setup page. The default user ID is most likely an administrative ID. This value enables you to set up a more secure user ID for the scope of the mapping.
Password	For each additional DN that you enter, add the corresponding password.

Installing Selected PeopleSoft-Specific Schema Extensions

Access the Schema Management page (select **PeopleTools** > **Security** > **Directory** > **Configure LDAP Directories** and click the **Schema Management** tab).

This example illustrates the fields and controls on the Configure Directory - Schema Management page.



Note: Unless you have installed the PeopleSoft Directory Interface product, you might not have any PeopleSoft schema extensions available to you.

Note: The Schema Management page enables you to add PeopleSoft-delivered object classes and attribute types to your directory. If you add attributes and object classes using the Schema Management page, you must also delete them using this page.

Field or Control	***Description***
Apply	Select this check box to apply the selected schema extension type to your directory.
Type	Displays the type of schema extension, either *Object Class* or *Attribute Type*.
Name	Displays the schema extension name.
Object Identifier	Displays the schema extension object identifier. The sequence 1.3.6.1.4.1.2810.20 identifies the object as a PeopleSoft object. The second to last number is either a 1 or a 2. A *1* indicates an object class type and a *2* indicates an attribute type. The last number indicates the sequence in which the extension was created.

Field or Control	*Description*
Revision	Displays the number of times the schema extension was revised.
Details	Click to display details about the selected schema extension in the **Details** region at the bottom of the page.
Select All	Click to select all the schema extensions to apply to your directory.
Deselect All	Click to deselect every schema extension.
Apply	Click to apply the selected schema extensions to your directory.

Details

When you click a schema extension **Details** button, the system displays the details of that extension. In addition to the object identifier and name, you may also be interested in the Superiors detail, which indicates which extensions, if any, are above this one in the hierarchy. Also of interest is the Type detail, which indicates whether the schema extension is a mandatory, optional, or auxiliary extension.

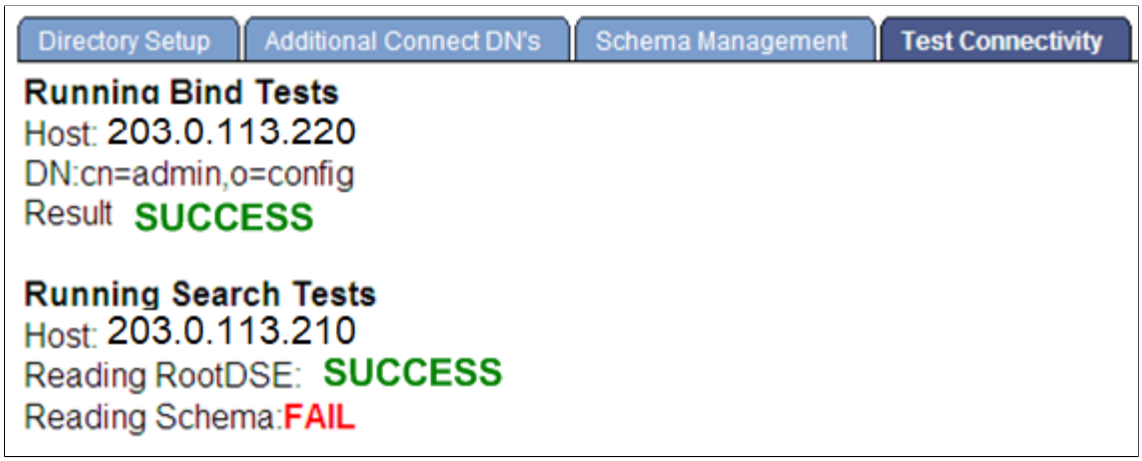
Schema Cache Information

For convenience, you can use the **Schema Cache Process** link to transfer you to the Schema Cache page so that you can invoke the Schema Cache process. **Last Update Date/Time** and **Last Update User ID** enable you to monitor the frequency of updates as well as the last administrator to run the process.

Testing Connectivity

Access the Test Connectivity page (select **PeopleTools** > **Security** > **Directory** > **Configure LDAP Directories** and click the Test Connectivity tab).

This example illustrates the fields and controls on the Configure Directory - Test Connectivity page.



The page displays the results (SUCCESS or FAIL) of the connectivity test. If connectivity fails, modify the connect information on the Directory Setup and Additional Connect DN's pages.

Caching the Directory Schema

You use the Cache Schema page to specify a directory server and invoke an Application Engine program designed to create a cache in the PeopleSoft database of the directory schema. This cache enables you to select names of object classes and attribute types when you create security maps.

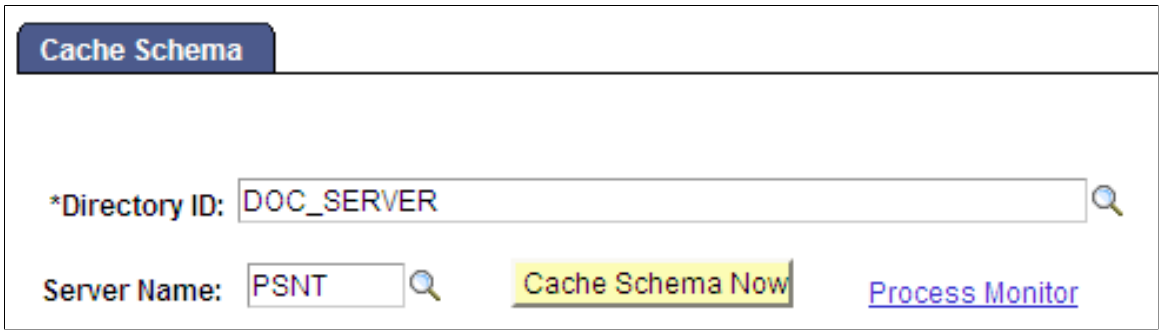
This section discusses how to create a cache of the directory schema.

Note: If you want to create a cache of the directory schema securely (using https), you should set the **Use Secure Socket Layer** option to *Yes* on the Authentication page (navigate to **PeopleTools** > **Security** > **Directory** > **LDAP Authentication Maps**). That is, select the **Use Secure Socket Layer** check box on the page.

Creating a Cache of the Directory Schema

Access the Cache Schema page (select **PeopleTools** > **Security** > **Directory** > **Cache LDAP Directory Schema**).

This example illustrates the fields and controls on the Cache Schema page.



Field or Control	*Description*
Directory ID	Select the directory ID to identify the directory that the system should connect to and retrieve schema information from.
Server Name	Search for the Process Scheduler server on which the Cache Schema process should run.
Cache Schema Now	Click this button to cache the LDAP schema data to tables within the PeopleSoft database. Typically, you use this option during initial setup and any time that the schema has changed.
Process Monitor	After invoking the process, you can monitor the progress by clicking this link.

Creating Authentication Maps

Use the Authentication page only if you are implementing directory authentication as opposed to storing authentication information in the PeopleSoft database. You create authentication maps to define mappings to one or more directories that the PeopleSoft system relies on for authenticating users. You can activate multiple authentication maps. Your PeopleSoft LDAP system authenticates users against all active authentication maps.

Authentication maps are used to specify the following information for LDAP authentication:

- The identity of all the LDAP servers to be searched and their credentials.
- The locations where the search has to be performed inside the LDAP.
- The attribute of the entries that must be matched with the signon user ID.

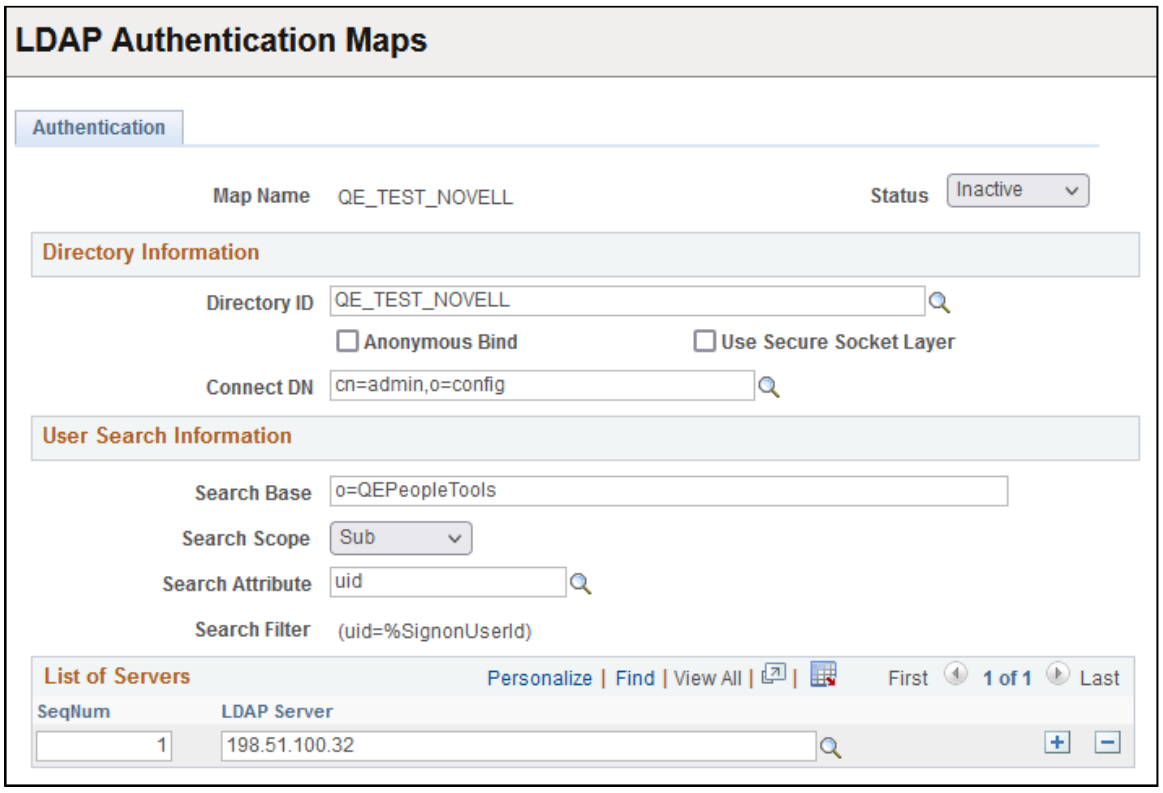
This section discusses how to:

- Define an authentication map.
- Use the Search Attribute field in authentication maps.

Defining an Authentication Map

Access the Authentication page (select **PeopleTools** > **Security** > **Directory** > **LDAP Authentication Maps**).

This example illustrates the fields and controls on the Authentication page.



Field or Control	*Description*
Status	Activate the authentication map by selecting *Active.* To disable an authentication map, select *Inactive.*

Directory Information

Field or Control	*Description*
Directory ID	Select the directory ID of the directory that you intend to use for authentication.
Anonymous Bind	If all directory data required for authentication and user profile maintenance is visible to an anonymous connection, select this check box.
Use Secure Socket Layer	Select this option if you are implementing an SSL connection between PeopleSoft and the directory. If you did not specify a port number for the directory, the system uses the default LDAPS port.
Connect DN	This value is the default connect DN that you specified on the Directory Setup page. To select one of the DNs specified on the Additional Connect DN's page, click the search button. **Note:** If **Anonymous Bind** is selected, the Connect DN is ignored.

User Search Information

Field or Control	*Description*
Search Base	Enter the root of the directory information tree under which the system should search for user information.

Field or Control	*Description*
Search Scope	Select the search scope for this search. Values are: *Base:* Not applicable. You should not use *Base* on the authentication map. *One:* The query searches only the entries one level down from the entry in the **Search Base** field. *Sub:* The query searches the entire sub tree beneath the search base entry.
Search Attribute	When a user signs in using LDAP Authentication, the system searches the directory to find the user's user entry. The search attribute is used to construct the LDAP search filter used in finding the person's user entry. The value in the **Search Attribute** field is entered by the user when the user signs in. Enter the attribute to be returned by the search, such as user ID (uid) or customer ID (cid). See Using the Search Attribute Field in Authentication Maps. **Important!** If you specify a different value here than the **User ID Attribute** value that you plan to specify on the Mandatory User Properties page, users will not be able to switch to another application from the Go menu in PeopleSoft Windows clients such as Application Designer. The second application expects to automatically authenticate a user with the value of %SignonUserId, the system variable that contains the value entered by the user in this field. However, the value of the **User ID Attribute** field is used to populate the OPRID field in PSOPRDEFN. Because the value of OPRID is different from the value of %SignonUserId, the authentication fails with an error message. Users can still access any PeopleSoft Windows client by launching it directly and signing in using the value of this field as the user ID.
Search Filter	Displays the LDAP search filter that the system uses to search the directory for equal entries.

List of Servers

Field or Control	*Description*
SeqNum (sequence number)	Set the order in which the system should access the list of servers for authentication.
LDAP Server	Select the name of the LDAP server. Use the plus button to enter additional servers.

Using the Search Attribute Field in Authentication Maps

The purpose of the Search Attribute prompt on the authentication maps page is to map a value that is used for the User ID on the login page. For example, if you want users to log in with their mailID, then mail attribute should be given in the prompt.

Example

Consider an entry corresponding to the user *adamclark* in the LDAP directory.

```
dn: uid=adamclark, dc=peoplesoft, dc=com
cn: adamclark
uid: adamclark123
description: peoplesoft user
mail: adamclark@example.com
telephone: 12345678
objectclass: person
password: PASSWORD
```

If the user is to log in with *adamclark/PASSWORD,* then the Search Attribute prompt value should be *cn*. If the user wants to log in with *adamclark@example.com/PASSWORD,* then the Search Attribute prompt value should be *mail*.

Creating User Profile Maps

This section provides an overview of user profile options and discusses how to:

- Specify mandatory user properties.
- Specify optional user properties.
- Associate user IDs and user profile maps.

Understanding User Profile Options

If you are going to authenticate users with the directory server, a PeopleSoft user profile is still required. That is, a row is still required in the table in which PeopleSoft user information is stored (PSOPRDEFN). In this context, you cache LDAP user information inside your PeopleSoft system. The properties that you specify on the Mandatory and Optional User Properties pages are the columns in PSOPRDEFN that the system populates with values from your directory server. These values comprise your user profile options.

PeopleSoft applications use this cache of user information, not your directory server. Whenever a transaction requires user information, the application refers to the local PSOPRDEFN table as opposed to querying the directory server. This improves performance.

After a user signs in to the system and the Signon PeopleCode is carried out, PeopleSoft creates a row for that user in the user definition table by retrieving the LDAP information and creating a local cache. Signon PeopleCode maintains this row automatically; manual updates are not necessary. Any changes made in the directory server are reproduced in the local cache.

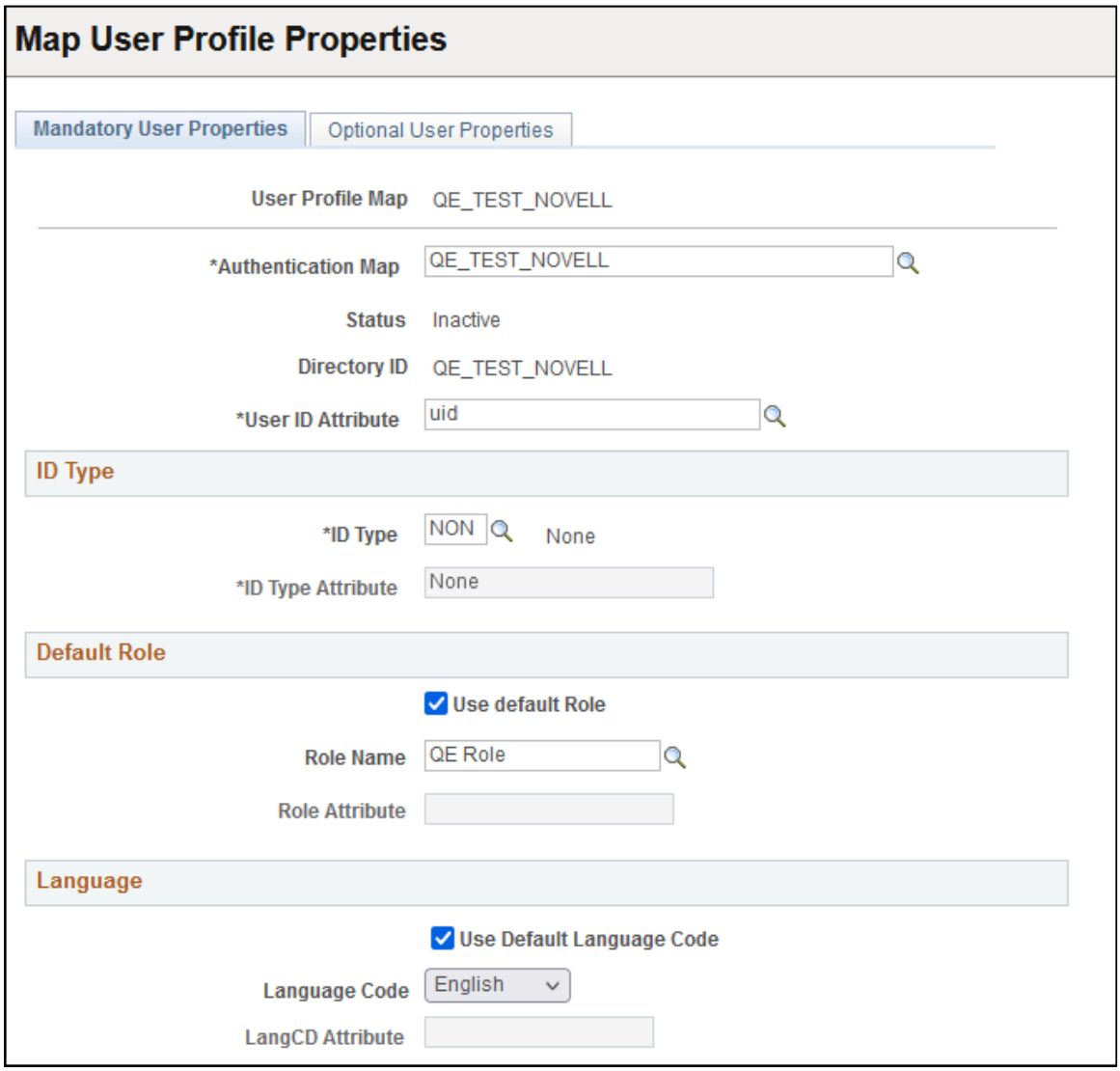
Some properties are required when creating a PeopleSoft User Profile; these properties appear on the Mandatory User Properties page. Other properties are optional; these properties appear on the Optional User Properties page.

Note: You must supply user properties to Signon PeopleCode only if you intend to authenticate users with your LDAP directory.

Specifying Mandatory User Properties

Access the Mandatory User Properties page (select **PeopleTools** > **Security** > **Directory** > **Map User Profile Properties** and click the **Mandatory User Properties** tab).

This example illustrates the fields and controls on the User Profile Map - Mandatory User Properties page.



Field or Control	*Description*
Authentication Map	Select the authentication map to associate with this user profile mapping. The server and connection information are taken from the authentication map.
Status	Displays the status of the selected user profile map. **Note:** Only one user profile map should be active at any time.
Directory ID	Displays the directory ID associated with the authentication mapping.

Field or Control	*Description*
User ID Attribute	Specify the LDAP attribute used to populate the OPRID (user ID) field on PSOPRDEFN. **Important!** If you specify a different value here than the **Search Attribute** value that you specified on the Authentication page, then users will not be able to switch to another application from the Go menu in PeopleSoft Windows clients such as Application Designer. The second application expects to automatically authenticate a user with the value of %SignonUserId, the system variable that contains the user ID that was used to sign in. However, because the value of OPRID is different from the value of %SignonUserId, the authentication fails with an error message. Users can still access any PeopleSoft Windows client by launching it directly and signing in using the same **Search Attribute** value for the user ID.

ID Type

Field or Control	*Description*
ID Type	Enter the default ID type for new users, such as Employee ID, Customer ID, and so on. This field is similar to Symbolic ID.
ID Type Attribute	Specifies the LDAP attribute in the directory that holds the selected ID value. For instance, the ID value might be Employee ID. Some ID types require additional data when creating a profile of that type. LDAP User Profile Management can retrieve that data from the LDAP directory if it is available.

Default Role

Field or Control	*Description*
Use Default Role	Select this option if you want to use the default role. If you enable this option, the Default Role field becomes available for entry while the Role Attribute field becomes unavailable for entry. You either specify a default role or specify an LDAP attribute on the user entry that holds the valid name of a PeopleSoft role.

Field or Control	*Description*
Role Name	Enter the name of a default role to be assigned to new users. This value applies to users the first time that they sign in and have not had any roles dynamically assigned to them. Typically, this role has only basic access authorizations, such as for only the self-service pages. Users should get most of their permissions through dynamically assigned roles.
Role Attribute	Instead of specifying only a single default role for each and every user, you can enter a value for the LDAP attribute that holds the name of a PeopleSoft role to be assigned to the user.

You can enable your application to automatically apply a role for the user. When signing in to the application, the user provides a value for the search attribute you specified in the authentication map. The system uses that attribute value to search for the user's entry in the LDAP directory, and then imports the groups containing the entry to the PSOPRDEFN table as the user's role.

To enable this automatic role import feature:

1. Define LDAP groups with names that exactly match the roles defined for your application and assign the user to groups.
2. Deselect the **Use Default Role** check box on this page.
3. Leave the **Role Name** and **Role Attribute** fields on this page blank.

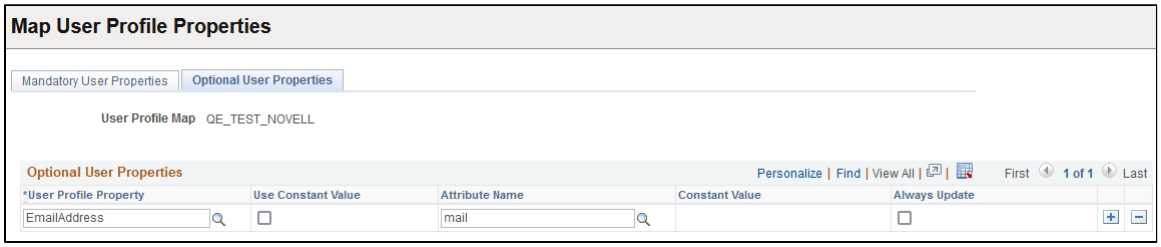
Language

Field or Control	*Description*
Use Default Language Code	Select if you do not maintain language codes in the directory.
Language Code	If the default language code is not stored in the directory, select a default value from the drop-down list box.
LangCD Attribute (language code default)	The name of the LDAP attribute containing a valid language code. The value retrieved from the attribute must be a valid PeopleSoft language code.

Specifying Optional User Properties

Access the Optional User Properties page (select **PeopleTools** > **Security** > **Directory** > **Map User Profile Properties** and click the **Optional User Properties** tab).

This example illustrates the fields and controls on the User Profile Map - Optional User Properties page.



Field or Control	*Description*
User Profile Property	Select the user profile property that you want to add to the local cache. These properties are described in the following table.
Use Constant Value	To supply a constant value for each user, select this option.
Attribute Name	Add the name of the attribute as it is represented in your LDAP schema.
Constant Value	Appears only if you selected **Use Constant Value.**
Always Update	Select this option if you always want the system to update the local user cache to reflect the data stored in the directory server every time the user signs in. If **Always Update** is not selected, the data will be taken from the directory only when the profile is first created.

Click the User Profile Property search button to select one of the following optional user profile properties:

Field or Control	*Description*
CurrencyCode	If the user deals with international prices, set the currency code to reflect the native or base currency so that values appear in the currency with which the user is familiar.
EmailAddress	Select if a user is part of your workflow system or you have other systems that generate emails for users.
MultiLanguageEnabled	Select if the user is set up to use PeopleSoft with multiple languages.
NavigatorHomePermissionList	Displays the homepage permission list that is associated with PeopleSoft Workflow (Navigator Homepage).

Field or Control	*Description*
PrimaryPermissionList	PeopleSoft determines which data permissions to grant a user by examining the primary permission list and row security permission list. Which one is used varies by application and data entity (employee, customer, vendor, business unit, and so on). Consult your PeopleSoft application documentation for more details. PeopleSoft also determines mass change and definition security permissions from the primary permission list.
ProcessProfilePermissionList	The process profile contains the permissions that a user requires for running batch processes through PeopleSoft Process Scheduler. For example, the process profile authorizes users to view output, update run locations, restart processes, and so on. Only the process profile comes from this permission list, not the list of process groups.
RowSecurityPermissionList	See explanation for the **Primary Permission List** field.
SymbolicID	If the symbolic ID is required for the user, select this option.
UserDescription	Typically, displays the name of the user, such as an employee name or a vendor name.
UserIDAlias	In some cases, the user ID is an alias in the form of an email address. If so, select this option.

Associating User IDs and User Profile Maps

When a user is authenticated, a user profile must be created in the PeopleSoft database without a password. Every user profile map will be associated with an authentication map. When a user is logged in through a authentication map, the profile is updated with the values in the corresponding user profile map. All the information that populates the user profile comes from the user profile map. You can specify the role, languageCD, description, and so on in the user profile map.

The user ID of the profile that the system creates corresponds to the **User Profile Map - User ID Attribute** field, which contains an LDAP attribute name.

Consider an entry corresponding to the user *adamclark* in LDAP:

```
dn: uid=adamclark, dc=peoplesoft, dc=com
cn: adamclark
uid: adamclark123
description: peoplesoft user
mail: adamclark@example.com
telephone: 12345678
objectclass: person
password: PASSWORD
```

Example 1

Authentication Map Search Attribute: *cn*

User Profile Map User ID Attribute:*mail*

You must log in as *adamclark/PASSWORD*, while the system creates the user profile with the name *adamclark@example.com*.

Example 2

Authentication Map Search Attribute: *uid*

User Profile Map User ID Attribute:*telephone*

You must log in as *adamclark123/PASSWORD* while the system creates the user profile with the name *12345678*.

Note: The Search Attribute value in the authentication map and the User ID Attribute value in the user profile map need not be the same.

Creating Role Membership Rules

Use the Role Policy page to define the rules that are read by Dynamic Role Rule PeopleCode and populate PeopleSoft roles with members. The rules return the DNs of "people" directory entries, which supply the system with the user IDs specified on the user profile mapping.

This section provides an overview of role membership rules and discusses how to define role membership rules.

Understanding Role Membership Rules

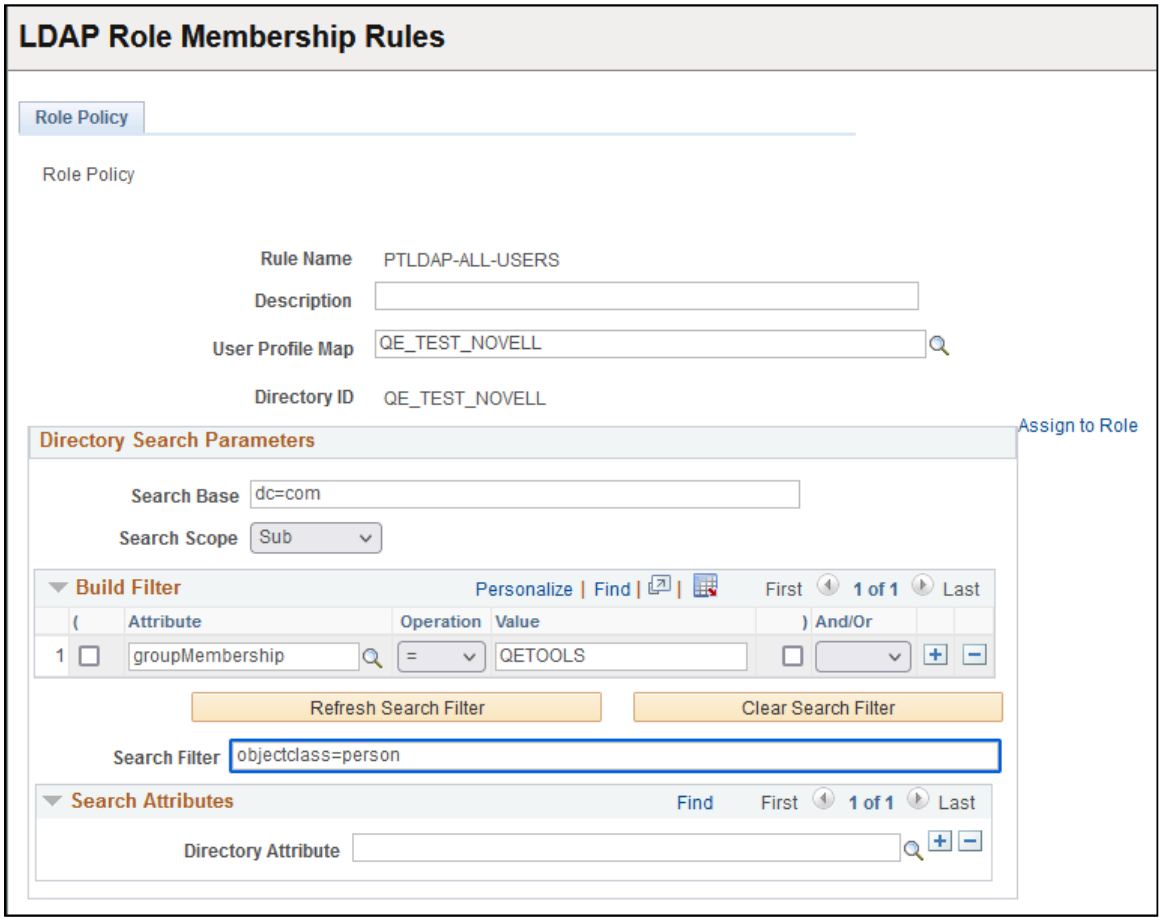
PeopleSoft security roles are comparable to LDAP directory groups. Roles enable you to group user IDs in logical sets that share the same security privileges. PeopleSoft enables you to keep your external directory groups synchronized with the data stored within the PeopleSoft database.

Important! You must keep the data within PeopleSoft consistent with any changes made to the structure or content of the external directory server, especially when you are dealing with security data. The Role Membership Rules page enables you to modify a PeopleSoft role based on directory criteria.

Defining Role Membership Rules

Access the Role Policy page (**PeopleTools** > **Security** > **Directory** > **LDAP Role Membership Rules**).

This example illustrates the fields and controls on the Role Policy page.



Field or Control	*Description*
Rule Name	Displays the directory search name that you entered on the search page.
Description	Enter a short description of the rule.
User Profile Map	Select the user profile map to associate with the rule.
Directory ID	Displays the directory associated with the user profile map that you select.
Assign to Role	Click this link to automatically start the Dynamic Members page in the Roles component of the Security menu. On that page, select Directory Rule Enabled and specify the server on which to carry out the rule.

Directory Search Parameters

Field or Control	*Description*
Search Base	Enter the entry (or container) at which to begin the search.
Search Scope	Select the search scope for this search from the following options: *Base:* The query searches only the value in the **Search Base** field. *One:* The query searches only the entries one level down from the value in the **Search Base** field. *Sub:* The query searches the value in the **Search Base** field and all entries beneath it.

Build Filter

Field or Control	*Description*
( )	Parentheses; on either side of the filter expression select the check boxes below the parentheses to group expressions.
Attribute	Select the attribute that the system will filter.
Operation	Assign an operator to your rule, such as <, <=, <>, =, >, or >=.
Value	Enter the value to assign to the attribute that you specified.
And/Or	To add another line to your rule, select *AND* or *OR,* depending on your rule logic. Select *END* to signify the end of the search. Select *NONE* if you are not using this kind of filter.
Refresh Search Filter	After you make changes using the **Build Filter** options, click this button to update the **Search Filter** edit box to reflect the changes.
Clear Search Filter	Click this button to delete all values from the **Search Filter** edit box and the Build Filter selections.

Field or Control	*Description*
Search Filter	The purpose of this field depends on whether you also specify values in the **Directory Attribute** field, as follows: • No directory attributes specified. Enter a name=value pair that identifies a key field and value on the user record. The system applies this criterion to search for an individual user, regardless of group membership. • One or more directory attributes specified. Enter a name=value pair that the system applies to the search for the DN of the defined container or group. This value typically displays the directory object class of the container in the form “objectclass = GroupOfUniqueNames”, for example. This indicates what type of container to search. To retrieve the correct container DNs, the system adds the name of the container to the search filter at runtime.

Search Attributes

Field or Control	*Description*
Directory Attribute	Select attributes that identify the user to add to this membership. The system searches only for members within the group that is specified by the **Search Filter** field.

Note: You can also write PeopleCode to determine group membership using any arbitrary LDAP search criteria.

Deleting Directory Configurations

You can delete the entire directory configuration or just parts of it.

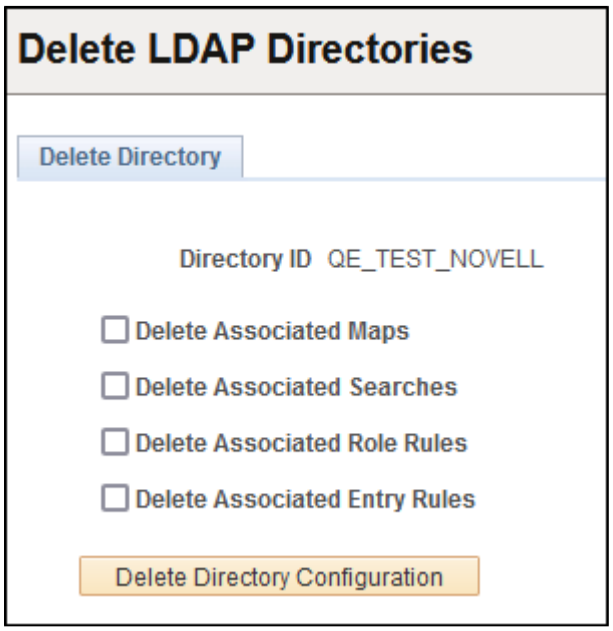
This section discusses how to:

- Delete the directory configuration.
- Work with the workflow address book.

Deleting the Directory Configuration

Access the Delete Directory page (**PeopleTools** > **Security** > **Directory** > **Delete LDAP Directories**).

This example illustrates the fields and controls on the Delete Directory page.

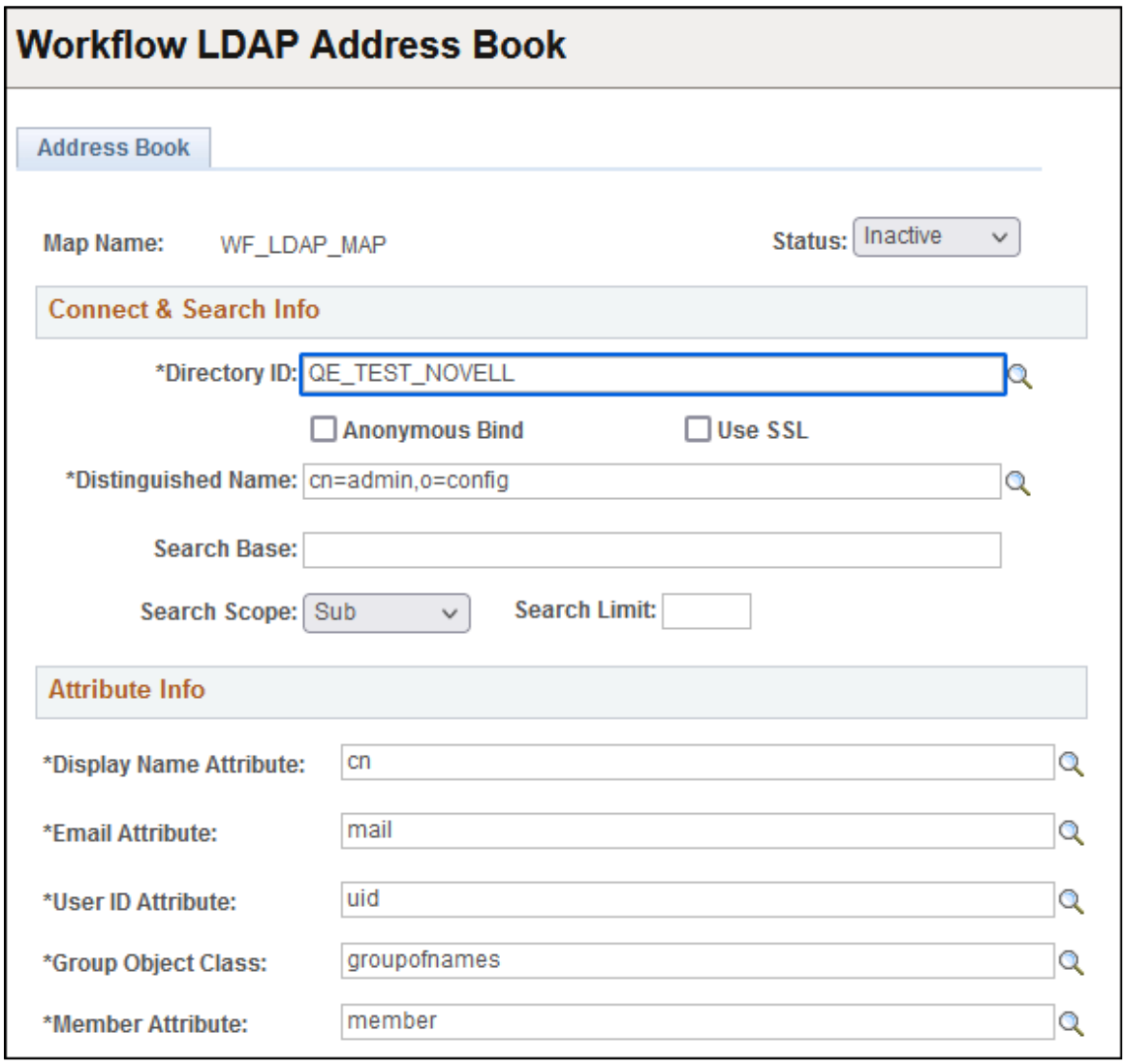


Field or Control	*Description*
Delete Associated Maps	Deletes the authentication and user profile maps from the configuration.
Delete Associated Searches	Deletes any searches related to the directory configuration.
Delete Associated Role Rules	Deletes any role rules that you have specified for a configuration.
Delete Associated Entry Rules	Applies to the PeopleSoft Directory Interface product only.
Delete Directory Configuration	After you have made the appropriate choices, click this button to perform the delete process. If you click this button with nothing selected, the system deletes only the directory ID and leaves all of the other configuration information intact.

Working with the Workflow Address Book

Access the Address Book page (**PeopleTools** > **Security** > **Directory** > **Workflow LDAP Address Book**).

This example illustrates the fields and controls on the Address Book page.



Use the Address Book page for configuring LDAP address lookups for use with user-initiated notifications in PeopleSoft Workflow. This page contains the controls needed to retrieve the necessary addresses from the directory. This page applies only if you store user information in a directory.

Field or Control	***Description***
Map Name	Displays the name of the workflow address book map.
Status	Select *Active* or *Inactive.*

Connect & Search Info

Field or Control	*Description*
Directory ID	Select the directory ID of the directory that you intend to use for authentication.
Anonymous Bind	If all directory data required for authentication and user profile maintenance is visible to an anonymous connection, select this check box.
Use Secure Sockets Layer	Select this option if you are implementing an SSL connection between PeopleSoft and the directory.
Distinguished Name	Enter the distinguished name (DN) associated with the directory ID where you want to start the workflow address book search.
Search Base	Enter the root of the directory information tree under which the system should search for user information.
Search Scope	Select the search scope for this search. Values are: *Base:* Not applicable. You should not use *Base* on the authentication map. *One:* The query searches only the entries one level down from the entry in the **Search Base** field. *Sub:* The query searches the entire sub tree beneath the search base entry.
Search Limit	Enter the maximum number of search results to return. The maximum is *99999*.

Attribute Info

Field or Control	*Description*
Display Name Attribute	Select the attribute to associate to the display name in the workflow address book.
Email Attribute	Select the attribute to associate to the email in the workflow address book.

Field or Control	*Description*
User ID Attribute	Select the attribute to associate to the user ID in the workflow address book.
Group Object Class	Select the attribute to associate to the group object class in the workflow address book.
Member Attribute	Select the attribute to associate to the member attribute in the workflow address book.

Related Links

"Understanding Events and Routings" (Workflow Technology)
"Setting Workflow Routing Options" (Workflow Technology)

Enabling Signon PeopleCode for LDAP Authentication

Access the Signon PeopleCode page (**PeopleTools** > **Security** > **Security Objects** > **Signon PeopleCode**).

LDAP Authentication runs as Signon PeopleCode that must be enabled and configured to be carried out with proper permissions.

To enable Signon PeopleCode:

1. Click the **Invoke As** option that applies to your configuration.

   Do you want to use a default user ID, or do you want the Signon PeopleCode to be invoked by the user ID of the user who happens to be signing on to the system? Either way, the value for the user ID and password must be a valid PeopleSoft User ID and password.

   For LDAP authentication, you may need to use Invoke As if the value entered on the Signon Page is not also a valid PeopleSoft OPRID. For instance, if someone signs on using an EmailID, Invoke as must be used since the email ID is not a valid PeopleSoft OPRID.

2. Locate the row for the LDAP_Authentication function on the Record FUNCLIB_LDAP.

3. Select the **Enabled** check box (if it is not already selected by default).

4. Ensure that the **Exec Auth Fail** check box is selected; if PeopleSoft authorization fails, then Signon PeopleCode is carried out.

   PeopleSoft authorization always fails if you are using LDAP authentication.

5. Click **Save** at the bottom of the page.

6. Reboot any application servers running against the local database.

Note: If you intend to use the User Profile Map, you also need to enable LDAP_PROFILESYNCH. The same options apply.

Using LDAP Over SSL (LDAPS)

This section provides an overview of SSL and discusses SSL between PeopleSoft and LDAP.

Understanding SSL

SSL is a protocol developed by Netscape that defines an interface for data encryption between network nodes. To establish an SSL-encrypted connection, the nodes must complete the SSL handshake. These are the simplified steps of the SSL handshake:

1. Client sends a request to connect.
2. Server responds to the connect request and sends a signed certificate.
3. Client verifies that the certificate signer is in its acceptable certificate authority (CA) list.
4. Client generates a session key to be used for encryption and sends it to the server encrypted with the server's public key (from the certificate received in Step 2).
5. Server uses its private key to decrypt the client generated session key.

Establishing an SSL connection requires two certificates: one containing the public key of the server (server certificate or public key certificate) and another to verify the CA that issued the server certificate (trusted root certificate). The server needs to be configured to issue the server certificate when a client requests an SSL connection, and the client needs to be configured with the trusted root certificate of the CA that issued the server certificate.

The nature of those configurations depends on both the protocol being used and the client and server platforms. In most cases, you replace HTTP with LDAP. SSL is a lower level protocol than the application protocol, such as HTTP or LDAP. SSL works the same regardless of the application protocol. To connect to a directory server over LDAPS from a PeopleSoft application, SSL has to be configured in the directory server and PeopleSoft application.

Note: Establishing LDAPS is not related to web server certificates or certificates used with PeopleSoft integration.

SSL Between PeopleSoft and LDAP

You can use LDAP Business Interlink to establish a secure LDAP connection between the application server and the LDAP server.

To establish the secure connection between the PeopleSoft application server and the LDAP server you will need the following certificates:

- A server certificate for the LDAP server.
- The trusted root certificate from the CA that issues the server certificate.

Installing and Removing Root CA Certificates in PeopleSoft Databases

To install Root CA Certificates into PeopleSoft databases:

1. Select **PeopleTools** > **Security** > **Security Objects** > **Manage Digital Certificates**.

   The list of installed certificates appears.

2. Click the insert row button (+) in the last row of the displayed certificates.

   A blank row appears.

3. Select *Root CA* from the **Type** drop-down list box.

4. Enter a meaningful name as the alias of this certificate in the **Alias** field.

5. Click the **Issuer Alias** field prompt button.

   The name of the Alias automatically populates the **Issuer Alias** field.

6. Click the **Add Root** link.

   The Add Root Certificate page appears. Minimize the browser window.

7. Open the root CA certificate with a text editor and copy the contents.

8. Maximize the browser and paste the contents into the text box.

9. Click the **OK** button to see the new digital certificate.

10. Reboot the application server.

To remove root CA certificates from PeopleSoft databases:

1. Select **PeopleTools** > **Security** > **Security Objects** > **Manage Digital Certificates**.

   The list of installed certificates appears.

2. Click the delete row (–) button in the row of the certificate you want to remove.

   A Delete Confirmation message box appears.

3. Click the **OK** button to confirm the deletion.

4. Reboot the application server.

Enabling LDAP Authentication Over SSL in PeopleSoft Applications

To enable LDAP authentication over SSL in PeopleSoft applications:

1. Follow the documentation for your directory server to add the server certificate to your directory server.

2. Install the root CA certificate into the PeopleSoft database.

3. Select **PeopleTools** > **Security** > **Directory** > **Configure LDAP Directories** > **Directory Setup** to access the Directory Setup page.

The **SSL Port** field must reflect the correct LDAPS port for the directory server.

4. Click the Test Connectivity tab.

   You must see *SUCCESS* for the SSL transactions to work. If you see *FAILURE* here, the LDAP authentication will not succeed over SSL.

5. Select **PeopleTools** > **Security** > **Directory** > **LDAP Authentication Maps** to access the Authentication Map page, and select the **Use Secure Sockets Layer** check box.

6. Enable the LDAP_AUTHENTICATION Signon PeopleCode.

   See Enabling Signon PeopleCode.

7. Reboot the application server.

Viewing SSL for LDAP Transactions Setup Examples

For the LDAP transactions between PeopleSoft and a directory server, SSL must be configured in both PeopleSoft and the directory server. This section provides a sample SSL configuration between directory servers such as Oracle Internet Directory, Active Directory Server, Sunone, and PeopleSoft applications.

Important! The procedures outlined in this section are provided as examples. They may not necessarily apply to all situations. Verify the appropriate documentation for further details.

Setting Up SSL for Oracle Internet Directory (OID)

To set up SSL for OID:

1. Create certificate request in the wallet.
2. Create a new configuration set for SSL in Oracle Directory Manager.
3. Configure OID with the newly created configuration set.

Creating the Certificate Request in the Wallet

To create the certificate request:

1. Open Oracle Wallet Manager and select Operations, Add Certificate Request.
2. Fill in the fields and click the **OK** button.
3. Select Wallet, Save. (By default, it is stored in C:\Wallets.)

Creating a New Configuration Set for SSL in Oracle Directory Manager

To create a new configuration set for SSL in Oracle Directory Manager:

1. Open the Oracle Directory Manager and log in as an *admin*.
2. From the Server management section on the left pane, select the *Default Configuration Set*.

The Default Configuration Set properties appear in the right pane.

3. From the tool bar, click the **Create Like** icon.

   A new configuration set will be created.

4. In this new configuration set, change these properties:

   Number of Child Processes = *4*

   Non SSL Port = <*Any number other than 389*>. For example, 399.

5. Click the SSL Settings tab and enter the following values:

   SSL Authentication = *SSL Server Authentication*.

   SSL Enable = *Both SSL and Non SSL*.

   SSL Wallet = <*path of the Wallet*>. For example, file:C:\wallets.

   SSL Port = <*any number other than 636*>. For example, 646.

Note: The port numbers for both SSL and non-SSL can be changed to *any* values other than the default configuration set port values.

Configuring OID with the Newly Created Configuration Set

To configure OID with the newly created configuration set:

1. Restart the oidldapd server by navigating to <Oracle_Home>\ldap\admin and running the following commands in the command prompt:

   ```
   oidctl connect=<database SID> server=<OID server type value> instance=<instanc⇒
   e number value> stop
   ```

   Example: `oidctl connect=orcl server=oidldapd instance=1 stop`

2. Start the OID with the new configuration set (configset=1). The default configuration set is demoted (configset=0).

   ```
   oidctl connect=<database SID> server=<OID server type value> instance=<instanc⇒
   e number value> configset=<new configset value> start
   ```

   Example: `oidctl connect=orcl server=oidldapd instance=1 configset=1 start`

3. Close the Oracle Directory Manager and log in through SSL.

   Enter the wallet path and the wallet password in the login dialog.

   Note: If the SSL is incorrectly configured, you will not be able to log in.
   The wallet path should be given as file:C:\wallets. The path of the wallet is sufficient; the wallet name is unnecessary.

Setting up SSL for Active Directory Server

Any utility or application that creates a valid PKCS #10 request can be used to form the SSL certificate request. The following example uses *certreq.exe* to form the request.

To set up SSL for Active Directory Server (ADS):

1. Find the Fully Qualified Domain Name (FQDN).
2. Request a server authentication certificate.
3. Verify an LDAPS connection.

To create certificate request, the Fully Qualified Domain Name (FQDN) of the Domain Controller (DC) is needed.

Finding the FQDN

To find the FQDN:

1. Select **Start** > **Programs** > **Administrative Tools** > **DNS**.

   The dnsmgmt window opens.

2. Double-click the host name of your machine, and you will see the FQDN.

Requesting a Server Authentication Certificate

To request a server authentication certificate:

1. Copy and paste the following text into a new text file and save it as *request.inf*:

```
; ----------------- request.inf -----------------

[Version]

Signature="$Windows NT$"

[NewRequest]

Subject = "CN = LAB-SUMAHADE-WF.adserver.coretools”
; replace with the FQDN of the DC
KeySpec = 1
KeyLength = 1024
; Can be 1024, 2048, 4096, 8192, or 16384.
; Larger key sizes are more secure, but have
; a greater impact on performance.
Exportable = TRUE
MachineKeySet = TRUE
SMIME = False
PrivateKeyArchive = FALSE
UserProtected = FALSE
UseExistingKeySet = FALSE
ProviderName = "Microsoft RSA SChannel Cryptographic Provider"
ProviderType = 12
RequestType = PKCS10
KeyUsage = 0xa0

[EnhancedKeyUsageExtension]

OID=1.3.6.1.5.5.7.3.1 ; this is for Server Authentication
```



```
;-----------------------------------------------
```

2. Provide the fully qualified DNS name of the domain controller in the request. The semicolon (;) is used to indicate that the following text through the end of the line is a comment.
3. Create the request file and then, in a command prompt, navigate to the path where the request is and type the following command:

   ```
   certreq -new <Name of the inf file> <name of the request file>
   ```

   Example: `certreq -new request.inf request.req`

   A new *request.req* is created in the current directory. This is the base64-encoded request file.
4. Submit the request to a CA for a server certificate. Save the server certificate, *servercert.cer*, on your machine. The saved certificate must be base64–encoded.
5. Accept the issued certificate by opening a command prompt, navigating to the path where the server certificate is stored, and executing the following command:

   ```
   certreq -accept <Name of the server certificate>
   ```

   Example: `certreq -accept servercert.cer`
6. Now the certificate is installed in your personal store. A private key is associated with this certificate. Verify this key by referring to the ADS documentation.
7. Restart the domain controller by restarting the server.

Verifying an LDAPS Connection

To verify an LDAPS connection:

1. Start the Active Directory Administration Tool (ldp.exe) by selecting **Start** > **Run** > **ldp.exe.**
2. On the Connection menu, click **Connect**.
3. When prompted, enter the name of the domain controller (enter the FQDN) to which you want to connect and the SSL port number.
4. Click OK.

   The RootDSE information should appear in the right pane, indicating a successful connection.

Setting up SSL for Sunone Directory Server (iPlanet)

1. Open the Sunone Directory Server console and select **Manage Certificates** from the Tasks tab.
2. Select **Request** and then **Next**.
3. Enter your computer name (or server name) and other organizational details.
4. Enter a password and click **Next**.

   The system creates a certificate request.

5. Click the **Copy to Clipboard** button to copy this request to the clipboard, or save the request to a file.
6. Submit the Certificate Request to a trusted CA and download the server certificate, for example, *servercert.cer*.
7. In the directory server, open the Manage Certificates page.
8. On the Server Certs tab, click the **Install** button.
9. Select this local file. Click the **Browse** button and select the server certificate, *servercert.cer*. Click **Next** on each of the following two pages.
10. Enter a name and a password and then click **Done**.

Setting Up SSL in PeopleSoft Applications

This section discusses how to configure the LDAP business interlink to establish SSL encrypted LDAP connections. The LDAP business interlink uses a root CA certificate that you install in the PeopleSoft database through the Digital Certificates page.

To enable SSL, the SSL parameter in the LDAP business interlink should be set to *YES* either:

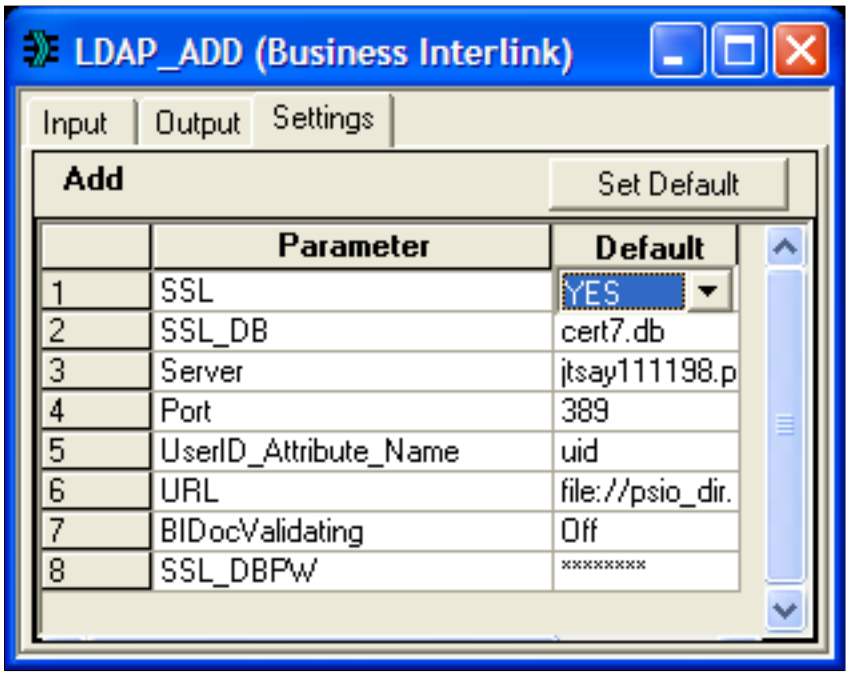
- Manually in Application Designer.
- Programmatically through PeopleCode.

Setting the Business Interlink SSL Parameter in Application Designer

To set the SSL parameter in Application Designer:

1. Open an existing instance of the LDAP business interlink, or create a new instance.
2. Select the Settings tab.
3. Set the SSL parameter to *YES*.
4. Save the business interlink.

This example shows the correct setting of the SSL parameter for the LDAP_ADD business interlink.



Note: This example shows the LDAP_ADD business interlink transaction, but it applies to all LDAP business interlink transactions.

Setting the Business Interlink SSL Parameter Programmatically

To set the business interlink SSL parameter programmatically:

1. Drag the business interlink definition into the PeopleCode editor. The following code is created:

```
/* ===>
This is a dynamically generated PeopleCode template to be used only as a helpe⇒

r
to the application developer.
You need to replace all references to '<*>' OR default values with references ⇒

to
PeopleCode variables and/or a Rec.Fields.*/
/* ===> Declare and instantiate: */
Local Interlink &LDAP_SEARCH_1;
Local BIDocs &inDoc;
Local BIDocs &outDoc;
Local boolean &RSLT;
Local number &EXECRSLT;
&LDAP_SEARCH_1 = GetInterlink(INTERLINK.LDAP_SEARCH);


/* ===> You can use the following assignments to set the configuration paramet⇒

ers.
*/
&LDAP_SEARCH_1.SSL = "NO";
&LDAP_SEARCH_1.SSL_DB = "cert7.db";
&LDAP_SEARCH_1.URL = file://psio_dir.dll";
&LDAP_SEARCH_1.BIDocValidating = "Off";
...
```

Note: This example uses the search transaction, but the principle applies to any transaction.

2. Change the SSL parameter setting to indicate that SSL should be used. For example:
   ```
   &LDAP_SEARCH_1.SSL = "YES";
   ```

Note these points:

- The SSL parameter setting in PeopleCode takes priority over the setting in Application Designer. For example, setting *YES* in Application Designer and *NO* in PeopleCode will result in a non-SSL transaction.
- The application server binds as a client to the LDAP server as part of the authentication, so it is only necessary to have access to the root certificates. The LDAP administrator at your site should have already installed a server (Node) certificate on the LDAP Server.
- Whenever you enable or disable Signon PeopleCode, reboot the application server domain.
- Whenever you install or uninstall a certificate, reboot the application server.

Chapter 9

Employing Signon PeopleCode and User Exits

Understanding the Delivered External Authentication Solutions

PeopleSoft delivers the most common authentication solutions and packages them with our application for you to use. This saves you the trouble of developing your own solutions and saves you time with your security implementation.

Note: The traditional method, where the user submits signon credentials that the system compares to a row in the PSOPRDEFN table, is a valid means of authentication; however, it is not a recommended method for increased scalability and manageability as you deploy applications to the internet.

The authentication solutions are delivered PeopleCode programs that you can include in your Signon PeopleCode. The following table describes each function that appears on the Signon PeopleCode page:

Function	*Exec Auth Fail*	*Description*
WWW_Authentication	Not Required	Applies when you want the browser to pass the client certificate to the web server for authentication by mutual authentication Secure Sockets Layer/ Transport Layer Security (SSL/TLS) at the web server level (also known as client authentication). In this situation, you configure PeopleSoft to "trust" the authentication performed by a third-party system at the web server. The function performs the following: 1. Extracts the user's distinguished name (DN) from the client certificate passed to the application server by the HTTP server. 2. Sets a global variable to the DN for a subsequent call to the LDAP_ ProfileSynch function. 3. Converts the DN to a PeopleSoft user ID and sets the current user context.

Function	*Exec Auth Fail*	*Description*
LDAP_Authentication	Required	Applies when you want the user to submit signon credentials at the signon page, and then the system passes the credentials to the directory to perform authentication. This function performs the following: 1. Searches the directory for all entries that match the entered user name. 2. Attempts to bind to the directory for each found DN using the entered password. 3. Sets a global variable to the bound DN for a subsequent call to LDAP_ ProfileSynch. 4. Converts the DN to the appropriate PeopleSoft Username and sets the current user context.
SSO_Authentication	Not Required	Applies in situations where you have single signon configured. The system authenticates the user's single signon token, which has already been issued by another database (node). This function performs the following: 1. Converts the PeopleSoft User ID to a DN. 2. Sets a global variable for a subsequent call to LDAP_ ProfileSynch.

Function	*Exec Auth Fail*	*Description*
LDAP_ProfileSynch	Not Required	Applies in situations where PeopleSoft user profiles need to be created or updated with data stored in an LDAP directory. The function requires that the global variable &global_DN has been initialized by one of the previous authentication functions. Remember that regardless of how a user is authenticated, each user populates a row in PSOPRDEFN to which applications can refer during transactions (if necessary). The LDAP _ProfileSynch updates that row in PSOPRDEFN (or user profile cache) with the most current information. As delivered, this function performs the following: 1. Retrieves the LDAP entry specified by &global_DN. 2. Either creates or updates the corresponding PeopleSoft user profile. **Note:** One of the XXX_Authentication functions needs to be carried out prior to running LDAP_ProfileSynch. PeopleSoft provides disabled example Signon PeopleCode with this function. If you work with the NDS, Active Planet, or iPlanet directories, you can use this Signon PeopleCode to assign roles dynamically at sign-on time. See LDAP_ProfileSynch Considerations.

When using any of the delivered external authentication solutions, the following items apply:

- All functions get the LDAP server configuration from specifications in **PeopleTools** > **Security** > **Directory** > **Configure LDAP Directories**.
- All functions support a single database—multiple databases are not required.

This section discusses:

- WWW_Authentication considerations.
- LDAP_Authentication considerations.
- SSO_Authentication considerations.
- LDAP_ProfileSynch considerations.

WWW_Authentication Considerations

If you intend to authenticate your users at the web server level using mutual authentication SSL/TLS (also known as client authentication), the users that are authenticated at the web server level must signon to the system using a different web site than users of the other authentication methods.

When you configure a PeopleSoft site to enable public access, a public user ID and password in the web profile provide automatic authentication. Keep in mind that this enables public access for the entire site. The web server always passes the specified public user ID and password to the application server. So, if you want some users to be authenticated by PeopleSoft rather than at the web server level, they must sign in through a PeopleSoft site that has public access disabled.

Important! The PeopleCode **RevalidatePassword( )** and **SwitchUser( )** built-in functions don't work during a user session for which you're using WWW_Authentication.
In WWW_Authentication, PeopleSoft performs no validation of users and their passwords. The Signon PeopleCode simply accepts the web server's word that the user was properly authenticated. Your PeopleSoft application has no way to revalidate the user's password in this case, so you shouldn't call **RevalidatePassword** or **SwitchUser** after WWW_Authentication has been used.
You can determine whether WWW_Authentication has been used by examining a global variable. The Signon PeopleCode for WWW_Authentication sets the PeopleCode global variable called *&authMethod* to the value *WWW* when a successful signon occurs. In PeopleCode where you want to call **RevalidatePassword** or **SwitchUser**, first examine *&authMethod*. If it's not equal to *WWW*, you can call those functions.

Related Links

“RevalidatePassword” (PeopleCode Language Reference)

LDAP_Authentication Considerations

When using LDAP_Authentication, the default searching behavior can be overridden by entering *attribute*=%UserId% in the Search Attribute edit box on the Directory Setup page. When you insert this syntax, the system constructs the DN of the user by concatenating the search attribute plus the entered user name with the search base.

For example, given the setup depicted in the following example, if the user entered *Sschumacher* in the User Name edit box of the signon page, the DN would be:

```
uid=Sschumacher,ou=Inkoop,o=ccb.com
```

This constructed DN would be used for the bind attempt rather than searching the directory with the search filter of:

```
uid=Sschumacher
```

SSO_Authentication Considerations

If you are using SSO_Authentication and LDAP_ProfileSynch to automatically generate profiles, then the value of the LDAP attribute mapped to User ID *must be* unique throughout the directory.

The PeopleSoft User ID uniquely identifies a person within PeopleSoft, and a DN uniquely identifies a person within the directory. PeopleSoft maps the PeopleSoft User Profile to a directory entry by specifying the directory attribute that holds the value of the PeopleSoft User ID.

You specify the appropriate mapping between the PeopleSoft system and your directory using the User Profile Caching component. On the Mandatory User Properties page, you must equate the PeopleSoft User ID attribute with an LDAP attribute. For example, in many cases the PeopleSoft User ID is mapped to the LDAP attribute of uid.

With a single signon token, the system can provide the Signon PeopleCode with only a user ID value to identify a person. Then the system must search the directory to find the corresponding DN. If multiple entries within the scope of the search have the same value on the User ID attribute, then PeopleSoft is unable to determine which entry corresponds to the user.

Note: It is not required to use these functions to enable single-signon within PeopleSoft. The SSO_ Authentication combined with the LDAP_ProfileSynch applies only to situations where you want cache profile data from a directory if the user presents a single-signon token during signon.

LDAP_ProfileSynch Considerations

If you work with the NDS, Active Directory, or iPlanet directories and would like to assign roles dynamically at sign-on time, you can use the disabled example Signon PeopleCode that PeopleSoft has provided with this function. Directory-specific information is included in the comments of the code.

Note: This Signon PeopleCode provides a basic framework for dynamically assigning roles at sign-on time. If you want to dynamically assign roles at sign-on time, you must modify this code to work specifically with your NDS, Active Directory, or iPlanet directory schema. You should attempt this only if you are familiar with your directory schema and with writing PeopleCode.

Using Signon PeopleCode

This section provides overviews of Signon PeopleCode and Signon PeopleCode permissions, and discusses how to:

- Modify Signon PeopleCode.
- Enable Signon PeopleCode.
- Access X.509 certificates.

Understanding Signon PeopleCode

Signon PeopleCode runs whenever a user signs in to a PeopleSoft application. The main purpose of Signon PeopleCode is to copy user profile data from a directory server to the local database whenever a user signs in. This ensures that the local database has a current copy of the user profile. Because Signon PeopleCode runs at each signon, you are not required to maintain the local copy of the user information.

Signon PeopleCode is not limited to Lightweight Directory Access Protocol (LDAP) integration. You can also use Signon PeopleCode and business interlinks to synchronize a local copy of the user profile with any data source when a user signs in. Because the signon program is written in PeopleCode, you can customize it any way that suits your site requirements.

The basic process flow of Signon PeopleCode is as follows:

1. A user enters user ID and password on the signon page.
2. PeopleTools attempts to authenticate a user with the local PeopleSoft password.
3. Signon PeopleCode runs.

   It verifies the user and password, and then updates the local cache of user profiles stored in the PeopleSoft database.

Signon PeopleCode runs only when a user logs on through Pure Internet Architecture, the portal, or a three-tier Windows workstation.

Note: If you are using LDAP authentication, the PeopleSoft authentication process will fail because the user password is not stored within the PeopleSoft database. Because of this, if you are using LDAP authentication, you set your Signon PeopleCode program to run when PeopleSoft authentication fails.

Understanding Signon PeopleCode Permissions

Signon PeopleCode scripts run with full permissions of the user they're invoked as. This includes access to the database using Structured Query Language (SQL), access to the file system, business interlinks, component interfaces application messaging, and so on. A developer could conceivably write a Signon PeopleCode program that exposed or corrupted sensitive information. To minimize this risk, you should follow these guidelines:

- You should limit access to the Signon PeopleCode setup page to trusted administrators only.

  This will prevent people from configuring un-trusted PeopleCode programs to run at sign-on time.

- If you aren't implementing external authentication at your site (all your users are authenticated based on an existing user ID and password with the PeopleSoft database), you should not have the "Exec Auth Fail" column selected for any Signon PeopleCode scripts.
- After a trusted administrator configures the list of functions that should run at sign-on time, you should use Object Security to restrict access to the record objects that contain the programs.

  Only trusted developers should be allowed to modify the PeopleCode on these records.

- Even for trusted developers, it is a good idea to have a second person review the code before testing and moving to production.
- No developer or administrator should have access to the Signon PeopleCode setup page, or the records that contain the Signon PeopleCode functions in a production system.

Note: The password that the user types on the signon page is never visible to the Signon PeopleCode developer. It is impossible to write a script that captures a password entered by a user, and store it in a file or database table.

Modifying Signon PeopleCode

Signon PeopleCode is record PeopleCode, which you view and edit on the record with which the program is associated. PeopleSoft applications deliver a PeopleCode program for directory authentication. It is intended for production use but it can also be used as a sample that shows many of the technologies you can include within a Signon PeopleCode program. You can find the delivered PeopleCode program on

the following record: FUNCLIB_LDAP.LDAPAUTH (FieldDefault). You can customize it as needed for testing or production use.

Open the record in PeopleSoft Application Designer, and view the PeopleCode with the PeopleCode Editor. The delivered PeopleCode accommodates as many different directory scenarios as possible; it demonstrates use of the business interlink and component interface technologies. You may want to modify the authentication PeopleCode to improve login performance or to accommodate any special directory authentication needs. The delivered program that ships with PeopleTools has the following general flow:

1. Searches the directory server for the user profile of the user signing in.
2. Using the password the user entered at the signon page, the program attempts to bind (or connect) to the directory server.

   If the connect succeeds, then the password is valid.
3. Retrieves the user profile of the user signing in.

   The program gets the profile from the directory server and creates a local cache copy within the PeopleSoft database. This improves performance by enabling the PeopleSoft applications to access the user profile locally, rather than making a call to the LDAP server every time they need user profile data. If a locally cached copy already exists for the user signing in, the local cache is updated according to the current user in the directory server.

Note: To see what the Signon PeopleCode program does, use the PeopleCode debugger. This enables you to step through the program step-by-step.

The following table presents the key PeopleCode constructs that you use with Signon PeopleCode. Click the function to view more details in the PeopleCode product documentation:

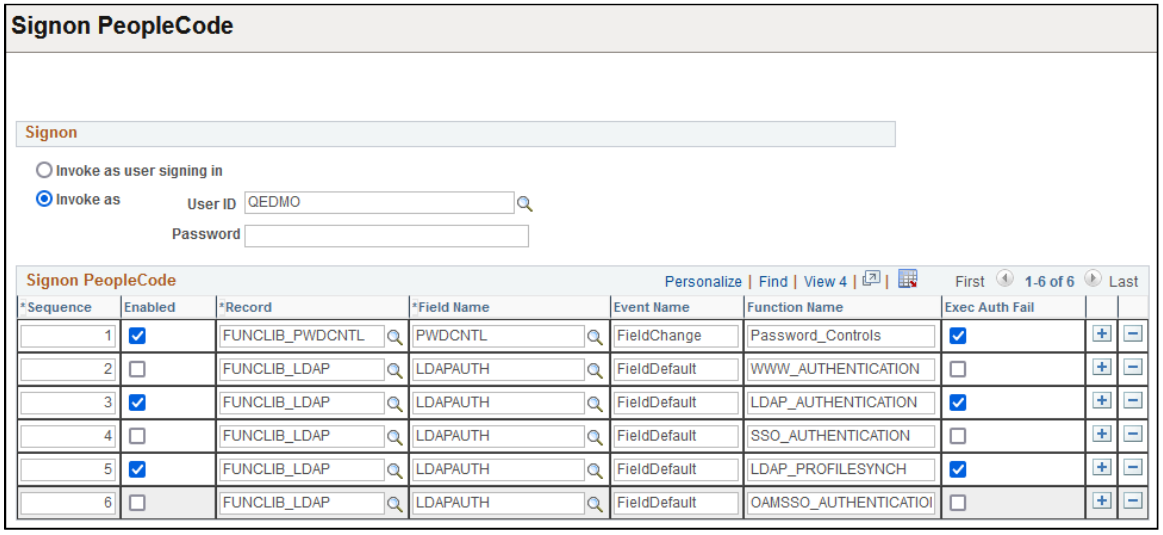
PeopleCode Function	***Description***
See “%PSAuthResult” (PeopleCode Language Reference).	Returns the result (boolean) of PeopleSoft authentication.
See “SetAuthenticationResult” (PeopleCode Language Reference).	Verifies customers who log on to the system even if the PeopleSoft authentication fails.
See “%SignonUserId” (PeopleCode Language Reference).	User ID value entered by the user on the Signon page. This applies to Pure Internet Architecture and Windows signon.
See “%SignOnUserPswd” (PeopleCode Language Reference).	User password value the user entered at the Signon page. This value is encrypted. This applies to Pure Internet Architecture and Windows signon.
See “%Request” (PeopleCode Language Reference).	The HTML request that comes from the browser. In the case of security, this includes any information submitted at the Signon page, such as user ID, password, and any additional fields if you have extended the Signon page. This applies only to Pure Internet Architecture.

Note: Do not use the `%SwitchUser` variable in Signon PeopleCode.

Enabling Signon PeopleCode

Access the Signon PeopleCode page (**PeopleTools** > **Security** > **Security Objects** > **Signon PeopleCode**).

This example illustrates the fields and controls on the Signon PeopleCode page.



Signon PeopleCode is different from other PeopleCode in that you specify which Signon PeopleCode you want to have on a specific Signon PeopleCode page. Notice that the PeopleSoft Password Controls program, which is written in PeopleCode, is also on this page.

By default, some of the Signon PeopleCode programs are disabled. You enable them on this page. You can also enable them by enabling password controls on the Password Controls page or by enabling directory authentication on the Directory Authentication component. After enabling each option on the appropriate page, the system enables the associated PeopleCode program on the Signon PeopleCode page.

Note: Using PeopleSoft password controls is valid only if you are *not* using LDAP authentication. When you're using LDAP authentication, the directory server, not PeopleSoft, controls the password.

You can add your own PeopleCode programs, but you must add them to another record, and then add them to this page. You add and remove rows from the grid using the plus and minus buttons.

Field or Control	*Description*
Invoke as user signing in/ Invoke as…	When a PeopleCode program runs, it has to have a context of a user. This is how you indicate to the system which user is executing the program. This is important because the user ID provided must have access to all of the objects that your signon program uses. For example, if you are using LDAP, notice that the Signon PeopleCode contains a business interlink and a component interface. If the user ID provided does not have the appropriate authority to business interlinks or component interfaces, the program fails. Whether you use the value of the user signing in or you create a default user ID for all signon attempts depends on your implementation. For example, if your Signon PeopleCode creates local copies of users, you have to configure that program to be “Invoked as” an existing user in the system. In this case, you should create a new user within PeopleSoft that only has authority to access the objects required within your PeopleCode program. You should then enter this user as the “Invoke As” user.
Sequence	Displays the sequence in which the signon programs run. You can change the sequence by changing the numerical value in the edit box. The application server runs all programs in the ascending order in which they appear.
Enabled	To enable a program to run at signon, select this check box. If it is not selected, then the system ignores the program at signon.
Record	Specify the record on which your record PeopleCode exists.
Field Name	Enter the specific field that contains the PeopleCode.
Event	Enter the event that triggers a particular program.
Function Name	Enter the name of the function to be called.
Exec Auth Fail (execute authentication fails)	Select this check box to "execute if PeopleSoft authentication fails." In other words, if PeopleSoft does not successfully authenticate the user based on the password within the PeopleSoft database, you still want the program to run. For example, you want the LDAP authentication program to run after PeopleSoft denies access so that your program can authenticate the user instead. Also, you can leave this option clear to further secure your system. If you aren't using LDAP authentication, leaving this option unchecked prevents any program or script from running if your PeopleSoft authorization fails.

Accessing X.509 Certificates

X.509 certificates are used to authenticate a user at the web server level—SSL/TLS with client-side authentication. You can use PeopleCode to access X.509 certificates.

When you use certificate authentication with PeopleSoft, users do not see the PeopleSoft signon page and enter a user ID. Because of this, the X.509 certificate needs to be available in the Signon PeopleCode so you can write PeopleCode that maps the certificate to a PeopleSoft user ID.

The following sample PeopleCode shows how you access X.509 certificates in Signon PeopleCode:

```
Local string &clientDN;
&clientDN = %Request.GetParameter("com.peoplesoft.tools.ssl_client_dn");
```

The value of &clientDN might be similar to the following:

```
E=tom_sawyer@peoplesoft.com, C=US, S=California, L=Pleasanton, O=PeopleSoft, OU=Peo⇒

pleTools, CN=Tom Sawyer
```

Using the Web Server Security Exit

This section provides an overview of the web server security exit and discusses how to:

- Create a public access user.
- Modify the web profile.
- Write a Signon PeopleCode program.
- Sign in through the web server.

Understanding the Web Server Security Exit

Part of the integration technology PeopleSoft delivers is to ensure that our security or authentication system is open and flexible. Because the PeopleSoft applications are now designed for internet deployment, many sites must take advantage of the authentication services that exist at the web server level.

Note: The exits described here are offered in addition to the Signon PeopleCode running on the application server, which itself provides integration. There are no PeopleSoft user ("psuser") exits on the application server; Signon PeopleCode replaces that functionality. On the client side, the functionality is the same as previous releases. You should use Signon PeopleCode when developing new signon integration. The topics in this section support previous implementations.

This section describes a procedure that enables you to configure your implementation so that PeopleTools authentication logic "trusts" the authentication performed at the web server level. The following list presents examples of some of the third-party authentication technologies with which you may want to integrate:

- Web single signon or authorization or authentication solutions.
- Client-side SSL/TLS authentication provided by web servers.

- Public Key Infrastructures, either stand-alone or embedded as part of the network operating system environment.

Note: The previous list is not a list of certified integration points, just examples of authentication technologies that exist in the industry.

For the web server exit configuration to work successfully, the following assumptions should be true:

- You want to authenticate the user at the web server level only, not within the PeopleSoft application server.

  (The configuration discussed in this section enables you to authenticate users within the web server instead of the default configuration, where the application server controls the authentication logic.)

- Your web server environment includes a mechanism to identify and authenticate a user.

  This may be through a sign in page with a user ID and password, through a digital certificate, or through one of several industry-standard authentication methods.

- Your web server has the capability of passing the user ID to the application server through the HTTP request PeopleCode object.

  For this you can use an HTTP header variable, a cookie, or a form field.

Note: Configuring the following authentication system is not a delivered feature. It requires development outside of your PeopleSoft application, and because of that, you should have the appropriate level of internet development expertise to make sure that you are passing the appropriate information to the PeopleSoft system.

Creating a Public Access User

You create a public or default user profile by using PeopleTools Security. This user profile does not require any roles or permission lists. You should consider creating a long password that is difficult to guess.

For this example, we create a user profile with these parameters:

- User ID: PUBUSER
- Password: *passwordpasswordpassword*

See Working With User Profiles.

Modifying the Web Profile

After you create the default user, you can modify the web profile to include the default user sign in information.

To modify the web profile to include the default user sign in information, you first must enable public access to the portal. In the **Public Users** section of the Web Profile Configuration - Security page, select **Allow Public Access** to indicate that the system should not prompt users to sign in when they click a direct link to a page. When this is selected, the PeopleSoft system does not display the password page to

the user. Instead, the system authenticates users with the values specified in the **User ID** and **Password** fields in the same section of the page.

Note: In the following discussion, notice that the user is never actually signed in as "PUBUSER." The user ID you specify is just a temporary value used to initiate a secure connection to the application server. The application server then determines the correct user ID using Signon PeopleCode. The correct user ID is contained in the request object, and all the other user information, such as language code, roles, and so on, is already stored in the PeopleSoft system or an LDAP directory server.

Besides selecting the **Allow Public Access** check box, you also must set the user ID and password parameters to reflect the user ID created in the previous step. For example, set the **User ID** field value to *PUBUSER*, and the **Password** field to *passwordpasswordpassword*.

Because you hard-code the signon values in the web profile, no end user ever needs to know them—their use is transparent.

You should limit access to and knowledge of the public access user ID and password values. You can do this by sharing this information only with a small number of trusted security administrators. Also, you should make sure that only these select few have read access to the web profile.

Even if somebody does discover the public access user ID and password values, he or she won't be able to sign in to the PeopleSoft system. Recall that the PUBUSER doesn't have any roles or permission lists. Alternatively, a sophisticated hacker could attack the application server directly by sending it a connection request formatted in the Oracle Tuxedo/Jolt protocol and potentially assume the identity of a user. You should use network and firewall products to restrict the origin of requests sent to the application server.

Note: To prevent a user ID from being the default user on the sign in page, set the **Days to Autofill User ID** property on the Web Profile Configuration - Security page to *0*.

Related Links

"Configuring Web Profiles" (Portal Technology)

Writing a Signon PeopleCode Program

In addition to creating a default user and enabling public access, you also must write a Signon PeopleCode program that:

- Uses data within the HTTP request to determine the real user ID.

  Your web server authentication system should be configured to insert the USERID of an authenticated user into the HTTP request as a header, a form field, or cookie.

- Creates or updates the local copy of the user profile within the PeopleSoft database.

The programs developed to perform this task vary depending on where the web server inserted the user ID in the HTTP request and where the user profiles are stored. For example, some systems use an HTTP header to store the user ID, while others use cookies or form fields.

If the web server security product uses LDAP as a backend data store for user profiles, you can reuse some of the LDAP authentication PeopleCode to copy the profile from LDAP to the local database. The user profile may also be stored in another database, or a Windows domain registry. In either case, you must write PeopleCode to retrieve the value and make a local copy.

Note: You can't use the LDAP Authentication PeopleCode program as delivered. This program performs LDAP authentication and copies the user profile from an LDAP directory to the local database. You can, however, use the code that copies the profile from the directory, as a template for the code you need in this case.

The following is sample PeopleCode with the External_Authentication function. It is a simple example of retrieving the user ID from a form field named UserID:

```
/*////////////////////////////////////////////*/
Function External_Authentication()

   /*  This application server "trusts" the authentication performed by the web ser⇒

ver */
   /*  retrieve the USERID from the HTTP request and pass it to SetAuthentication R⇒

esult */

   &UserID = %Request.GetParameter("UserID");
   SetAuthenticationResult( True, &UserID, "", False);

End-Function;
```

After you have written the program, you must set the Signon PeopleCode program to run only if authentication is successful. On the Signon PeopleCode page, you set the running as follows:

- Clear the **Exec Auth Fail** check box; it must *not* be selected.

  You want this PeopleCode to run only if the connection to the application server originates from a web server that presents a valid user ID and password. In this case, the user ID is PUBUSER and the associated password. You should only select the **Exec Auth Fail** check box when the PeopleCode itself authenticates the user, not when the program relies on the web server to perform authentication.

- You must set **Invoke as** to a user profile that has the appropriate roles and permissions to do all the operations in the External_Authentication function.

  For example, if External_Authentication creates a local copy of the user profile using the User Profile component interface, signon_peoplecode_user must have permission to use this component interface. The Signon PeopleCode program runs under the signon_peoplecode_user user ID.

Note: Before running the PeopleCode, the application server authenticates the **User ID** and **Password** field values in the Public Users section of the Web Profile Configuration - Security page.

Signing In Through the Web Server

This section provides a step-by-step example of the steps that occur within the system after you have it configured to trust authentication performed at the web server level:

Step	*Component*	*Description*
1	Browser	The user clicks a link to the PeopleSoft application, for example http://serverXYZ/servlets/psportal/peoplesoft8/?cmd=start.

Step	Component	Description
2	Web server	The web server receives the request for the uniform resource locator, authenticates the user, and adds the user ID to the HTTP request for the resource. The method the system uses to authenticate the user and the method the web server uses to add the user ID to the HTTP request depends on your implementation. For example, it could be a third-party web single signon or authorization solution, a PKI/ digital certificate, or SSL/TLS with client-side authentication.
3	Servlet	The PeopleSoft servlet receives the HTTP request, which includes the user ID in a header, cookie, or form field, and connects to the application server using the public user ID and password from the web profile.
4	Application server	The application server authenticates the connection from the web server by checking the public access user ID and password against the values stored in PSOPRDEFN. The user ID and password must be valid for the connection to succeed and for Signon PeopleCode to run. **Note:** The password verification prevents a sophisticated hacker from connecting to the application server directly and carrying out service requests.
5	Signon PeopleCode	Signon PeopleCode runs, under the context of the signon _peoplecode_user, with all the permissions of this user. It grabs the "real" user ID from the HTTP request and creates a copy of the user profile in the local database (if appropriate). It also calls the PeopleCode built-in SetAuthenticationResult and passes the user ID, and an AuthResult of "true." The PeopleCode program always passes "true" for AuthResult because the application server is "trusting" the authentication logic of the web server. The Pure Internet Architecture session is set to the user ID of whatever you pass into SetAuthenticationResult. For example: `SetAuthenticationResult(True, "TSAWYER"," ", False);` In this case, the system sets the session to TSAWYER. The user can access all the pages to which TSAWYER has access.

Using the Windows Security Exit

This section provides an overview of Windows security exits and discusses how to:

- Customize PSUSER.DLL.

- Implement a customized PSUSER.DLL.

Understanding Windows Security Exits

Almost all end users will access PeopleSoft applications by using a browser, so you may not need to implement any client-side Windows exits. However, you can provide this functionality, perhaps for developers.

The Windows client-side exits are:

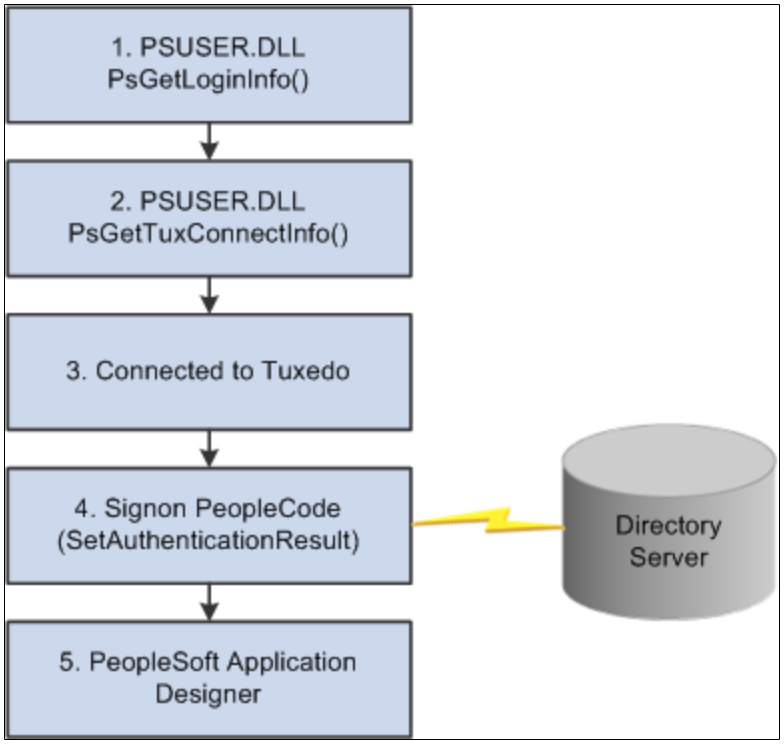
- PsGetTuxConnectInfo(): Used only for three-tier Microsoft Windows workstations running PeopleSoft Application Designer or Query, for example.
- PsGetLogonInfo(): Used for Microsoft Windows workstations in both a two-tier and three-tier environment.

Use these functions to create a customized PSUSER.DLL. These exits are used primarily for the PeopleTools Development Environment, PeopleSoft Query users, or PeopleSoft Tree Manager users. Unless you intend to deploy PeopleSoft applications to Microsoft Windows workstations, these exits are seldom used.

PsGetLogonInfo was used for the Microsoft Windows Client in previous releases to fill in the signon screen programmatically without displaying it to the user.

With the three-tier Microsoft Windows Client signon you can also bypass the PeopleSoft Signon window by modifying the PsGetLogonInfo() function as with the two-tier connection. But because you are connecting to the database through Tuxedo, there are some other authorizations that need to occur.

This diagram shows the Microsoft Windows Client three-tier signon exits.

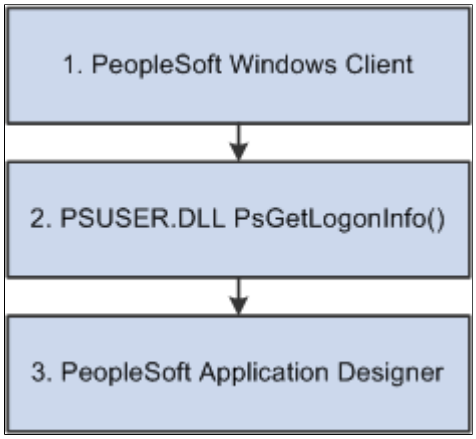


The required authorizations are as follows:

1. The PsGetLogonInfo function must specify APPSERV as the szDBType parameter to bypass the PeopleSoft Signon dialog box.
2. To connect to the Tuxedo application server, the PsGetTuxConnectInfo function retrieves authentication information from directory server.
3. If the authentication information is valid, Tuxedo allows connection.
4. Tuxedo must connect to the database server.

   The application server verifies the authentication information passed by the PsGetTuxConnectInfo function.
5. If the authentication is successful, the user is connected to PeopleTools.

The following diagram illustrates the results produced by customizing the PSUSER.DLL PsGetLogonInfo function to bypass the PeopleSoft Signon dialog box.



In this case, the sequence of events is as follows:

1. From the workstation the user runs PSTOOLS.EXE. PSTOOLS.EXE calls the PSUSER.DLL.
2. The PsGetLogonInfo function supplies user signon information.

   If information is validated by the RDBMS, the user is connected as User ID or Connect ID, and then after the security profile is retrieved and validated the user is connected as Access ID.
3. If the signon information is valid, the PeopleSoft system connects the user to the specified PeopleTool.

Customizing PSUSER.DLL

If your site has implemented a security system external to the PeopleSoft system, you can use that external system to validate your Microsoft Windows Client PeopleSoft users, also. This is done through the user exit (PSUSER.DLL), which also enables you to specify your own encryption for use in encrypting passwords.

To enable these options, you must modify several procedures in the PSUSER.C, and recompile to create a new PSUSER.DLL. Then you must install the new DLL file wherever users run the PeopleSoft executable files, such as <PS_HOME> on the file server.

In this section, we discuss the security functions that we provide and how you can tailor them for use in your own system. To successfully complete any customizations with these functions, you must be familiar with the C programming language.

PsGetLogonInfo

The PsGetLogonInfo function is always called when the PeopleSoft system is started. If you're already controlling which users can access the PeopleSoft applications—through a custom security solution—you may want to use this function to let those users start the PeopleSoft system directly without being prompted for PeopleSoft signon information. This function can also be overridden to provide information to the three-tier exit, PSGetTuxConnectInfo.

As delivered, PsGetLogonInfo returns a FALSE value and is ignored. However, if it returns a TRUE value, the PeopleSoft signon dialog box is bypassed and the information that you've coded into the function is used as the signon parameters.

You'll find this function in your *PS_HOME*\src\PSUSER\PSUSER.C file. The code initially looks like this:

```
/************************************************
* Function:     PsGetLogonInfo                                  *
*                                                               *
* Description:  Sample routine to get logon information.        *
*                                                               *
* Returns:      TRUE if logon information returned              *
*               FALSE to ignore                                 *
************************************************/

PS_EXPORT(BOOL) PsGetLogonInfo(LPPSLOGINFO lpPsLogInfo)
{

/*--------------- BEGIN SAMPLE CODE -----------------

// ask for user input only when it is the first signon
if (!lpPsLogInfo->bSubsequentSignon)
    {
    // test auto logon
    strcpy(lpPsLogInfo->szDBChange, "NO");
    strcpy(lpPsLogInfo->szDBType, "DB2");
    strcpy(lpPsLogInfo->szDBName, "C9442A");
    strcpy(lpPsLogInfo->szServerLogonSec, "NO");
    strcpy(lpPsLogInfo->szOprId, "C944201");
    strcpy(lpPsLogInfo->szOprPswd, "C944201");
    return(TRUE);
    }

----------------- END SAMPLE CODE ----------------*/

return(FALSE);

}
```

To activate the automated signon feature, you must comment out the "false" return and uncomment the "true" return line. The return value is historical and ignored. The user exit bypasses the screen only if it receives enough information.

Then you must code the appropriate logic to fill in the values for the parameters to the PSGetLogonInfo routine. If you provide all of the appropriate field values, the system proceeds directly to your default initial window specified in the PeopleSoft Configuration Manager Startup tab. Your procedure might look something like this:

```
PS_EXPORT(BOOL) PsGetLogonInfo(LPPSLOGINFO lpPsLogInfo)
{
```



```
/* test auto logon */
//strcpy(lpPsLogInfo->szDBChange, "NO");
strcpy(lpPsLogInfo->szDBType, "ORACLE");
strcpy(lpPsLogInfo->szDBName, "PSORADB");
strcpy(lpPsLogInfo->szServerLogonSec, "NO");
strcpy(lpPsLogInfo->szOprId, "MGR2");
strcpy(lpPsLogInfo->szOprPswd, "password");
return(TRUE);

//return(FALSE);
}
```

Note: If any required signon parameters are omitted, the signon screen appears and the missing values are set by default to the settings found in the registry. One way to control whether the signon dialog displays is to have PSUSER.DLL provide (or not provide) the user's password.

All parameters except bSubsequentSignon, which is Boolean, are of the data type CHAR and are defined as follows:

Parameter Name	***Description and Values***
bSubsequentSignon	An initial or subsequent signon. Values are: FALSE: Initial signon. User just started the PeopleSoft system. TRUE: Subsequent signon. User probably selected an item from the Go menu in the Development Environment (PSIDE.EXE).
szDBChange	Change database name or type. Values are: TYPE: Allow to change type and name. YES: Allow to change name only. NO: Do not allow change to either.
szDBType	Database type. Values are: DB2: DB2 z/OS through Centura Gateway. DB2ODBC: DB2 z/OS through ODBC. MICROSFT: Microsoft SQL Server. ORACLE: Oracle Server. APPSERV: Application Server.
szDBName	Database name or application server name.

Parameter Name	*Description and Values*
szServerLogonSec	The Change Password feature. Values are: YES: enabled. NO: disabled.
szOprId	User ID.
szOprPswd	User password.

PsGetTuxConnectInfo

When operating in three-tier mode, PsGetTuxConnectInfo is called after PsGetLogonInfo and just before connecting to Tuxedo. Use this function to pass authentication data (key) to the server. Use this to either supplement or replace PeopleSoft's standard authentication process.

You'll find this function in your *PS_HOME*\src\PSUSER\PSUSER.C file. The delivered code looks like this:

```
/*****************************************************************
* Function:     PsGetTuxConnectInfo                                *
*                                                                  *
* Description:  This function is called from PeopleTools just prior to *
*               connecting to Tuxedo.  The PeopleTools client sends  *
*               the data in *ppData to the PeopleSoft Tuxedo         *
*               authentication service (PSAUTH), where it can be used  *
*               as an alternative or supplement to the default       *
*               PeopleTools authentication (see PsTuxAuthExit in     *
*               pssite.c).                                           *
*                                                                  *
* TO DO:        Add logic to obtain client authentication information. *
*               An example might be NT or DCE signon information.    *
*                                                                  *
* Returns:      TRUE if logon information returned                   *
*               FALSE to ignore                                      *
*****************************************************************/

PS_EXPORT(BOOL) PsGetTuxConnectInfo(NETEXTAUTH *pExtAuth)
{

/*------------------------ BEGIN SAMPLE CODE --------------------------

// set the auth information size and allocate space for auth information
pExtAuth->nLen = 25;
pExtAuth->pData = (unsigned char *) malloc(pExtAuth->nLen);

// set your authentication string
memcpy(pExtAuth->pData, "NATHAN HORNE\0\0PEOPLESOFT\0", pExtAuth->nLen);

return(TRUE);

--------------------------- END SAMPLE CODE -------------------------*/

return(FALSE);

}
```


Implementing a Customized PSUSER.DLL

To rebuild and implement PSUSER.DLL:

1. Compile PSUSER.C and create PSUSER.DLL.

   To do this for Windows platforms, run NMAKE while in the *PS_HOME*\src\PSUSER\WINX86 directory. You must use a Microsoft Visual C++ 6.x compiler.

   On UNIX, run the shell script psuser.sh in *PS_HOME*\src\psuser.

   The resulting file, PSUSER.DLL, is used by PeopleTools (PSTOOLS.EXE), and the Windows COBOL interfaces. For Windows operating systems, you must copy this file into your COBOL directory.

2. Distribute PSUSER.DLL to workstations.

If your workstations run the PeopleSoft executable files from a common file server, you must ensure that your new PSUSER.DLL is copied to that file server. If any of your workstations run the PeopleSoft executable files locally, PSUSER.DLL must be distributed to such workstations.

Chapter 10

Implementing Single Signon

Understanding Single Signon

This section discusses:

- Single signon options.
- The PS_TOKEN cookie.

Understanding Single Signon Options

Single signon refers to the ability of users to navigate freely within a system of multiple applications after only being authenticated once. There are three different ways to configure single signon, depending on the participating applications that you have installed. The following table displays the single signon options.

Single Signon Option	***Description***
PeopleSoft-only	This option enables single signon only between multiple PeopleSoft applications, such as PeopleSoft Human Capital Management and PeopleSoft Customer Relationship Management. After a user is authenticated by one PeopleSoft application, an in-memory value gets set in the browser (PS_TOKEN cookie) that the next PeopleSoft application uses for a user credential. If you have only PeopleSoft applications, use this option. **Note:** This option is the same single signon feature offered in previous PeopleSoft releases. See Implementing PeopleSoft-Only Single Signon.
PeopleSoft and Oracle applications	If you have Oracle applications and PeopleSoft applications being used in your organization, users who have been authenticated by the Oracle system can freely access PeopleSoft applications without having to be re-authenticated. This option is tailored for sites running their PeopleSoft applications on Oracle WebLogic . This option applies to all previous PeopleTools 8.x versions. For example, if you intend to incorporate applications running on Enterprise PeopleTools 8.53, you can implement this option. See Implementing Oracle Access Manager as the PeopleSoft Single Signon Solution.

Note: You must ensure that before users attempt to use the single signon functionality, a valid user profile is defined for each user in each participating application database. You can accomplish this in a variety of ways, such as automatically generating user profiles based on users' LDAP information, replicating user profiles through Integration Broker at initial sign in, or manually defining user profiles for the authorized users before going live.

Note: Many single signon solutions require that you create a generic user profile with minimal permissions and set up this user as the default or public access user. You find information for creating the public access user profile in this documentation. You find information for specifying the identity of the public access user in the product documentation for *PeopleTools: Portal Technology*.

Related Links

Working With User Profiles
Creating a Public Access User
“Configuring Web Profiles” (Portal Technology)

Understanding the PS_TOKEN Cookie

When the system authenticates a user, it distributes the PS_TOKEN cookie to the browser. The PS_TOKEN cookie holds user authentication information in the browser that a PeopleSoft system uses to verify user access. Having the token in the browser memory allows the user to navigate freely within the system without having to provide user credentials repeatedly.

The key security features of the PS_TOKEN cookie authentication are:

- The cookie exists in memory; it is not written to disk.
- There is no password stored in the cookie.
- You can set the expiration of the cookie to be a matter of minutes or hours; so if a cookie is intercepted it will only be usable for the duration you specify.

The following table presents the fields that appear in the PeopleSoft authentication token:

Field	***Description***
UserID	The user ID of the user to which the server issued the token. When the browser submits this token for single signon, this is the user that the application server logs on to the system.
Language Code	Specifies the language code of the user. When the system uses his token for single signon, it sets the language code for the session based on this value.

Field	*Description*
Date and Time Issued	Specifies the date and time the token was first issued. The system uses this field to enforce a time out interval for the single signon token. Any application server that accepts tokens for signon has a timeout minutes parameter configured at the system level. A system administrator sets this parameter using the PeopleTools, Security, Single Signon page. The value is in Greenwich Mean Time (GMT) so it does not matter which time zone the application server is in.
Issuing System	Shows the name of the system that issued the token. When it creates the token, the application server retrieves this value from the database. Specifically, it retrieves the defined Local Node. You configure a node only to trust single signon tokens from specific nodes. Consequently, an application server needs the name of the issuing system so that it can check against its list of trusted nodes to see if it trusts the issued token. **Note:** Single signon is not related to Integration Broker, except for the fact that single signon functionality leverages the use of nodes and local nodes.

Signature	This field contains a digital signature that enables the application server using a token for single signon to ensure that the token hasn't been tampered with since it was originally issued. The system issuing the token generates the signature by concatenating the contents of the token (all the fields that appear in this table) with the node definition password for the local node. Then the system hashes the resulting string using the SHA1 hash algorithm. For example ("+" indicates concatenation),
	
	`signature = SHA1_Hash ( UserID + Lang +⇒`
	
	`Date Time issued + Issuing System + Local Nod⇒`
	
	`e Pswd )`
	
	There is only one way to derive the 160 bits of data that make up the signature, and this is by hashing exactly the same User ID, Language, Date Time, Issuing System, and node password.
	
	Note: If you are using digital certificate authentication, the signature of the digital certificate occupies this space. The above description applies to using password authentication only.

Note: Single signon does not depend on Lightweight Directory Access Protocol (LDAP) directory authentication. You can implement single signon and not LDAP, you can implement LDAP and not single signon, or you can implement both LDAP and single signon.

Implementing PeopleSoft-Only Single Signon

This section provides an overview of PeopleSoft-only single signon and discusses:

- Defining nodes for single signon.
- Working with the Single Signon page.
- Defining authorized sites for single signon.
- Setting up certificate authorization
- Single signon transaction example.
- PeopleSoft-only single signon configuration considerations.
- PeopleSoft-only single signon configuration examples.
- Securing the PeopleSoft single signon token.
- Using the single signon API.
- Configuring single signoff.

Note: In this configuration, you must create PeopleSoft node definitions for each of the participating applications. You can run any of the participating applications on Oracle WebLogic. You can use passwords or digital certificates for single signon authentication.

Understanding PeopleSoft-Only Single Signon

PeopleSoft applications supports single signon among other PeopleSoft applications. Within the context of your PeopleSoft system, single signon means that after a user has been authenticated by one PeopleSoft application server, then that user can access other PeopleSoft application servers without entering an ID or a password. Although the user is actually accessing different applications and databases—recall that each suite of PeopleSoft applications, such as HCM or CRM, resides in its own database—the user navigates seamlessly through the system.

Note: The PeopleSoft-only single signon solution applies only to PeopleSoft applications. Single signon requires that user profiles exist in all databases involved in single signon.

The user profiles to utilize single signon must be defined on all participating databases. For example, for user Marcia Brady to be able to use single signon to access *Database A, Database B,* and *Database C,* her user profile must be defined in each of the three databases.

Single signon is critical for PeopleSoft portal implementations because the portal integrates content from various data sources and application servers and presents them in a unified interface. When the users sign in through the portal, they always take advantage of single signon. Users need to signon once and be able to navigate freely without encountering numerous signon screens. Because single signon is so integral to the portal, you always need to configure it before deploying a live portal solution.

Authentication Token (PS_TOKEN)

After the first application server/node authenticates a user, the system delivers a web browser cookie containing an authentication token (PS_TOKEN). PeopleSoft uses web browser cookies to store a unique access token for each user after they are authenticated initially. When the user connects to another PeopleSoft application server/node, the second application server uses the token in the browser cookie (as long as the token is valid) to re-authenticate users automatically so they don't have to sign in repeatedly.

Note: The browser cookie is an in-memory cookie and is never written to disk. The cookie is also encrypted to prevent snooping and digitally signed to prevent tampering.

Check Token IDs

A check token ID verifies that the PS Token is still valid at the originator site or at the last site visited.

Each PeopleSoft database participating in single signon must define a check token ID on the default local node definition on the local database, as well as define the check token ID of their single signon participants on the remote node definitions of each.

You can create a system-generated check token ID by using the **Create CheckTokenID** button on the Nodes - Node Definitions page. When you click the button, the system generates a random 184 byte/248 character value and populates the value in the Check TokenID field. Note that the **Create CheckTokenID** button appears only on the Nodes - Node Definitions page for the local default node. As an alternative you can create your own custom check token ID. Custom check token IDs have a 254-character limit. Be sure to copy the ID before saving the component. You must provide this value to your single-signon participants, as they must define this value on their local database on the remote node definition that represents your database. Once you save the component, a mask appears in the field.

See the section "Defining Nodes for Single Signon" later in this section for more information.

Understanding Setting Up PeopleSoft-Only Single Signon

The following table list steps for setting up single signon among PeopleSoft systems. Note that additional configuration may be required based on your business and security requirements.

Step	*Page/Navigation*	*Description*
1. Configure the default local node definition.	Nodes - Node Definitions page. **PeopleTools** > **Portal** > **Portal Node Definitions.** Select the default local node and click the Nodes Definition tab.	• Define the Authentication Option. The valid options for single signon are *Password* or *Certificate.* You must define the same value on the remote PeopleSoft nodes participating in single signon. • Generate and define a check token ID. Click the CheckTokenID button to create a system-generated ID. The system automatically populates the value in the Check TokenID field on the Node Definition page. As an alternative, create a custom ID, or create a custom ID of up to 256 characters. Make a note of the ID before saving the page. You must provide a copy of the ID to your single signon participants, who must in turn define that value on their databases on the Nodes – Node Definitions page for the remote node definition that represents your database.
	Nodes - Portal page. **PeopleTools** > **Portal** > **Portal Node Definitions.** Select the default local node and click the Portal tab.	• Define the PeopleTools release. In the Tools Release field enter the PeopleTools release running on the local database. For example, *8.56.00* • Define the Content URI. In the Content URI Text field enter the uniform resource identifier (URI) of the pscontent servlet (psc) for the local default node. • Define the Portal URI. In the Portal URI Text field enter the URI of the the portal servlet (psp) for the local default node

Step	*Page/Navigation*	*Description*
2. Configure remote PeopleSoft node for each node participating in single signon.	Nodes - Node Definitions page. **PeopleTools** > **Portal** > **Portal Node Definitions.** Select the remote PeopleSoft node and click the Nodes Definition tab.	• Define the Authentication Option. The valid options for single signon are *Password* or *Certificate.* You must define the same value as defined on the local default node of the single signon participant. • Define the check token ID. Enter the check token ID as provided by the single signon participant.
	Nodes – Portal page. **PeopleTools** > **Portal** > **Portal Node Definitions.** Select the remote PeopleSoft node and click the Portal tab.	• Define the PeopleTools release. In the Tools Release field enter the PeopleTools release running on the single signon partner database. For example, *8.56.00* • Define the Content URI. In the Content URI Text field enter the URI of the pscontent servlet (psc) for the single signon participant's default local node. • Define the Portal URI. In the Portal URI Text field enter the URI of the the portal servlet (psp) for the single signon participant's default local node.
3. Add nodes/databases participating in single signon to the Single Signon page.	Single Signon page. **PeopleTools** > **Security** > **Security Objects** > **PeopleSoft Single Signon**	Add nodes participating in single signon to the Trust Authentication Tokens Issued by These Nodes grid. Click the Lookup button to search for and select nodes to participate in single signon. The default local node appears in the grid by default.

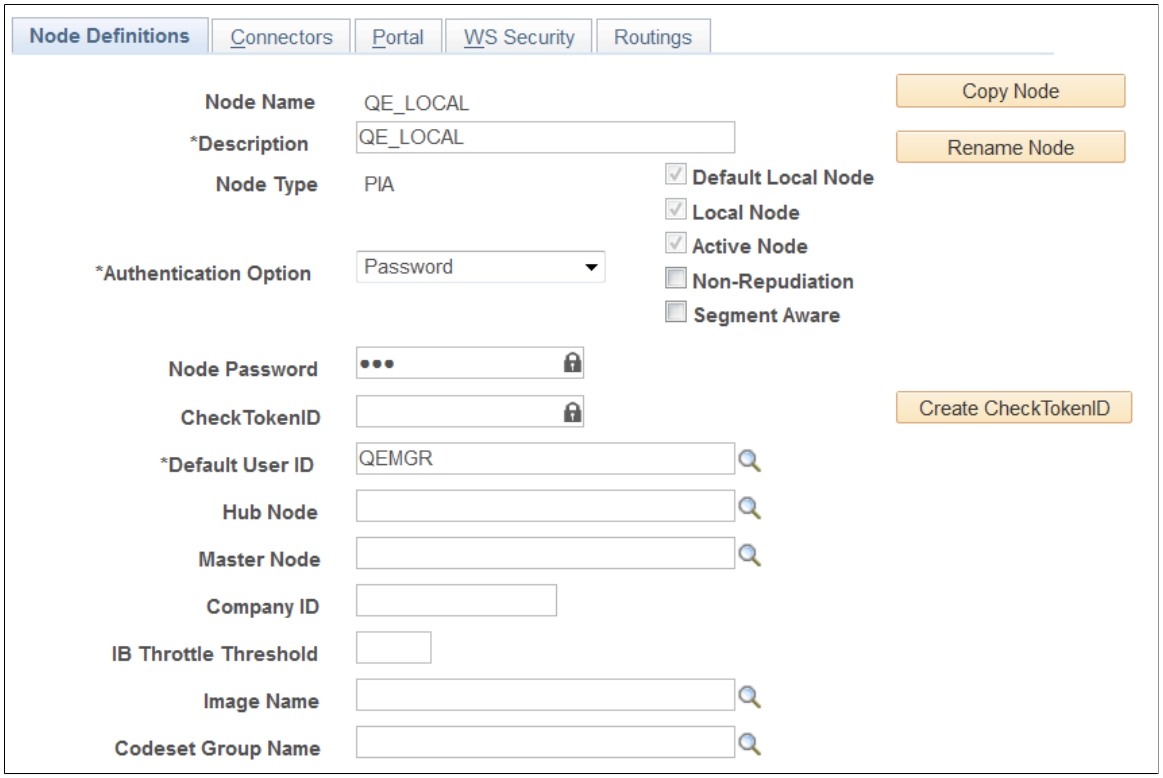
Step	*Page/Navigation*	*Description*
4. Add sites participating in single signon to the Authorized Sites page.	Authorized Sites page. **PeopleTools** > **Web Profile** > **Authorized Sites.**	You can add sites two ways: • Allow Domain Compare. In the CheckToken section of the page, select the Allow Domain Compare box. Selecting this option allows access for all sites within a defined authentication domain, including their sub-domains. For example, an authentication domain of *example.com* will include *myserver1.example.com, myserver2.example.com* and so on. • Allowlist sites to participate in single signon. In the Authorized Sites grid, add a row for each site and select the CheckToken box to enable single signon for the site.

Defining Nodes for PeopleSoft-Only Single Signon

Defining General Node Properties for PeopleSoft-Only Single Signon

Access the Node Definitions page (**PeopleTools** > **Portal** > **Portal Node Definitions**).

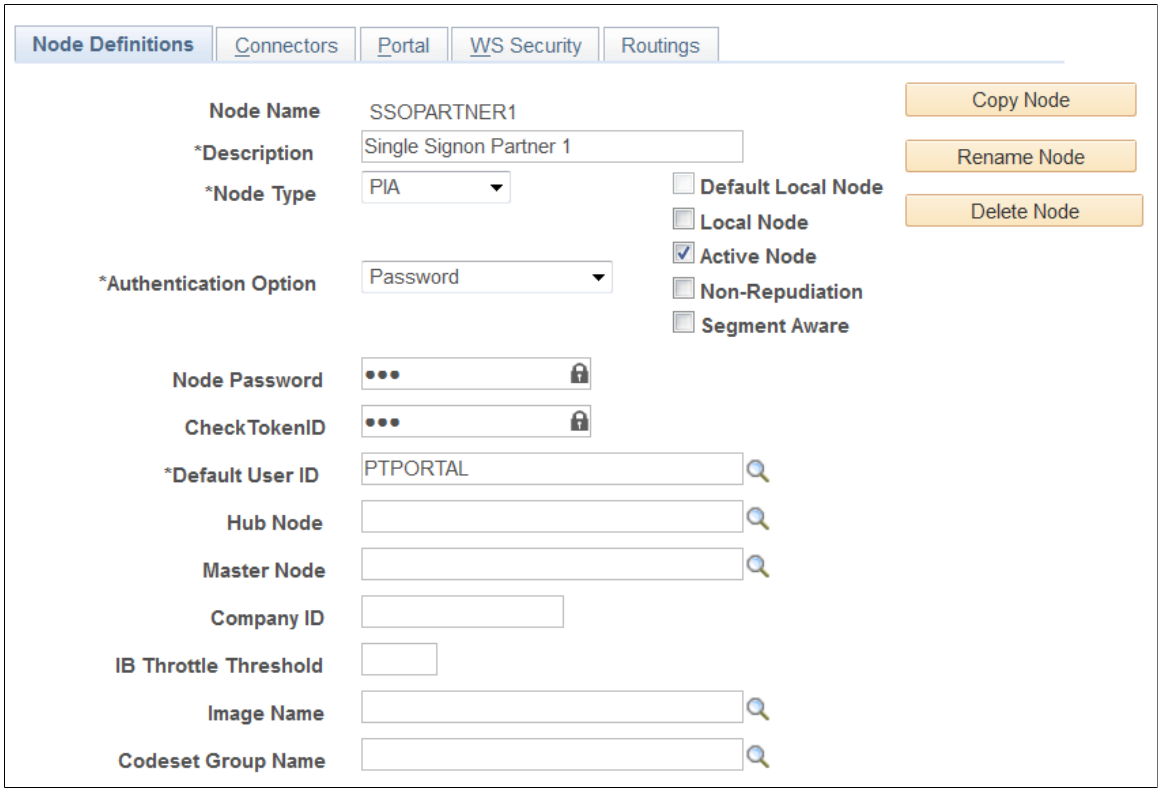
This example illustrates the fields and controls on the Nodes - Node Definitions page for the default local node.



The following example shows the Nodes - Node Definitions page for a remote PeopleSoft node participating in single signon as defined on the local database:

Important! The node name that you define for a remote PeopleSoft node to participate in single signon must be the same name as the default local node name on the participant's local database.

This example illustrates the fields and controls on the Nodes - Node Definitions page a remote PeopleSoft node.



The fields and controls required for single signon are described in the following table. Note that your business and system requirements may warrant additional configuration of this page.

Field or Control	***Description***
Description	Enter a description for the node.
Node Type	Select *PIA* from the drop-down list. The Node Type for local and remote PeopleSoft nodes participating in single signon must be *PIA*.

Field or Control	*Description*

Authentication Option	Determines how nodes in a single signon configuration authenticate other nodes in the same configuration. The valid options for single signon are:
	• *Password:* Indicates that each node in the single signon configuration authenticates other nodes by way of knowing the password for each node. For example, if there are three nodes (A, B, and C), the password for node A needs to be specified in its node definition on nodes A, B, and C.
	• *Certificate:* Indicates that a digital certificate authenticates each node in the single signon configuration. For certificate authentication, you need to have the following in the key store in the database for each node:
	  • Certificate for each node.
	  Note: For SSO, avoid using certificate key size 4096 due to browser limitation.
	  • Root certificate for the CA that issued the certificate.
	  Important! For single signon, the alias for the certificate of a node needs to be the *same* as the node name. Also, you must request and set up your digital certificates before you set the authentication option to certificate authentication.
	While the option *None,* which signifies no authentication between nodes, is included in the drop-down list, it is not a valid option for single signon nodes.

Default Local Node	When defining the default local node on the local database, select the **Default Local Node** option. Indicates that the current node represents the database you're signed in to. The default local node is used specifically for setting up single signon. The options you set for single signon should be made on the default local node.

Field or Control	*Description*
Node Password	• Default local node definition. Enter a password for the node. The value you enter is limited to 88 characters. • Remote PeopleSoft node definitions. Enter the password for the single signon participant's default local node, as provided by the participant. **Note:** You must reboot the application server and the web server after you define or change this value.
Default User ID	• Default local node definition. Enter or select a default user ID to associate with the node. • Remote PeopleSoft node definitions. Enter or select the default user ID defined for the single signon participant's default local node, as provided by the participant.
Create CheckTokenID	The **Create CheckTokenID** button appears only: • On the definition for the default local node. • when the **Authentication Option** type is *Password* or *Certificate.* Click the button to create a system-generated check tokenID for use in conjunction with single signon among PeopleSoft systems. When you click the button, the system populates the **Check TokenID** field with the generated value.

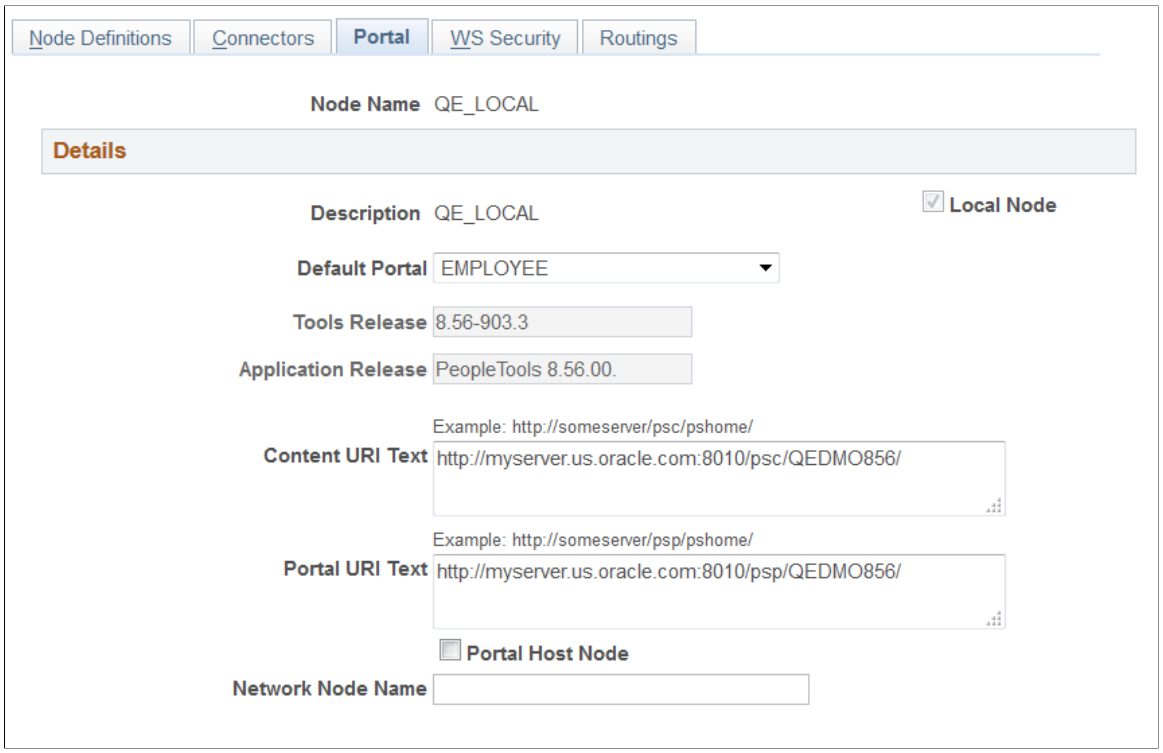
Field or Control	*Description*
Check TokenID	• Default local node definition: This field displays the check token ID value generated after clicking the **Create CheckTokenID** button. This ID is used in conjunction with single signon among PeopleSoft systems. Alternatively, you can create a custom ID up to 256 characters. Copy the value in the field before saving the page. You must provide this value to other PeopleSoft partners/ nodes participating in single-sigon, as they must define this value on their database on the remote node definition that represents your database. • Remote PeopleSoft node definitions. On definitions for remote PeopleSoft nodes participating in single signon, enter the value provided by your single signon partner. After you save the page the field becomes masked, regardless of whether a value is defined in the field or not. **Note:** You must reboot the application server and the web server after you define or change this value.

Defining Portal Node Properties for PeopleSoft-Only Single Signon

Access the Nodes - Portal page (**PeopleTools** > **Portal** > **Portal Node Definitions** and click the Portal tab).

The following example shows the Nodes - Portal page for the local default node.

This example illustrates the fields and controls on the Nodes - Portal page for a default local node.



The fields and controls required for single signon are described in the following table. Note that your business and system requirements may warrant additional configuration of this page.

Note: References to “remote PeopleSoft nodes” in the descriptions refer to remote node definitions that you must define for each PeopleSoft system participating in single signon.

Field or Control	*Description*
Tools Release	• Default local node definition. When defining properties for the default local node, enter the Tools release version installed on the local database. On most browsers, you can press CTRL + J to locate the PeopleTools release installed on the database. • Remote PeopleSoft node definitions. When defining properties for remote PeopleSoft nodes, enter the Tools release version installed on the single signon participant’s database.

Field or Control	*Description*
Content URI Text	• Default local node definition. When defining properties for the default local node, enter the URI of the pscontent servlet for the local database. • Remote PeopleSoft node definitions. When defining properties for remote PeopleSoft nodes, enter the URI of the pscontent servlet (psc) for the single signon participant's default local node.
Portal URI Text	• Default local node definition. When defining properties for the default local node, enter the URI of the portal servlet (psp) for the local database. • Remote PeopleSoft node definitions. When defining properties for remote PeopleSoft nodes, enter the URI of the portal servlet (psp) for the single signon participant's default local node.

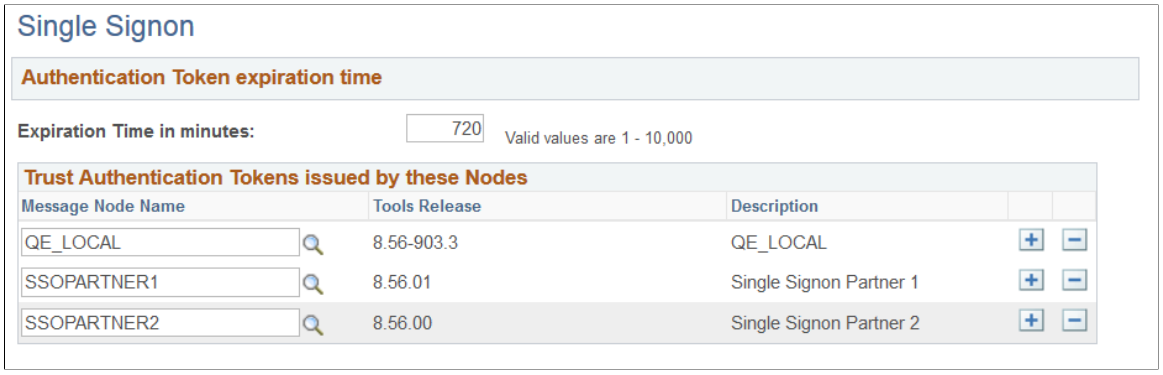
Related Links

"Understanding Nodes" (Integration Broker Administration)
"Implementing Nonrepudiation" (Integration Broker Administration)

Working with the Single Signon Page

Access the Single Signon page (select **PeopleTools** > **Security** > **Security Objects** > **PeopleSoft Single Signon**).

This example illustrates the fields and controls on the Single Signon page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Expiration time in minutes	You need to set an expiration time for tokens this system accepts for authentication. Otherwise, once the user is authenticated, the user could be authenticated and signed on to the system with the token for as long as it stays up and running. You can set the authentication interval to be minutes, hours, or days depending on your signon strategy. The value is in minutes. For example, 480 minutes is 8 hours. This is global setting for all users of your PeopleSoft system that get issued the cookie. A short expiration period is more secure, but less convenient because users need to enter their passwords more frequently. The system accepting the token controls the expiration time, not the issuing system. For example, Node HCM_WEST, which has an expiration time of 100 minutes, issues a token to a user. The user attempts to use that token to sign in to Node FIN_EAST, which has an expiration time set to 60 minutes. If a period greater than 60 minutes has transpired, Node FIN _EAST rejects the token. When a node rejects a single signon token, the system prompts the user to enter a user ID and password on the standard signon screen. **Note:** This expiration time is separate from the timeouts you specify in the Permission Lists and the web server configuration files.
Message Node name	Shows the name of the Message Node. In order to share authentication tokens between nodes, the nodes need to trust each other. By adding a node to this grid, you indicate that a particular node is known to the system and trusted. When a node is trusted, the local node accepts tokens issued by it. By default, no nodes appear in the trusted nodes list. If you want to implement single signon, you need to explicitly configure your system to support it by adding trusted nodes. First, you need to add the local node to the grid as a node must be able to trust its own tokens. When you sign in to the portal, the system authenticates users with a single signon token issued by the local system. The portal won't be able to sign in unless the local node is trusted. Then you add the names of other nodes in the system that should be trusted. **Note:** You define nodes in Portal, Portal Node Definitions.
Tools Release	Displays the PeopleTools release of the node, as defined on the Nodes - Portal page.
Description	Displays the node name description, as defined on the Nodes - Node Definitions page.

Note: After you update the list of trusted nodes, the system automatically recognizes the new list. Rebooting the application server is not required.

Defining Authorized Sites for Single Signon

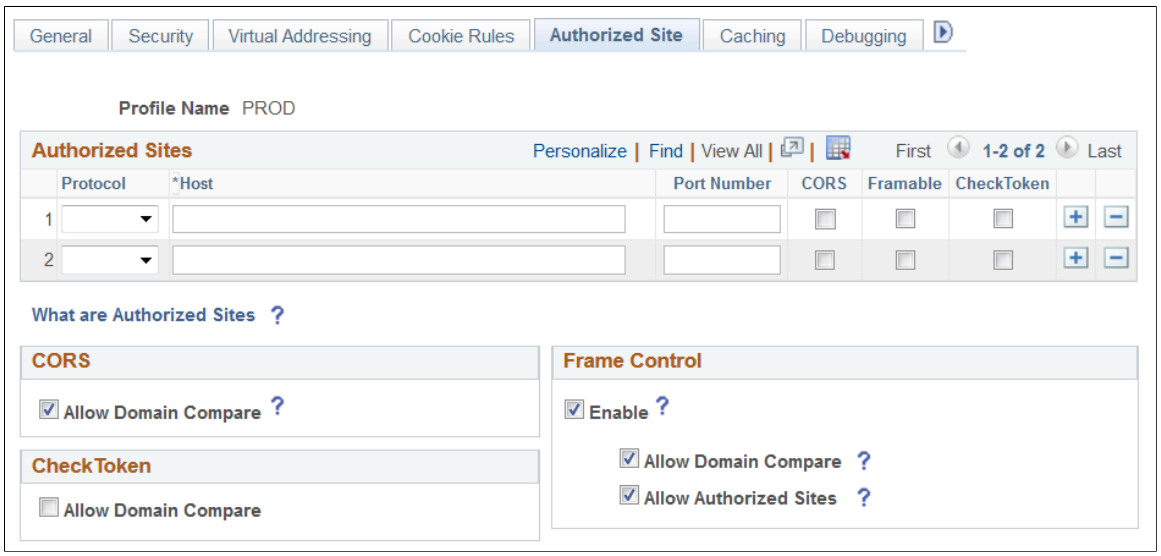
You can authorize the sites that users can access using single signon. The sites must be located on nodes/databases defined on the Single Signon page (PeopleTools, Security, Security Objects, PeopleSoft Single Signon).

There are two ways to authorize sites for single signon:

- Allow access to all sites configured on the domains of nodes defined on the Single Signon page.
- Create an allowlist of sites of the domains defined on the Single Signon page.

Use the Authorized Sites page to define sites authorized for single signon. To access the page select **PeopleTools** > **Web Profile** > **Authorized Sites.**

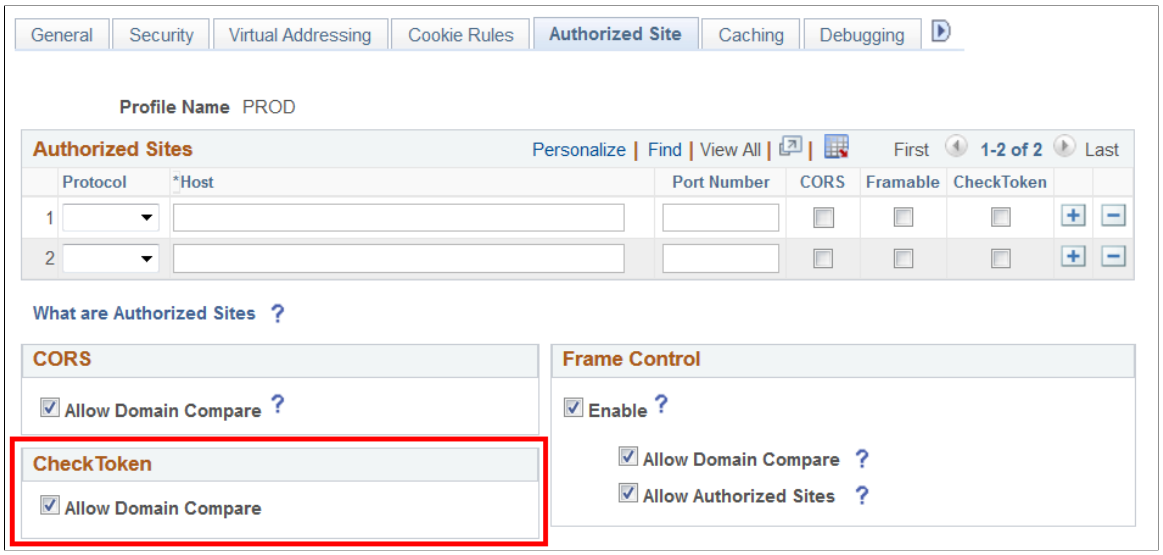
This example illustrates the fields and controls on the Authorized Sites page. The fields and controls related to defining authorized sites for single signon are described later in this section.



Authorizing Single Signon Sites Across All Defined Single Signon Domains

To authorize sites to use single signon across all of the domains listed on the Single Signon page, in the CheckToken section of the Authorized Sites page, select Allow Domain Compare, as shown in the following example:

This example illustrates the Authorized Sites page with the Allow Domain Compare option highlighted.



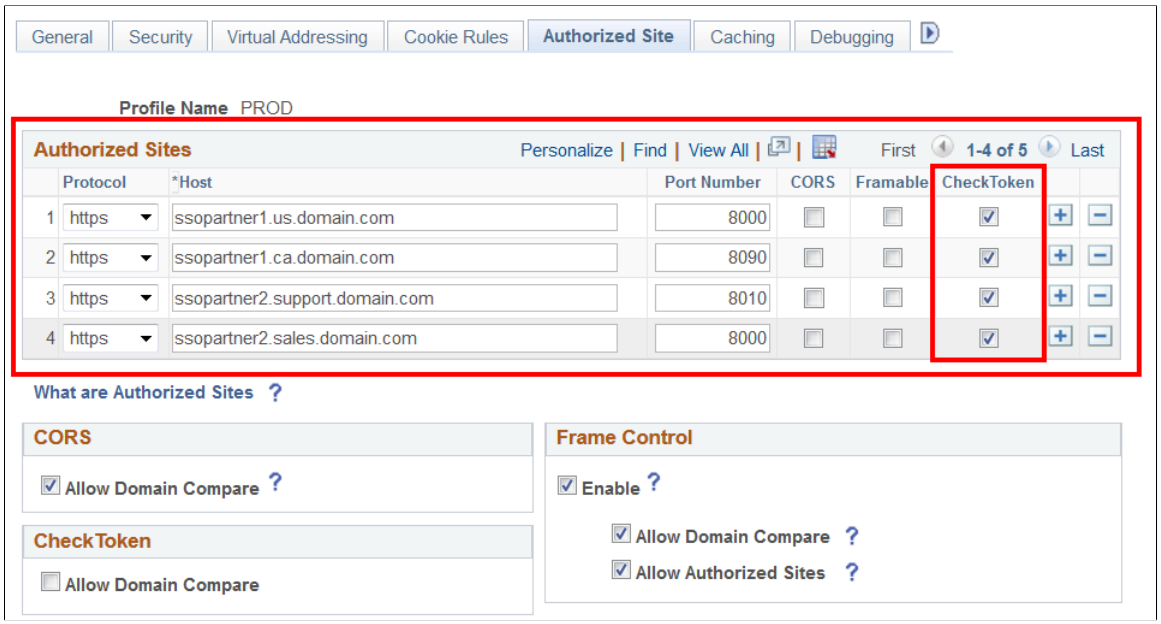
When you select the Allow Domain Compare option, the system allows single signon across all authentication domains and sub-domains of the nodes that you have listed on the Single Signon page.

Note: You must reboot the application server and the web server after you enable or disable this option.

Creating an Allowlist of Sites Authorized for Single Signon

To create an allowlist of sites authorized for single signon, define the sites in the Authorized Sites grid on the Authorized Sites page. The Authorized Sites grid is shown in the following example:

This example illustrates the Authorized Sites page with the Authorized Sites grid highlighted.



To define an allowlist of sites authorized for single signon, in the Authorized Sites grid at the top of the page:

1. From the Protocol drop-down list, select the protocol used on the site. The options are:
   - *http.*
   - *https.*
2. In the **Host** field, enter the domain of the site.
3. In the **Port Number** field, enter the port number of the domain.
4. Select the **Check Token** box.
5. Repeat steps 1 to 5 to define additional sites.
6. Click the **Save** button.
7. Reboot the application server and the web server.

Note: It's important that you remember to select the **Check Token** option in the Authorized Sites grid for each site you want to allowlist for single signon. The Authorized Sites grid can contain site information for other PeopleTools functionality. Selecting the **Check Token** option enables the allowlist functionality for the site to be used during token validation.

Note: You must reboot the application server and the web server after you add or remove a site from the list.

Setting up Certificate Authentication

This section provides additional details and steps to assist the configuration of certificate authentication used in a single signon implementation.

In the following scenario, you are configuring single signon between these two PeopleSoft systems.

Database	***Node Name***	***Local Node***	***Remote Node***
PeopleSoft Portal (master)	PSPORTAL	PSPORTAL	PSHCM
PeopleSoft HCM (content)	PSHCM	PSHCM	PSPORTAL

Perform these steps:

1. Set certificate authentication option in master database.
2. Define the portal node and establish trust in content database.
3. Create the private key and install the digital certificate for the local node in master database.
4. Install the digital certificate for the remote node in the content-side database.

Setting Certificate Authentication Option in Master Database

To set certificate authentication option in master database:

1. Sign in to the Portal database.
2. Select **PeopleTools** > **Portal** > **Portal Node Definitions**.
3. Select PSPORTAL from the list of nodes.
4. Verify that it is the local node.
5. Select *Certificate* from the **Authentication Option** drop-down list box.
6. Save the page.
7. Click the **Return to Search** button.
8. Verify that PSHCM exists as a remote node.

Defining Portal Node and Establishing Trust in Content Database

To define the portal node and establish trust in content database:

1. Sign in to the HCM database.
2. Select **PeopleTools** > **Portal** > **Node Definition.**
3. Click the **Add a New Value** link.
4. Enter *PSPORTAL* and click the **Add** button.
5. Select *Certificate* from the **Authentication Option** drop-down list box.
6. Save the page.
7. Select **PeopleTools** > **Security** > **Security Objects** > **PeopleSoft Single Signon** and add the PSPORTAL message node to the list of trusted nodes in the **Trust Authentication Tokens issued by these Nodes** group box.
8. Save the page.

Creating the Private Key and Installing the Digital Certificate for Local Node

To create the private key and install the digital certificate for the local node:

1. Sign in to the Portal database.
2. Select **PeopleTools** > **Security** > **Security Objects** > **Digital Certificates.**

   Note: Make sure that Root CA with Issuer Alias of *PeopleTools* is available.

3. Click the Add a new row button (+).
4. Select *Local Node* as the **Type.**
5. Enter *PSPORTAL* in the **Alias** field.

6. Select *PeopleTools* as the **Issuer Alias.**
7. Click the **Request** link.
8. Fill in the form

   Note: For UNIX application servers, use 512 as the Key Size and PSPORTAL as the common name.

9. Click the **OK** button.
10. Select all of the text, copy the request, and click the OK button.
11. Request a certificate from your certificate provider.
12. Request the certificate using a base-64-encoded CMC or PKCS #10 file, or submit a renewal request by using a base-64-encoded PKCS #7 file.
13. When you receive the certificate, download and save it to C:\temp as *newcert.cer.*
14. Open the certificate with a text editor.
15. Select all of the text and copy the certificate.
16. Sign in to the Portal database.
17. Select **PeopleTools** > **Security** > **Security Objects** > **Digital Certificates.**
18. Click the **Import** link for the PSPORTAL alias.
19. Paste the certificate into the text box.

    Note: Make sure that there is no space after END CERTIFICATE, otherwise, you are not allowed to save.

20. Click the OK button.

Installing Digital Certificate for the Remote Node in the Content-Side Database.

To install the digital certificate for the remote node in the content-side database:

1. Sign in to the HCM database.
2. Navigate to **PeopleTools** > **Security** > **Security Objects** > **Digital Certificates.**
3. Click the **Add a new row** button (+).
4. Select *Remote Node* as the **Type.**
5. Enter *PSPORTAL* in the **Alias** field.
6. Select *PeopleTools* as the **Issuer Alias.**
7. Click the **Import** link.
8. Open the certificate that you downloaded to C:\temp\newcert.cer with a text editor.
9. Copy the text and paste the digital certificate into the empty edit box.

10. Click the **OK** button.

Single Signon Transaction Example

Now that you have a general understanding of why a single signon implementation is useful, and some of the details involved with PeopleSoft-only single signon, this section presents an example of how the PeopleSoft-only single signon scheme works.

In this scenario there are two databases, or nodes: an HCM database and Financials database. Recall that the terms database and node are synonymous. Each database has one application server and one web server. The following steps describe the “back-end” events that occur when a user signs in to the HCM database, completes a transaction, and then clicks a link that targets a page in the Financials database.

Step 1: User Signs In to an HCM Application

The following occurs:

1. The user PTDMO clicks this link: http://hcm.myserver.com/psp/hcmprod/?cmd=login&languageCd=ENG
2. The user enters ID and Password at the sign in page and clicks the **Sign In** button.

Step 2: Application Server Authenticates User

The following occurs:

1. The web server relays sign in request to the HCM application server.
2. The HCM application server authenticates the user.

Step 3: Application Server Generates Single Signon Token

The following occurs:

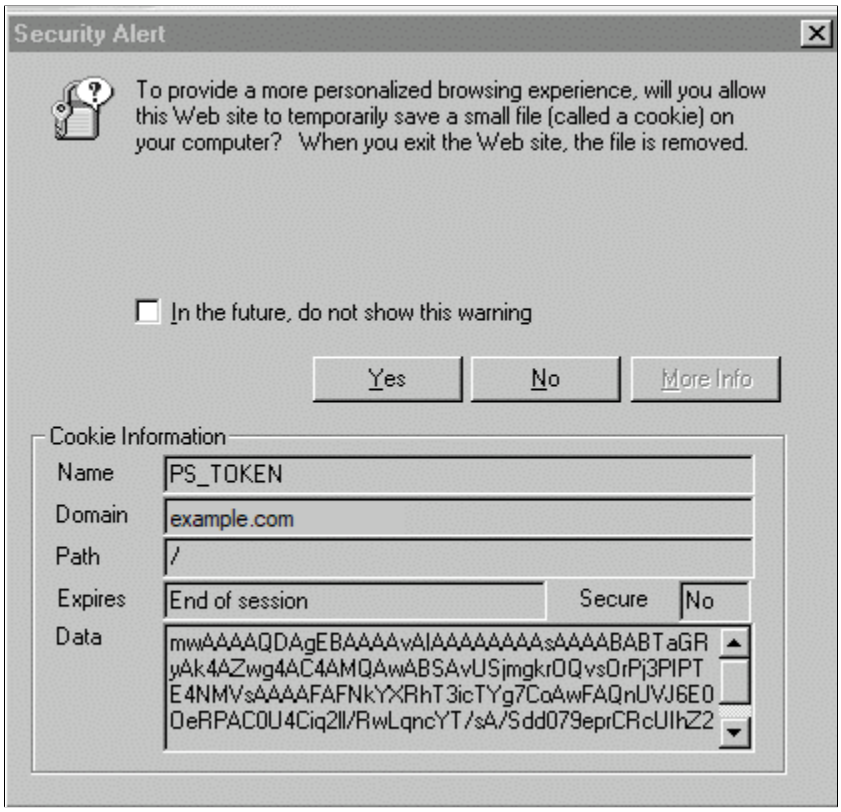
1. If the user is authenticated by the application server, then it generates a single signon token.
2. The application server encrypts and encodes the token (base 64).
3. The application server sends the token to the web server, along with a return code indicating that the system authenticated the user.

Step 4: Web Server Creates Cookie in User's Browser

When the web server receives the single signon token from the application server, it creates a cookie and inserts the cookie in the user's browser.

If the browser is configured to show the Security Alert dialog, then the user sees a message similar to the following example. In most cases, you don't configure browsers to show this dialog; this dialog box is just an example of the data that the browser receives.

This example illustrates the Security Alert dialog box.



The cookie that the web server distributes for PeopleSoft single signon is named PS_TOKEN. In this case the domain mydomain.example.com set the cookie.

Notice that the cookie expires at the end of session. This indicates that the system never writes the cookie to disk, the cookie exists in browser memory for the duration of the session only.

The web server inserts the single signon token within the Data field of the cookie. So that the system can send the binary data across the HTTP protocol, the token data is encrypted and base 64 encoded.

Step 5: User Needs to Access Financial Application

After the user completes a few transactions in the HCM system, suppose they arrive at a page containing a link to the Financial system. The user clicks the link, and because they've already entered their credentials for the HCM system they don't need to sign in again.

The browser sends the PS_TOKEN cookie to the Financials web server.

Step 6: Financials Web Server Receives PS_TOKEN Cookie

The Financials web server *does* detect that the user hasn't been authenticated by the Financials system yet. However, because the web server received the signon cookie it does not display the sign in page.

To retrieve the page the user requested (by way of the link in the HCM application), the Financials web server attempts to connect to the Financials application server. It passes only the Data field from the PS_TOKEN cookie because the application server needs only the information in the Data portion.

Step 7: Financials Application Server Authenticates PS_TOKEN

Before allowing the user to connect, the Financials application server evaluates the PS_TOKEN Data field in the following flow:

1. **Is the forwarding node trusted?**

   The application server checks to see that the message node name listed as the Issuing System is a trusted node. The list of trusted nodes for the Financials system resides in the PSTRUSTNODES table. You configure the list using PeopleTools, Security, Security Objects, PeopleSoft Single Signon. The Single Signon page enables the administrator of the Financials system to "trust" authentication tokens generated from HCM as well as any other nodes deemed trusted.

2. **Has the token expired?**

   The application server checks that the authentication token hasn't expired. Using the Issued Date and Time field within the token, the Financials application server makes sure that the token was issued within the interval between the timeout minutes value and the current time. You configure a token's expiration time on the Single Signon page.

   Note: It is important to note that the expiration parameter specified in the Financials system is the relevant value, not the expiration value specified in HCM. This enables the Financials administrator to control the maximum age of an acceptable token. It's also important to consider that all times are in Greenwich Mean Time (GMT), so it doesn't matter what time zones the systems are in.

3. **Has the signature been tampered with?**

   The application server checks that the signature is valid. The Financials application server takes all the fields in the token and the Node password for the issuing node and generates a hash. The token is valid only if the signature within the token *exactly* matches the one generated by the Financials application server. Because an exact match is the only acceptable situation, Financials can be sure that HCM generated the token, and that it hasn't been tampered with since it was generated. If a hacker intercepted the token in transit and changed the User ID, Language, and so on, the signatures wouldn't match and as a result the Financials application server would reject the token.

   Note: You should use digital certificate authentication when implementing single signon.

PeopleSoft-Only Single Signon Configuration Considerations

The following topics describe some items you might want to consider as you implement your single signon configuration.

Single Authentication Domain Limitation

Web servers must be assigned to the same authentication domain—the server name in the URLs used to access them must contain the same domain name. A browser sends a cookie back only to the same domain from which it received the cookie.

In PeopleSoft applications, an authentication domain is not the same thing as an internet protocol (IP) address. An authentication domain is a logical URL address that you specify during Pure Internet Architecture setup, and its purpose is to associate different web servers (even at different physical locations) so that they appear to be at the same location to the PeopleSoft applications that use those web servers.

Important! Specifying authentication domains incorrectly for multiple Pure Internet Architecture installations can produce single signon errors.
If you want to keep two PeopleSoft applications from erroneously attempting to employ single signon, make sure that the authentication domain you specify for one application's web server is not a subset of the authentication domain you specify for the other. For example, if your CRM web server has an authentication domain of *.crm.mycompany.com*, your Financials web server authentication domain must not be *.mycompany.com* (the parent of the CRM server domain) or *.fin.crm.mycompany.com* (a child of the CRM server domain). It can, however, be *.fin.mycompany.com* (or any child of the *mycompany.com* domain).
If you *do* want two PeopleSoft applications to employ single signon, you must ensure that each application contains a definition of the other as a trusted node, and you must specify the same authentication domain for both applications' web servers during Pure Internet Architecture setup.

Furthermore, the web server that generates the cookie must have the domain that shares the PS_TOKEN cookie specified in the web profile of the local Pure Internet Architecture web site. For example, in the context of our HCM to Financials example, the web profile for the HCM web server must contain the value of *.example.com* in the **Authentication Domain** property.

Note: You must specify the leading dot (.).

The single domain issues occur in the following situations:

- You're using straight Pure Internet Architecture, as in you are deploying applications but not by way of the portal.
- You're using the portal with frame-based templates. All PeopleSoft portal solutions products (Enterprise, Employee, Customer, Supplier portals) are built using frame-based templates.

Frame-based templates aren't proxied automatically. Proxying refers to when the system rewrites the URL to point to a location on the portal servlet, rather than the original location of the URL.

Single Signon Between Machines without DNS Entries

If you're setting up single signon between machines that don't have DNS entries, you need to modify the hosts file on the machine that's running the web browser. For example, let’s say that you are using machine a.example.com to signon to the web server a.example.com, and then access b.example.com using single signon. In this situation, you would need to update the hosts file on a.example.com as follows.

```
# Copyright (c) 1993-1999 Microsoft Corp.
#
# This is a sample HOSTS file used by Microsoft TCP/IP for Windows.
#
# This file contains the mappings of IP addresses to host names. Each
# entry should be kept on an individual line. The IP address should
# be placed in the first column followed by the corresponding host name.
# The IP address and the host name should be separated by at least one
# space.
#
# Additionally, comments (such as these) may be inserted on individual
# lines or following the machine name denoted by a '#' symbol.
#
# For example:
#
#      192.0.2.1     myserver.example.com          # source server
#       203.0.113.1     myclient.example.com              # x client host

192.0.2.8       localhost
203.0.113.4  a.example.com
```



```
198.51.100.5  b.example.com
```

Domain Names

You need to use a fully qualified domain name when addressing the web server in your browser.

- This is an example of a *correctly* formatted URL: *http://hcm.example.com/myapplication/signon.html*
- This is an example of a *incorrectly* formatted URL: *http://hcm/myapplication/signon.html*

When using the portal, the domain name that you specify in the **Portal URI Text** edit box on the Content Provider administration pages must match the fully qualified domain name you enter as the authentication domain. For example, you must specify myserver.example.com/servlets, not myserver/servlets.

Cross Domain Single Signon

The current PeopleSoft single signon solution deals mainly with systems where there is only one DNS domain. Many sites need to deploy the PeopleSoft Portal in multi-domain environments. For example, you might want to have the portal in one domain such as, *www.myserver.com,* and the HCM database in another domain, such as *www.yourcompany.com*.

You can configure your environment to support cross-domain single signon by completing the following configuration tasks.

- Setup a third-party web security product that supports multi-domain single signon and supports LDAP user profiles.

  There are several industry-standard products on the market.

- Configure the portal and content provider web servers to trust the web server for authentication.

  For PeopleSoft applications , this involves creating and enabling the public access user.

- Set up the PeopleSoft applications to download the user profiles from the same LDAP server that the web security product uses.

  This means that the DN that comes from the subject field of the certificate has to be a valid DN for the directory that the LDAP_profilesynch function references. Because of this you need to build a user profile cache map that points to the same directory that generated the subject's DN.

Note: This cross-domain limitation does not apply to the portal if the content from the provider in a different domain is wrapped in an HTML template. However, this limitation does apply for any content in the portal that is wrapped in a frame template. Because the Enterprise, Customer, Supplier, and Employee portals that ship with PeopleTools all include frame templates as defaults, you'll need to perform the extra configuration steps to support cross-domain single signon in multi-domain environments. This limitation also applies to Pure Internet Architecture-to-Pure Internet Architecture (iClient-to-iClient) single signon.

PeopleSoft-Only Single Signon Configuration Examples

The following topics describe examples of single signon configurations and the steps required to implement them.

One Database and Two Web Servers

In this scenario there is one database and two or more web servers. While single signon is configured at the database level (that is, you specify timeout minutes and trusted nodes for the entire database), it's actually used any time two different PeopleSoft servlets connect to the same database.

To set up single signon with one database and multiple web servers:

1. Select **PeopleTools** > **Portal** > **Portal Node Definitions** and make sure that at least one node is defined as the Default Local Node.

   In the results on the search page, you can determine this by looking for a *Y* in the **Default Local Node** column.

2. Select **PeopleTools** > **Security** > **Security Objects** > **PeopleSoft Single Signon** and set the following:

   - Make sure the Default Local Node appears in the list under **Trust Authentication Tokens issued by these Nodes.**
   - Set the timeout minutes to an appropriate value (the default is 720).

3. Access the web profile for each web server and modify the Authentication Domain property.

   Because single signon is implemented using browser cookies, it must be configured so that the user's browser sends the single signon cookie to each web server machine involved. By default, the browser only sends cookies back to the machine that set the cookie. So if web server a.example.com sets a cookie after the user is authenticated, the browser (by default) only sends the cookie to a.example.com. By default, the browser would not send the cookie to b.example.com. To make the browser send the single signon cookie to all servers at in a domain (example.com), access the Web Profile Configuration - General page and set a value of *.example.com* for the Authentication Domain property.

   Note: You need the leading period (.) before the domain. It should appear as ".example.com," not "example.com."
   If you use only one web server, you *don't* need to modify the Authentication Domain property. A web server is designed to accept the cookies it distributes.

Two Databases and Two Web Servers

To set up single signon with multiple databases and multiple web servers:

1. Select **PeopleTools** > **Portal** > **Portal Node Definitions**.

   For *each* node that you want to involve in the single signon configuration and check the following:

   - Make sure that at least one node definition is defined as the Default Local Node for each database.

     In the results on the search page, you can determine this by looking for a *Y* in the Default Local Node column.

   - Make sure that each database contains a node definition for the other nodes in the single signon configuration.

- Make sure that the **Authentication Option** is set correctly.

  For example, if you are using password authentication make sure that the node password for node ‘X’ is the same in each node definition for node ‘X’ in each database.

  If you use digital certificate authentication, make sure the certificates are properly installed in the PeopleSoft Keystore before setting the node’s Authentication Option to Certificate.

2. Select **PeopleTools** > **Security** > **Security Objects** > **PeopleSoft Single Signon** and set the following:

   - Make sure the Default Local Node appears in the list under **Trust Authentication Tokens issued by these Nodes.**
   - Set the timeout minutes to an appropriate value (the default is 720).

3. Access the web profile on your web server and modify the **Authentication Domain** property.

   Because single signon is implemented using browser cookies, it must be configured so that the user's browser sends the single signon cookie to each web server machine involved. By default, the browser only sends cookies back to the machine that set the cookie. So if web server a.example.com sets a cookie after the user is authenticated, the browser (by default) only sends the cookie to a.example.com. By default, the browser would not send the cookie to b.example.com. To make the browser send the single signon cookie to all servers at in a domain (example.com), modify the authentication domain as follows.

See Understanding SSL/TLS and Digital Certificates, Understanding the PeopleSoft LDAP Solution.

Single Signon with Third Party Authentication

This section presents a simple example of how to implement single signon when you have implemented a third-party authentication system at the web server level. This applies to both portal and intranet web servers.

This discussion assumes that you have enabled public user access in the web profile for the appropriate site.

See Creating a Public Access User.

Note: While this example does not cover authentication, it assumes that you have set up your third-party authentication correctly. Third-party authentication is out of the scope for PeopleSoft support and documentation.

For PeopleSoft application single signon, the PeopleSoft system needs to know the user ID to be used for the web session. If implementing this configuration, you are required to address the following steps:

1. Authenticate the web user.
2. Determine which PeopleSoft user ID to use for this web user.
3. Send the user ID to the PeopleSoft application server.
4. Write Signon PeopleCode to retrieve the user ID from the location, as indicated in step 3.

5. Reauthenticate the user ID during Signon PeopleCode.

6. Indicate to the PeopleSoft application server to use the user ID for all subsequent service requests.

The following examples address steps 3, 4, and 6.

The following HTML applies to step 3 above. You can change the JavaScript function to set the cookie name and value that you want. Also, change the location to point to the PeopleSoft page to which you want to redirect users, for example:

```
<html>
<head>
<title>PeopleSoft 8 Single Signon Example</title>
</head>

<!--
PeopleSoft 8 Single Signon Example

In this example, security is non-existent.  In a production
system, the UserId could come from your site's single signon
tool.  Other information could also be included.  For this
example, only the UserId is saved into cookie.  This cookie then
gets sent to the PIA Web Servlet which passes it on to the
PeopleSoft Application Server.  A piece of Signon PeopleCode is
needed to extract the UserId from the cookie and call
SetAuthorizationResult to "sign on" the user.

- Change the domain value of the cookie to your domain.
- Change the location ref to the target URL within your PeopleSoft site.
//-->

<body>
<script language=JavaScript>
var cookie = "ThirdPartyUserId=PS; Domain=.example.com; path=/; MaxAge=1";
document.cookie = cookie;
location="https://hcm.mycompany.com/psp/hcmprod/EMPLOYEE/HCM/c/ROLE_EMPLOYEE.TIME_O⇒

FF.GBL?FolderPath=PORTAL_ROOT_OBJECT.EE_SELF_SERVE.EE_TIMEOFF_GBL&IsFolder=false&Ig⇒

noreParamTempl=FolderPath%2cIsFolder"</script>
</body>

</html>
```

The following Signon PeopleCode example applies to steps 4 and 6 above. The Signon PeopleCode needs to retrieve &UserID from where the third-party portal put it in the HTTP Request. For example,

```
Function SSO_EXAMPLE()

/*This is step 4*/
   &TPUserId = %Request.GetCookieValue("ThirdPartyUserId");
   /*This is step 6*/
   If &TPUserId <> "" Then
      SetAuthenticationResult( True, &TPUserId, "", False);
   End-If
End-Function;
```

After you write the program, you need to enable the program using the Signon PeopleCode page (**PeopleTools** > **Security** > **Security Objects** > **Signon PeopleCode**).

Securing the PeopleSoft-Only Single Signon Token

PeopleSoft single signon functionality also applies at the web server level. For example, let's say that you have two web servers: server X and server Y. Assume that web server X is an SSL/TLS site, and assume

that web server Y is not. In these situations, many organizations want server Y to trust the authentication token, PS_TOKEN, issued by server X. This requires that the PS_TOKEN be set to be secure.

If the PS_TOKEN is not marked as secure, then when a user signs in through server Y, the browser sends PS_TOKEN to server Y over the unencrypted, non-SSL/TLS link. This is typical behavior for browsers when dealing with non-secure cookies. Potentially, in this situation a hacker could identify this token from the clear network and use it to signon to the SSL/TLS-secure server X.

Another important use of this feature relates specifically to the PeopleSoft Interaction Hub. When the portal proxies content with an HTML template, it should forward PS_TOKEN cookies that are marked secure only over SSL/TLS connections.

To resolve this potential security issue, select the **Secure Cookie with SSL** check box on the Web Profile Configuration - Security page. You use this property to control the secure attribute of the single signon cookie. If you enable the property, and the scheme of the current request is HTTPS (an SSL/TLS server), the system sets the secure attribute of the single signon cookie (PS_TOKEN) to true. This prevents the single signon token from travelling over an insecure network.

Note: If you enable this property, you are effectively disabling single signon to any non-SSL/TLS servers.

If, at your site, you want users to sign in to an HTTPS server, and then want to do single signon with HTTP servers, set this property to false, which allows single signon between HTTPS and HTTP servers.

Note: If you can tolerate the security risk, and want single signon between secure and non-secure links, you can set this flag to false. However, before doing this make sure you are aware of all the security implications, such as the security of the HTTPS server may be compromised.

Using the Single Signon API

PeopleSoft provides a component interface named PRTL_SS_CI that enables external applications to seamlessly integrate a single signon solution with the PeopleSoft portal applications. This ensures that users who have already signed in to the portal don't have to sign in again for every system you reference in your portal.

To take advantage of the Single Signon API, you need to create a custom API, which includes building the dynamic link libraries, classes, and registry settings necessary to enable an external application to communicate with PeopleSoft software.

Note: Due to constraints imposed by the PeopleCode **SwitchUser** built-in function, PRTL_SS_CI does not work properly when called from PeopleCode. Only external applications, such as Java, Visual Basic, and C/C++ programs, can access PRTL_SS_CI.

The files of your custom API need to reside on the client machine; that is, the web server for ASP, and the machine running the Java program for Java. The registry file may also need to be used to update the registry with the new libraries.

Understanding the Signon Process with the API

The PRTL_SS_CI Component Interface contains two user-defined methods:

- **Authenticate**

Your external authentication program distributes an authentication token that can be retrieved from a cookie in the browser. The Authenticate function determines if an authentication token is valid.

- **GetUserID**

  If the token is valid, you use the GetUserID function to retrieve the User ID associated with the authentication token.

Before we describe the development requirements of your API, PeopleSoft recommends that you take a moment to examine the steps that occur internally when you use the API in conjunction with the delivered PRTL_SS_CI.

Step	***Description***
1	The user enters the User ID and password into the PeopleSoft portal sign in page.
2	If the login on portal application server is successful, the server generates a single signon token. The web server receives the single signon token from the application server, and issues a cookie to the browser.
3	The user navigates in the portal and encounters a link to the external system. The user clicks the link.
4	The browser passes the PS_TOKEN cookie to your external web server.
5	The external web server checks for the PS_TOKEN cookie before displaying a sign in page.
6	Once it is determined that the user is accessing your application through the PeopleSoft portal, you retrieve the authentication token and send it to the PRTL_SS_CI component interface to verify authentication.
7	After the system authenticates the token, the system can then make calls to the PRTL_SS_CI.Get_UserID function to return the appropriate User ID.

Developing your External Application to Support Single Signon

Developers of the external applications need to alter the signon process to conform to the following requirements.

1. Check for the PS_TOKEN cookie.

   If the cookie doesn't exist, continue with your normal signon process. Otherwise, bypass the sign in page.

2. Retrieve the authentication token from the PS_TOKEN cookie.

3. Make a connection to the PeopleSoft system through the PRTL_SS_CI API.

4. Pass the authentication token to the `Authenticate()` function of the API.

5. If the function returns True, you then the `Get_UserID()` function retrieves the user ID associated with the authentication token.

Note: The component interface is not mapped to data because the key field for the data would be the authentication token. This token is dynamically assigned when the user signs in to the portal, and it is not stored anywhere in the system as data. Therefore, there are no key fields and the token is passed directly to the user defined functions.

Configuring PeopleSoft-Only Single Signoff

In addition to single signon, the PeopleSoft system also signs the user off of content providers when the user signs off. However, there are some exceptions to the sign-off functionality.

The portal only signs out content providers that meet the following criteria:

- Content providers are accessed only through HTML templates.
- Content providers are all PeopleSoft 8.x or higher applications.

This means that for content providers accessed through frame templates, single sign off is not automatically enabled when you configure single signon. This section describes the steps you need to complete to configure single sign-off for content providers being accessed through frame templates, which includes all of the PeopleSoft Portal solutions (Employee, Customer, and so on).

The following procedure covers inserting an HTML image tag <img> containing a logout command into a set of files on the web server. When the user signs off, the browser attempts to download the images using an "HTTP get," which causes the system to send the logout command to each specified content provider.

This procedure is not appropriate for content that is *never* accessed using a frame, as in it is accessed from the content source using an iScript and a business interlink, such as Lotus Notes integration.

To configure single sign-off for frame content:

1. On your web server, locate and open signin.html.

2. Open signon.html, select Save As, and enter the name *signout.html.*

3. Open signout.html, expire.html, and exception.html.

4. Add the following image tags to these files.

   You need to add one image tag to each of these files for each content provider that requires single signoff.

   Add the tags just before the closing body tag, as shown:

   ```
   <! add tags here>
   </body>
   ```

   If you have three content providers that require single signoff, such as HCM, FIN, and HTML Access, you need to add three image tags to each file.

For example:

```
<IMG src="http://hcm.myserver.com/servlets/psp/ps/hrdb/?cmd=logout"
height=0 width=0 border=0>
<IMG src="http://fin.myserver.com/servlets/psp/ps/hrdb/?cmd=logout"
height=0 width=0 border=0>
<IMG src="http://htmlaccess.example.com/html_access/system/init_asp/
logout.asp?cmd=dummy"
height=0 width=0 border=0>
```

The previous code is an example. To determine the exact URL you need to add for your implementation, right-click the logout link of each content provider. You can usually view the logout link when accessing the application outside of the portal. Examine the properties of this link, and add the specified URL to the image tag.

Note: The string "cmd=dummy" is required in the image tag for HTML Access to make sure that the browser doesn't attempt to cache the image, which would prevent it from issuing the logout command.

---

5. Select **PeopleTools** > **Web Profile** > **Web Profile Configuration** > **Look and Feel** on your web server.

   In the **Signon/Logout Pages** group box, change the value of the **Logout Page** field to *signout.html*.

Implementing Oracle Access Manager as the PeopleSoft Single Signon Solution

PeopleSoft applications support Oracle Access Manager as the single signon solution.

To implement Oracle Access Management Access Manager (Oracle Access Manager) as the PeopleSoft single signon solution:

1. Install and configure Oracle Access Manager.

   See the Oracle Access Manager installation documentation.

   See https://docs.oracle.com/en/middleware/idm/access-manager/12.2.1.4/index.html

2. In the PeopleSoft application, create *OAMPSFT* as a new user profile and associate a low security role such as PeopleSoft User.

   See Creating a New User Profile.

3. In the user profile, access the ID page and select *NONE* as the ID type.

   See Defining User Profile Types.

4. Save the user profile.

5. Access the web profile and enter *OAMPSFT*as the public access user ID.

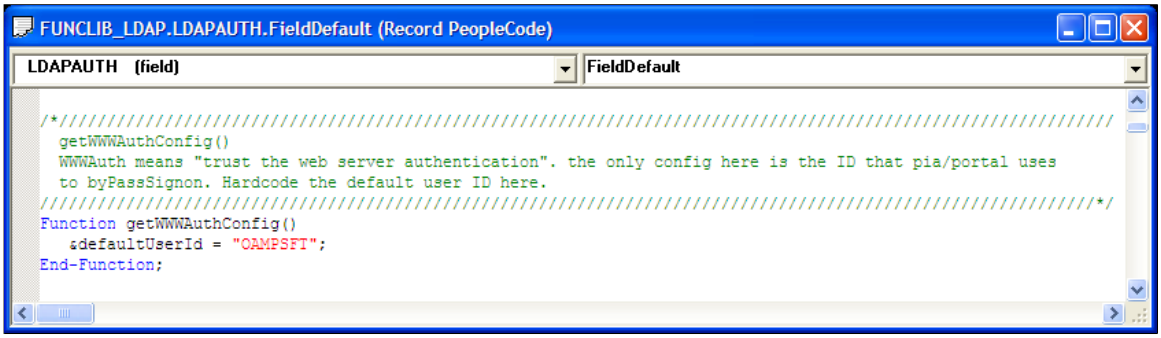
   See “Configuring Web Profiles” (Portal Technology).

6. Using PeopleSoft Application Designer, open the FUNCLIB_LDAP record.

7. Right-click the LDAPAUTH field and select **View PeopleCode.**

8. Find the `getWWWAuthConfig()` function and replace the value that is assigned to the `&defaultUserId` with *OAMPSFT*.

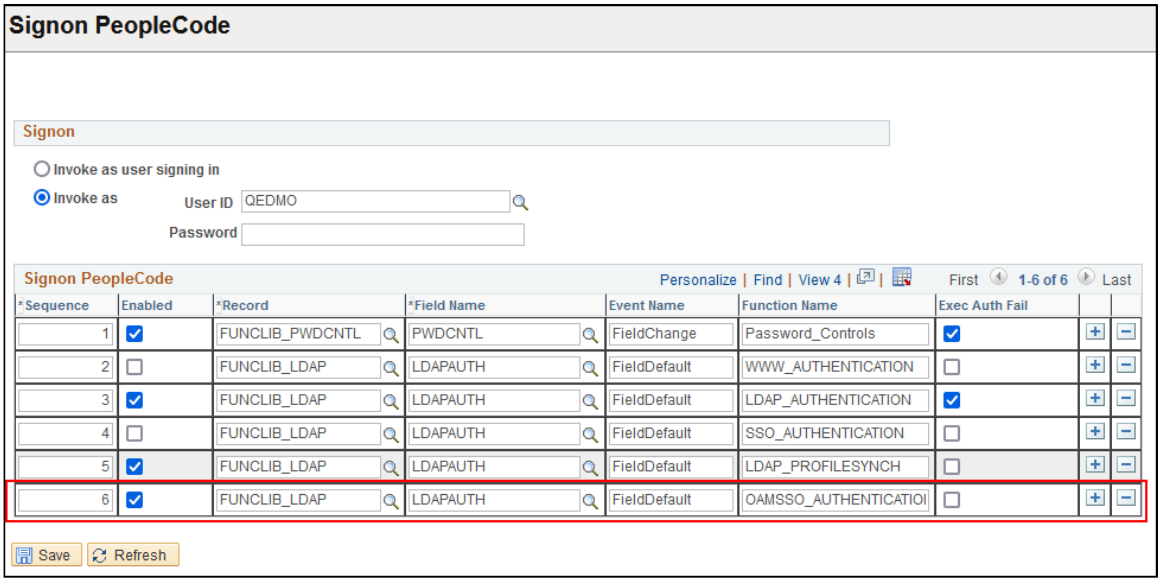
   This example illustrates the getWWWAuthConfig() Function showing modified user ID.



9. Save the record definition.

10. Access the Signon PeopleCode page (PeopleTools, Security, Security Objects, Signon PeopleCode) and enable the OAMSSO_AUTHENTICATION function—the Signon PeopleCode for Oracle Access Manager single signon.

    This example illustrates the Signon PeopleCode page with the OAMSSO_AUTHENTICATION function enabled.



11. Save the page.

12. WebLogic users must disable basic authentication.

    Access <*PS_CFG_HOME*>\webserv\<*domain_name*>\config and modify the config.xml file by adding this tag: `<enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-credentials>`

For Example:

```
<security-configuration xmlns:xacml="http://www.bea.com/ns/weblogic/90/securit⇒

y/xacml">
    <name>peoplesoft</name>
    <realm>myrealm</realm>

.......

    <credential-encrypted>{3DES}dOa1fqoTbXxxxxxxxxxxxxxxx</credential-encrypt⇒

ed>
    <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-cred⇒

entials>

  </security-configuration>
```

13. Be sure the logout page and expire page are configured correctly to work with the Oracle Access Manager logout mechanism.

    See “Configuring Web Profiles” (Portal Technology).

    See the Oracle Access Manager documentation.

14. Stop and restart the application server, web server, and HTTP server.

Chapter 11

Using Web Services for Object and Row-Level Data Authorization

Understanding Using Web Services for Object and Row-Level Data Authorization

PeopleSoft provides a security authorization service that you can use to authorize access to certain PeopleSoft objects and row-level data on local and remote PeopleSoft nodes.

Note: The terms *security authorization service* and *authorization service* are used interchangeably in this documentation.

Object Authorization

You can use the security authorization service to authorize basic security access to content references, components and pagelets. You can also use the service to get the authorization for users to run PeopleSoft queries and iScripts.

Row-Level Data Authorization

The security authorization service enables you to authorize row-level data access to data on local and remote PeopleSoft nodes.

For example, in the related content framework, you can create related services out of components residing on a remote node and assign them as related actions to a component on the local node. You can use the security authorization service to determine if a user can access the services using the related actions on the local node.

Basic security to a content reference or component must be cleared before the system tests for and authorizes row-level data access.

Understanding Developing and Invoking the Security Authorization Service

This section provides an overview of developing and invoking the security authorization service.

Developing and Invoking the Security Authorization Service for Object Authorization

This section provides the high-level steps for developing and invoking authorization services to authorize user access to content references, components, pagelets, PeopleSoft queries, and iScripts.

Object Authorization on Local Nodes

For basic data authorization on a local node:

- Develop a SOAP request message.
- Invoke the service by performing a direct application class method invocation with the request message

Object Authorization on Remote Nodes

For object authorization on a remote node:

- Develop a SOAP request message.
- Invoke the service by sending a SyncRequest to the remote node.

Developing and Invoking the Security Authorization Service for Row-Level Data Authorization

This section provides the high-level steps for developing and invoking authorization services to authorize row-level data access to components and content references.

Row-Level Data Authorization on Local Nodes

For row-level data authorization on a local node:

- Develop a SOAP request message.
- Develop an application class.
- Use the Authorization page to configure the component or content reference for using the authorization service application class.
- Invoke the service operation by calling the authorization service application class method OnAuthRequest().

Row-Level Data Authorization on Remote Nodes

For row-level data authorization on a remote node:

- Develop a SOAP request message.
- Develop an application class.
- Use the Authorization page to configure the component or content reference for using the authorization service application class.

- Invoke the service operation by performing a SyncRequest to the remote node.

Understanding Security Authorization Service Metadata

The following table describes the delivered authorization service metadata.

Note: Developers must create request, response, and any fault messages to use with this service.

Object	*Description*	*Comments*
Service	PTCS_HANDLER	NA
Service operation	PTCS_GETAUTHORIZATION	This is a synchronous service operation. By default this service operation is delivered with no security. By default this service operation is added to permission list PTPT1000.
Application Class Handler	PTCS_ HANDLER:DefaultSecurityHandler	The onAuthRequest method is used with this handler.
Application class	PTCS_ SECURITY:Security:AuthRequest	Methods used with application class: • AuthRequest • GetParameterValue
Application class interface	PTCS_ SECURITY:Security:SecurityHandler	This base interface has only one method, GetAuthorization(), which needs to be implemented by all the child classes.

Understanding Authorization Service Code Examples

This topic contains pseudocode examples to help illustrate using services to authorize object and row-level data access. The code examples are for illustrative purposes only and are not intended to be used in a production environment.

The code examples for authorization service request messages feature all required elements. They may also feature some, but not necessarily all, optional elements. Please refer to the table in the Authorization Service Request Message Elements section for a list of all required and optional elements for authorization service request messages.

See Request Message Elements for the Security Authorization Service.

Prerequisites for Developing Services for Object and Row-Level Authorization

To develop services for object and row-level authorization you should have a general understanding of the PeopleSoft services-oriented architecture and PeopleSoft Integration Broker.

In addition, the following items must be set to use the authorization service:

- Target and schema namespaces.

  See “Understanding Configuring PeopleSoft Integration Broker for Handling Services” (Integration Broker Administration).

- Service operation permissions.

  See “Setting Permissions to Service Operations” (Integration Broker).

- Authentication domain.

  See “Configuring General Portal Properties” (Portal Technology).

- WS-Security.

  See “Implementing Web Services Security” (Integration Broker Administration).

Developing Request Messages for the Security Authorization Service

This section discusses:

- Request message elements for the security authorization service.
- Request messages for authorizing access to content references.
- Request messages for authorizing access to components.
- Request messages for authorizing access to PeopleSoft queries.
- Request messages for authorizing access to PeopleSoft pagelets.
- Request messages for authorizing access to iScripts.

Understanding Developing Request Messages for the Security Authorization Service

An authorization service request message contains a SOAP header followed by a number of authorization request elements.

Inside the message envelope is the PARAMARRAY element. The PARAMARRAY element can contain none to many PARAMS elements. Each PARAMS element corresponds to a separate authorization request. You can bundle multiple requests into a single request.

The following pseudocode shows an example of a request message for the authorization service containing two authorization requests. Each request is contained in a PARAMS element:

```
<!-- Begin SOAP header -->
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsa="http://
schemas.xmlsoap.org/ws/2003/03/addressing/" xmlns:xsd="http://www.w3.org/2001/
XMLSchema/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance/">
  <soapenv:Header xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security soap:mustUnderstand="1" xmlns:soap="http://schemas.xmlsoap.org/
     wsdl/soap/" xmlns:wsse="http://docs.oasis-
      open.org/wss/2004/01/oasis-200401-wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken>
        <wsse:Username>PTDMO</wsse:Username>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <FindAccess xmlns="http://xmlns.oracle.com/Enterprise/Tools/schemas/
     PTCSSecurityReq.v1">

<!-- End SOAP header -->
<!-- Begin message envelope -->

     <PARAMARRAY>
        <PARAMS>
           <SERVICEID>1</ SERVICEID >
           <SERVICE_TYPE>UPGE</SERVICE_TYPE>
           <NODE>PT_LOCAL</NODE>
           <MENU>APPLICATION_ENGINE</MENU>
           <COMPONENT>AE_TOOLS</COMPONENT>
           <MARKET>GBL</MARKET>
           <COMP_ITEM_NAME>SCPERSONALDICT</COMP_ITEM_NAME>
           <KEYVAL>ACTION=U</KEYVAL
           <KEYVAL>SET_ID=S3</KEYVAL>
           <KEYVAL>CUSTOMERID=CATHYPACIFIC</KEYVAL>
        </PARAMS>

        <PARAMS>
           <SERVICEID>2</ SERVICEID >
           <SERVICE_TYPE>CREF</SERVICE_TYPE>
           <PORTAL>EMPLOYEE</PORTAL>
           <NODE>PT_LOCAL</NODE>
           <CREFID>SCPERSONALDICT</CREFID>
           <KEYVAL>NAME=SAMPLEUSER</KEYVAL>
           <KEYVAL>NAME=ARTHI</KEYVAL>
           <KEYVAL>SET_ID=S3</KEYVAL>
        </PARAMS>
     </PARAMARRAY>

     <!-- End message envelope -->

    </FindAccess>
  </soapenv:Body>
</soapenv:Envelope>
```

Important! If the service is invoked on a remote node, it will run on the context of the user ID provided in the <wsse:Username> element defined in the request message header. If the service is invoked on a local node by creating an application class object, the system ignores the <wsse:Username> element value and it implements the code in the context of the user.

Request Message Elements for the Security Authorization Service

The following table describes elements and their usage for request messages used in the security authorization service:

Element	*Usage*	*Comments*
SERVICEID	Differentiates different requests in the incoming message.	Required element. This element is also used to map request messages to response messages, and is particularly useful for mapping sub-requests to sub-responses.
SERVICE_INSTID	Used by PeopleTools internally when multiple instances of the service are used.	Optional element.
SERVICE_TYPE	Service type for which authorization is required.	Required element. Valid values are: • *CREF.* Content reference. • *UPGE.* Component. • *PEP.* Pagelet. (Embedded). • *POP.* Pagelet. • *UQRY.* Query. • *USCR.* iScript. If none of the valid values are defined for the SERVICE_TYPE element in the request message an “Invalid Service Type” message appears in the response message.
NODE	Name of the service provider.	Optional element. When specified the value is passes to the authorization application class. It does not play any other role in determining the security.
CREFID	Content reference ID for the content reference for which authorization is needed.	Required element for service type *CREF.* This element is used to get the CREF authorization in the FindCrefById() function.
MENU	Menu name of the component.	Required element for service type *UPGE.*

Element	*Usage*	*Comments*
COMPONENT	Component name.	Required element for service type *UPGE*
COMP_ITEM_NAME	Item name of the component.	Optional element. The process the system uses to derive this value if one is not specified is described elsewhere in this topic. See Request Messages for Authorizing Access to Components. See “Implementing a Security Authorization Handler” (PeopleCode API Reference)
MARKET	Market name of the transaction.	Optional element. If this element is empty or if a node is not supplied, the value of this field defaults to *GBL,* (global).
PORTAL	Portal name of the provider system.	Optional element used for the following service types: • *CREF.* • *UPGE.* If no value is defined for this element or if there is no value defined for the NODE element, the default portal of the default node is used as the value.

Element	*Usage*	*Comments*

KEYVAL	Key/value pairs to pass to the authorization service.	Optional element use for the following service types to authorize row-level security access:
		• *CREF.*
		• *UPGE.*
		The system uses this element mainly in data security to pass parameters to the Authorization class. It can also be used in basic authorization to send the action mode.
		Use key/value pairs in the following scenarios:
		• Pass key/value pairs to the service.
		• In the Related Content framework, use this element to specify keys of a component.
		• In the Related Content framework and other cases, use this element to pass an action mode, using the key value *ACTION.*
		There can be one or more values for each KEYVAL element. For example:
		`<KEYVAL>AE_PRODUCT=S3</KEY⇒`
		`VAL>`
		`<KEYVAL>CUSTOMERID=CATHYPA⇒`
		`CIFIC</KEYVAL>`
		Note: The value must not contain more than one equal sign (=). If more than one equal sign is specified for the element an error occurs and the system returns a message element (MSG) containing the message *"Invalid Keyval value."*
		For *UPGE* service types only, a special key/value with the key name *ACTION* is available through which action mode can be passed. The *ACTION* key/value specifies the action mode in which to check the authorization.
		For the Related Content framework this value is passed as a service element as follows:
		`<KEYVAL>ACTION=U<KEYVAL>`
		The valid values for the ACTION element are:

Element	*Usage*	*Comments*
		• *A,* Add. Constant value: *%Action_Add* • *U.* Update/Display. Constant value: *%Action_ UpdateDisplay* • *L.* Update/Display All. Constant value: *%Action_ UpdateDisplayAll* • *C.* Correction. • *E.* Data entry. Constant value: *%Action_DataEntry* If you do not define a value for this element the systems ascertains in what mode, of all the available modes, the user has access to the component. If the user has access in multiple modes, the systems uses the mode with the greatest privilege. Though it makes no difference while determining the authorization, it will be of use inside the security application class , into which the action mode is passed via the Authorization Request object.
PAGELETID	Pagelet ID of the pagelet.	Required element for the following service types: • *PEP.* • *POP.* In cases where the pagelet ID is not available but the content reference ID (CREFID) is available, you can authorize pagelet access by selecting *CREF* as the service type and specify the CREFID of the pagelet.
QUERY	Query name.	Required element for service type *UQRY.*
RECORD	iScript record name.	Required element for service type *USCR.*
FIELD	iScript field name.	Required element for service type *USCR.*
FUNCTION	iScript function name.	Required element for service type *USCR.*

Request Messages for Authorizing Access to Content References

The following pseudocode shows an example of the PARAMS section of a request message for authorizing access to a content reference:

```
<PARAMARRAY>
   <PARAMS>
      <SERVICEID>1</SERVICEID>
      <SERVICE_TYPE>CREF</SERVICE_TYPE>
      <NODE>PT_LOCAL</NODE>
      <CREFID>SCPERSONALDICT</CREFID>
       <PORTAL>EMPLOYEE</PORTAL>
   </PARAMS>
</PARAMARRAY>
```

If no value is supplied for the PORTAL element the service uses the value of the default local portal assigned to the node.

Request Messages for Authorizing Access to Components

This section discusses request messages for authorizing access to components and provides code examples of request messages.

IsMenuItemAuthorized

If menu, component and component item name are available, the IsMenuItemAuthorized function call can be used to get authorization. Note that barname and itemname are obtained using menu, market and component name.

If component item name is not available, then the IsMenuItemAuthorized function is invoked for each component item name (page) in the component. The user is provided access even if he or she has access to one of the pages in the component.

Action mode (Update, Update/Display) and other service parameters that need to be passed on to the authorization service application class can be passed to the IsMenuItemAuthorized function through the KEYVAL element with the keyname *ACTION* . See the Authorization Service Request Message Elements chart presented earlier in this section for additional information about using the KEYVAL element and the key name *ACTION*.

Component Authorization Request Messages: Component Name and Action Mode are Available

The following pseudocode shows an example of the PARAMS section of a request message for authorizing access to a component when the component item name and action mode are available:

```
<PARAMARRAY>
   <PARAMS>
      <SERVICEID>1</SERVICEID>
      <SERVICE_TYPE>UPGE</SERVICE_TYPE>
      <MENU>APPLICATION_ENGINE</MENU>
      <COMPONENT>AE_TOOLS</COMPONENT>
      <COMP_ITEM_NAME>COMP_ITEM_NAME</COMP_ITEM_NAME>
      <MARKET>GBL</MARKET>
      <KEYVAL>ACTION=U</KEYVAL>
      <NODE>PT_LOCAL</NODE>
   </PARAMS>
</PARAMARRAY>
```


Component Authorization Request Messages: Action Type is Not Available

The following pseudocode shows an example of the PARAMS section of a request message for authorizing access to a component when the action type is not available. In such cases the action type is determined by the code:

```
<PARAMARRAY>
   <PARAMS>
      <SERVICEID>1</SERVICEID>
      <SERVICE_TYPE>UPGE</SERVICE_TYPE>
      <MENU>APPLICATION_ENGINE</MENU>
      <COMPONENT>AE_TOOLS</COMPONENT>
      <COMP_ITEM_NAME> COMP_ITEM_NAME</COMP_ITEM_NAME>
      <MARKET>GBL</MARKET>
   </PARAMS>
</PARAMARRAY>
```

Component Authorization Request Messages: Component Item Name is Not Available

If a component item name is not present then it is derived as follows: For each of the pages in the component the IsMenuItemAuthorized function is invoked by passing the component item name of each page; if the user has access to the component for at least one of the pages in the component the authorization service will return true.

The following pseudocode shows an example of the PARAMS section of a request message for authorizing access to a component when the component item name is not available, but values for PORTAL and MARKET elements are available:

```
<PARAMARRAY>
   <PARAMS>
      <SERVICEID>1</SERVICEID>
      <SERVICE_TYPE>UPGE</SERVICE_TYPE>
      <NODE>PT_LOCAL</NODE>
      <MENU>APPLICATION_ENGINE</MENU>
      <COMPONENT>AE_TOOLS</COMPONENT>
      <MARKET>GBL</MARKET>
      <PORTAL>EMPLOYEE</PORTAL>used ]
   </PARAMS>
</PARAMARRAY>
```

The following pseudocode shows an example of the PARAMS section of a request message for authorizing access to a component when no values for COMP_ITEM_NAME, PORTAL or MARKET elements are specified. The value for PORTAL is defaulted to the portal of the default provider node; the value for MARKET is defaulted to GBL.

```
<PARAMARRAY>
   <PARAMS>
      <SERVICE_TYPE>UPGE</SERVICE_TYPE>
      <NODE>PT_LOCAL</NODE>
      <MENU>APPLICATION_ENGINE</MENU>
      <COMPONENT>AE_TOOLS</COMPONENT>
   </PARAMS>
</PARAMARRAY>
```

Request Messages for Authorizing Access To PeopleSoft Queries

The following pseudocode shows an example of the PARAMS section of a request message for authorizing access to a PeopleSoft query:

```
<PARAMARRAY>
   <PARAMS>
```



```
      <SERVICE_TYPE>UQRY</SERVICE_TYPE>
      <QUERY>MESSAGE_FOR_MESSAGESET</QUERY>
   </PARAMS>
</PARAMARRAY>
```

The authorization service uses the Query API to get the query authorization for the user.

Request Messages for Authorizing Access to PeopleSoft Pagelets

There are three types of PeopleSoft pagelets:

- Pagelet wizard pagelets.
- Component-based pagelets.
- iScript-based pagelets.

This section provides code examples of the PARAMS section requests messages for authorizing access to these types of PeopleSoft pagelets.

Request Messages for Authorizing Access to Pagelet Wizard Pagelets

To authorize a user for a pagelet wizard pagelet, you must pass the pagelet ID. The following pseudocode example shows passing the pagelet ID:

```
<PARAMARRAY>
   <PARAMS>
      <SERVICE_TYPE>POP</SERVICE_TYPE>  OR  <SERVICE_TYPE>PEP</SERVICE_TYPE>
      <PAGELETID>PAGELET_ID</PAGELETID>
   </PARAMS>
</PARAMARRAY>
```

Request Message for Authorizing Access to Component and iScript Pagelets

To authorize a user to access a component or iScript-based pagelet used the service type *CREF* instead of *POP* or *PEP* and pass the CREFID like any other *CREF* service type request:

```
<PARAMARRAY>
   <PARAMS>
      <SERVICEID>1</SERVICEID>
      <SERVICE_TYPE>CREF</SERVICE_TYPE>
      <CREFID>PAGELET_CREF_ID</CREFID>
   </PARAMS>
<PARAMARRAY>
```

The authorization service queries PeopleTools security data to get the permission lists that can access this iScript. It then checks if the user has access to the permission list.

Request Messages for Authorizing Access to iScripts

The following pseudocode shows an example of the PARAMS section of a request message to authorize access to a PeopleSoft iScript:

```
<PARAMARRAY>
   <PARAMS>
      <SERVICE_TYPE>USCR</SERVICE_TYPE>
      <NODE>PT_LOCAL</NODE>
      <RECORD>WEBLIB_RPT</RECORD>
      <FIELD>ISCRIPT1</FIELD>
      <FUNCTION>IScript_Test</FUNCTION>
```



```
    </PARAMS>
</PARAMARRAY>
```

The authorization service uses the Pagelet Wizard security data to get pagelet authorization for a user.

Working with Response Messages for the Security Authorization Service

This section discusses how to:

- Read authorization status in response messages.
- Evaluate response messages that contain multiple responses.
- Read validation and error information in response messages.

Reading Authorization Status in Response Messages

An authorization service response message contains the element ACCESS which can contain the following values:

- *T.* User can access the content reference, menu, pagelet, query, iScript or row-level data.
- *F.* User is denied access to the content reference, menu, pagelet, query, iScript or row-level data.

Evaluating Response Messages that Contain Multiple Responses

If the request message has three (3) PARAMS elements that correspond to three (3) requests, the response message also contains three (3) PARAMS elements. Each PARAMS element in the response message contains an ACCESS element to convey the authorization status for each corresponding request.

In cases where there are multiple sub requests in a single request, the sub responses do not appear in the same order in the response message as the sub requests in the request message. Use the SERVICEID element value to map the sub responses to the sub requests.

The following examples show how the SERVICEID element maps sub-requests to sub-responses:

The following example shows requests in the order *SVC_1, SVC_2,* and *SVC_3:*

```
<PARAMARRAY>

   <PARAMS>
     <SERVICEID>SVC_1</SERVICEID>
   </PARAMS>

   <PARAMS>
     <SERVICEID>SVC_2</SERVICEID>
   </PARAMS>

   <PARAMS>
     <SERVICEID>SVC_3</SERVICEID>
   </PARAMS>

<PARAMARRAY>
```


The following example shows that the PARAMS elements in the response are not in the same order as in the request:

```
<PARAMARRAY>

   <PARAMS>
     <SERVICEID>SVC_3</SERVICEID>
     <ACCESS>F</ACCESS>
   </PARAMS>

   <PARAMS>
     <SERVICEID>SVC_1</SERVICEID>
     <ACCESS>F</ACCESS>
   </PARAMS>

   <PARAMS>
     <SERVICEID>SVC_2</SERVICEID>
     <ACCESS>F</ACCESS>
   </PARAMS>

<PARAMARRAY>
```

Use the service ID value in each PARAMS element to map the sub responses to the sub requests.

Reading Validation and Error Information in Response Messages

A MSG element is contained within each PARAMS element when the system must convey validation or error information. For example, if a required element is missing from a request message, such as SERVICE_TYPE, or if an exception has occurred, a MSG element that contains information about the validation or error is included in the response.

The following example shows a response message for the authorization service. The information contained in each MSG element conveys validation or error information for the request:

```
<?xml version="1.0"?>
<soapenv:Envelope xmlns:soapenc="http://schemas.xmlsoap.org/soap/encoding/"
xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:wsa="http:
//schemas.xmlsoap.org/ws/2003/03/addressing/" xmlns:xsd="http://www.w3.org/
2001/XMLSchema/" xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance/">
  <soapenv:Header xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <wsse:Security soap:mustUnderstand="1" xmlns:soap="http://schemas.xmlsoap.
    org/wsdl/soap/" xmlns:wsse="http://docs.oasis-open.org/wss/2004/01/oasis-200401
    -wss-wssecurity-secext-1.0.xsd">
      <wsse:UsernameToken>
        <wsse:Username>PTDMO</wsse:Username>
      </wsse:UsernameToken>
    </wsse:Security>
  </soapenv:Header>
  <soapenv:Body xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <FindAccess xmlns="http://xmlns.oracle.com/Enterprise/Tools/schemas/
     PTCSSecurityReq.v1">

<PARAMARRAY>
   <PARAMS>
        <SERVICEID>2</SERVICEID>
        <SERVICE_INSTID/>
        <ACCESS>F</ACCESS>
        <SERVICE_TYPE>CREF</SERVICE_TYPE>
        <MSG>Market name is defaulted to 'GBL'</MSG>
        <MSG>Portal name is defaulted to the default portal EMPLOYEE of the node
          PT_LOCAL</MSG>
         <MSG>Invalid Cref</MSG>
   </PARAMS>

   <PARAMS>
```



```
          <SERVICEID>1</SERVICEID>
          <SERVICE_INSTID/>
          <ACCESS>T</ACCESS>
          <SERVICE_TYPE>CREF</SERVICE_TYPE>
                   <MSG>Market name is defaulted to 'GBL'</MSG>
          <MSG>Portal name is defaulted to the default portal EMPLOYEE of the node
            PT_LOCAL</MSG>
          <MSG>Basic Security Cleared</MSG>
   </PARAMS>
</PARAMARRAY>

 </FindAccess>
</soapenv:Body>
</soapenv:Envelope>
```

Developing the Security Authorization Service Application Class

This section describes how to:

- Develop the authorization service application class.
- Use the Authorization Request object.

Developing the Authorization Application Class

The application class for the authorization service must be created from the base interface PTCS_SECURITY:Security:SecurityHandler. This base interface has only one method, GetAuthorization, which must be implemented by all child classes. This method receives an array of AuthRequest objects as parameters.

Note: You need to develop the security authorization application class when you are performing row-level authorization.

```
Import PTCS_SECURITY:Security:*;
Class SampleSecurityAppclass extends PTCS_SECURITY:Security:SecurityHandler
   /*method AuthRequestHandler(&arrAuthReq As array of PTCS_SECURITY:Security:*/
   /*AuthRequest);*/
   method GetAuthorization(&arrAuthReq As array of PTCS_SECURITY:Security:AuthReque⇒

st);

end-class;

/*method AuthRequestHandler*/
method GetAuthorization
   /+ &arrAuthReq as Array of PTCS_SECURITY:Security:AuthRequest +/
   /+ Extends/implements PTCS_SECURITY:Security:SecurityHandler.GetAuthorization +/

   Local integer &i;
   Local string &val, &userid;

/* Setting the Access Property in the AuthRequest object */
   For &i = 1 To &arrAuthReq.Len
      &arrAuthReq [&i].Access = "T";
   End-For;

/* Reading the Keyvalue from the AuthRequest object */
    &val = &arrAuthReq [1].GetParameterValue("CUSTOMER");

/* Reading the userid from the AuthRequest object */
end-method;
```


Related Links

"Implementing a Security Authorization Handler" (PeopleCode API Reference)

Using the Authorization Request Object

The different parameters of an authorization request that are present in each PARAMS element in a request message are encapsulated in an AuthRequest object. The AuthRequest object stores the key values of the request in an array. Use the GetParameterValues method to retrieve a particular value by passing the key name.

The AuthRequest object has an Access property that you use to set the authorization access for the user. A value of *T* (true) authorizes access and a value of *F* (false) denies access. The value of the Access property is set to *F* by default. You can set the property to *T* from inside the security application class as dictated by business requirements.

Configuring Content Types to Use the Security Authorization Service

This section discusses how to:

- Configure a security application class to map multiple instances of content types to use the security authorization service.
- Configure related content services for content types with the security authorization service configured.

Understanding Configuring Security Application for Security Authorization Service

You can use security authorization service to authorize row-level security for a content type. You can configure a security application class and map one or more than one content type to it. Service related information is passed to the security application class that determines the access permissions of the user. You can configure row level security for the following content types:

- Content Reference
- Component
- Application Class
- iScript
- Pagelets
- PS-Query

You can also set attributes on related content services for a content type that uses the security authorization service.

Related Links

Understanding Using Web Services for Object and Row-Level Data Authorization

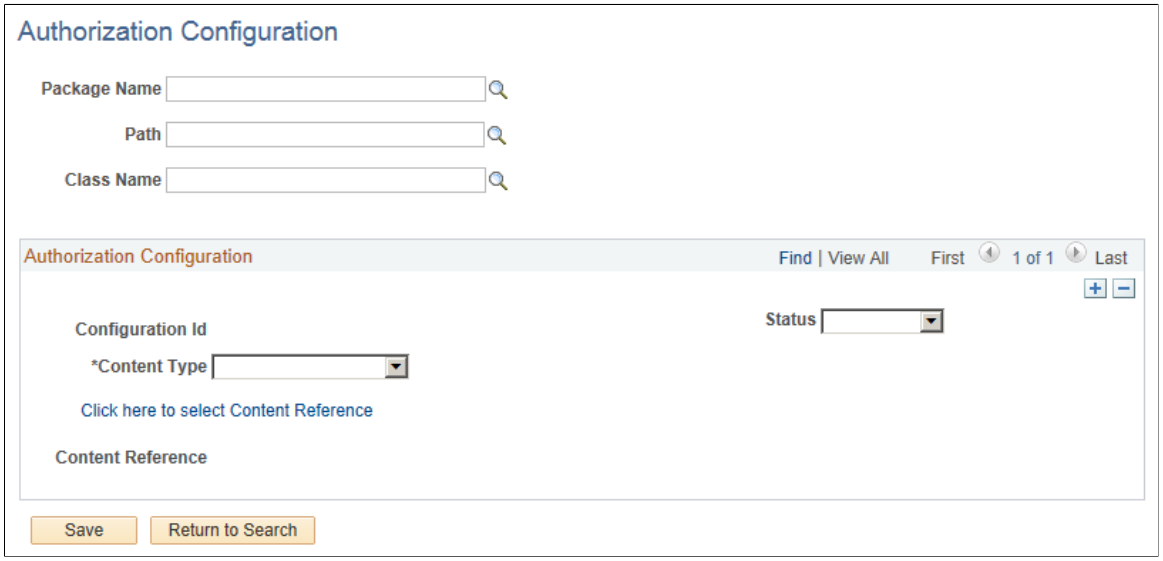
Configuring Security Application Classes for Multiple Content Types When Using the Security Authorization Service

Use the Authorization Configuration page (PTCAC_AUTH_CONFIG) to map content types to a security application class for security authorization services. You can map a single application class to multiple content types on a single page. Enter the information related to a security application class one time and associate different content types to it. This functionality saves time in audit and maintenance tasks.

To access the page, select **PeopleTools** > **Security** > **Security Authorization Service**. The Search Authorization Configuration page appears. You can search for existing authorization configurations using the **Package Name** or **Class Name**. You can also select any of the content type options provided under the **Search In** drop down list.

You can click a search result to display the authorization configuration page loaded with the security application class and associated content types. You can select the **Create new Authorization Configuration** link to create a new authorization configuration page under a single security application class with multiple content types associated to it.

This example illustrates the fields and controls that appear on the Authorization Configuration page.



To create a new authorization configuration for multiple content types:

1. Select the Application package from the **Package Name** field.
2. Enter the qualifying path in the **Path** field.
3. Select the Application Class Name from the **Class Name** field.
4. Select the content type from the **Content Type** drop-down list inside the **Authorization Configuration** group box. The fields and controls in the **Authorization Configuration** group box change with each content type.
5. Click the **Add a new row** icon to add another content type under the same Package Name.

6. Add Content Type details in the Authorization Configuration group box.

7. Click the **Save** button.

The following sections elaborate the fields and controls that are displayed in the Authorization Configuration group box when you select a content type.

Application Class

Select the **Package Name**, **Path**, and **Class Name** to configure an application class for security authorization services. Select **Active** from the **Status** field to apply the security application class.

Click the **Save** button to generate a **Configuration ID**.

Component

Select the **Market**, **Menu Name**, and **Component Name** to configure a component for security authorization services. Select **Active** from the **Status** field to apply the security application class.

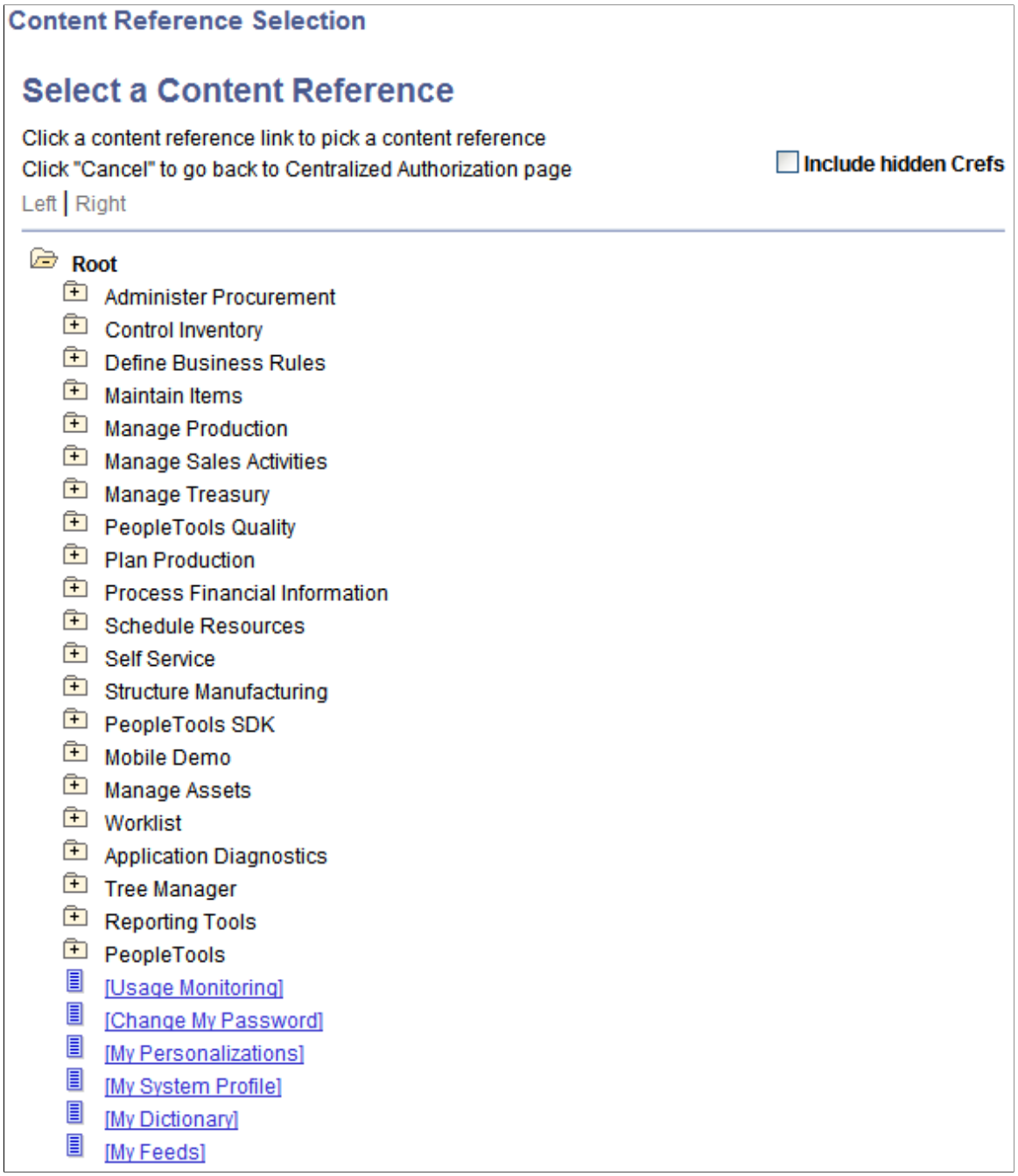
Click the **Save** button to generate a **Configuration ID**.

Content Reference

In the Authorization Configuration group box, select the content reference to configure the service and the provider portal on which the content reference being authorized resides.

Select the **Click here to select Content Reference** link to select the content reference. When you select the link the Select a Content Reference page (PTCAC_CRFURL_SELCT) appears as shown in the following example:

This example illustrates the fields and controls on the Selecting a Content Reference page.



Expand and collapse the folders on the page to select a content reference. The page also features an **Include hidden Crefs** check box. Select the check box to show and select from hidden content references. When you select a content reference, the system displays the Authorization Configuration page and it populates the component value for the content reference.

IScript

Select the **Record**, **Event Name**, **IScript Field**, and **IScript Function** to configure an IScript for security authorization services. Select **Active** from the **Status** field to apply the security application class.

Click the **Save** button to generate a **Configuration ID**.

PS Query

In the Authorization Configuration group box select PS Query as the content type. It displays the **Query Name** field. Select a Query Name that populates the Description. Click the **Save** button to generate the **Configuration ID**.

Pagelet

In the Authorization Configuration group box select Pagelet as the content type. It displays the **Pagelet ID**. Select a Pagelet ID that populates the **Pagelet Title**. Click the **Save** button to generate the Configuration ID.

Related Links

“Understanding Application Classes” (PeopleCode API Reference)
“Administering Content References” (Portal Technology)
“PeopleSoft Query Overview” (Query)
“Understanding Internet Script Classes” (PeopleCode API Reference)
“Understanding Pagelets” (Portal Technology)

Configuring Related Content Services on Content Types with Security Authorization as a Service

You can modify attributes of the related content service that is associated with a content type and has security authorization service configured on it. You can add or update mouse-over text, modify service labels, or enable and disable an action, as an user on the related content service. AuthRequest object, *PTCS_SECURITY:Security:AuthRequest* is provisioned with a SetExtraInfo property which is an array to hold the extra information attributes. You set this property using the SetExtraAttr method.

This SetExtraAttr method is invoked from the security application class with GetAuthorization method on an AuthRequest object. Refer to the table to add the corresponding PeopleCode on a related content service in the Application Designer.

Action	***PeopleCode***
Enable and disable any action.	`&arrAuthReq [1].SetExtraInfo("Enable", "⇒ Y");`
Add or update mouse-over text.	`&arrAuthReq [1].SetExtraInfo("MouseoverT⇒ ext", "This is mouse over text");`
Update service labels of actions.	`&arrAuthReq [1].SetExtraInfo("ServiceLab⇒ el", "This is a Label");`

Related Links

“Creating and Managing Related Content Service Definitions” (Portal Technology)
“AuthRequest Class” (PeopleCode API Reference)

“SecurityHandler Class Methods” (PeopleCode API Reference)

Testing and Debugging the Security Authorization Service

Use the following utilities to test and debug the authorization service:

- Handler Tester Utility.

  Use this utility to test the authorization service application class that you develop.

- Generate SOAP Template Utility.

  Use this utility to test SOAP messages.

See Integration Broker Testing Utilities and Tools.

Chapter 12

Working with SSL/TLS and Digital Certificates

Understanding SSL/TLS and Digital Certificates

The PeopleSoft system takes advantage of HTTPS, Secure Sockets Layer/Transport Layer Security (SSL/TLS), and digital certificates to secure the transmission of data from the web server to an end user's web browser and also to secure the transmission of data between PeopleSoft servers and third-party servers (for business-to-business processing) over the internet.

PeopleSoft customers can implement PeopleSoft software using HTTP or HTTPS. The native SSL/TLS support in commercially available web browsers and web servers is used to provide HTTPS communication between the web browser and web server.

Understanding SSL/TLS

With business-to-business applications, where systems communicate with each other over the internet, data must flow securely. As such, system-to-system authentication is critical. PeopleSoft uses HTTPS and digital certificates for secure transmission of data between systems and system-to-system authentication. PeopleTools use the inherently supported SSL/TLS implementation provided with JRE.™

The PeopleSoft system uses Extensible Markup Language (XML) messaging over HTTPS for our Integration Broker and Business Interlink technologies to deliver system-to-system integration over the internet. HTTPS is used to guarantee secure transmission of the XML message. The digital signature of the XML message is used for authentication between systems. With digital certificates, XML messages are digitally signed to prove that the message came from the server that created and signed the message and to prove that the message has not been altered.

The following table lists the PeopleSoft technologies that use HTTPS (HTTP over SSL/TLS) and how it is implemented for each technology.

Technology	***How HTTPS (HTTP over SSL/TLS) is Implemented***
PeopleSoft Portal Solutions	Secure page transport — Uses web server platform to provide server side SSL/TLS. Secure access to remote content providers — Application server uses JRE to provide the client side of SSL/TLS connection to gateway. Uses web server platform to provide server side SSL/TLS.

Technology	*How HTTPS (HTTP over SSL/TLS) is Implemented*
PeopleSoft Integration Broker (application messaging)	Secure message transport to remote nodes — Application server uses JRE to provide client side of SSL/TLS connection to gateway. Uses web server platform to provide server side SSL/TLS.
PeopleSoft Business Interlinks	Secure calls to remote data sources or modules — Application server uses JRE to provide client side of SSL/TLS connection to gateway. Uses web server platform to provide server side SSL/TLS.
User Authentication	Certificate-based client authentication — Uses web server SSL/TLS client authentication. Certificate data is passed to application server. The application server trusts the web server's authentication. Distinguished name of the certificate is used to log on to PeopleSoft system.

Understanding Certificate Authorities

Anytime you implement SSL/TLS with mutual authentication (both client and server authenticate each other) you need the following three items:

- Server Certificate (issued by some trusted third party or certificate authority).
- Client Certificate (issued by the same trusted third party or certificate authority).
- Client and server both need a copy of a root certificate for the trusted third party. The root certificate has the crypto keys (public and private key) of the authority. Using these keys and the client and server certificates, each party is able to authenticate the other.

When you log on to an SSL/TLS server using your browser, you don’t have to worry about a Root Certificate because they come bundled with the browser. You don’t have to worry about having a client certificate because the web server doesn’t require “Client Side Authentication”.

Important! When you are importing a digital certificate, you may receive an error message if you attempt to import the digital certificate immediately after downloading it from a certificate authority. This is due to issues related to "valid from" dates and times, and the inconsistencies in time settings between different computers. You should save the certificate to a Microsoft Windows workstation, right click on it using Microsoft Windows Explorer, and select Open. This opens the Certificate dialog box. Examine the information regarding the “valid from” and “to” dates. Make sure those dates are valid on the application server the certificate will be installed on. The Details tab on the Certificate dialog presents the most thorough information.

Configuring Digital Certificates

Select **PeopleTools** > **Security** > **Security Objects** > **Manage Digital Certificates**.

The Digital Certificates page displays your inventory of server-side digital certificates. This page also enables you to import new certificates from a certificate authority.

Note: For user certificates, no redundant setup of user certificates is required. With a few lines of Signon PeopleCode, you can reuse the existing PKI server that you have in place.

Note: Currently, root CA key size is limited to 1024 bits.

To view details regarding a particular certificate, click **Details.**

Field or Control	*Description*
Type	Select the type of certificate. *Cert.* Select this option when you are adding a new certificate to your key store. *Local Node.* Select this option when you are setting up a local node for the PeopleSoft messaging system (PeopleSoft Integration Broker). *Root CA.* Select this when you are adding a new Root CA to your key store. *Remote.* Select this option when you are setting up a remote node for the PeopleSoft messaging system (PeopleSoft Integration Broker). *SSH.* Select this option when you are adding a Secure Shell certificate to your key store. The keys can be used for SFTP file transfer using File Attachment PeopleCode methods
Alias	Enables you to add a custom alias for identification purposes.
Issuer Alias	Contains the alias of the authority that issued the certificate.
Valid To	Shows how long the certificate is valid for use.
Detail	Launches a sub-page with more certificate information. The Certificate Detail page reveals subject and certificate information so you can determine such characteristics as the serial number, the fingerprint, the encryption algorithm, and so on. **Note:** Depending on the type of certificate you're adding, this link might be displayed as **Add Root, Import,** or **Request.**

Note: When adding a Local Node certificate and you click the Import link, the Request New Certificate page appears in which you need to add Subject information (Organization, Locality, and so on) and Key Pair information (encryption algorithm, and key size).

Installing Application Server-Based Digital Certificates

This section discusses how to:

- Install application server-based digital certificates.
- Access certificate properties.
- Export and convert certificates.

Understanding Installing Application Server-Based Digital Certificates

This section discusses how to install digital application server-based digital certificates.

Use the procedures discussed in this section for generating and installing digital certificates for use with nonrepudiation and certificated-based node authentication. Installing digital certificates for these security technologies requires that you install digital certificates in the application server keystore on each system participating in an integration.

However, while the process for generating application server-based digital certificates is the same for nonrepudiation and certificate-based node authentication technologies, generate and install separate certificates for each technology.

To install application server-based digital certificates on the PeopleSoft system use the Digital Certificates page (ADMINISTER_CERTS). This page enables you to:

- Install root certificates.
- Install signed public key certificates.
- Install a remote certificate.
- Export a certificate.

To obtain and import a local node certificate, use the Request New Certificate page (CERT_REQ_SBP).

Certificate Types

Each node requires three types of certificates:

- One root certificate from a trusted CA.

  This certificate contains the CA's digitally signed public key. Each root certificate is stored in a record of type *Root CA* in the keystore.

- One certificate containing the default local node's public key, signed by the same trusted CA.

  The CA's root certificate must be installed before you install the default local node's certificate, which is stored in a record of type *Local Node* in the keystore.

- One or more certificates containing the public keys of the remote nodes that participate in nonrepudiation or certificate-based node authenticated messaging.

Each of these certificates is stored in a record of type *Remote.*

Remote Node Certificates

Any participating third-party system must have a set of certificates complementary to those installed at the PeopleSoft nodes.

Installing Application Server-Based Digital Certificates

This section discusses how to:

- Add CA authorities and install root certificates.
- Install signed public key certificates.
- Resolve root certificate mismatches.
- Install remote certificates.

Adding CA Authorities and Installing Root Certificates

PeopleSoft delivers a number of root certificates. Before you begin this process, check to see if your root certificate already exists. If it does, there is no need to perform this step.

If your root certificate does not exist, contact your CA for information on how to obtain the root certificate for importing into PeopleSoft.

To install a new root CA certificate:

1. Select **PeopleTools** > **Security** > **Security Objects** > **Manage Digital Certificates**. The Digital Certificates page displays.
2. Add a CA authority:
   a. Click the plus button (+). A new row appears.
   b. From the **Type** drop-down list, select **Root CA**.
   c. In the **Alias** field, enter the alias name for the certificate.
   d. In the **Issuer Alias** field, enter an alias for the issuer. Click the **Lookup** button to select the certificate alias as the issuer alias.
3. Add the root certificate.
4. Click the **Add Root** link near the plus button (+). The Add Root Certificate page displays.
5. Copy the contents of the certificate into the text box.

   You must include the begin section (-----BEGIN CERTIFICATE-----) and end section (-----END CERTIFICATE -----).
6. Click the **OK** button.

7. Click the **Refresh** button.

Install Signed Public Key Certificates for Application Server-Based Digital Certificates

To section discusses how to:

- Add local node certificates to the PeopleSoft system and generate CSRs.
- Submit local node certificates to CAs for signing.
- Import signed local node certificates into the PeopleSoft system.

To install a signed public key certificate, you must define a local node certificate row in the keystore, then obtain the signed certificate from a CA whose root certificate is installed. To do this, you generate a CSR, submit the CSR to the CA, then retrieve and import the content of the signed certificate into your certificate row.

To add a local node certificate and generate a CSR:

1. Select **PeopleTools** > **Security** > **Security Objects** > **Manage Digital Certificates**. The Digital Certificates page displays.
2. Click the plus button (+). A new row appears.
   a. From the **Type** drop-down list, select **Local Node**.
   b. In the **Alias** field, enter the name of the local node.

      Note: The name you enter must exactly match the name of the local node.

   c. In the **Issuer Alias** field, click the **Lookup** button to select the issuer alias.
3. At the end of the row, click the **Request** link. The Request New Certificate page displays.
4. In the Subject Information section, enter the following information:

   These fields represent attributes of the default local node's DN. The CA to whom you submit the CSR might require values for any or all of the fields. The DN is also stored on the Detail page of the local node certificate. For the common name, enter the name of the PeopleSoft Integration Broker default local node.

Field or Control	***Description***
Company Name.	Enter the default local node name (with no underscore).
Org Unit(organizational unit)	Enter the name of the organizational unit.
Organization	Enter the name of the organization.

Field or Control	*Description*
Locality	Enter the location of the organization.
State/Province	Enter the state or province name.
Country	Enter the two-character country code.

5. In the Key Pair Information section, from the **Algorithm** drop-down list, select a value. The values are:
   - *MD5 with RSA encryption.*
   - *SHA1 with DSA encryption.*
   - *SHA1 with RSA encryption.* (Default.)
   - *SHA256 with RSA encryption.*
6. From the **Key Size** drop-down list, select a key size (bits):.
   - *1024.* (Default.)
   - *2048.*
   - *4096*
   - *768.*
   - *512.*
7. Click the **OK** button.

   In addition to generating the CSR, which contains the default local node's public key, this step also creates the matching private key, which is automatically installed in the same row of the node's keystore.

To submit a local node certificate for signing:

1. After you click the **OK** button as described in the previous section, the CSR is generated. Copy the CSR and submit it to your CA for signing.

   The process of obtaining digital certificates varies, depending on the CA. Typically, a CA requires you to paste the content of the PEM-formatted CSR into a form that you submit online.

   The CA may send you the signed public key certificate by email or require you to download it from a specified web page.

   When you submit the CSR for signing, you must include the begin section (-----BEGIN NEW CERTIFICATE REQUEST-----) and the end section (-----END NEW CERTIFICATE REQUEST-----).

2. When you receive the signed certificate back, copy it to a temporary directory. For example:

   ```
   c:\temp\newcert.cer
   ```

After you generate a CSR for the local node certificate and obtain a signature, you import the signed certificate into PeopleSoft.

To import signed local node certificates into a PeopleSoft system:

1. Select **PeopleTools** > **Security** > **Security Objects** > **Manage Digital Certificates**. The Digital Certificates page displays.
2. Locate the row that contains the local certificate.
3. At the end of the row, click the **Import** link. The Import Certificate page displays.
4. Open the signed certificate you received back from the CA, copy it and paste it into the text box. The content you paste must include the begin section (-----BEGIN CERTIFICATE-----) and end section (-----END CERTIFICATE-----).
5. Click the **OK** button.
6. Click the **Refresh** button.

Three outcomes are possible:

- The Digital Certificates page appears and the new certificate's row now contains a Detail link. In this case, the certificate has been successfully installed, and you can proceed to install remote certificates for the node.

  Note: The new certificate's row may contain a different issuer alias than the one that you selected for it. This indicates that the keystore contains a root certificate signed by the same CA that signed the new certificate, but it wasn't the one with the issuer alias that you selected (the issuer alias of a root certificate doesn't always reflect which CA actually signed the certificate). PeopleSoft Integration Broker has changed the issuer alias for the new certificate to correctly reflect which root certificate is its parent.

- The following message may appear: *Could not decode PEM-formatted certificate data.* This indicates either that the pasted content isn't formatted properly as a certificate, or that the certificate is not yet valid.

  Every signed digital certificate has a period of time during which it can be used, specified by its internal timestamp fields, **Valid From** and **Valid To,** which are set by the signing CA. The timestamps were inserted by the CA's certificate server. You can't import the certificate content until the **Valid From** time has passed on your default local node's application server, which may lag by several minutes, depending on the relative clock accuracy of the two servers. Note that time zones are automatically accounted for and have no effect on this issue. You must examine the**Valid From** field in the certificate's properties dialog box to determine when the certificate can be imported.

  See Accessing Certificate Properties.

- The following message may appear: *The certificate signature is not valid. The certificate is corrupt or has been modified.* This indicates either that the certificate has been tampered with, or that the keystore contains no root certificate signed by the same CA.

The issuer alias of a root certificate doesn't always reflect which CA actually signed the certificate. Therefore it's possible that the CA to which you submitted your CSR didn't sign any of your installed root certificates. The local certificate in your keystore must be accompanied by a root CA certificate signed by the same CA.

Resolving Root Certificate Mismatches

To import a signed public key certificate to the application server keystore as a row of type *Local Node* on the Digital Certificates page, a root certificate signed by the same CA that signed the public key certificate must already exist as a *Root CA* row on that page.

If you cannot import a signed public key certificate because no matching root certificate exists, you can resolve the deficiency by installing the root certificate of the CA that *did* sign your public key certificate. Then you obtain a new signed public key certificate from that CA.

To resolve a root certificate mismatch:

1. Export the embedded root certificate from the signed public key certificate file.

   See Exporting and Converting Certificates.

2. Define a new root CA certificate in the keystore.

   Refer to the previous procedure for establishing a root certificate.

3. Delete the local node row from the keystore's Digital Certificates page.

4. Add a new local node certificate to the keystore using the same issuer alias as the new root CA certificate.

   Refer to the previous steps for installing a signed public key certificate.

Installing Remote Certificates for Application Server-Based Digital Certificates

To section discusses setting up remote certificates for nonrepudiation and certificated-based node authentication and describes how to:

- Export remote node certificates.
- Add remote node CAs and import remote node certificates into the local node system.

To establish two-way authentication or nonrepudiation, each node must possess copies of the other participating nodes' public keys. You accomplish this with a certificate row of type *Remote* in the default local node's application server keystore, which contains a certificate exported from the row defined as *Local Node* in a remote node's keystore. You define one remote certificate for each participating remote node.

Note: Each remote certificate is a copy of the local node certificate and is installed on the remote node that it represents. As a result, you must first establish a root CA certificate and install a local node certificate on node A before you can export a copy of that certificate to node B. The simplest approach is to first install a certificate of type *Root CA* and a certificate of type *Local Node* on each of the participating nodes. Then you can export each of the local node certificates and import them to the other nodes as type *Remote.*

The following requirements apply:

- The remote system's local node certificate must already be installed.

  Refer to the previous steps for installing a signed public key certificate.

- The local system must have a root certificate installed with the same issuer alias (and actual issuer) as the remote system's local node certificate.

  Refer to the previous steps for establishing a root certificate.

Note: For the purposes of this discussion, assume that both local and remote nodes are PeopleSoft applications. If the remote node is a third-party system, the same requirements must still be satisfied—the third-party system must provide a copy of its signed public key certificate to the PeopleSoft node.

To export a remote node certificate:

1. On the remote node system, select **PeopleTools** > **Security** > **Security Objects** > **Manage Digital Certificates**. The Digital Certificates page displays.
2. Locate the row that contains the default local node, and click the **Detail** link at the end of the row. The Certificate Details page displays.
3. Click the **Export** button and copy the content in the edit box.
4. Click **Cancel**.

To add a remote node CA and import a remote node certificate into the local node system:

1. On the local node system, select **PeopleTools** > **Security** > **Security Objects** > **Manage Digital Certificates**. The Digital Certificates page displays.
2. Click the plus button (+). A new row appears.
   a. From the **Type** drop-down list, select **Remote Node**.
   b. In the **Alias** field, enter the name of the remote node.

      Note: The name you enter must exactly match the name of the remote node.

      ---

   c. In the **Issuer Alias** field, click the **Lookup** button to select the issuer alias.
3. Click the **Refresh** button.
4. At the end of the remote node row, click the **Import** link. The Import Certificate page displays.
5. Paste the certificate that you exported in the previous section into the text box. You must include the begin section (-----BEGIN CERTIFICATE-----) and the end section (-----END CERTIFICATE-----).
6. Click the **OK** button.
7. Click the **Refresh** button.

Accessing Certificate Properties

When you need to install a signed public key certificate in a keystore, you need the issuing CA's root certificate in the keystore as well. Your public key certificate is more than a single certificate; the same file contains the issuing CA's root certificate as well. If you do not receive a separate root certificate from the CA, you can access it from the public key certificate properties.

When you need to export a root certificate or examine the certificate's valid dates—or when you need to convert a certificate between DER and PEM formats—use the security extensions on a Windows machine to access the certificate properties dialog box .

To access certificate properties:

1. Double-click any certificate file with a .DER (binary format) extension or a .CER (PEM format) extension.

   This invokes the Windows extensions for security management, which open a dialog box so you can inspect the certificate properties.

2. (Optional.) Access the properties of the embedded root certificate.

   a. Select **Certification Path**.

      A tree structure appears, showing the hierarchical chain of trust between the public key certificate and its issuer root certificate. Your certificate has the common name that you supplied for it, and the issuer root certificate (its parent) has the name of its issuing CA.

   b. Select the root certificate, and click **View Certificate**.

      A dialog box display the properties of the root certificate.

3. (Optional.) Select **Details**.

   A list of fields appears. Click a field name to examine its value. This is especially useful for determining the certificate's **Valid From** and **Valid To** date and time.

Exporting and Converting Certificates

You might need to export an embedded root certificate or convert an existing certificate from DER format to PEM format. You can export certificates from:

- DER or PEM formatted certificate files.
- Certificate rows in a PeopleSoft application server keystore.

To export or convert a certificate from a file:

1. Access the properties dialog box of the certificate to export or convert.

   See Accessing Certificate Properties.

2. In the certificate properties, select **Details**, then click **Copy to File**.

   The Certificate Export Wizard launches.

3. Click **Next**, then select a format.

   Base64-encoded X.509 (.CER) is the PEM format option, which is recommended. The *DER encoded binary X.509 (.CER)* option may also work, depending on the environment.

4. Click **Next**, and then browse to select a location and file name for the new certificate file.

   Specify the same location as the certificate. Ideally, you should give an exported root certificate file the same name as the issuing CA.

5. Click **Next**, then **Finish** to save the root certificate file.

   A message indicates when the export is successful.

To export a certificate from an application server keystore:

1. In the PeopleSoft Pure Internet Architecture, sign on to the application database and select **PeopleTools** > **Security** > **Security Objects** > **Manage Digital Certificates**.

   The Digital Certificates page appears.

2. Click the **Detail** link of the desired certificate, then click **Export**.

   The Export Certificate page appears, containing the exportable certificate content in a long edit box.

3. Copy the entire certificate content and sign out of the database.

   Note: Save this certificate content to a file with a .CER extension.

Installing Web Server-Based Digital Certificates

This section discusses how to install digital certificates on Oracle WebLogic web servers.

Understanding Web Server-Based Digital Certificates

In addition to using the information in this section to generate and install web server-based digital certificates, you can use this information to generate and install gateway-based digital certificates for:

- Integration gateway encryption.
- Client authentication.
- WS-Security.

Note that for integration gateway encryption if the integration gateway is installed on a web server that has SSL/TLS implemented, the integration gateway and web server can share the digital certificates. As a result, you do not need to install separate integration gateway certificates. However, if the integration gateway is installed on a web server where SSL/TLS is not implemented, you must generate and install digital certificates on that web server.

For more information about generating and installing integration gateway-based digital certificates see

- “Installing Integration Gateway-Based Digital Certificates” (Integration Broker Administration)

- The “Setting Up Gateway Private Keys (WebLogic)” section later in this topic.

Understanding Installing Web Server-Based Digital Certificates

You must install web server-based digital certificates to implement web server SSL encryption.

You use utilities provided with the Oracle WebLogic software to install web server-based certificates for SSL encryption. This authentication secures inbound messages. The web server requires three elements:

- The web server's private key.
- A certificate containing the web server's public key, digitally signed by a trusted certificate authority (CA).
- A root certificate from the CA that signed the web server's public key.

The information in section outlines the basic steps required to obtain and install the certificates and keys that you need. Oracle WebLogic provides their own interface and methodology for establishing SSL encryption—you should refer to the documentation supplied with the web server software for detailed information about this process. In addition, refer to the information supplied by the selected CA.

Note: PeopleSoft delivers a number of certificate authorities and root certificates. If your certificate authority or root certificate is not listed, you need to add it to the PeopleSoft system.

You use the web server software to generate its own private key. At the same time, it also generates a certificate signing request (CSR), which contains the web server's public key. You submit the CSR to the selected CA, which creates, digitally signs, and returns your web server's public key certificate to you. This certificate might be in standard DER-encoded binary format; however, it can be converted to PEM format if necessary. You then install both signed certificates, and you register them and your private key with your web server, so that the web server recognizes and uses them.

Understanding the PSKeyManager Utility

PSKeyManager is a command-line utility delivered with PeopleTools that you use to generate and import digital certificates into the keystore. The location of the PSKeyManager utility is:

```
<PIA_HOME>\webserv\peoplesoft\piabin
```

The basic syntax of PSKeyManager is:

```
pskeymanager -command
```

Note: The first time you launch a command using the PSKeyManager utility you are prompted to define a unique keystore password.

Each command can be followed by a variety of options. Both the command and the keyword for each option that you invoke with it must be preceded by a hyphen, and most options must be followed by a value.

When you navigate to the PSKeyManager utility, start it with the command *pskeymanager* and hit the *Enter* key, a list of all commands and their options is displayed. The PSKeyManager utility provides ten or so commands, but you'll use only two of the options for this task:

```
pskeymanager -create
```



```
pskeymanager -import
```

Note: The `pskeymanager -create` command supports the Subject Alternate Name (SAN) attribute, which allows you to specify more than one host name, IP address, or other value for a single SSL certificate.

Keystore Location for SSL/TLS Digital Certificates

The keystore location for SSL/TLS digital certificates is:

```
<PIA_HOME>\webserv\peoplesoft\piaconfig\keystore
```

In addition, integration gateway, client authentication, and WS-Security certificates are stored in this location.

Installing Digital Certificates for SSL/TLS Encryption on Oracle WebLogic

This section describes how to install digital certificates for SSL/TLS encryption for the Oracle WebLogic environment and discusses how to:

- Generate and import public keys.
- Generate private keys and CSRs.
- Submit CSRs to CAs for signing.
- Import signed private keys into keystores.
- Set up gateway private keys.
- Set up Oracle WebLogic Console for SSL.

Generating and Importing Public Keys (WebLogic)

Before you can generate and import public keys into PeopleSoft, you must access and download the signed public key from your CA. The process for accessing and downloading the signed public key varies, depending on your CA. Contact your CA for information on how to perform these tasks.

To generate and import public keys:

1. Place the public key from your CA in the keystore. The location of the keystore is:

   ```
   <PIA_HOME>\webserv\<DOMAIN>\piaconfig\keystore
   ```

2. Open a command prompt and navigate to the keystore:

   ```
   <PIA_HOME>\webserv\peoplesoft\piaconfig\keystore
   ```

3. Enter the following at the prompt:

   ```
   pskeymanager -import
   ```

4. At the **Enter current keystore password** prompt, enter the password and press **Enter**.

5. At the **Specify an alias for this certificate** prompt, enter the alias name and press **Enter**.

The alias name you enter must be the same one you entered when you generated the private key.

6. At the **Enter the name of the certificate file to import** prompt, enter the path and name of the certificate to import, and press **Enter**.

7. At the **Trust this certificate** prompt, enter *Yes* and press **Enter**.

Generating Private Keys and CSRs (WebLogic)

You use PSKeyManager to generate private keys. PSKeyManager is a wrapper to Sun Microsystem's Keytool for managing keys and certificates.

While using PSKeyManager, press the **Enter** key to select any of the default values presented.

To generate the private key and the CSR on Oracle WebLogic:

1. Open a command prompt and navigate to the keystore:

   ```
   <PIA_HOME>\webserv\peoplesoft\piaconfig\keystore
   ```

2. Enter the following at the prompt:

   ```
   pskeymanager -create
   ```

3. Enter the current keystore password and press **Enter**.

4. At the **Specify an Alias for this Certificate <host_name>?** prompt, enter the certificate alias and press **Enter**.

   The default certificate alias is the local machine name.

5. At the **What is the common name for this certificate <host_name>?** prompt, enter the host name for the certificate. For example:

   ```
   <host_name>.corp.example.com
   ```

   Press **Enter**.

   Enter the exact name as it will be accessed in a browser URL. For example, for a URL of https://server.example.com/ps/signon.html, enter server.example.com. The default common name is the same as the alias.

6. At the **What is the Subject Alternate Name for this certificate?** prompt, enter one or more host names.

   Enter a Subject Alternate Name (SAN) in the format type:value, where type can be any of domain name server (DNS), IP address, EMAIL, URI, or an arbitrary object identifier (OID). For example, DNS:server.example.com or IP:192.0.2.1. The default SAN is DNS:common_name.

   To enter more than one value, separate the type:value entries with commas and no spaces. For example: DNS:server.example.com,DNS:server2.example.com,IP:192.0.2.1.

7. Enter the appropriate information at the following prompts. Press **Enter** after each entry.

   a. Organization unit.

   b. Organization.

c. City of locality.

d. State or province.

You must spell out the entire state name. Do not enter an abbreviation.

e. Country code.

f. Number of days the certificate should be valid.

The default value is *90*.

g. Key size to use.

The default value is *1024*.

h. Key algorithm.

The default value is *RSA*.

i. Signing algorithm.

The default value is *SHA256withRSA*.

8. At the **Enter a private key password** prompt, enter the password or press **Enter** to use the keystore password.

9. Verify that the values you entered are correct, and press **Enter**. To go back and change any values, enter *No* and press **Enter**.

PSKeyManager generates a private key and provides the certificate signing request (CSR) that you will provide to the CA for signing. The following example shows a sample CSR.

```
-----BEGIN NEW CERTIFICATE REQUEST----- MIIBtDCCAR0CAQAwdDELMAk
GAxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxFD
ASxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxDV
QQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx43
lCxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxMGfk
/jYxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx2F
zdxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxEE
BQxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxqI
hoxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxQT
qsxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxaxYI=
-----END NEW CERTIFICATE REQUEST-----
```

The CSR is written in as a text file to the <PIA_HOME>\webserv\peoplesoft directory. The file name is *<host_name>_certreq.txt*.

Submitting CSRs to CAs for Signing (WebLogic)

After you generate the private key and a certificate signing request (CSR), you must submit the CSR to the certificate authority (CA) for signing.

The process of obtaining the signature varies, depending on the CA that you select. Typically, a CA requires you to paste the content of the PEM-formatted CSR into a form that you submit online. However, the CA may send the signed public key (root) certificate to you by email or require you to download it from a specified web page. The CA may also provide its root certificate or instructions for retrieving it.

Use the appropriate method to submit a CSR for signing as determined by your CA.

When you do submit the CSR for signing the content you provide must include the begin section (-----BEGIN NEW CERTIFICATE REQUEST-----) and the end section (-----END NEW CERTIFICATE REQUEST-----) of the CSR.

The CA will return the signed certificate to you that you must import into the keystore.

Importing Signed Private Keys into Keystores (WebLogic)

You use PSKeyManager to import a server-side private key into the keystore.

1. Open a command prompt and navigate to the keystore:

   ```
   <PIA_HOME>\webserv\peoplesoft\piaconfig\keystore
   ```

2. Enter the following at the prompt:

   ```
   pskeymanager -import
   ```

3. At the **Enter current keystore password** prompt, enter the password and press **Enter**.

4. At the **Specify an alias for this certificate** prompt, enter the alias name and press **Enter**.

   The alias name you enter must be the same one you entered when you generated the private key.

5. At the **Enter the name of the certificate file to import** prompt, enter the path and name of the certificate to import, and press **Enter**.

6. At the **Trust this certificate prompt**, enter *Yes* and press **Enter**.

Setting Up Gateway Private Keys (WebLogic)

To set up private keys for gateways, follow the procedures outlined in the following topics presented earlier in this section:

- Generating Private Keys and CSRs.
- Submitting CSRs to CAs for Signing.
- Importing Server-Side Private Keys into Keystores.

The only difference is that for the following prompts you enter names that are gateway-specific:

Prompt	***Sample Values***
Certificate alias.	Enter an alias, such as *PT860GATEWAY*.
Common name for this certificate.	Enter a name, such as *PT860GATEWAY*.

Setting Up Oracle WebLogic for SSL/TLS Encryption

This section describes how to set up Oracle WebLogic for SSL/TLS encryption.

Note: Several pages and fields mentioned in this section reference only SSL. These pages and fields are also used for setting up TLS.

To set up Oracle WebLogic for SSL/TLS:

1. Login to WebLogic Console.
    a. Open a web browser.
    b. In the URL or address field, enter *http://localhost/index.html* and press **Enter**. The Web Server Index Page displays.
    c. Click **Access WebLogic Server Console**. The signon page for WebLogic Server Administration Console appears.
    d. Enter the **Username** and **Password** and click *Sign In*. WebLogic Administration Console displays.

       The username and password are those that you specified when you installed PeopleSoft Pure Internet Architecture.

2. Navigate to the PIA server Configuration page using one of these methods:
    - In the WebLogic Server Console In the left navigation area, navigate to **PeopleSoft** > **Servers** > **PIA**.
    - In the WebLogic Server Console, in the Domain Configuration section, click **Servers**. The Servers page displays. In the table that appears on the page, click the **PIA** link.

3. Click the Keystores and SSL tab.
4. In the Keystore Configuration section, on the right side of the page, click the **Change** link. The Specify Keystore Type page displays.
5. From the **Keystores** drop-down list, select *Custom Identity and Custom Trust*.
6. Click the **Continue** button. The Configure Keystore Properties page displays.
7. In the Custom Identity section complete the following fields:
    a. In the **Custom Identity Key Store File Name** field, enter *keystore/pskey*.
    b. In the **Custom Identity Key Store Type** field, enter *JKS*.
    c. In the **Custom Identity Key Store Pass Phrase** field, enter *password*.
    d. In the **Confirm Custom Identity Key Store Pass Phrase** field, enter *password* again.
    e. Click the **Continue** button. The Review SSL Private Key Settings page displays.
8. In the Review SSL Private Key Setting page, review the information and click the **Continue** button.
9. Click the **Finish** button. You will restart the web server at a later time. You are returned to the Keystore Configuration tab.

10. Scroll down the page to the Advanced Options section and click the **Show** link.
11. In the Server Attributes section, from the **Two Way Client Cert Behavior** drop-down list box, select *Client Certs Requested and Enforced*.

    Note: Set this option only if the node is set up for certificate-based authentication or non-repudiation, or if required for two-way SSL.

12. Click the **Apply** button.
13. Restart the web server.

Implementing Web Server SSL/TLS Encryption

This section provides an overview of web server SSL/TLS encryption and discusses how to:

- Configure web server SSL/TLS encryption.
- Implement web server SSL/TLS encryption.

Understanding Web Server SSL/TLS Encryption

This section discusses:

- Outbound web server SSL/TLS encryption.
- Inbound web server SSL/TLS encryption.

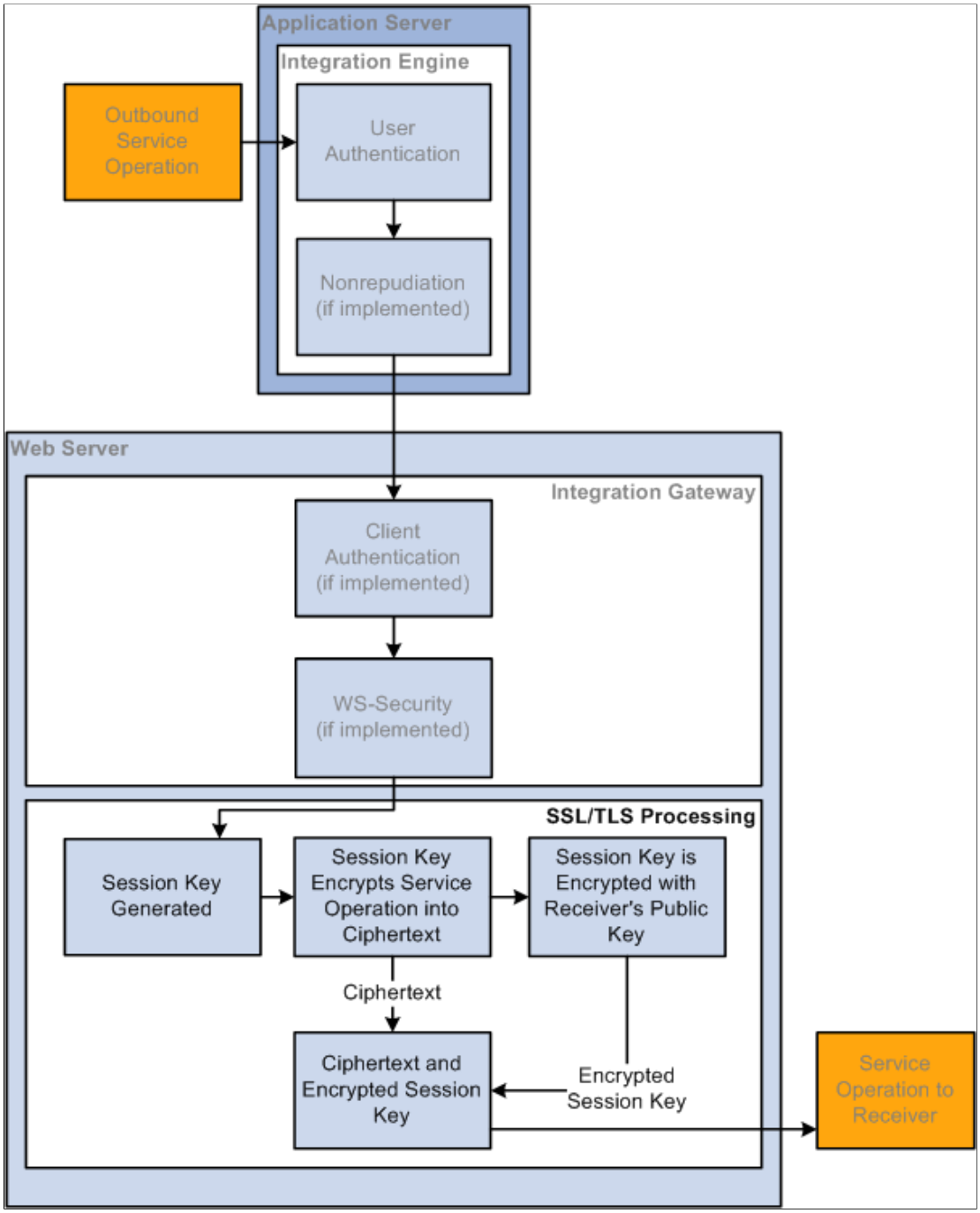
Outbound Web Server SSL/TLS Encryption

This section discusses outbound web server SSL/TLS encryption.

Before the integration starts, your integration partner generates a key pair that consists of a private key and a public key. The private key is placed in its web server keystore. The public key is placed in a digital certificate.

You contact the integration partner's site using a secured URL that begins with HTTPS. The integration partner's site responds by sending you its web server digital certificate, which contains the public key of the key pair it generated prior to initiating the integration.

This diagram shows the processing that occurs on outbound transactions when web server SSL/TLS encryption is implemented. The diagram shows all possible security processing for an outbound integration to show where in the processing flow SSL/TLS encryption occurs. However, the SSL/TLS encryption steps are highlighted.



Your web server generates a session key to encrypt the plain text outbound request contents into ciphertext. Then the web server encrypts the ciphertext and session key using your integration partner's public key that was sent to you in the digital certificate.

The session is now secure and all communication is encrypted and can only be decrypted by you and your integration partner.

When the request arrives at your integration partner's web server, the integration partner's web server uses its private key to decrypt the ciphertext and session key. It then uses the session key to decrypt the ciphertext and extract the service operation contents in plain text.

Inbound Web Server SSL/TLS Encryption

This section discusses inbound web server SSL/TLS encryption.

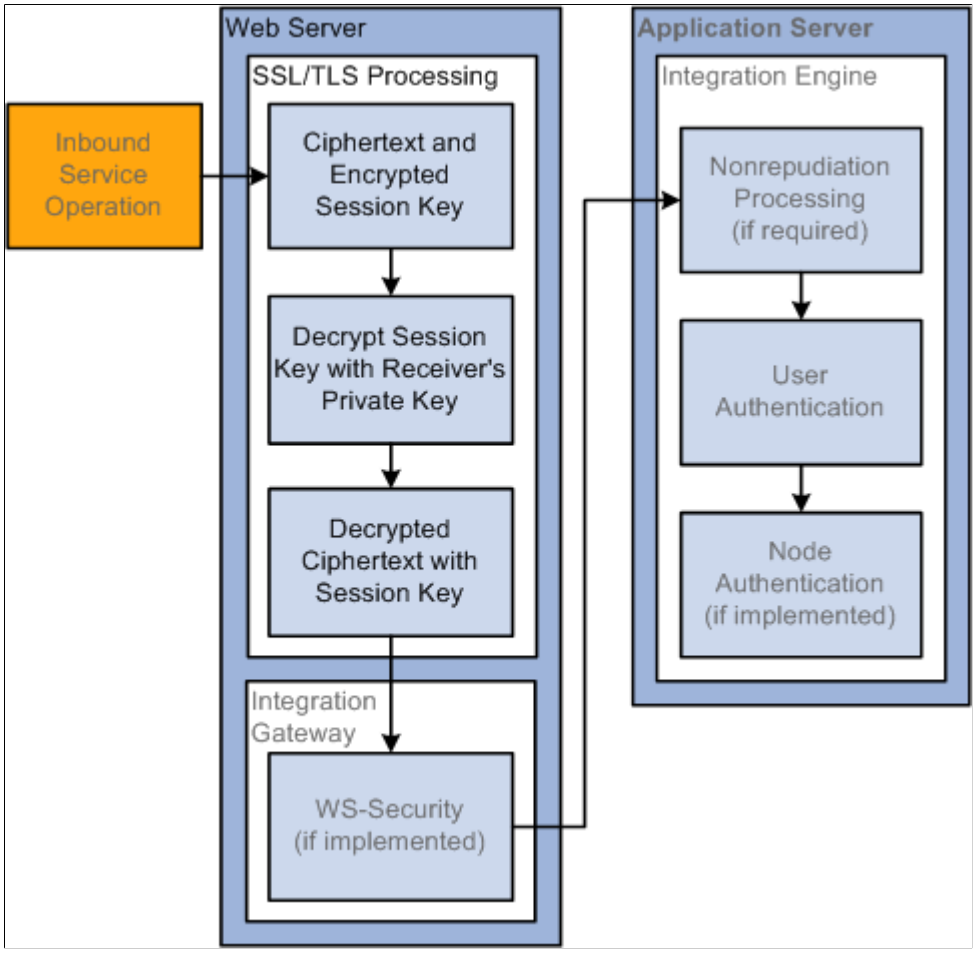
Before the integration starts, you generate a key pair that consists of a private key and a public key. You place the private key in your web server keystore and the public key gets placed in a digital certificate.

For inbound web server SSL/TLS encryption processing, your integration partner contacts you using a secured HTTPS URL. Your web server responds by sending the integration partner a web server digital certificate that contains your public key. The integration partner's web server goes through the outbound processing described in the previous section.

This diagram illustrates the processing that occurs on inbound transactions when web server SSL/TLS encryption is implemented. The diagram shows all possible security processing for an inbound integration to show where in the processing flow SSL/TLS encryption occurs, with SSL/TLS encryption processing highlighted in the foreground.

The SSL/TLS encryption processing steps highlighted in the diagram are the receipt of the ciphertext and encrypted session key, the system's descryption of the session key and receiver's private key, and the decryption of the ciphertext with the session key.

After this SSL/TLS encryption processing, the service operation passes to the application server for additional processing.



When the service operation arrives on your web server, it is one package that contains the ciphertext (encrypted service operation contents) and the encrypted session key that decrypts the ciphertext.

Your web server uses its private key to decrypt the ciphertext and session key. It then uses the session key to decrypt the ciphertext into a plain text service operation.

Prerequisites for Implementing Web Server SSL/TLS Encryption

You must set up web server-based digital certificates to implement web server SSL/TLS encryption.

See Installing Web Server-Based Digital Certificates.

Configuring Web Server SSL/TLS Encryption

Configuring web server SSL/TLS encryption involves the following tasks:

- Supply the digital certificates containing the public and private keys required for encrypted transactions. You install these certificates in the web server keystore. You configure the web server to recognize and use its installed certificates for SSL/TLS transactions.
- Edit the Integration Gateway Certificates section of the integrationGateway.properties file to convey parameters for the web server certificates that you installed in the gateway keystore.

Integration Gateway Properties File Parameter	***Description***
ig.certificateAlias	Certificate alias.
ig.certificatePassword	Certificate alias password.

See “Using the integrationGateway.properties File” (Integration Broker Administration).

Implementing Web Server SSL/TLS Encryption

For outbound transactions you must change the value of the HTTP target connector's PRIMARYURL;URL property to start with *https://* instead of *http://*. You can apply this setting on a node-by-node basis, or apply it to the gateway as a whole, which will use it as the default setting for all nodes. The HTTP target connector makes the necessary SSL/TLS connection at runtime.

Receipt of HTTPS requests is nearly automatic. When the integration gateway's HTTP listening connector receives an HTTPS request, it is forwarded to the default local node for authentication.

Implementing Client Authentication

This section provides an overview of client SSL authentication, prerequisites, and discusses how to implement client authentication.

Understanding Client Authentication

As mentioned previously in this topic, outbound transactions connect from the PeopleSoft application server to the integration gateway using an HTTP over MIME connection. Client SSL encryption allows you to secure this connection. Client SSL encryption is not implemented on inbound transactions from the integration gateway to the application server, since this connection is made using a Jolt connection.

Client SSL encryption is typically implemented when the application server and integration gateway each reside on separate machines.

Client SSL encryption is implemented using digital certificates and you must have them installed on the integration gateway.

Note: You must have web server SSL encryption set up and implemented to use client SSL authentication. With web server SSL set up and implemented, client SSL authentication will fail.

After digital certificates are installed, there are no other steps required to implement client SSL authentication.

Related Links

“Installing Integration Gateway-Based Digital Certificates” (Integration Broker Administration)

Chapter 13

Working with Web Service Security (WS-Security)

Understanding WS-Security

By implementing the Web Service Security (WS-Security) standard, PeopleSoft provides the ability to leverage emerging XML security technologies to address web services security requirements. WS-Security provides:

- A way for applications to construct secure SOAP message exchanges.
- A general-purpose mechanism for associating security tokens with SOAP messages.
- XML message integrity and confidentiality.

By providing WS-Security capabilities, you can leverage the standard set of SOAP extensions, which you use when building secure web services, to implement message content integrity and confidentiality. WS-Security provides a way to insert and convey security tokens in SOAP messages. The ability to leverage WS-Security standards provides for better interoperability and improved usability, enabling the implementation of robust security within a WSRP-capable environment. The solutions being provided through the PeopleSoft WS-Security implementation include:

- Enable web service security between WSRP consumer and producer.

  The web services consumer passes the appropriate identification to a producer as part of the SOAP message, so that producer can verify the identity in order to process requested web services on behalf of the user without requiring a user to log in. Single signon between the web services consumer and producer is currently supported in PeopleSoft WSRP Portal and PeopleSoft Integration Broker.

- SOAP message integrity.

  Ensuring that messages have not been tampered with.

- SOAP message confidentiality.

  Guaranteeing that messages are protected against eavesdroppers.

The WS-Security Username Token Profile defines a standard way to associate user ID and password information in the SOAP messaging for web services interoperability.

The Security Assertion MarkUp Language (SAML) token uses assertions to define a standard way to associate common information such as issuer ID, `NotBefore` and `NotOnOrAfter` conditions, assertion ID, subject, and so on.

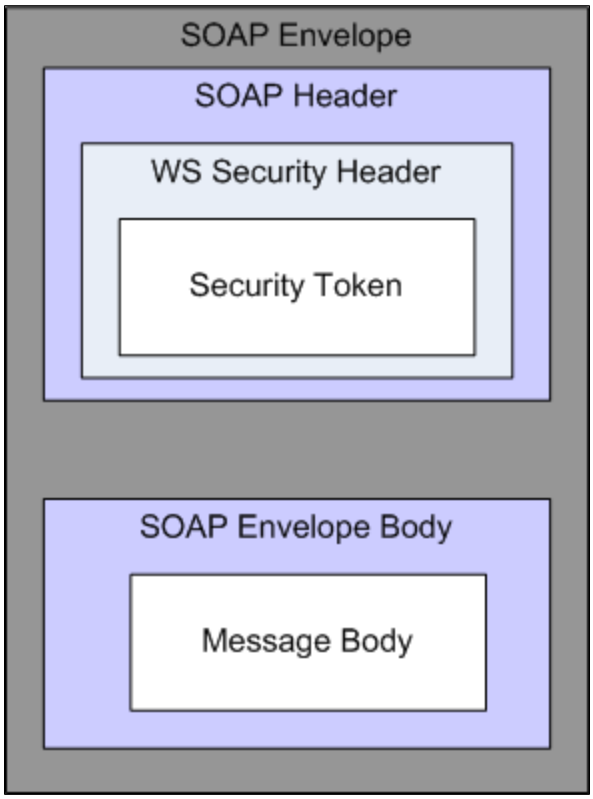
The OASIS WS-Security specification is the open standard for web services security. Its goal is to let applications secure SOAP message exchanges by providing encryption, integrity, and authentication

support. It provides authentication support for SOAP messaging. WS-Security offers these general-purpose mechanisms for associating security tokens with message content:

- Username token.
- SAML token.

Note: PeopleSoft provides multiple levels of security for WSRP. These levels, or options, are discussed in the following topic. PeopleSoft recommends that you determine the level that is appropriate for your needs before implementing WS-Security. Using SSL/TLS connections to secure transmissions may be sufficient.

This figure shows how WS-Security inserts and conveys security tokens in SOAP messages.



Implementing WS-Security for WSRP

If using the web services for remote portals (WSRP) technology, you implement WS-Security.

See “Understanding the PeopleSoft WS-Security for WSRP” (Portal Technology).

Note: WSRP and WS-Security for WSRP will be desupported in a future release.

Implementing WS-Security for PeopleSoft Integration Broker

If using PeopleSoft Integration Broker, you configure WS-Security to ensure secure transmissions.

See “Understanding Setting Up Secure Integration Environments” (Integration Broker Administration).

Chapter 14

Applying Digital Signatures to PDF Report Output

Understanding Applying Digital Signatures to PDF Report Output

BI Publisher for PeopleSoft report output in PDF format can be digitally signed to verify the authenticity of the report output that you send and receive, and to validate that the output has not been altered since the PDF was created and digitally signed.

Note: As of the PeopleTools 8.54 release you can apply digital signatures to BI Publisher for PeopleSoft PDF output only.

This feature digitally signs PDF report output using Personal Information Exchange (PFX) certificates.

Participants, Permission Lists, and Roles in Applying Digital Signatures to PDF Report Output

This section describes the participants, permission lists, and roles in applying digital signatures to PDF report output.

Participants

This section describes the participants in applying digital signatures to PDF report output.

Term	***Definition***
System or Security Administrator	Provides digital signature setup and maintenance
Report Developer	Person who develops the report.
Report Signer	Person whose digital signature is required for signing the report.
Report Operator	User or role that is running the report and that can apply the digital signature to the document.

Permission Lists

Permission list *PTPT4900* allows access to the External Digital Certificates page (described later in this topic), to create details for the digital certificate and stipulate users and roles that can apply digital signatures to documents.

The following participants should have access to the permission list:

- System or security administrator.
- Report signer.

If the system administrator or security administrator wants to limit access to the External Digital Certificates page and does not want a signer to access the page, he/she or someone else with appropriate permission list access can access the page to define the signer of the document.

Roles

The *XMLP_REPORT_DEVELOPER* role allows access to properties on the Report Definitions – Property page to enable, place, and specify the digital signature to apply to a document.

The report signer, report operator, and anyone else who should be able to apply a digital signature to a report should be assigned this role.

Process Overview

To use this feature:

- Use the External Digital Certificates page define the digital ID, load signed PFX certificates to the PeopleSoft database, define the signer of the certificate, and define the users and roles that can apply the digital signature to documents.

  Information about using the External Digital Certificates page and performing these tasks is provided later in this topic.

- Create a PeopleCode application class to exchange reporting, signing, and digital certificate information between BI Publisher for PeopleSoft and the application that requires digitally signed PDF report output.

  The application class should also include logic to extract the PFX certificate from the database keystore and provide BI Publisher for PeopleSoft an encrypted password to open the certificate file and perform the signing of the report.

  Information about the interface to use for this application class is described later in this topic and also in the *PeopleTools: PeopleCode API Reference* documentation.

- In BI Publisher for PeopleSoft use the Report Definition – Properties page to enable digital signatures for PDF output, define the position of the digital signature in the document, and specify the application class to exchange reporting, signing, and digital certificate with the application.

  Using the Report Definition – Properties page is described in the *PeopleTools: BI Publisher for PeopleSoft* documentation.

- Run the report.

Running BI Publisher for PeopleSoft reports is described in the *PeopleTools: BI Publisher for PeopleSoft* documentation.

PeopleTools for Applying Digital Signatures to PDF Report Output

This section describes PeopleTools used for applying digital signatures to PDF report output.

Term	***Definition***
PeopleSoft Security	Use the External Digital Certificates page in the PeopleSoft Pure Internet Architecture to: • Define the digital ID. • Upload digital certificates (PFX files) for interacting with BI Publisher and PeopleSoft applications. • Define distinguished name (DN) properties for certificates. • Define users and roles that can apply digital signatures to PDF report output.
PeopleCode	Several application class interfaces, application classes, and methods are provided for storing and retrieving PFX files and passwords, mapping digital signatures to input properties, digitally signing reports, and so on.
BI Publisher for PeopleSoft	Use BI Publisher for PeopleSoft to: • Enable digital signatures for PDF report output. • Define the location of digital signatures in PDF output. • Generate digitally signed PDF reports.

Prerequisites for Applying Digital Signatures to PDF Report Output

Before you can apply digital signatures to BI Publisher for PeopleSoft PDF output documents, the following should be done:

- Obtain a digital ID from a public certificate authority. To use signed documents for internal use you can obtain a digital ID from a private or internal certificate authority.
- Store the digital ID inside the PFX file.
- Install the PFX file, including its password, into the database keystore. Note that the minimum length for the password is eight (8) characters.

- The development team of the application group that is requesting the signed PDF report output must create the application class to exchange reporting and signing information between BI Publisher for PeopleSoft and the application.

Considerations for Applying Digital Signatures for PDF Report Output

Consider the following points when applying digital certificates to PDF report output:

- Only a single digital ID can be used to sign a PDF document. Thus, only one digital signature can sign a document.
- The digital signature is enabled at the report level. As a result, multiple templates assigned to the same report will share the digital signature properties.

Developing PeopleCode for Applying Digital Signatures for PDF Report Output

This topic provides a high-level overview of PeopleCode for applying digital signatures to PDF report output.

See the *PeopleTools: PeopleCode API Reference* documentation for more detailed information about the application classes and methods for implementing this feature.

PeopleCode for Applying Digital Signatures to PDF Report Output

The following PeopleCode is used in conjunction with applying digital signatures to PDF report output.

Term	***Definition***
IPT_PDFSIGNATURE_INT	Application developers requesting report signing should create an application class implemented with this interface. This application class should have application-specific logic which maps a digital signature ID to application class input properties. The application class should interact with the PT _SECURITY_DIGITALCERTSTORE application class to retrieve digital certificate information. On the Report Definition – Properties page, the report developer defines the name of this application class to map the generated PDF output to the signing authority.

Term	*Definition*
PT_SECURITY_DIGITALCERTSTORE	Use this application class to store and retrieve digital certificates in the database keystore. This application class contains the logic to: • Provide an external caller with information regarding digital certificates (PFX file) that are stored in the PeopleSoft database keystore. • Extract a PFX file to a specified location and provide the caller with an encrypted password to open/use the certificate.

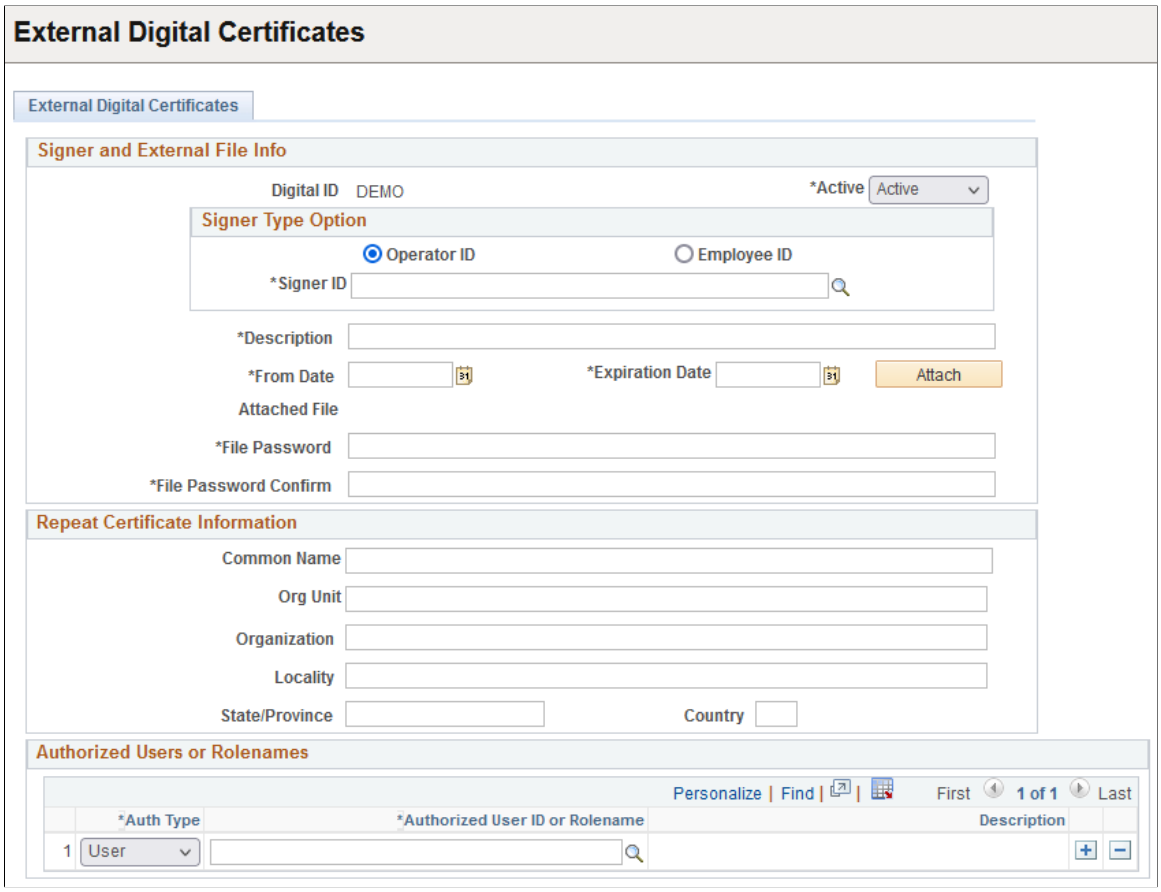
Using the External Digital Certificates Page

Use the External Digital Certificates page (PT_CERT_EXTFILE) to:

- Load a signed PFX digital certificate to the PeopleSoft database.
- Define the certificate signer. This is the person whose digital signature is to be used on the document.
- Define the users and roles that can apply the digital signature to documents.

To access the page select **PeopleTools** > **Security** > **Security Objects** > **External Digital Certificates.**

This example illustrates the fields and controls on the External Digital Certificates page. You can find definitions for the fields and controls later on this page.



Use the Signer and External File Info section of the page to define the certificate signer, upload the PFX certificate to the database keystore, and define the validity dates for the certificate. This table describes the fields and controls in the Signer and External File Info section of the page.

Field or Control	***Description***
Digital ID	Displays the digital ID for the certificate.
Active	Use the field to activate or deactivate the certificate. The valid values are: • *Active.* (Default.) Activate the certificate. • *Inactive.* Deactivate the certificate.
Operator ID	Click the radio button to specify the certificate signer (in the **Signer ID** field) by user ID.
Employee ID	Click the radio button to specify the certificate signer (in the **Signer ID** field) by employee ID.

Field or Control	*Description*
Signer ID	Click the **Lookup** button to select the certificate signer. The values that appear for selection depend on whether you've selected *Operator ID* or *Employee ID* as the signer type.
Description	Enter a description for the certificate.
From Date	Enter the beginning validity date for the certificate.
Expiration Date	Enter the expiration date for the certificate.
Attach	Click the button to attach the PFX certificate file. The file should include the certificate and its password.
File Password	Enter the PFX certificate password.
File Password Confirm	Enter the PFX certificate password again.

This table describes the fields and controls in the Repeat Certificate Information section of the page.

These are standard signature attributes that define the owner and details of the signature file as required by a certificate authority, audits, and reporting as required by local regulatory systems.

Field or Control	*Description*
Common Name	Enter the common name defined in the attached PFX certificate.
Org Unit	Enter the organizational unit defined in the attached PFX certificate.
Organization	Enter the organization defined in the attached PFX certificate.
Locality	Enter the locality defined in the attached PFX certificate.
State/Province	Enter the state/province defined in the attached PFX certificate.
Country	Enter the country defined in the attached PFX certificate.

Use the Authorized Users or Rolenames section of the page to define the users and roles that can apply the digital signature to PDF report output. This table describes the fields and controls in the Authorized Users or Rolenames section of the page.

Field or Control	*Description*
Auth Type	Choose the authorization type from the drop-down list. The valid values are: • *Role.* • *User.* (Default.)
Authorized User ID or Rolename	Click the Lookup button to select the ID. The values that appear for selection defined on the authorization type selected.

Click the Add Row button (+) to add additional users and roles that can apply the digital signature.

Digitally Signing Historical Reports

Historical (old) reports should be signed by a user with current credentials; expired PFX files cannot be used for signing historical reports.

Chapter 15

Encrypting Text With PSCipher

Understanding the Advanced Encryption Standard (AES) Implementation

The PSCipher utility provides Advanced Encryption Standard (AES) for increased data security. When you install PeopleSoft PeopleTools on your application servers and web servers, a default, AES encryption key is provided. For security reasons, you must generate a unique encryption key using PSCipher command line utility as described in this documentation. The new key can be used to re-encrypt passwords included in your PeopleSoft environment files.

The version of the default encryption key is version 2.1, or {V2.1}. If you generate a unique key, the current version used by the system would be {V2.2}. Each time you generate a new key, the system increments the current version number.

Using the PSCipher Utility

The PSCipher feature encrypts and decrypts text used in your PeopleSoft system. System administrators interact with PSCipher through a Java command-line utility located on the web server, which enables you to encrypt text, such as user IDs and passwords, stored in configuration files. PSCipher also involves a runtime element, running on the application server, that decrypts the encrypted text. The runtime element requires no user interaction.

In previous releases, PSCipher was used, for example, to secure the node IDs and node passwords used in conjunction with PeopleSoft Integration Broker configurations. The following sections describe how to use PSCipher for these actions:

- Text encryption: Encrypt (AES) a variety of text values stored in various configuration files throughout your system.
- Key generation: Generate a new key using PSCipher command line utility.
- Version maintenance: The key file maintains a version history of all previous versions of the key file, which enables text encrypted with a previous version to be decrypted.
- Key maintenance: Update an encryption key.

Note: The current version of PSCipher is able to understand encrypted text from previous releases.

The PSCipher utility is located in the <*PS_CFG_HOME*>\webserv\<*domain_name*>\piabin directory.

Encrypting Text

To encrypt text, you submit text values in the form of arguments that PSCipher encrypts and then displays in its encrypted form.

To encrypt a password on Microsoft Windows, change to the directory where PSCipher resides. Enter the command and supply the text to be encrypted:

```
pscipher <clear_text_value>
```

To encrypt a password on UNIX, change to the directory where PSCipher resides. Enter the command and supply the text to be encrypted:

```
PSCipher.sh <clear_text_value>
```

PSCipher returns the encrypted form of these submitted text values, which you can then copy to a configuration file to assign to a configuration parameter.

Note: Due to the limitations of certain characters in command-line interfaces on different platforms, such as Microsoft Windows, Linux and so on, you may not be able to use these characters while running PSCipher for encryption. Therefore, you need to check the submitted text values for the character that is causing the incorrect encrypted result.

Note: This same procedure will need to be applied whenever you intend to encrypt text using PSCipher. Note that in the following sections of this document it is assumed that you understand how to encrypt the text value.

Generating a Unique Encryption Key

You use the PSCipher Java utility's `buildkey` command to build new AES encryption keys. The `buildkey` command adds a new AES encryption key stored in the psvault file (the key file). If you generate new versions of the key file, the system appends the new version of the key to the end of the key file.

To invoke the command on a Microsoft Windows server, change to the directory where PSCipher resides and enter:

```
pscipher -buildkey
```

To invoke the command on UNIX, change to the directory where PSCipher resides and enter:

```
PSCipher.sh -buildkey
```

Select one web server in your system to generate the new version of the key file. The pscipher.bat and PSCipher.sh utilities only run in the Java environment of the web server. After you have created the new key file, you then copy the new version of psvault from the initial server to the appropriate directories on all the appropriate servers in your system. The psvault file is stored in different directories depending on your web server vendor. Oracle WebLogic is the supported web server software for the current release. On the application server the psvault file resides in <*PS_HOME*>\secvault.

Note: If you are not using the default encryption key and you have generated a unique encryption key, note that each time you add a new server to your system, you will need to copy the key file to the appropriate location on that server. For example, if you are using the default key version ({V2.1}), any server you add to the system and install PeopleTools on will also have the default key version ({V2.1}). As such, no further steps are required. However, if you have generated a new key, giving the version number a value of {V2.1} or greater, then you need to make sure to copy that key file to the added server(s). Also, each time you update the key, you need to ensure that the new version of the key file is copied to the additional servers in your system.

Warning! When you upgrade to new PeopleTools releases, as in PeopleTools 8.58 to PeopleTools 8.60, you will need to back up any modifications you have made to the key file using PSCipher in the previous release and reapply that same key file to the appropriate servers onto which you have installed the new PeopleTools release.

Renewing the Existing Cipher Text

To renew an existing Cipher text using PSCipher utility, use the PSCipher Java utility's **renew** command.

On Microsoft Windows, change to the directory where PSCipher resides. Enter the command and supply the existing encrypted value:

```
pscipher -renew <existing-cipher-text>
```

On UNIX, change to the directory where PSCipher resides. Enter the command and supply the existing encrypted value:

```
PSCipher.sh -renew <existing-cipher-text>
```

Existing encrypted values for the passwords in web server files can be replaced with the encrypted password created using PSCipher utility.

This table lists web server files and passwords to check.

Location	***File***	***Example of Password Text***
<*PS_CFG_HOME*>\webserv\<*domain_name*>\applications\peoplesoft\PSIGW.war\	integrationGateway.properties	• ig.isc.NODE_NAME.password={V2.2}XXXXXX • ig.isc.NODE_NAME.DomainConnectionPwd={V2.2}XXXXXX • Others as needed. There are several sections including passwords. Be sure to update all that apply to your environment.
<*PS_CFG_HOME*>\webserv\<*domain_name*>\applications\peoplesoft\PSIGW.war\WEB-INF	gatewayUserProfile.xml	<password>{V2.2}XXXXXX</password>

Location	*File*	*Example of Password Text*
<*PS_CFG_HOME*>\webserv\<*domain_name*>\applications\peoplesoft\PORTAL.war\ WEB-INF\psftdocs\<*site_name*>	configuration.properties	• KeyStorePwd={V2.2}XXXXXX • DomainConnectionPwd={V2.2}XXXXXX • WebUserId={V2.2}XXXXXX • WebPassword={V2.2}XXXXXX

A new key of version {V2.2} is generated successfully.

Updating the Encryption Key on Oracle WebLogic

With Oracle WebLogic, PSCipher.bat or PSCipher.sh is stored in the location

<*PS_CFG_HOME*>\webserv\<*domain_name*>, and the psvault is stored in the location <*PS_CFG_HOME*>\webserv\<*domain_name*>\piaconfig\properties.

Generating the Encryption Key on Oracle WebLogic

To update the encryption key:

1. Change directory to <*PS_CFG_HOME*>\webserv\<*domain_name*>\piabin.

   For example, where the web server domain is peoplesoft:

   ```
   cd PS_CFG_HOME\webserv\peoplesoft\piabin
   ```

2. Run PSCipher –buildkey to create a new key in the key file.

   For example, on Microsoft Windows:

   ```
   PSCipher.bat -buildkey
   Your environment has been set.
   A new key of version {V2.1} is generated successfully
   ```

   For example. on UNIX:

   ```
   PSCipher.sh -buildkey
   Your environment has been set.
   A new key of version {V2.1} is generated successfully
   ```

3. Copy <*PS_CFG_HOME*>\webserv\<*domain_name*>\piaconfig\properties\psvault to the equivalent location on all other web server hosts and to <*PS_HOME*>\secvault\psvault on all application servers in your system.

4. Modify the encrypted text fields as described in the following sections.

Updating the Web Profile

The configuration.properties file is located in the following directory:

<*PS_CFG_HOME*>\webserv\<*domain_name*>\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs\<*PIA_site_name*>

The following encrypted text values in the configuration.properties file need to updated:

```
WebUserId=encrypted_password
WebPassword=encrypted_password
```

Submit the values for these properties to PSCipher, and copy the generated encrypted text to the WebUserID and WebPassword properties in the configuration.properties file, overwriting any previous value assigned to the property.

Updating the Integration Gateway

On the Integration Gateway, you need to modify the following files:

- gatewayUserProfile.xml
- integrationGateway.properties

The gatewayUserProfile.xml file is located in the following directory:

<*PS_CFG_HOME*>\webserv\<*domain_name*>\applications\peoplesoft\PSIGW.war\WEB-INF

In the gatewayUserProfile.xml file, update the following text value:

```
<password>{V2.1}encrypted_password</password>
```

Note: There can be more than one password field in this file. There could be different `<password> </password>` entries for different users. You should use PSCipher to encrypt all `<password> </password>` entries.

Submit the values for these properties to PSCipher, and copy the generated encrypted text to the `<password> </password>` entry in the gatewayUserProfile.xml file, overwriting any previous value.

The integrationGateway.properties file is located in the following directory:

<*PS_CFG_HOME*>\webserv\<*domain_name*>\applications\peoplesoft\PSIGW\WEB-INF

Update the following text values stored in the integrationGateway.properties file.

Note: If you are not currently assigning a value to one of the following properties, you don't need to supply a value.

- `ig.isc.password=`
- `ig.isc.$NODENAME.password=`
- `#ig.certificatePasswd=`
- `secureFileKeystorePasswd=`

- `#ig.jms.JMSTargetConnector.JMSProvider.Password=`
- `# ig.jms.Queue1.Password=`
- `# ig.jms.Topic1.Password=`
- `#ig.jms.Topic1.NodePassword=`

Submit the values for these properties to PSCipher, and copy the generated encrypted text to the corresponding entries in the integrationGateway.properties file, overwriting any previous value.

Updating WSRP/WSS

You need to update the wss.properties file used for Web Services Remote Portal (WSRP) and Web Services Security (WSS).

The wss.properties file needs to be updated in the following locations:

- <*PS_CFG_HOME*>\webserv\<*domain_name*>\applications\peoplesoft\PORTAL\WEB-INF\classes\
- <*PS_CFG_HOME*>\webserv\<*domain_name*>\applications\peoplesoft\pspc\WEB-INF\classes\

Update the following text entry in the wss.properties file in both locations:

`org.apache.ws.security.crypto.merlin.keystore.password=`

Submit each password value to PSCipher, and copy the generated encrypted text to the corresponding entries in the wss.properties file, overwriting any previous value.

Securing the External Key File

The encryption key used by PSCipher is stored in a key file named psvault. This file is critical to your system security. It is very important to protect this file using *at least* the concepts discussed in this section.

Setting up Operating System File Security

The key file should be secured and protected by your operating system with the appropriate file access permissions on all platforms. The recommended file access permissions are:

- File ‘read’ access for only the administrators that need to run the PSCipher command-line utility to encrypt text.
- File ‘read’ access for the only the administrators that need to start the application servers and web servers.
- File ‘write’ access for only the administrators that need to run PSCipher –buildkey to create a new PSCipher key.

Backing Up the Key File

It will be a time-consuming task to recover your system if you accidentally damage or delete the key file. Therefore, it is important to save a backup of your key file. It is recommended that every time you build a new key that you backup your latest key file to a safe location.

Note: You only need to keep the latest version of your key file for your backup. The latest version contains a version history of previous keys.
For additional information on upgrade of psvault, refer to the upgrade documentation.

Chapter 16

Securing Data with PeopleSoft Encryption Technology

Understanding Data Security

To understand PeopleSoft Encryption Technology, it's first necessary to understand the types of data security that cryptography in general can provide.

Data security comprises the following elements:

- Privacy – keeping data hidden from unauthorized parties.

  Privacy is normally implemented with some type of encryption.

- Integrity – keeping transmitted data intact.

  Integrity can be accomplished with simple checksums, or better, with more complex cryptographic checksums known as one-way hashes. Many times, checksums are combined with a type of asymmetric cryptography to produce digital signatures. These signatures, when verified, assure you that the data has not changed.

- Authentication – verifying the identity of an entity that is transferring data.

  Authentication can also be accomplished using digital signatures, which makes them an obvious choice for data security.

Privacy Through Encryption

There may be certain regulatory, certification, or legal requirements to store certain data in a secure manner. For instance credit card numbers should not be stored in clear text form. Many businesses use encryption technology to secure this data.

Encryption is the scrambling of information such that no one can read it unless they have a piece of data known as a key. Using the key, the sender encrypts *plaintext* to produce *ciphertext.* The recipient also uses a key to decrypt the ciphertext, producing the original plaintext. The type of key at either end of this transaction, and the way it's applied, constitute an encryption algorithm. In all cases, the security of an encryption algorithm should *not* rely on its secrecy. Rather, it should rely on how well the operations involved affect the input data.

Data encryption algorithms come in two major forms: Symmetric cryptography and asymmetric cryptography. Symmetric cryptography falls into two categories: Block ciphers and stream ciphers. The bulk of cryptographic research has gone into block ciphers, which are employed by PeopleSoft Encryption Technology.

Symmetric Encryption

Symmetric encryption involves both encrypting and decrypting a piece of data using the same key, which is stored on the sending and receiving entities. To make it a bit harder to crack symmetric encryption schemes, they can be applied in a number of encryption *modes*. These modes provide ways of applying encryption sequentially to blocks of data, such that each block is encrypted by a combination of the encryption key and the previously encrypted block. Of course, when encrypting the first block, a previously encrypted block isn't available, so the encryption software applies a random *initialization vector* (IV) to get the process started. This IV does not have to be secret.

The most popular symmetric encryption modes currently in use are:

- Electronic Code Book (ECB).
- Cipher Block Chaining (CBC).
- Cipher Feed Back (CFB).
- Output Feed Back (OFB).

For information on these modes, see the OpenSSL online documentation.

There's a drawback with symmetric cryptography: The recipient of symmetrically encrypted ciphertext must possess the same key to decrypt it that you used to encrypt it. Because of this, you'll need a secure method of transmitting the key. This can be done a number of ways. You can send the key electronically over a private line that cannot be tapped; you can personally hand the key to your recipient; or you can use a courier to deliver the key. None of these approaches is foolproof or very efficient. A partial solution to this problem is asymmetric encryption.

Authenticated Symmetric Encryption

Authenticated symmetric encryption builds on symmetric encryption, by allowing you to provide additional authentication data for the encryption and decryption algorithms.

This release supports these Authenticated Encryption Algorithm modes:

- CCM mode — Counter with CBC-MAC
- GCM mode — Galois/Counter Mode

See Authenticated Encryption Algorithms in OpenSSL Algorithms.

For an overview of these modes, see the OpenSSL online documentation.

Asymmetric Encryption

Asymmetric encryption involves the use of a pair of complementary keys, in which one key is used to encrypt a piece of data and the other key is used to decrypt it. This system uses *public key encryption* technology. The encryption key is called the public key and is widely distributed. The decryption key is the private key, which its owner must never reveal or transmit. Asymmetrically encrypted ciphertext is readable only by the owner of the private key. Anyone who wants to send ciphertext to that party needs only to have a copy of the recipient's freely available public key to perform the encryption.

Although asymmetric encryption is by design an excellent way for strangers to exchange data, it requires more computing power and capacity than symmetric encryption. Because of this, symmetric and

asymmetric encryption are typically used in combination, to take advantage of the strengths of each system.

You apply the more efficient symmetric encryption to your data using a randomly generated symmetric key, which leaves only the problem of transmitting your symmetric key (also known as the *content encryption key*) to the recipient, who can use it to decrypt the ciphertext. You use the recipient's public key as a *key encryption key,* to apply asymmetric encryption to your symmetric key, not to your already encrypted ciphertext. The ciphertext and your symmetric key can now both be transmitted to the recipient. The recipient's private key is used to decrypt your symmetric key, which in turn is used to efficiently decrypt the ciphertext.

Integrity Through Hashing

Integrity can be provided with a *cryptographic hash.* There are several well-known hash types, including MD2, MD4, MD5, SHA1, and RIPEMD160. These hash types have the following properties in common:

- They're one-way.

  You cannot reverse the operation and get back the text that produced the hash. Indeed, this is obvious since most hashes have values that are 128-256 bits long. The size of a typical message will far exceed this, so it's extremely unlikely that the hash could contain all of the original information.

- They're collision resistant.

  There's almost no possibility of finding two meaningful messages that produce the same hash. Each hash algorithm has a different degree of collision resistance.

To use hashing, you generate a hash value from your data and include it when you transmit the data. The recipient uses the same hash algorithm to generate a hash value from the received data. If the result matches the transmitted hash, the data wasn't altered in transit.

Authentication Using Digital Signatures

Authentication can be accomplished in a number of ways. These include:

- Fixed passwords.
- Time-variant passwords.
- Digital signatures.

Digital signatures are by far the most popular and most reliable method of authentication. Digital signatures usually combine a hash with another cryptographic operation (typically asymmetric encryption) to produce a type of check that not only verifies that the data was not altered in transit, but also assures that the named sender is, in fact, the actual sender of the data.

For example, if we provide a digital signature based on SHA1 with RSA encryption, this means that an SHA1 hash of the message was encrypted with the private key of the sender. Because the SHA1 hash is very collision resistant, and assuming the private key of the sender is known only by the sender, then verifying such a signature indicates that the message was not altered and that it was sent by the named sender.

Understanding PeopleSoft Encryption Technology

PeopleSoft Encryption Technology provides a way for you to secure critical PeopleSoft data and communicate securely with other businesses. It enables you to extend and improve cryptographic support for your application data, giving you strong cryptography with the flexibility to change and grow, by incrementally acquiring stronger and more diverse algorithms for encrypting data.

You can use PeopleSoft Encryption Technology to secure data in flat files or in database tables.

PeopleSoft Encryption Technology Features

You can encrypt any data used in your application by invoking PeopleCode to apply your preferred encryption algorithms. You can obtain these algorithms from various vendors' cryptographic libraries, using the capabilities you want from each library.

The features of PeopleSoft Encryption Technology include:

- Access to a robust set of algorithms (symmetric and asymmetric ciphers, password-based encryption, hashes, MACs, signatures, enveloping, encoding, and writing/processing secured messages).
- The ability to encrypt, decrypt, sign, and verify fields in a database.
- The ability to encrypt, decrypt, sign, and verify external files.
- A secure keystore for encryption keys of widely varying types.
- The ability to convert data from one encryption scheme to another.

PeopleSoft Encryption Technology Concepts

This section describes key PeopleSoft Encryption Technology concepts.

Term	***Definition***
Encryption Algorithm	An encryption algorithm encrypts and decrypts data. As described in the previous sections of this documentation, PeopleSoft supports symmetric and asymmetric encryption algorithms.
Encryption chain	An encryption chain is a sequence of encryption algorithms.

Term	*Definition*
Encryption Profile	An encryption profile is a specific implementation of an encryption chain. When you create an encryption profile definition, you review the algorithm chain to identify all the algorithms and parameters that are required for the task. You must supply values for all of the parameters for the encryption profile to be viable for use. The design of the encryption profile allows you to reuse algorithms across many different encryption chain definitions. And you can implement the encryption chain definitions in many different encryption profiles, with each profile having its own distinct set of parameter values.
Encryption Algorithm Parameters	Some encryption algorithms may require input parameters. These input parameters may come from keysets or may be entered directly into the encryption profile definition.
Keyset, Keyset ID, and Keyset Value	A keyset is a definition that associates a keystore certificate alias or private key to an encryption algorithm. The definition is identified by a user-defined *keyset ID*. The *keyset value* is the certificate alias or private key defined. Some encryption algorithms may require a keyset ID as an input parameter. At runtime the keyset ID is used to get the keyset value that is used in the algorithm. A keyset can also be a SYMMETRIC KEY value

PeopleSoft Encryption Technology Development

The functional elements of PeopleSoft Encryption Technology are:

- A DLL for each supported encryption library, which uses C glue code to convert each cryptographic library's API into a unified plug-in with an API accessible from PeopleCode.
- A universal keystore that handles all forms of encryption keys, protected with row-level security.
- A sequence, or chain, of algorithms that you define for a specific type of encryption task.

  These algorithms are applied in turn to transform data from its original form into a desired final form.
- An encryption profile, which you define as an instance of an algorithm chain, applicable to a specific encryption task.
- The PeopleCode crypt class for accessing the algorithm chains that you define.

To develop and use an encryption profile:

1. Obtain an encryption library.

The current release of PeopleTools includes the *OpenSSL* encryption library.

2. Develop API glue code to access the encryption library's algorithms.

   PeopleTools includes glue code already developed to support the delivered OpenSSL encryption library, as well as glue code to support the *PGP* encryption library.

   The glue code combines with each library to create a plug-in accessible from PeopleCode. The plug-in can be an independent DLL file, or it can be incorporated into the encryption library file, which is the case with the delivered OpenSSL library.

   You can develop glue code to produce plug-in wrappers for other encryption libraries of your choice. The plug-ins make their APIs accessible to PeopleCode, and the new algorithms become as easily available as the delivered algorithms. You can find development information and examples of glue source code in *PS_HOME*\src\pspetssl.

3. Load the encryption library's algorithms into the PET database, generate accompanying encryption keys, and insert them into the PET keystore.

4. Define a chain of algorithms by selecting from the algorithms in the database.

   Because all algorithms are accessed from PeopleCode, you can combine algorithms from different libraries regardless of their source.

5. Define an encryption profile, which is an instance of an algorithm chain applicable to a specific encryption task.

   With an encryption profile you can apply parameter values that differ from the default values.

6. Test the encryption profile using the Test Encryption Profile page.

7. Write PeopleCode to invoke the encryption profile.

   With the delivered glue code, you can take advantage of the capabilities of these libraries through a single PeopleCode object. The PeopleCode crypt class provides an interface into all algorithms loaded from the underlying encryption libraries.

Encryption Algorithm Libraries

This section describes encryption algorithm libraries and those libraries supported by PeopleSoft.

Algorithm Libraries

An algorithm library is computer code provided from a vendor that provides access to a collection of encryption algorithms. As an example, PGP and OpenSSL are algorithm libraries. These vendor algorithms are stored in tables within the PeopleSoft system and become part of the organized collection of PET data (or PET database).

Accessing Algorithm Libraries

PeopleSoft delivers the open source OpenSSL library as well as the glue code to interact with the library.

For other third-party libraries, such as PGP, you must separately obtain a license and install the product.

Access to the delivered OpenSSL library is obtained through the PeopleSoft Internet Architecture using the pages in the Encryption component (ALGORITHM_PFRL). These pages are discussed in later sections of this documentation.

Algorithm Library Glue Code

PeopleSoft delivers the glue code to interact with OpenSSL and PGP libraries. The location of the glue code is one of the following:

```
<PS_HOME>\src\pspetinc
<PS_HOME>\src\pspetssl
```

The OpenSSL glue code has been tested on all supported PeopleSoft platforms with PKCS7 and AES. The glue code to interact with the PGP library has been tested on the Microsoft Windows platform only.

For other third-party libraries you must develop the glue code, using the PeopleSoft glue code as a guide.

PGP Library Considerations

If you license the PGP encryption library, you must ensure that its installed location is included in the paths used by both the application server and PeopleSoft Process Scheduler, as follows:

- Using the PSADMIN utility, add the full installed path of the PGP SDK to the *Add to PATH* parameter.

  See “Domain Settings” (System and Server Administration).

- In the Oracle Tuxedo Settings section of the Process Scheduler configuration file, add the full installed path of the PGP SDK to the *Add to PATH* parameter.

  See “Understanding the PeopleSoft Process Scheduler Configuration File” (Process Scheduler).

Note: The path added must be the directory which contains the .dll and .lib files. There can be no intermediate subdirectory between the path setting and these files.
PGP operations are supported only on platforms where the PGP SDK is supported: Microsoft Windows, Oracle Solaris, and Red Hat Linux. Note that the glue code provided by PeopleTools is tested on Microsoft Windows only.

Understanding Documentation for PeopleSoft Encryption Technology

This documentation discusses how to use an encryption library for which glue code has already been developed and compiled, such as OpenSSL and PGP.

Understanding the Supported Algorithms

This section discusses the minimum set of encryption algorithms supported by PeopleTools. Support for these algorithms is provided through the OpenSSL and PGP plug-ins, and internally through the PeopleCode crypt class.

Note: You use the crypt class to open an encryption profile, which comprises the chain of algorithms that you want to invoke. The crypt class then invokes the algorithms and applies their parameters as specified by the profile.

Some algorithms have accompanying parameters, some with default values, which are stored along with the algorithms in the database. You supply appropriate parameter values in an encryption profile, and they are used when the algorithm is invoked.

Each algorithm returns data appropriate to its purpose, using properties provided by the crypt class. The Result property is used to make output data available from algorithms that produce or transform data by encoding, decoding, encryption, decryption, generating hash values, or generating signatures. The Verified property conveys the success or failure of algorithms that verify the input data.

Related Links

Defining Encryption Profiles

"Understanding the Crypt Class" (PeopleCode API Reference)

Internal Algorithms

Support for the following algorithms is provided by the PeopleCode crypt class. They are automatically available for inclusion in your algorithm chains.

Algorithm	***Description***
PSUnicodeToAscii	Convert Unicode text to ASCII.
PSAsciiToUnicode	Convert ASCII text to Unicode.
PSHexEncode	Convert octets (bytes) into ASCII hex nibbles.
PSHexDecode	Convert ASCII hex nibbles (with a leading 0x) into binary octets (bytes).
PSUnicodeToAscii_Generic_ENC	Convert Unicode text to ASCII **Note:** Use when encrypting data across multiple platforms where one platform is Db2 z/OS. This algorithm functions the same as PSUnicodeToAscii on all platforms other than Db2 z/OS.
PSAsciiToUnicode_Generic_DEC	Convert ASCII text to Unicode **Note:** Use when performing cross-platform decryption where one platform is Db2 z/OS. This algorithm functions the same as PSAsciiToUnicode on all platforms other than Db2 z/OS.

OpenSSL Algorithms

This section describes the algorithms supported by the OpenSSL plug-in, including encoding algorithms, hashing algorithms, symmetric encryption algorithms, digital signature algorithms, and the individual secure messaging algorithms. These algorithms are available when you load the OpenSSL encryption library into the PET database.

Encoding Algorithms

Following are the supported OpenSSL encoding algorithms.

Algorithm	***Description***
base64_encode	Encode data in base64 format.
base64_decode	Decode data from base64 format.

Hashing Algorithms

Following are the supported OpenSSL hashing algorithms.

Algorithm	***Description***
md2_generate	Generate an MD2 hash value from the input data.
md4_generate	Generate an MD4 hash value.
md5_generate	Generate an MD5 hash value.
sha1_generate	Generate an SHA1 hash value.
ripemd160_generate	Generate a RIPEMD160 hash value.
hmac_sha1_generate	Generate a hash message authentication code SHA1 hash value.

HMAC encryption takes a SECRETKEY parameter. The parameter is not required, but if supplied it must be defined in the keyset (similar to SYMMETRIC_KEY for other algorithms). The value specified must begin with 0x. The value should be at least 16 Hex characters (0-9, A-F). It should be random but its secrecy isn't critical. For example: 0x0102030405060708. The longer the value the more secure the hash output.

See Defining Algorithm Keysets.

Symmetric Encryption Algorithms

The following tables describe symmetric encryption algorithms that implement triple Data Encryption Standard (DES) encryption with various key sizes and modes.

This table lists triple DES symmetric encryption algorithms that use a key size of 112 bits.

Algorithm Name	***Description***
3des_ks112_ecb_encrypt	Encrypt data using a key size of 112 bits, in electronic code book mode.
3des_ks112_ecb_decrypt	Decrypt data using a key size of 112 bits, in electronic code book mode.
3des_ks112_cbc_encrypt	Encrypt data using a key size of 112 bits, in cipher block chaining mode.
3des_ks112_cbc_decrypt	Decrypt data using a key size of 112 bits, in cipher block chaining mode.
3des_ks112_cfb_encrypt	Encrypt data using a key size of 112 bits, in cipher feedback mode.
3des_ks112_cfb_decrypt	Decrypt data using a key size of 112 bits, in cipher feedback mode.
3des_ks112_ofb_encrypt	Encrypt data using a key size of 112 bits, in output feedback mode.
3des_ks112_ofb_decrypt	Decrypt data using a key size of 112 bits, in output feedback mode.

This table lists triple DES symmetric encryption algorithms that use a key size of 168 bits.

Algorithm Name	***Description***
3des_ks168_ecb_encrypt	Encrypt data using a key size of 168 bits, in electronic code book mode.
3des_ks168_ecb_decrypt	Decrypt data using a key size of 168 bits, in electronic code book mode.

Algorithm Name	*Description*
3des_ks168_cbc_encrypt	Encrypt data using a key size of 168 bits, in cipher block chaining mode.
3des_ks168_cbc_decrypt	Decrypt data using a key size of 168 bits, in cipher block chaining mode.
3des_ks168_cfb_encrypt	Encrypt data using a key size of 168 bits, in cipher feedback mode.
3des_ks168_cfb_decrypt	Decrypt data using a key size of 168 bits, in cipher feedback mode.
3des_ks168_ofb_encrypt	Encrypt data using a key size of 168 bits, in output feedback mode.
3des_ks168_ofb_decrypt	Decrypt data using a key size of 168 bits, in output feedback mode.

The following tables describes Open SSL symmetric encryption algorithms that implement Advanced Encryption Security (AES) encryption.

This table lists AES encryption algorithms that use a key size of 128 bits:

Algorithm Name	*Description*
aes_ks128_cbc_decrypt	Decrypt data using a key size of 128 bits, in cipher block chaining mode.
aes_ks128_cbc_encrypt	Encrypt data using a key size of 128 bits, in cipher block chaining mode.
aes_ks128_cfb_decrypt	Decrypt data using a key size of 128 bits, in cipher feedback mode.
aes_ks128_cfb_encrypt	Encrypt data using a key size of 128 bits, in cipher feedback mode.
aes_ks128_ecb_decrypt	Decrypt data using a key size of 128 bits, in electronic code book mode.

Algorithm Name	*Description*
aes_ks128_ecb_encrypt	Encrypt data using a key size of 128 bits, in electronic code book mode.
aes_ks128_ofb_decrypt	Decrypt data using a key size of 128 bits, in output feedback mode.
aes_ks128_ofb_encrypt	Encrypt data using a key size of 128 bits, in output feedback mode.

The following table describes AES encryption algorithms that use a key size of 192 bits:

Algorithm Name	*Description*
aes_ks192_cbc_decrypt	Decrypt data using a key size of 192 bits, in cipher block chaining mode.
aes_ks192_cbc_encrypt	Encrypt data using a key size of 192 bits, in cipher block chaining mode.
aes_ks192_cfb_decrypt	Decrypt data using a key size of 192 bits, in cipher feedback mode.
aes_ks192_cfb_encrypt	Encrypt data using a key size of 192 bits, in cipher feedback mode.
aes_ks192_ecb_decrypt	Decrypt data using a key size of 192 bits, in electronic code book mode.
aes_ks192_ecb_encrypt	Encrypt data using a key size of 192 bits, in electronic code book mode.
aes_ks192_ofb_decrypt	Decrypt data using a key size of 192 bits, in output feedback mode.
aes_ks1928_ofb_encrypt	Encrypt data using a key size of 192 bits, in output feedback mode.

The following table describes AES encryption algorithms that use a key size of 256 bits:

Algorithm Name	*Description*
aes_ks256_cbc_decrypt	Decrypt data using a key size of 256 bits, in cipher block chaining mode.
aes_ks256_cbc_encrypt	Encrypt data using a key size of 256 bits, in cipher block chaining mode.
aes_ks256_cfb_decrypt	Decrypt data using a key size of 256 bits, in cipher feedback mode.
aes_ks256_cfb_encrypt	Encrypt data using a key size of 256 bits, in cipher feedback mode.
aes_ks256_ecb_decrypt	Decrypt data using a key size of 256 bits, in electronic code book mode.
aes_ks256_ecb_encrypt	Encrypt data using a key size of 256 bits, in electronic code book mode.
aes_ks256_ofb_decrypt	Decrypt data using a key size of 256 bits, in output feedback mode.
aes_ks2568_ofb_encrypt	Encrypt data using a key size of 256 bits, in output feedback mode.

Most of these algorithms use the same two parameters:

- *IV* (Initialization Vector)

  This parameter isn't used by the listed ECB mode algorithms. When defining an encryption profile, specify the IV as a hex encoded value, which is used to alter the first plaintext block of data before it's encrypted. This value serves as an encryption seed value, which must be applied for both encryption and decryption. The value must be the length of the cipher's blocksize — 8 bytes for 3 DES and 16 bytes for AES (that is, 32 hex). It should be random but its secrecy isn't critical. For example: *000102030405060708090A0B0C0D0E0F*.

  See Defining Encryption Profiles.

- *SYMMETRIC_KEY*

  Specify as a string the keyset ID of the symmetric encryption key to be used with this algorithm. This parameter must identify a key that's stored in the PET keyset database.

  See Defining Algorithm Keysets.

  You can use any third-party key generation utility capable of producing hex encoded keys of the required length for the algorithm that you are using. However, using a key generation utility is not a

requirement. You can build a hex encoded string manually by stringing together any combination of the numbers (0-9) and letters (A-F) to the appropriate length.

The appropriate length of the key to store in the keyset database is determined by the Key Size (..._ksnnn_...) portion of the algorithm name. Divide the nnn (Bit length) portion by 4 to get the number of hex chars (0-9, A-F) to enter following "0x". So an algorithm with ..._ks256_... in the algorithm name, would need a key containing "0x" + 64 hex chars.

Note: All algorithm chains that use symmetric and authenticated encryption algorithms must include either the base64_encode or PSHexEncode algorithm as a step in the encryption algorithm chain. All algorithm chains that use symmetric and authenticated decryption algorithms must include the corresponding base64_decode or PSHexDecode algorithm as a step in the decryption algorithm chain.

Authenticated Encryption Algorithms

The following table describes authenticated encryption algorithms that use a key size of 128 bits:

Algorithm Name	***Description***
aes_ks128_ccm_decrypt	Decrypt data using a key size of 128 bits, in CCM mode (Counter with CBC-MAC).
aes_ks128_ccm_encrypt	Encrypt data using a key size of 128 bits, in CCM mode (Counter with CBC-MAC).
aes_ks128_gcm_decrypt	Decrypt data using a key size of 128 bits, in Galois/Counter mode.
aes_ks128_gcm_encrypt	Encrypt data using a key size of 128 bits, in Galois/Counter mode.

The following table describes authenticated encryption algorithms that use a key size of 192 bits:

Algorithm Name	***Description***
aes_ks192_ccm_decrypt	Decrypt data using a key size of 192 bits, in CCM mode (Counter with CBC-MAC).
aes_ks192_ccm_encrypt	Encrypt data using a key size of 192 bits, in CCM mode (Counter with CBC-MAC).
aes_ks192_gcm_decrypt	Decrypt data using a key size of 192 bits, in Galois/Counter mode.

Algorithm Name	*Description*
aes_ks192_gcm_encrypt	Encrypt data using a key size of 192 bits, in Galois/Counter mode.

The following table describes authenticated encryption algorithms that use a key size of 256 bits:

Algorithm Name	*Description*
aes_ks256_ccm_decrypt	Decrypt data using a key size of 256 bits, in CCM mode (Counter with CBC-MAC).
aes_ks256_ccm_encrypt	Encrypt data using a key size of 256 bits, in CCM mode (Counter with CBC-MAC).
aes_ks256_gcm_decrypt	Decrypt data using a key size of 256 bits, in Galois/Counter mode.
aes_ks256_gcm_encrypt	Encrypt data using a key size of 256 bits, in Galois/Counter mode.

Authenticated encryption algorithms are essentially symmetric-key algorithms with an additional authentication component. The authentication is implemented in two parts. Encryption allows the specification of additional authentication data (clear text, non-secret), and will output an authentication tag (non-secret) in addition to the encrypted text. Decryption will require any additional authentication data and the authentication tag as input parameters.

All of the authenticated encryption algorithms use these two symmetric-key parameters:

- *IV* (Initialization Vector)

  This parameter is required for CCM and GCM algorithms. It cannot be blank.

  When defining an encryption profile, specify the IV as a hex encoded value, which is used to alter the first plaintext block of data before it is encrypted. This value serves as an encryption seed value, which must be applied for both encryption and decryption.

  See Defining Encryption Profiles.

  For CCM algorithms the value specified must begin with 0x, which must be followed by at least six hex characters (that is, 0x123456). It can be no more than 26 hex characters in addition to 0x (that is, 0x112233445566778899001122 33).

  For GCM algorithms the value specified must begin with 0x, which must be followed by at least two hex characters (that is, 0x12).

- *SYMMETRIC_KEY*

Specify as a string the keyset ID of the symmetric encryption key to be used with this algorithm. This parameter must identify a key that's stored in the PET keyset database.

See Defining Algorithm Keysets.

You can use any third-party key generation utility capable of producing hex encoded keys of the required length for the algorithm that you are using. However, using a key generation utility is not a requirement. You can build a hex encoded string manually by stringing together any combination of the numbers (0-9) and letters (A-F) to the appropriate length.

The appropriate length of the key to store in the keyset database is determined by the Key Size (..._ksnnn_...) portion of the algorithm name. Divide the nnn (Bit length) portion by 4 to get the number of hex chars (0-9, A-F) to enter following "0x". So an algorithm with ..._ks256_... in the algorithm name, would need a key containing "0x" + 64 hex chars.

In addition to the IV and SYMMETRIC_KEY parameters, the encrypt algorithms also use the *AAD* (Additional Authenticated Data) parameter. Specify data as clear text. The input data is authenticated but not encrypted. The output from the encrypt algorithms is accessed with the AuthTag property added to the PeopleCode Crypt object.

In addition to the IV and SYMMETRIC_KEY parameters, the decrypt algorithms also use these input parameters:

- *AAD* (Additional Authenticated Data)

  When defining an encryption profile, specify AAD with the same value you specified for the encrypt algorithm.

  See Defining Encryption Profiles.

- *AUTHTAG* (Authorization Tag)

  When defining an encryption profile, specify AUTHTAG with the string that was retrieved, through the AuthTag property, from the PeopleCode Crypt object during encryption. Add **0x** (zero x) to the beginning of the string before supplying it as input to the decrypt algorithm.

  See Defining Encryption Profiles.

  See “Crypt Class Properties” (PeopleCode API Reference).

Digital Signature Handling Algorithms

Following are the supported OpenSSL algorithms for generating signatures.

Algorithm Name	***Description***
rsa_md5_sign	Generate an RSA signature using an MD5 hash.
rsa_sha1_sign	Generate an RSA signature using an SHA1 hash.
dsa_sha1_sign	Generate a DSA signature.

The signing algorithms all use the same parameters:

- *SIGNERPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the signer's private key in the PET keyset database. The keyset entered value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*, depending on the algorithm.

  See Defining Algorithm Keysets.

- *SIGNERPKPASSPHRASE*

  When defining an encryption profile, specify SIGNERPKPASSPHRASE with the pass phrase used to decrypt and unlock the signer's private key. This parameter's value is the actual pass phrase.

  See Defining Encryption Profiles.

Note: The output of these algorithms must be a hex encoded signature if it is going to be used as the SIGNATURE parameter value for the Verify routine. To generate a Hex value a PSHexEncode algorithm must be the second to the last step in the chain.

Following are the supported OpenSSL algorithms for verifying signatures.

Algorithm Name	***Description***
rsa_md5_verify	Verify an RSA signature based on an MD5 hash.
rsa_sha1_verify	Verify an RSA signature based on an SHA1 hash.
dsa_sha1_verify	Verify a DSA-hashed signature.

The verifying algorithms all use the same parameters:

- *SIGNERCERT*

  Specify, as a string, the keyset ID that represents the signer's certificate in the PET keyset database. The actual certificate stored in the keyset database is an X.509 certificate. The keyset entered should begin "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

  Note: The API implementation of the rsa_sha1_verify algorithm requires that the Public Key be certified.

- *SIGNATURE*

  When defining an encryption profile, specify SIGNATURE as a string with the hex encoded signature that's delivered with the input data or that's returned as the result of invoking a signing algorithm.

  See Defining Encryption Profiles.

Note: The system expects all hex encoded values to begin with *0x*. If the hex encoded signature value does not begin with these two characters, you must manually prepend *0x* to it or the signature will be invalid.

Secure Messaging — pkcs7_encrypted_decrypt

The pkcs7_encrypted_decrypt algorithm decrypts an encrypted PKCS7 message. The parameters are:

- *RECIPIENTCERT*

  Specify, as a string, the keyset ID that represents the recipient's certificate in the PET keyset database. The keyset entered value in the keyset database should begin with the line "-----BEGIN CERTIFICATE-----"

  See Defining Algorithm Keysets.

- *RECIPIENTPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the recipient's private key in the PET keyset database. The keyset entered value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*.

  See Defining Algorithm Keysets.

- *RECIPIENTPKPASSPHRASE*

  When defining an encryption profile, specify RECIPIENTPKPASSPHRASE with the pass phrase used to decrypt and unlock the recipient's private key. This parameter's value is the actual pass phrase.

  See Defining Encryption Profiles.

Secure Messaging — pkcs7_encrypted_encrypt

The pkcs7_encrypted_encrypt algorithm generates an encrypted PKCS7 message.

This algorithm has one parameter: *SIGNERCERT*, which is the keyset ID that represents the signer's X.509 certificate in the PET keyset database. The keyset entered value should begin with the line "-----BEGIN CERTIFICATE-----".

See Defining Algorithm Keysets.

Secure Messaging — pkcs7_signandencrypt_decryptandverify

The pkcs7_signandencrypt_decryptandverify algorithm decrypts and verifies an encrypted PKCS7 message. The parameters are:

- *SIGNERCERT*

  Specify, as a string, the keyset ID that represents the signer's certificate in the PET keyset database. The keyset entered value stored in the keyset database is an X.509 certificate. Its value should begin "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

- *RECIPIENTCERT*

  Specify, as a string, the keyset ID that represents the recipient's certificate in the PET keyset database. The keyset entered value in the keyset database should begin with the line "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

- *RECIPIENTPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the recipient's private key in the PET keyset database. The keyset entered value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*.

  See Defining Algorithm Keysets.

- *RECIPIENTPKPASSPHRASE*

  When defining an encryption profile, specify RECIPIENTPKPASSPHRASE with the pass phrase used to decrypt and unlock the recipient's private key. This parameter's value is the actual pass phrase.

  See Defining Encryption Profiles.

Secure Messaging — pkcs7_signandencrypt_signandencrypt

The pkcs7_signandencrypt_signandencrypt algorithm generates a signed and encrypted PKCS7 message. The parameters are:

- *SIGNERCERT*

  Specify, as a string, the keyset ID that represents the signer's certificate in the PET keyset database. The keyset entered value stored in the keyset database is an X.509 certificate. Its value should begin "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

- *SIGNERPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the signer's private key in the PET keyset database. The keyset entered value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*.

  See Defining Algorithm Keysets.

- *SIGNERPKPASSPHRASE*

  When defining an encryption profile, specify SIGNERPKPASSPHRASE with the pass phrase used to decrypt and unlock the signer's private key. This parameter's value is the actual pass phrase.

  See Defining Encryption Profiles.

- *RECIPIENT*

  Specify, as a string, the keyset ID that represents the recipient's certificate in the PET keyset database. The keyset entered value stored in the keyset database is an X.509 certificate. Its value should begin "-----BEGIN CERTIFICATE-----".

See Defining Algorithm Keysets.

- *SYMMETRIC_ALGORITHM*

  When defining an encryption profile, specify SYMMETRIC_ALGORITHM with the name of the symmetric algorithm used for content encryption. This must be a symmetric encryption algorithm supported by an encryption plug-in.

  See Defining Encryption Profiles.

  See the “Symmetric Encryption Algorithms” section earlier in this topic for more information.

Secure Messaging — pkcs7_signed_sign

The pkcs7_signed_sign algorithm generates a signed PKCS7 message. The parameters are:

- *SIGNERCERT*

  Specify, as a string, the keyset ID that represents the signer's certificate in the PET keyset database. The keyset entered value stored in the keyset database is an X.509 certificate.The Keyset entered value should begin "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

- *SIGNERPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the signer's private key in the PET keyset database. The keyset entered value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*.

  See Defining Algorithm Keysets.

- *SIGNERPKPASSPHRASE*

  When defining an encryption profile, specify SIGNERPKPASSPHRASE with the pass phrase used to decrypt and unlock the signer's private key. This parameter's value is the actual pass phrase.

  See Defining Encryption Profiles.

Secure Messaging — pkcs7_signed_verify

The pkcs7_signed_verify algorithm verifies a signed PKCS7 message. The parameters are:

- *RECIPIENT*

  Specify, as a string, the keyset ID that represents the recipient's certificate in the PET keyset database. The actual certificate stored in the keyset database is an X.509 certificate.The keyset entered value should begin "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

- *SYMMETRIC_ALGORITHM*

When defining an encryption profile, specify SYMMETRIC_ALGORITHM with the name of the symmetric algorithm used for content encryption. This must be a symmetric encryption algorithm supported by an encryption plug-in.

See Defining Encryption Profiles.

See the “Symmetric Encryption Algorithms” section earlier in this topic for more information.

Emails — smime_encrypted_decrypt

The smime_encrypted_decrypt algorithm decrypts an encrypted email message. The parameters are:

- *RECIPIENTCERT*

  Specify, as a string, the keyset ID that represents the recipient's certificate in the PET keyset database. The keyset entered value in the keyset database should begin with the line "-----BEGIN CERTIFICATE-----"

  See Defining Algorithm Keysets.

- *RECIPIENTPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the recipient's private key in the PET keyset database. The keyset entered value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*.

  See Defining Algorithm Keysets.

- *RECIPIENTPKPASSPHRASE*

  When defining an encryption profile, specify RECIPIENTPKPASSPHRASE with the pass phrase used to decrypt and unlock the recipient's private key. This parameter's value is the actual pass phrase.

  See Defining Encryption Profiles.

Emails — smime_encrypted_encrypt

The smime_encrypted_encrypt algorithm generates an encrypted email message.

This algorithm has one parameter: *SIGNERCERT*, which is the keyset ID that represents the signer's X.509 certificate in the PET keyset database. The keyset entered value should begin with the line "-----BEGIN CERTIFICATE-----".

See Defining Algorithm Keysets.

Emails — smime_signandencrypt_decryptandverify

The smime_signandencrypt_decryptandverify algorithm decrypts and verifies an encrypted email message. The parameters are:

- *SIGNERCERT*

  Specify, as a string, the keyset ID that represents the signer's certificate in the PET keyset database. The keyset entered value stored in the keyset database is an X.509 certificate. Its value should begin "-----BEGIN CERTIFICATE-----".

See Defining Algorithm Keysets.

- *RECIPIENTCERT*

  Specify, as a string, the keyset ID that represents the recipient's certificate in the PET keyset database. The keyset entered value in the keyset database should begin with the line "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

- *RECIPIENTPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the recipient's private key in the PET keyset database. The keyset entered value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*.

  See Defining Algorithm Keysets.

- *RECIPIENTPKPASSPHRASE*

  When defining an encryption profile, specify RECIPIENTPKPASSPHRASE with the pass phrase used to decrypt and unlock the recipient's private key. This parameter's value is the actual pass phrase.

  See Defining Encryption Profiles.

Emails — smime_signandencrypt_signandencrypt

The smime_signandencrypt_signandencrypt algorithm generates a signed and encrypted email message. The parameters are:

- *SIGNERCERT*

  Specify, as a string, the keyset ID that represents the signer's certificate in the PET keyset database. The keyset entered value stored in the keyset database is an X.509 certificate. Its value should begin "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

- *SIGNERPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the signer's private key in the PET keyset database. The keyset entered value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*.

  See Defining Algorithm Keysets.

- *SIGNERPKPASSPHRASE*

  When defining an encryption profile, specify SIGNERPKPASSPHRASE with the pass phrase used to decrypt and unlock the signer's private key. This parameter's value is the actual pass phrase.

  See Defining Encryption Profiles.

- *RECIPIENT*

Specify, as a string, the keyset ID that represents the recipient's certificate in the PET keyset database. The keyset entered value stored in the keyset database is an X.509 certificate. Its value should begin "-----BEGIN CERTIFICATE-----".

See Defining Algorithm Keysets.

- *SYMMETRIC_ALGORITHM*

  When defining an encryption profile, specify SYMMETRIC_ALGORITHM with the name of the symmetric algorithm used for content encryption. This must be a symmetric encryption algorithm supported by an encryption plug-in.

  See Defining Encryption Profiles.

  See the “Symmetric Encryption Algorithms” section earlier in this topic for more information.

Emails — smime_signed_sign

The smime_signed_sign algorithm generates a signed email message. The parameters are:

- *SIGNERCERT*

  Specify, as a string, the keyset ID that represents the signer's certificate in the PET keyset database. The keyset entered value stored in the keyset database is an X.509 certificate.The Keyset entered value should begin "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

- *SIGNERPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the signer's private key in the PET keyset database. The keyset entered value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*.

  See Defining Algorithm Keysets.

- *SIGNERPKPASSPHRASE*

  When defining an encryption profile, specify SIGNERPKPASSPHRASE with the pass phrase used to decrypt and unlock the signer's private key. This parameter's value is the actual pass phrase.

  See Defining Encryption Profiles.

Emails — smime_signed_verify

The smime_signed_verify algorithm verifies a signed email message. The parameters are:

- *RECIPIENT*

  Specify, as a string, the keyset ID that represents the recipient's certificate in the PET keyset database. The actual certificate stored in the keyset database is an X.509 certificate.The keyset entered value should begin "-----BEGIN CERTIFICATE-----".

  See Defining Algorithm Keysets.

- *SYMMETRIC_ALGORITHM*

  When defining an encryption profile, specify SYMMETRIC_ALGORITHM with the name of the symmetric algorithm used for content encryption. This must be a symmetric encryption algorithm supported by an encryption plug-in.

  See Defining Encryption Profiles.

  See the “Symmetric Encryption Algorithms” section earlier in this topic for more information.

PGP Algorithms

This section describes the secure messaging algorithms supported by the delivered PGP glue code. The messaging algorithms are available when you license the PGP encryption library from PGP Corporation, compile the glue code, and load the library into the PET database.

Note that the delivered PGP glue code has been tested on the Microsoft Windows environment only.

pgp_signed_sign

The pgp_signed_sign algorithm generates a signed PGP message. The parameters are:

- *SIGNERPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the signer's private key in the PET keyset database. The actual key value in the keyset database should begin "-----BEGIN PGP PRIVATE KEY BLOCK-----".

- *SIGNERKID*

  Specify, as a string, the PGP key ID for the signer's key. It's a hex encoded 32 bit value, for example, *0xAB01D6A5*. You can obtain this value from the PGP-based tool that created the key.

- *SIGNERPKPASSPHRASE*

  Specify the pass phrase used to decrypt the signer's private key. This parameter's value is the actual pass phrase.

- *CLEARSIGN*

  Specify a numeric value indicating whether the message is to be *clearsigned.* A clearsigned message should remain readable. If you specify a value of *1*, the message remains as is and a radix 64 armored signature block is appended to the message. If you specify a value of *0*, the signature block is appended and the entire message is radix 64 armored.

pgp_signed_verify

The pgp_signed_verify algorithm verifies a signed PGP message. The parameters are:

- *SIGNERPUBLICKEY*

  Specify the keyset ID that represents the signer's PGP Public key in the PET keyset database. The value stored in the keyset database should begin with the line "-----BEGIN PGP PUBLIC KEY BLOCK-----".

- *SIGNERKID*

  Specify, as a string, the PGP key ID for the signer's key. It's a hex encoded 32 bit value, for example, *0xAB01D6A5*. You can obtain this value from the PGP-based tool that created the key.

pgp_encrypted_encrypt

The pgp_encrypted_encrypt algorithm generates an encrypted PGP message. The parameters are:

- *RECIPIENTPUBLICKEY*

  Specify, as a string, the keyset ID that represents the recipient's public key in the PET keyset database. The actual key value in the keyset database should begin "-----BEGIN PGP PUBLIC KEY BLOCK-----".

- *RECIPIENTKID*

  Specify, as a string, the PGP key ID for the recipient's key. It's a hex encoded 32 bit value, for example *0xAB01D6A5*. You can obtain this value from the PGP-based tool that created the key.

pgp_encrypted_decrypt

The pgp_encrypted_decrypt algorithm decrypts an encrypted PGP message. The parameters are:

- *RECIPIENTPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the recipient's private key in the PET keyset database. The actual value in the keyset database should begin "-----BEGIN PGP PRIVATE KEY BLOCK-----".

- *RECIPIENTPKPASSPHRASE*

  Specify the pass phrase used to decrypt the recipient's private key. This parameter's value is the actual pass phrase.

- *RECIPIENTPUBLICKEY*

  Specify, as a string, the keyset ID that represents the recipient's public key in the PET keyset database. The actual value in the keyset database should begin "-----BEGIN PGP PUBLIC KEY BLOCK-----".

- *RECIPIENTKID*

  Specify, as a string, the PGP key ID for the recipient's key. It's a hex encoded 32 bit value, for example *0xAB01D6A5*. You can obtain this value from the PGP-based tool that created the key.

pgp_signedandencrypted_signandencrypt

The pgp_signedandencrypted_signandencrypt algorithm generates a signed and encrypted PGP message. The parameters are:

- *SIGNERPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the signer's private key in the PET keyset database. The actual key value in the keyset database should begin "-----BEGIN PGP PRIVATE KEY BLOCK-----".

- *SIGNERKID*

  Specify, as a string, the PGP key ID for the signer's key. It's a hex encoded 32 bit value, for example *0xAB01D6A5*. You can obtain this value from the PGP-based tool that created the key.

- *SIGNERPKPASSPHRASE*

  Specify the pass phrase used to decrypt the signer's private key. This parameter's value is the actual pass phrase.

- *RECIPIENTPUBLICKEY*

  Specify, as a string, the keyset ID that represents the recipient's public key in the PET keyset database. The actual value in the keyset database should begin "-----BEGIN PGP PUBLIC KEY BLOCK-----".

- *RECIPIENTKID*

  Specify, as a string, the PGP key ID for the recipient's key. It's a hex encoded 32 bit value, for example *0xAB01D6A5*. You can obtain this value from the PGP-based tool that created the key.

- *CLEARSIGN*

  Specify a numeric value indicating whether the message is to be *clearsigned*. A clearsigned message should remain readable. If you specify a value of *1*, the message remains as is and a radix 64 armored signature block is appended to the message. If you specify a value of *0*, the signature block is appended and the entire message is radix 64 armored.

pgp_signedandencrypted_decryptandverify

The pgp_signedandencrypted_decryptandverify algorithm decrypts and verifies a signed and encrypted PGP message. The parameters are as follows:

- *RECIPIENTPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the recipient's private key in the PET keyset database. The actual value in the keyset database should begin "-----BEGIN PGP PRIVATE KEY BLOCK-----".

- *RECIPIENTPKPASSPHRASE*

  Specify the pass phrase used to decrypt the recipient's private key. This parameter's value is the actual pass phrase.

- *RECIPIENTPUBLICKEY*

  Specify, as a string, the keyset ID that represents the recipient's public key in the PET keyset database. The actual value in the keyset database should begin "-----BEGIN PGP PUBLIC KEY BLOCK-----".

- *RECIPIENTKID*

  Specify, as a string, the PGP key ID for the recipient's key. It's a hex encoded 32 bit value, for example *0xAB01D6A5*. You can obtain this value from the PGP-based tool that created the key.

- *SIGNERPUBLICKEY*

Specify, as a string, the keyset ID that represents the signer's public key in the PET keyset database. The actual key value in the keyset database should begin "-----BEGIN PGP PUBLIC KEY BLOCK-----".

- *SIGNERKID*

  Specify, as a string, the PGP key ID for the signer's key. It's a hex encoded 32 bit value, for example *0xAB01D6A5*. You can obtain this value from the PGP-based tool that created the key.

Related Links

Loading Encryption Libraries

Algorithm Chain Considerations

Although you can select any sequence of algorithms to define a chain, many possible sequences don't work because the cumulative effect of the algorithms doesn't make any sense. You must define sequences of compatible algorithms.

To apply any of the supported algorithms for symmetric encryption, hashing, encoding, or secure messaging, the input data must be in ASCII text format. Because PeopleSoft stores data in Unicode format, the first algorithm in most chains must be PSUnicodeToAscii or PSUnicodeToAscii_Generic_ENC, and the last algorithm must be PSAsciiToUnicode or PSAsciiToUnicode_Generic_DEC.

All algorithm chains that use symmetric and authenticated encryption algorithms must include either the base64_encode or PSHexEncode algorithm as a step in the encryption algorithm chain. All algorithm chains that use symmetric and authenticated decryption algorithms must include the corresponding base64_decode or PSHexDecode algorithm as a step in the decryption algorithm chain.

Cross Platform Algorithm Chain Considerations

When encrypting and decrypting data across multiple platforms where Db2 z/OS is one of two or more platforms, the PSUnicodeToAscii_Generic_ENC algorithm must be the first algorithm in the encrypting algorithm chain. Conversely, PSAsciiToUnicode_Generic_DEC must be the last algorithm in the decrypting algorithm chain.

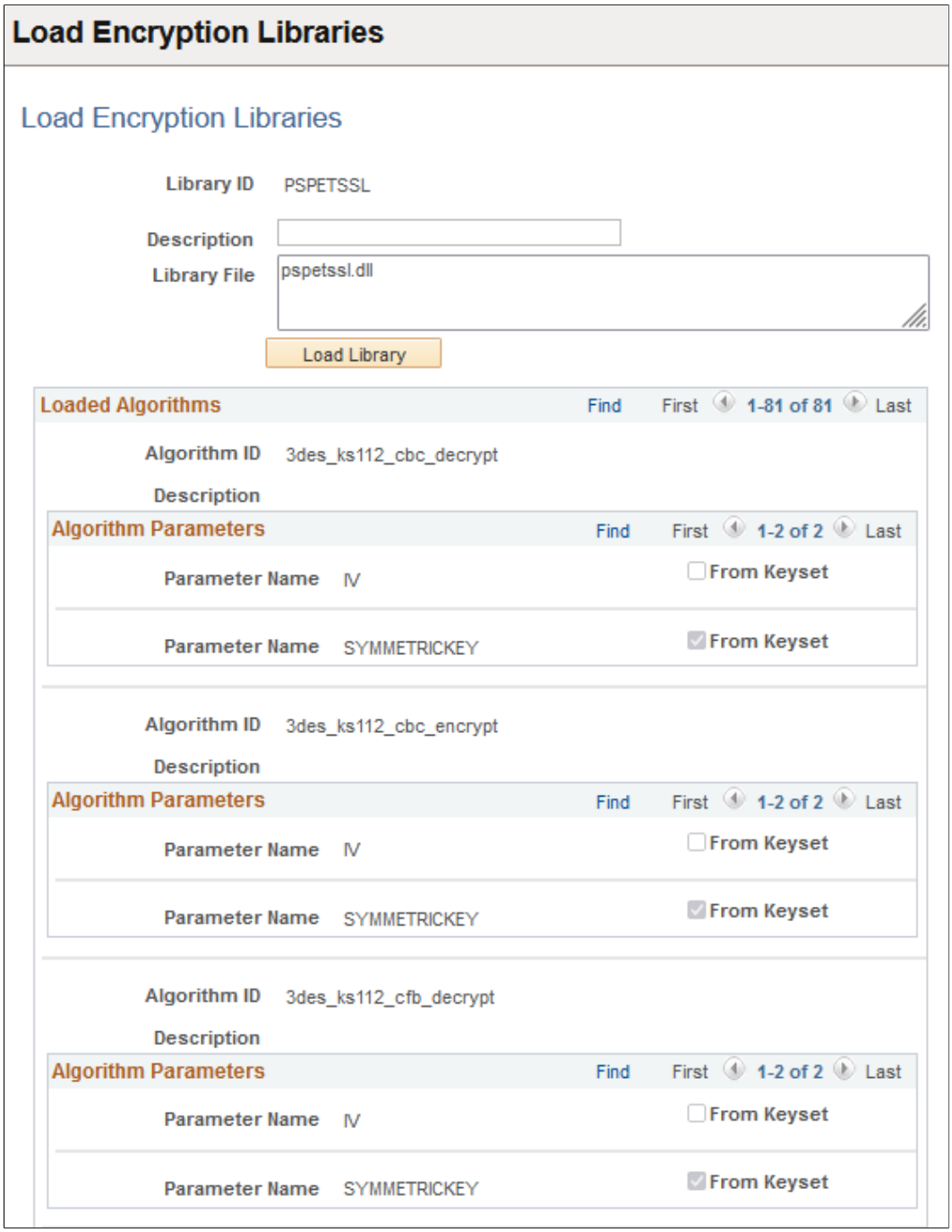
Note: If all participating encrypting and decrypting systems are on the Db2 z/OS platform, it is not necessary to use the generic algorithms. If none of the encrypting and decrypting systems in a cross platforms scenario are on the Db2 z/OS platform, the PSUnicodeToAscii_Generic_ENC algorithm functions exactly like the PSUnicodeToAscii algorithm and the PSAsciiToUnicode_Generic_DEC algorithm functions exactly like the PSAsciiToUnicode algorithm.

Important! If you modify current algorithm chains by replacing the PSUnicodeToAscii or the PSAsciiToUnicode algorithms with the PSUnicodeToAscii_Generic_ENC or the PSAsciiToUnicode_Generic_DEC algorithms, respectively, currently stored encrypted data on the Db2 z/OS DB must be unencrypted using the original decryption chain and reencrypted with the new encryption chain.

Loading Encryption Libraries

Access the Load Encryption Libraries page (**PeopleTools** > **Security** > **Encryption** > **Load Encryption Libraries**).

This example illustrates the fields and controls on the Load Encryption Libraries page.

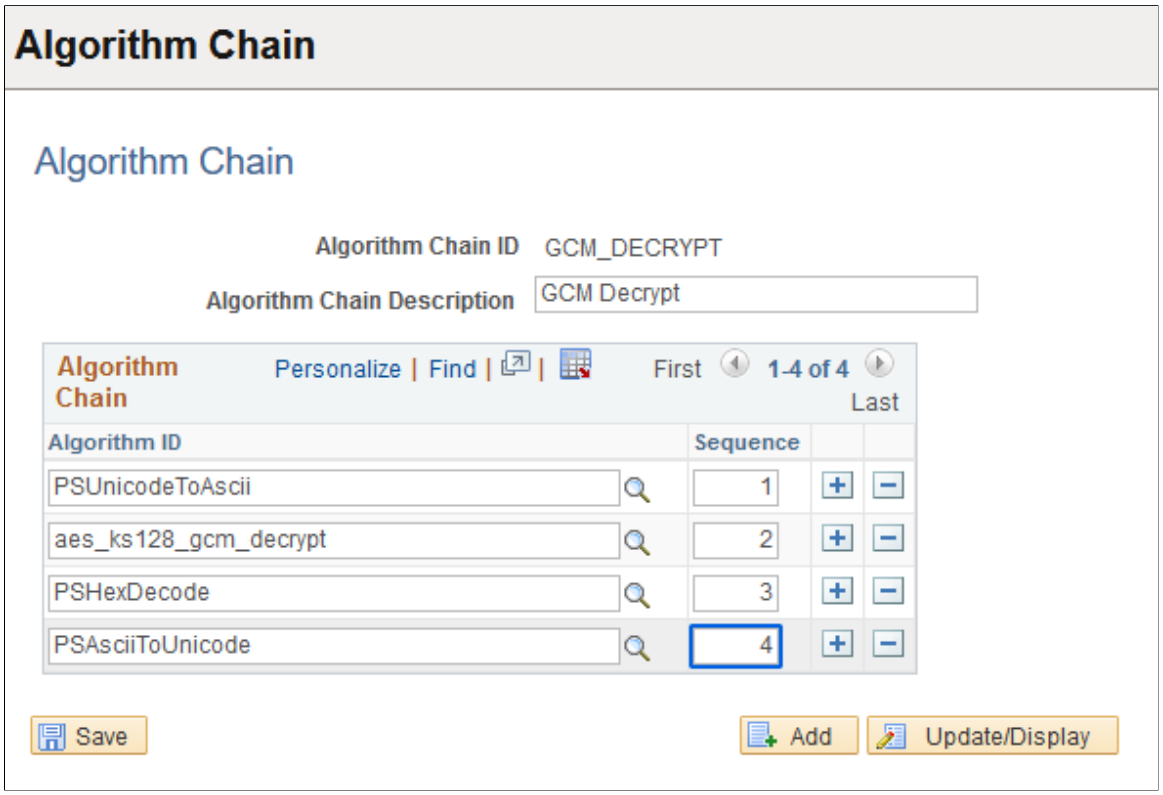


Field or Control	*Description*
Library File	Enter the filename of the selected encryption library for your operating system platform. The names of the delivered OpenSSL and PGP library files depend on the operating system platform where your application is installed. Following are the encryption library filenames for each supported platform: • Microsoft Windows OpenSSL: *pspetssl.dll* PGP: *pspetpgp.dll* • Red Hat Linux OpenSSL: *libpspetssl.so* • Sun Solaris OpenSSL: *libpspetssl.so* • HP Tru64 Unix OpenSSL: *libpspetssl.so* • HP-UX OpenSSL: *libpspetssl.sl* • IBM AIX OpenSSL: *libpspetssl.a*
Load Library	Click to load the specified encryption library. Each algorithm provided by the library appears in its own row with its algorithm ID. Its parameters each appear in a row, displaying the parameter's name and its default value. If the **From Keyset** check box is selected, the parameter represents an encryption key. The PeopleSoft Encryption Technology facility uses the parameter's value to access the encryption key from the PET keystore. **Important!** If the library you specify fails to load, you must sign out of your application, then shut down and restart the application server before signing back in. **Note:** You must create a valid openssl.cnf file *before* you load the PSPETSSL encryption libraries or the system removes the pkcs7 routines from the list of loaded encryption libraries. **Note:** When running multiple PS_HOME application server directories against the same database, each PS_HOME OpenSSL and PGP libraries and settings must be configured identically.

Defining Algorithm Chains

Access the Algorithm Chain page (**PeopleTools** > **Security** > **Encryption** > **Encryption Algorithm Chains**).

This example illustrates the fields and controls on the Algorithm Chain page.



Although you can select any sequence of algorithms to define a chain, many possible sequences don't work because the cumulative effect of the algorithms doesn't make any sense. You must define sequences of compatible algorithms.

To apply any of the supported algorithms for symmetric encryption, hashing, encoding, or secure messaging, the input data must be in ASCII text format.

Because PeopleSoft stores data in Unicode format, the first algorithm in *most* chains must be PSUnicodeToAscii when encrypting Unicode strings, and in *most* chains the last algorithm must be PSAsciiToUnicode when decrypting Unicode strings. However, chains may work better if you omit the PSUnicodeToAscii algorithm as the first step in the chain when encrypting non-Unicode strings, and omit the PSAsciiToUnicode algorithm as the last step in the chain when outputting non-Unicode strings from a decryption.

The following example shows an encryption string with PSUnicodeToAscii omitted as the first step:

```
3des_ks168_cbc_encrypt
Base64_encode
PSAsciiToUnicode
```

The following example shows a decryption string with PSAsciiToUnicode omitted as the last step:

```
PSUnicodeToAscii
```



```
Base64_decode
3des_ks168_cbc_decrypt
```

See Cross Platform Algorithm Chain Considerations.

To define an algorithm chain:

1. Open an existing algorithm chain or create a new one.
2. Select the algorithm IDs of the algorithms you want to use in your chain.

   Add a new row for each algorithm. The available algorithms depend on the encryption libraries you previously loaded. You can select the algorithms in any order.
3. Specify the operation sequence for your algorithm chain.

   Enter a number in the **Sequence** box for each algorithm. The lowest number designates the first algorithm, and the highest number designates the last. When you save the chain, the rows are resorted according to their sequence numbers.
4. Save your algorithm chain definition.

Delivered Algorithm Chains

PeopleSoft Encryption Technology includes the following predefined algorithm chains:

Algorithm Chain	***Algorithms***
3DES CBC B64 ENCRYPT	PSUnicodeToAscii 3des_ks168_cbc_encrypt base64_encode PSAsciiToUnicode
3DES CBC B64 DECRYPT	PSUnicodeToAscii base64_decode 3des_ks168_cbc_decrypt PSAsciiToUnicode
3DES CBC HEX ENCRYPT	PSUnicodeToAscii 3des_ks168_cbc_encrypt PSHexEncode PSAsciiToUnicode

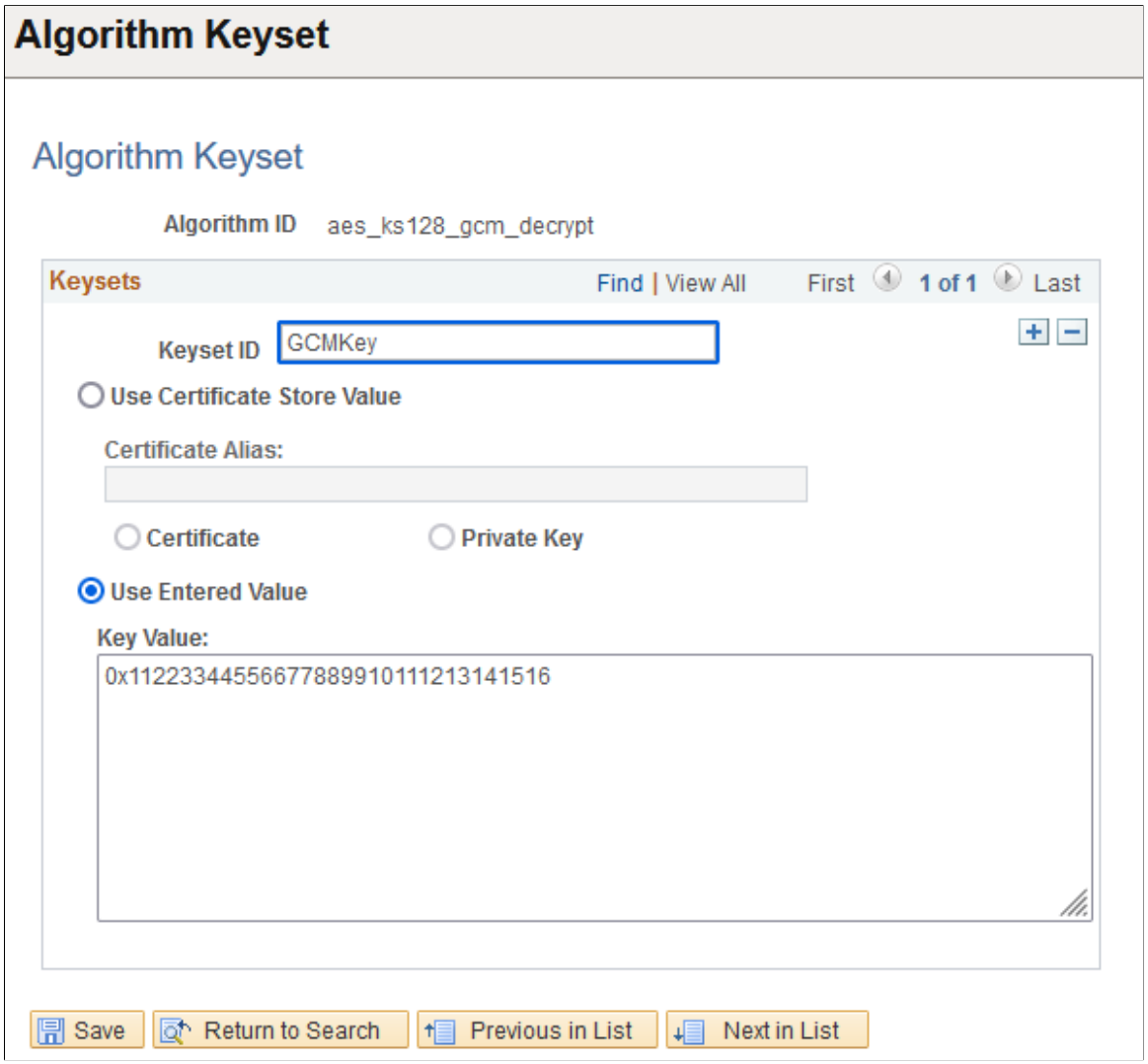
Algorithm Chain	*Algorithms*
3DES CBC HEX DECRYPT	PSUnicodeToAscii PSHexDecode 3des_ks168_cbc_decrypt PSAsciiToUnicode
PKCS7_ENCRYPTED	PSUnicodeToAscii pkcs7_encrypted_encrypt PSAsciiToUnicode
PKCS7_DECRYPTED	PSUnicodeToAscii pkcs7_encrypted_decrypt PSAsciiToUnicode
PKCS7_ENCRYPTED_SIGNED	PSUnicodeToAscii pkcs7_signedandencrypted_signandencrypt PSAsciiToUnicode
PKCS7_DECRYPTED_VERIFY	PSUnicodeToAscii pkcs7_signedandencrypted_decryptandverify PSAsciiToUnicode
PGP_ENCRYPTED	PSUnicodeToAscii pgp_encrypted_encrypt PSAsciiToUnicode
PGP_DECRYPTED	PSUnicodeToAscii pgp_encrypted_decrypt PSAsciiToUnicode

Algorithm Chain	*Algorithms*
PGP_ENCRYPTED_SIGNED	PSUnicodeToAscii pgp_signedandencrypted_signandencrypt PSAsciiToUnicode
PGP_DECRYPTED_VERIFY	PSUnicodeToAscii pgp_signedandencrypted_decryptandverify PSAsciiToUnicode
SMIME_DECRYPTED	PSUnicodeToAscii smime_encrypted_decrypt PSAsciiToUnicode
SMIME_DECRYPTED_VERIFY	PSUnicodeToAscii smime_signandencrypt_decryptandverify PSAsciiToUnicode
SMIME_ENCRYPTED	PSUnicodeToAscii smime_encrypted_encrypt PSAsciiToUnicode
SMIME_ENCRYPTED_SIGNED	PSUnicodeToAscii smime_signandencrypt_signandencrypt PSAsciiToUnicode
SMIME_VERIFY	PSUnicodeToAscii base64_decode smime_signed_verify PSAsciiToUnicode

Defining Algorithm Keysets

Access the Algorithm Keyset page (**PeopleTools** > **Security** > **Encryption** > **Encryption Algorithm Keysets**).

This example illustrates the fields and controls on the Algorithm Keyset page.



Choose an algorithm ID or description to view the keyset of any algorithm in the database.

Each row displays a key value. You can add, modify, or remove key values.

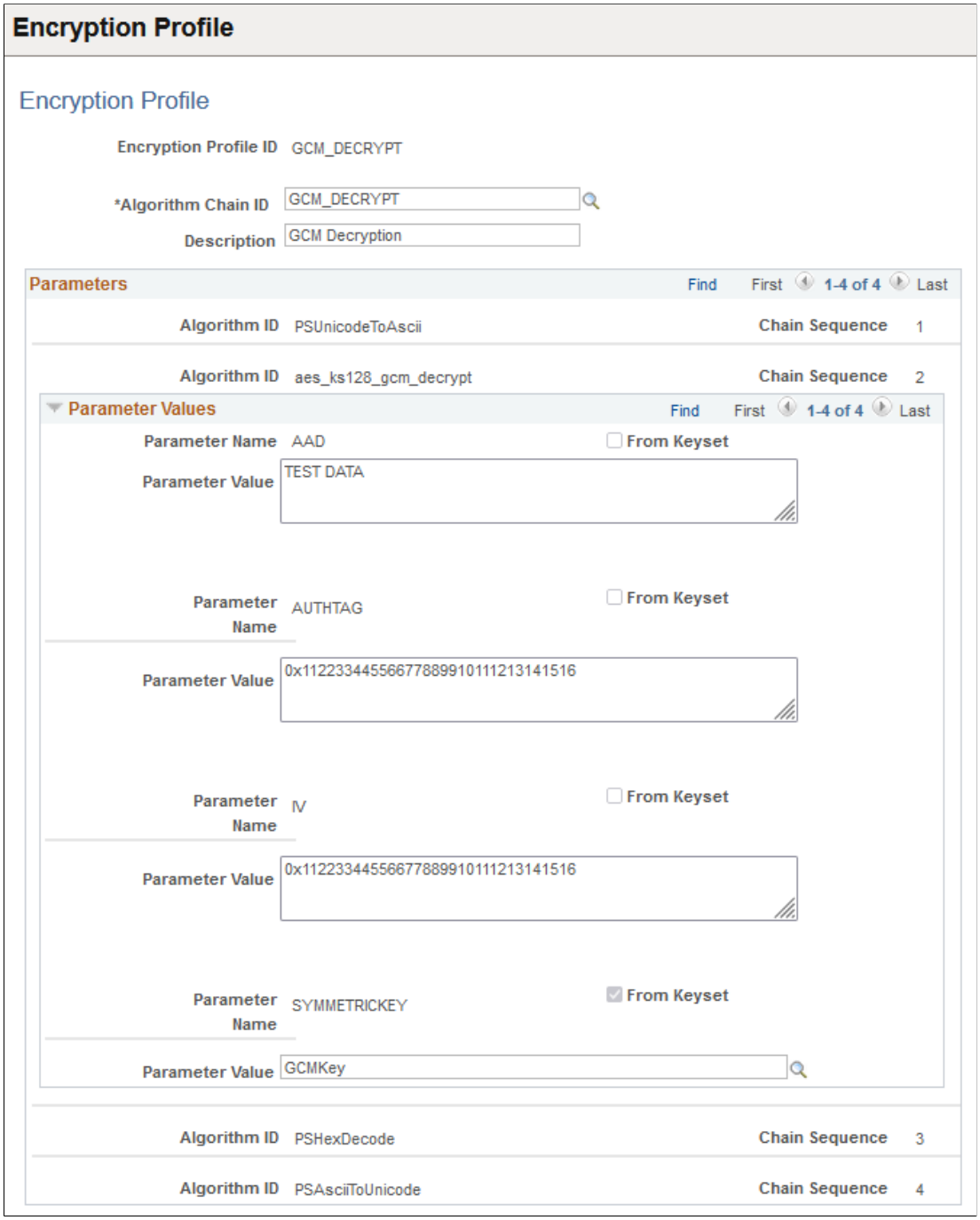
Field or Control	*Description*
Keyset ID	Enter a name for the key value in the current row. Each row must have a unique keyset ID for this algorithm.

Field or Control	*Description*
Use Certificate Store Value	This option enables you to take advantage of key values already stored in the PeopleSoft keystore. Select a certificate alias from the keystore, then indicate whether the alias represents a certificate or a private key. **Important!** The certificate must be a local node certificate. ***Warning!*** Certificates in the PeopleSoft keystore are in standard X.509 format, which is compatible for use with the internal and OpenSSL algorithms, but is *not* compatible with the PGP encryption library. If you're defining the keyset for a PGP algorithm, you must select the **Use Entered Value** radio button.
Use Entered Value	Select this option to use key values that aren't in the PeopleSoft keystore. Enter a key value that's formatted appropriately for the algorithm that you're configuring. This value will be entered into the PET keyset table, not the PeopleSoft keystore. See Understanding the Supported Algorithms. **Note:** The key value that you enter here is stored in the PET keyset table using a combination of the algorithm ID and the keyset ID as its identifier. Because this combination is unique for each algorithm, you can create identically defined keyset rows for multiple algorithms.

Defining Encryption Profiles

Access the Encryption Profile page (**PeopleTools** > **Security** > **Encryption** > **Define Encryption Profiles**).

This example illustrates the fields and controls on the Encryption Profile page.



To define a new encryption profile, specify a new profile ID, then select an algorithm chain ID. Each algorithm in the chain appears in order, in its own row with its algorithm ID and chain sequence number. Its parameters each appear in a row, displaying the parameter's name and default value, and indicating whether the parameter represents a key. You can override a parameter's default value by editing it in the **Parameter Value** edit box.

If you intend to enter a keyset as a parameter, check the From Keyset box and enter the keyset ID in the Parameter Value field. If the From Keyset box is checked you must enter the value using the Algorithm Keyset page (CRYPT_KEYSET) (PeopleTools, Security, Encryption, Algorithm Keyset). Keyset values that are implemented for the algorithm appear in the drop-down list.

Deleting an Encryption Profile

Access the Delete Encryption Profile page (**PeopleTools** > **Security** > **Encryption** > **Delete Encryption Profiles.**).

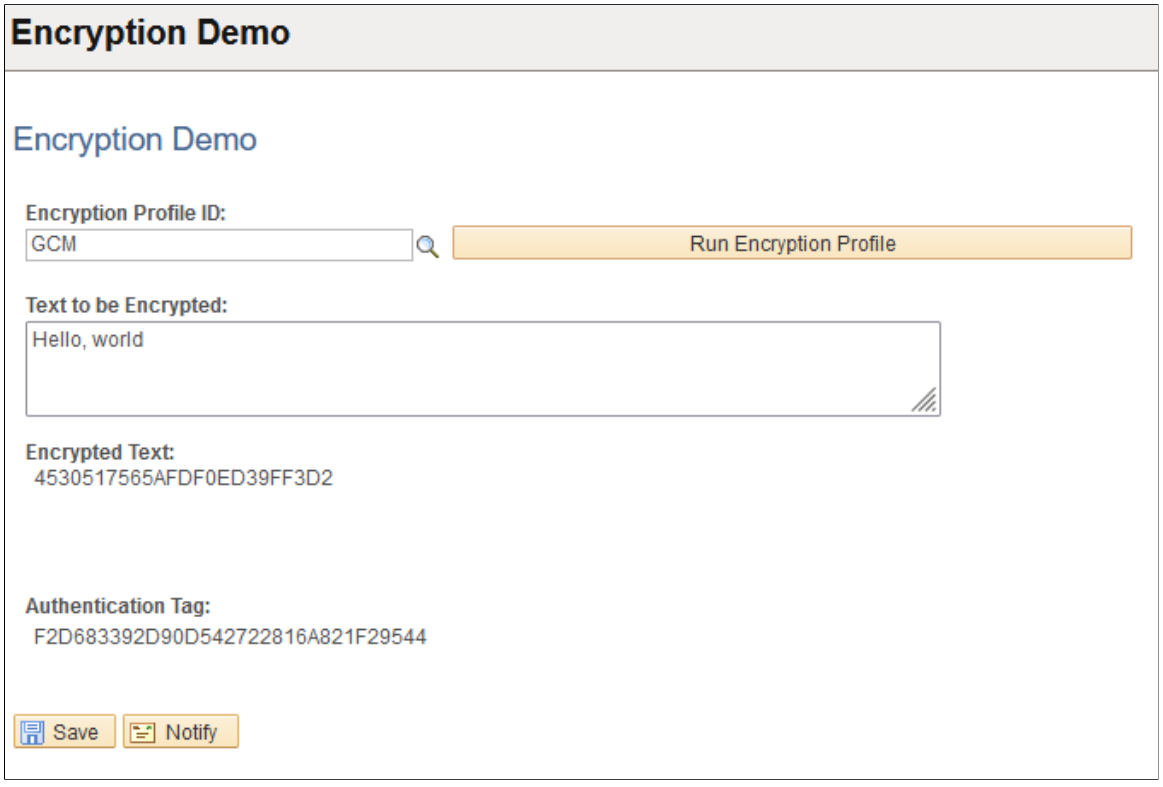
To delete an encryption profile:

1. Select the profile you want to delete
2. Click the **Delete** button.

Testing Encryption Profiles

Access the Encryption Demo page (**PeopleTools** > **Security** > **Encryption** > **Test Encryption Profiles**).

This example illustrates the fields and controls on the Encryption Demo page.



Use the Encryption Demo page to :

- Ensure that the encryption profiles produce the expected results.
- Determine the character length of the encrypted value.

Important! When planning to store encrypted data in fields on a table, you must consider that the length of the encrypted value is often *longer* than the unencrypted value.

To test an encryption profile:

1. Select the profile's encryption profile ID.
2. In the **Text to be Encrypted** field, enter or paste the input text.
3. Click **Run Encryption Profile**.

   The resulting output text appears in the **Encrypted Text** field.

You can use this page to test decryption as well. You can also test complementary pairs of profiles — one to encrypt, and the other to decrypt. By copying the result of the encryption profile test and pasting it as input to the decryption profile test, you can determine whether the text you get out is the same as the text you put in.

Invoking Encryption Profiles from PeopleCode

You access the encryption profile using the PeopleCode Crypt class.

This is an example of PeopleCode that invokes an encryption profile called CRYPT_WRK.CRYPT_PRFL_ID.

In the example the UpdateData method is the encrypt/decrypt command.

```
&cry = CreateObject("Crypt");
&bar = CRYPT_WRK.CRYPT_PRFL_ID;

/* &cry.Open(<profile name>) is a required command that must be used */
/* before any data can be encrypted or decrypted by the named profile.*/
&cry.Open(&bar);

/* UpdateData() is the encrypt/decrypt command. */
&cry.UpdateData(CRYPT_WRK.DESCRLONG);
DERIVED_CRYPT.DESCRLONG = &cry.Result;

/*If there is no Result, then maybe we are running a verify routine.*/

If None(DERIVED_CRYPT.DESCRLONG) Then
   DERIVED_CRYPT.DESCRLONG = &cry.Verified;
End-If;
```

Related Links

“Understanding the Crypt Class” (PeopleCode API Reference)

Upgrading or Migrating from One PET Encryption to Another PET Encryption

Use this template to create a PeopleCode program to upgrade or migrate one PET encryption library to a different PET encryption library.

The program changes the PET encryption library for a Record.Field. The comments in the template include parameter definitions and usage guidelines.

```
/*********************************************************/
/*  This program is built as a template for changing the    */
/*  PET encryption for a specific Record.Field.             */
```



```
/*                                                            */
/*  You Must Change the following values to fit your needs  */
/*  <RECORDNAME> = The Record containing the Field to have  */
/*                 Encryption Upgraded.                       */
/*  <FIELDNAME>  = The Field to have Encryption Upgraded.   */
/*  <DECRYPTPRFL>= The Profile needed to Decrypt the        */
/*                 current contents of the Field.           */
/*  <ENCRYPTPRFL>= The Profile to apply new encryption.     */
/************************************************************/

&sql = CreateSQL("%selectall(:1)", Record.<RECORDNAME>);
&rec = CreateRecord(Record.<RECORDNAME>);
&DecryptProfile = "<DECRYPTPRFL>";
&NewEncryptProfile = "<ENCRYPTPRFL>";

&Decrypt = CreateObject("Crypt");
&Encrypt = CreateObject("Crypt");

While &sql.Fetch(&rec)

   &Decrypt.Open(&DecryptProfile);
   /* If desired you can override any of the Profile Parameters.
      Should only be needed if Random IV, KEY, AAD, AUTHTAG were
      used/produced during Encryption of this Record.Field. You must
      uncomment and set GoToStep if you uncomment any SetParameter */
   rem   &Decrypt.GoToStep(n); /* Change n to Step number in decryption chain
*/
   rem   &Decrypt.SetParameter('IV', '<hex IV used during Encryption>');
   rem   &Decrypt.SetParameter('SYMMETRICKEY', '<keyset entry name used
during Encryption>');
   rem   &Decrypt.SetParameter('AUTHTAG', '<hex AuthTag produced during
Encryption>');
   rem   &Decrypt.SetParameter('AAD', '<AAD value used during Encryption>');
   /* Refer to the Documentation for Parameter Names and Values
   for the non-symmetric algorithms */
   rem   &Decrypt.SetParameter('<PARMNAME>', '<PARMVALUE>');
   &Decrypt.UpdateData(&rec.<FIELDNAME>.Value);
   &rec.<FIELDNAME>.Value = &Decrypt.Result;

   &Encrypt.Open(&NewEncryptProfile);
   /* If desired you can override any of the Profile Parameters.
      Should only be needed if Random IV, KEY, AAD, AUTHTAG are
      used during Encryption of this Record.Field.  You must
      uncomment and set GoToStep if you uncomment any SetParameter */
   rem   &Decrypt.GoToStep(n); /* Change n to Step number in encryption chain
*/
   rem   &Decrypt.SetParameter('IV', '<hex IV used during Encryption>');
   rem   &Decrypt.SetParameter('SYMMETRICKEY', '<keyset entry name used
during Encryption>');
   rem   &Decrypt.SetParameter('AAD', '<AAD value used during Encryption>');
   /* Refer to the Documentation for Parameter Names and Values for the
non-symmetric algorithms */
   rem   &Decrypt.SetParameter('<PARMNAME>', '<PARMVALUE>');
   &Encrypt.UpdateData(&rec.<FIELDNAME>.Value);
   &rec.<FIELDNAME>.Value = &Encrypt.Result;
   rem &rec.<FIELDNAME>.Value = &Encrypt.Verify; /* Used when running Verify
routine */
   rem &rec.<FIELDNAME>.Value = &Encrypt.AuthTag; /* Produced from AES CCM
and GCM modes - required for decryption */

   &rec.Update();
End-While;
```

Related Links

Understanding the Supported Algorithms

“Understanding the Crypt Class” (PeopleCode API Reference)

“Crypt Class Example” (PeopleCode API Reference)

“Encrypt” (PeopleCode Language Reference)
“Decrypt” (PeopleCode Language Reference)

Using Application Engine Programs to Encrypt and Decrypt Tables

There are two Application Engine programs that do full table encryption and decryption:

- PTENCRYPTPET

  If you use Data Mover to export data from a PeopleTools version that pre-dates PET to the current tools version, run PTENCRYPTPET on the target database after the import to encrypt the table data.

- PTDECRYPTPET

  If you use Data Mover to export PET table data from the current version into a version of PeopleTools that predates the introduction of the encrypt and decrypt field object methods, run PTDECRYPTPET on the source data prior to exporting to decrypt the table data.

  Run PTDECRYPTPET on encrypted tables before running any process that does *not* have the ability to implement PeopleCode, such as nVision, SQR, and so on.

  Note: It is recommended that you run PTENCRYPTPET after the system completes such processing.

Note: PET encryption and decryption works regardless of whether the keys are encrypted.

Related Links

“Running Application Engine Programs” (Application Engine)

Chapter 17

Using OAuth 2.0 for User Account Authorization

Understanding OAuth 2.0

OAuth (Open Authorization) is an open standard that allows an end user's account information to be used by third-party services without exposing the user's password. Initially, OAuth 2.0 was supported for Oracle Identity Cloud Service (IDCS) and Chatbot REST Services. Oracle subsequently added support for OAuth 2.0 to Azure, Okta, and Ping.

Currently, the only authentication options for provider REST services are Basic Authentication and PeopleSoft Token. OAuth 2.0 is the industry-standard protocol for authorization. The OAuth 2.0 authorization framework enables a third-party application to obtain limited access to an HTTP service, either on behalf of a resource owner by orchestrating an approval interaction between the resource owner and the HTTP service, or by allowing the third-party application to obtain access on its own behalf.

Item	***Description***
Representational State Transfer (REST)	REST is a style of software architecture for distributed hypermedia systems such as the World Wide Web.
OAuth 2.0	Authorization framework that enables applications to obtain limited access to user accounts on an HTTP service. PeopleTools will only use OAuth 2.0's AccessToken for this Authorization framework. This Access Token is not used for authentication purpose.
Access Token	• An Access Token is a credential that can be used by an application to access an API. • The purpose of this token is to inform the API that the bearer of the token has been authorized to access the API and perform a predetermined set of actions (which is specified by the scopes granted). • Access Tokens must never be used for authentication. It cannot tell us if the user has authenticated. • The only user information the Access Token possesses is the user ID, located in the sub claim.
Grant Access Type	Applications can request an access token to access protected endpoints in different ways, depending on the type of grant type specified in the client application.

Item	*Description*
Oracle Identity Cloud Service (IDCS)	Oracle's OAuth 2.0 Server.

Configuring Service Applications

The authorization server supported by PeopleTools is IDCS (Oracle Identity Cloud Service). The Oracle Identity Cloud Service REST APIs provide a way to integrate Oracle Identity Cloud Service with REST clients so that they can manage users, groups, applications, and settings, and perform federated single sign-on (SSO) and authorization in the cloud. PeopleTools supports Azure, Okta, and Ping as authorization servers.

Before you attempt to configure service applications for the supported authorization servers, ensure that you complete the registration on those servers because some details of the registration are required to configure the service applications.

Creating a Service Application

Access the Create OAuth2 Service Apps page (select **PeopleTools** > **Security** > **OAuth2 Administration** > **Create OAuth2 Service Apps**) and select the Add a New Value tab.

Field or Control	*Description*
OAuth Name	Specify a name for a service application.
Authorization Server	Oracle supports the following OAuth 2.0 server providers: • Azure: When you select Azure as an authorization server, you are presented with four application types: Email Client Only, Client Only, Resource Only, and Client and Resource. The Email Client Only application type is used with the Multichannel Framework product. For details of the Email Client Only application type, see "Setting Up MCF Email Using Azure" (MultiChannel Framework) • IDCS • Okta • Ping

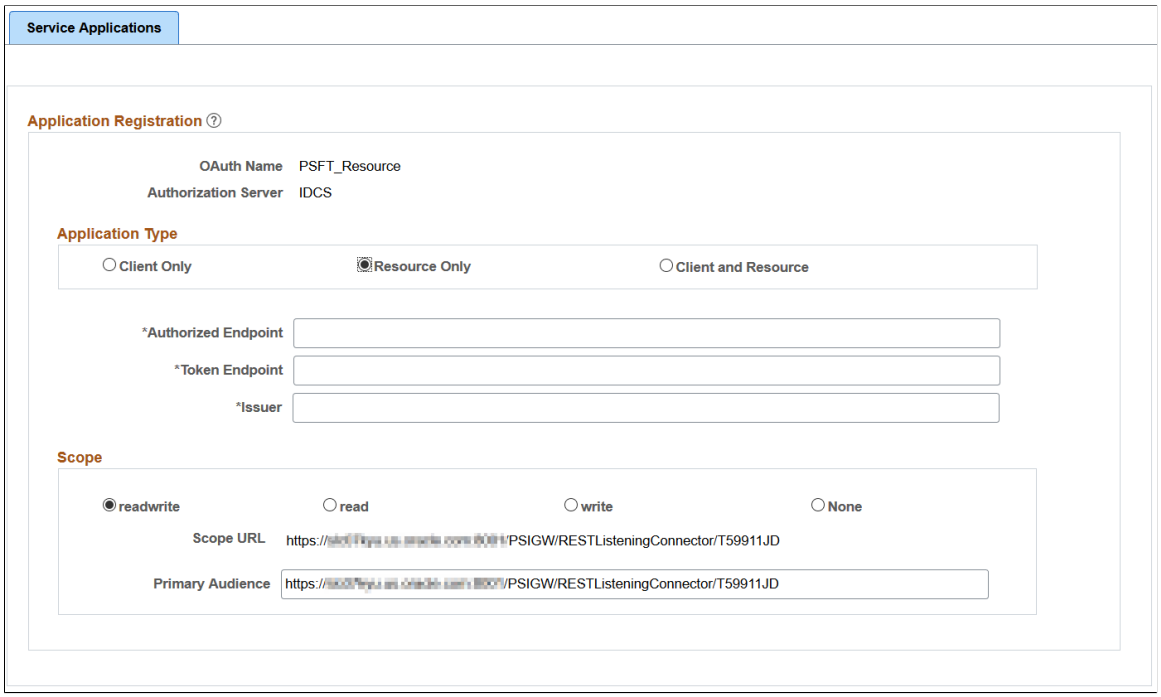
Service Application - REST Provider

This section describes the case where PeopleSoft is the provider of the service.

Access the Service Application page (select **PeopleTools** > **Security** > **OAuth2 Administration** > **Create OAuth2 Service Apps**) and select the Resource Only button.

Based on this selection, the necessary fields are displayed for the administrator to populate.

This example illustrates the fields and controls on the Service Applications - Resource Only page. You can find definitions for the fields and controls later on this page.



Term	*Definition*
Authorized Endpoint	It is an HTTP endpoint that clients can use to identify a user or obtain an authorization code (which is then later exchanged for an access token) to be able to post to their website. The metadata endpoints for the different authorization servers are listed here. The metadata endpoint is where the authorized endpoint is published. • IDCS - http://<tenantinstance>/.well-known/idcs-configuration • Azure - http://login.microsoftonline.com/<tenantinstance>/.well-known/openid-configuration • Okta - http://<tenantinstance>/.well-known/oauth-authorization-server • Ping - http://<tenantinstance>/.well-known/openid-configuration **Note:** If the system is upgraded from PeopleTools 8.58, the system administrator should ensure to enter the configuration data that is applicable to the specific OAuth server in use, because these values may change over time.

Term	*Definition*
Token Endpoint	It is an HTTP endpoint that clients can use to obtain an access token. The metadata endpoints for the different authorization servers are listed here. The metadata endpoint is where the token endpoint is published. • IDCS - http://<tenantinstance>/.well-known/idcs-configuration • Azure - http://login.microsoftonline.com/<tenantinstance>/.well-known/openid-configuration • Okta - http://<tenantinstance>/.well-known/oauth-authorization-server • Ping - http://<tenantinstance>/.well-known/openid-configuration **Note:** If the system is upgraded from PeopleTools 8.58, the system administrator should ensure to enter the configuration data that is applicable to the specific OAuth server in use, because these values may change over time.
Issuer	The issuer of the token.
Scope	This identifies the type of requests allowed. Valid choices are: • readwrite. • read. • write. • None.
Scope URL	This is an auto-generated URL based on the information defined on the Integration Broker Service Configuration page for Target Locations.
Primary Audience	This is the primary recipient where the token is processed. In this case it is the secure REST target location as defined on the Target Locations page. The administrator can append additional information to the URL to make it specific for a service operation.

Gateway Configuration

For configuration as a Resource, the integration.properties file needs to be updated with additional OAuth 2.0 information. The following properties must be specified:

- Cloud Site location used to validate OAuth2 token passed into REST Listening Connector:

- ig.JSONWebKey: cloud site's JSON Web Key
- ig.JSONWebKey=https://IDCS.tenant.instance.com/admin/v1/SigningCert/jwk

- OAuth2 Client used for creating JWT-Bearer during Assertion Grant Type in HttpTargetConnector:
  - ig.AssertionKeyAlias=Client's KeyAliasName
  - ig.AssertionKeyPassword=Client's KeyAlias Password

- Authorization Server used for Resources — currently supports IDCS, Azure, Okta, and Ping:
  - ig.AuthorizationServer:OAuth Server Type
  - ig.AuthorizationServer=IDCS

Service Operation Configuration

These instructions are applicable for the supported authorization servers: IDCS, Azure, Okta, and Ping.

For REST provider service operations, additional options are available in the **Req Verification** drop down list box on the Service Operations Definitions page for:

- **OAuth2 Authorization**: The integration partner must pass a valid access token.
- **OAuth2 Authorization and SSL**: The integration partner must pass a valid access token and transmit the service operation using SSL encryption.

See “Managing Provider REST Service Operations” (Integration Broker).

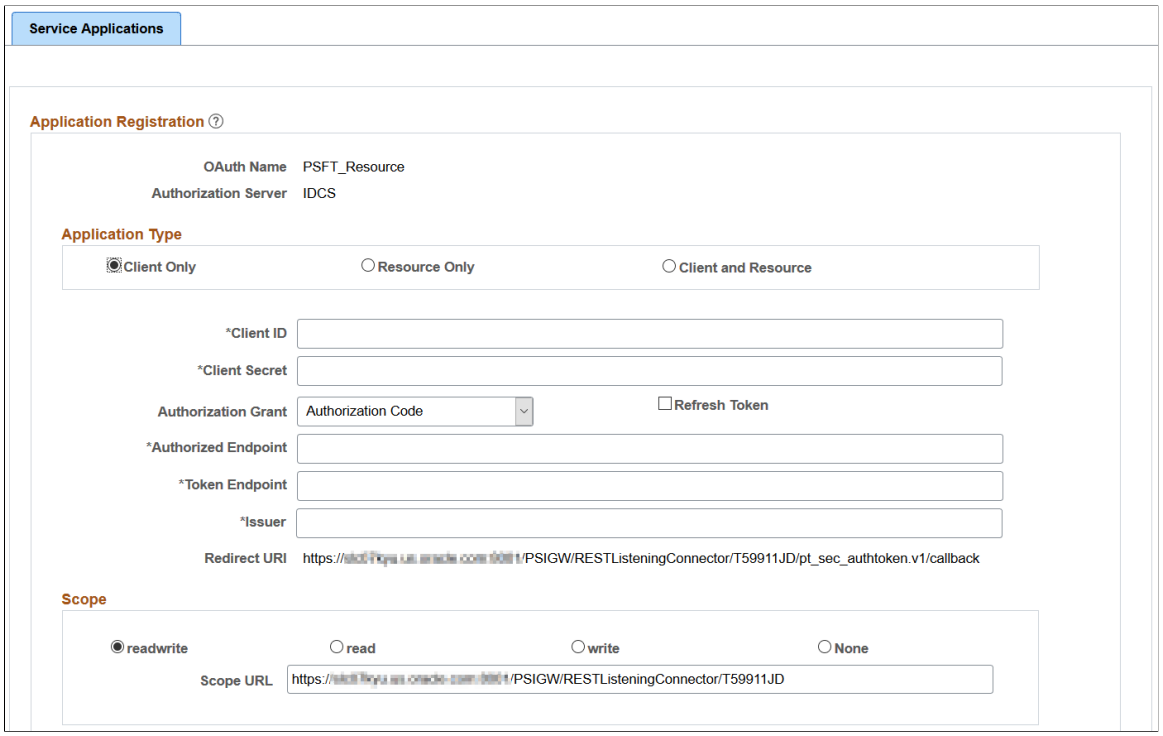
Service Applications - REST Consumer

Consumer REST calls requiring a token will be supported by the security framework if the authorization server is IDCS (or Microsoft Azure for authorization code only). For any other authorization server, it will be up to the developer to make the necessary calls to obtain an access token.

The administrator creates a Service Application and registers information from the provider’s authorization server.

Access the Service Application page (select **PeopleTools** > **Security** > **OAuth2 Administration** > **Create OAuth2 Service Apps**) and select the Client Only button.

This example illustrates the fields and controls on the Service Applications - Client Only page. You can find definitions for the fields and controls later on this page.



Term	*Definition*
Client ID	The API key value generated when you registered your application with the OAuth server.
Client Secret	The secret key value generated from the OAuth server.
Authorization Grant	The Authorization Grant drop-down options are available only for the IDCS authorization server. Applications can request an access token to access protected endpoints in different ways, depending on the type of grant type specified in the client application. A grant is a credential representing the resource owner's authorization to access a protected resource. Currently PeopleSoft supports the following grant types: • Authorization Code (3-legged) • Assertion (2-legged) • Client Credentials (2-legged) For Azure, Okta, and Ping, Authorization Code is the only available authorization grant. See Authorization Grant Types for more information.

Term	*Definition*
Refresh Token	Refresh tokens are credentials used to obtain access tokens. Refresh tokens are issued to the client by the authorization server and are used to obtain a new access token when the current access token becomes invalid or expires, or to obtain additional access tokens with identical or narrower scope (access tokens may have a shorter lifetime and fewer permissions than authorized by the resource owner). **Note:** The OAuth server needs to be configured to generate a refresh token along with an access token.
Authorized Endpoint	It is an HTTP endpoint that clients can use to identify a user or obtain an authorization code (which is then later exchanged for an access token) to be able to post to their website.
Token Endpoint	It is an HTTP endpoint that clients can use to obtain an access token.
Issuer	The issuer of the token.
Redirect URl	The URL is auto-generated based on the REST Target Location entry under Service Configuration. This URL should be copied and entered for IDCS to make the appropriate callbacks.
Scope	This identifies the type of requests allowed. Valid choices are: • readwrite. • read. • write. • None.
Scope URL	The scope URL is specific to the different authorization servers: For IDCS, the scope URL is auto-generated based on the information defined on the Service Configuration Target Locations page. See “Using the Target Locations Page to Set Target Locations for Services” (Integration Broker Administration). For Azure, it is defined as Azure Resource-Client-ID.

Gateway Configuration

For configuration as a client (consumer), when using the assertion grant type, the integration.properties file needs to be updated with additional information. This is applicable only to the IDCS authorization server.

The PeopleSoft private keys (client) used to retrieve the access token are:

- **ig.AssertionKeyAlias**: The alias name defined for the private key enter into the key store.
- **ig.AssertionKeyPassword**: The password defined for the private key.

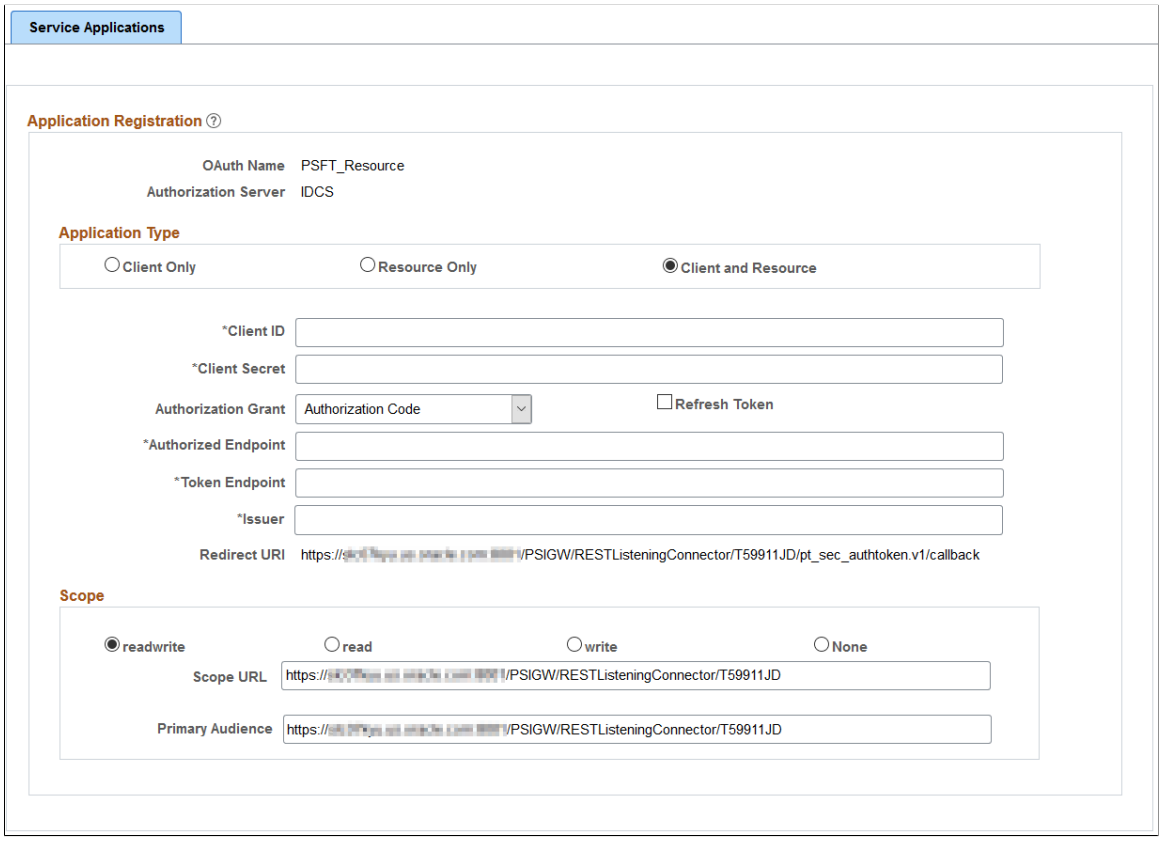
See “Using the integrationGateway.properties File” (Integration Broker Administration).

Service Applications - REST Provider and Consumer

For the case where the client is actually calling the same resource (local REST integrations), the administrator can select the application type as **Client and Resource**.

Access the Service Application page (select **PeopleTools** > **Security** > **OAuth2 Administration** > **Create OAuth2 Service Apps**) and select the Client and Resource button.

This example illustrates the fields and controls on the Service Applications - Client and Resource page. You can find definitions for the fields and controls later on this page.



Term	*Definition*
Client ID	The API Key value generated when you registered your application with the OAuth server.
Client Secret	The Secret Key value generated from the OAuth server.
Authorization Grant	The Authorization Grant drop-down options are available only for the IDCS authorization server. Applications can request an access token to access protected endpoints in different ways, depending on the type of grant type specified in the client application. A grant is a credential representing the resource owner's authorization to access a protected resource. Currently PeopleSoft supports the following grant types: • Authorization Code (3-legged) • Assertion (2-legged) • Client Credentials (2-legged) For Azure, Okta, and Ping, Authorization Code is the only available authorization grant. See Authorization Grant Types for more information.
Refresh Token	When you select the check box, it saves a refresh token to use when access token expires and retrieves an updated access token. **Note:** The OAuth server needs to be configured to generate a refresh token along with an access token.
Authorized Endpoint	It is an HTTP endpoint that clients can use to identify a user or obtain an authorization code (which is then later exchanged for an access token) to be able to post to their website.
Token Endpoint	It is an HTTP endpoint that clients can use to obtain an access token.
Issuer	The issuer of the token.
Redirect URl	The URL is auto-generated based on the REST Target Location entry under Service Configuration. This URL should be copied and entered for IDCS to make the appropriate callbacks.

Term	*Definition*
Scope URL	The scope URL is specific to the different authorization servers: For IDCS, the scope URL is auto-generated based on the information defined on the Service Configuration Target Locations page. For Azure, it is defined as Azure Resource-Client-ID.
Primary Audience	The primary recipient where the token is processed. In this case it is the secure REST target location as defined on the Target Locations page. The administrator can append additional information to the URL to make it specific for a Service Operation.

This information in this table provides the options for each grant type.

Authorization Grant Types

The following authorization grant types are supported:

- **Authorization Code - 3-legged-OAuth**: This grant type is used when you want to obtain an authorization code by using an authorization server as an intermediary between the client application and the resource owner.
- **Assertion - 2-legged-OAuth**: This grant type is used when you want to use an existing trust relationship expressed as an assertion and without a direct user approval step at the OAuth Authorization Server.
- **Client Credentials - 2-legged-OAuth**: This grant type is used when authorization type is limited to the protected resource s under the control of the client or to protected resources registered with the OAuth Authorization Server.

Grant Type	*OAuth Server*	*User Credential Required during Request*	*Trusted by Using Certificate*	*User Context in Access Token*	*Allowing Refresh Token*	*Browser Interaction*
Authorization Code (3–legged)	IDCS, Azure, Okta, and Ping	Y	N	Y	Y	Y
JWT User Assertion (2–legged)	IDCS only	N	Y	Y	Y	N
Client Credential (2–legged)	IDCS only	N	N	N	N	N

For more information about grants types, see the product documentation for Integrating and Extending Oracle Content and Experience in Oracle Help Center.

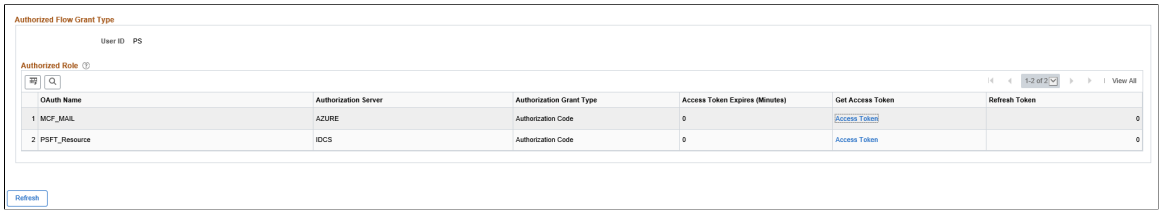
Retrieving Access Token

Use the Authorized Flow Grant Type page to retrieve an access token from the OAuth2 server. Access tokens are retrieved based on the logged in user ID and the selected OAuth2 service application. Therefore, users should ensure that they access this page and retrieve an access token for their user ID.

The page lists the OAuth2 service applications that are created in the PeopleSoft system.

Access the Authorized Flow Grant Type page (select **PeopleTools** > **Security** > **OAuth2 Administration** > **Retrieve OAuth2 Access Token**).

This example illustrates the fields and controls on Authorized Flow Grant Type page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
User ID	Displays the logged in user ID. **Note:** When you retrieve an access token using the Access Token link, you're retrieving an access token only for the logged in user.
Access Token Expires (Minutes)	Indicates the time in minutes when the access token expires. A value of 0 (zero) means that the token has expired or there is no stored token. **Important!** The value of 0 should be considered in conjunction with the value in the Refresh Token field because 0 also indicates a perfectly normal working state if a refresh token is used.
Get Access Token	Select the Access Token link to retrieve an access token for the selected service application.

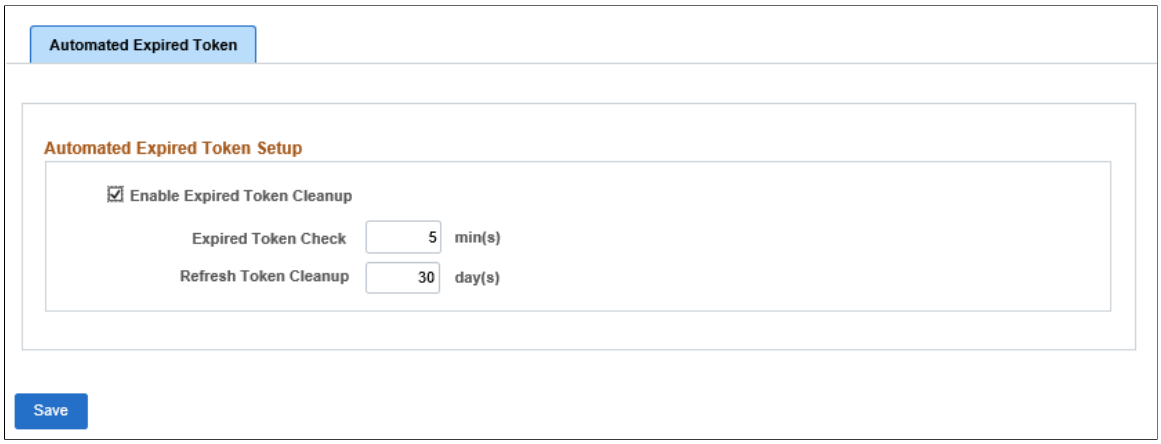
Field or Control	*Description*
Refresh Token	A value of 1 indicates a refresh token. **Important!** If a refresh token is part of the initial token request, a refresh token is used when an access token expires. For more information on refresh token, see Configuring Service Applications.

Deleting Expired Tokens

Use the Automated Expired Token page to automatically delete expired access tokens stored in the database. The next subsequent call to get an access token and refresh token if available will be used to make a call to the authentication server in order to retrieve a valid access token and an updated refresh token.

Access the Automated Expired Token page (select **PeopleTools** > **Security** > **OAuth2 Administration** > **OAuth2 Access Token Cleanup**).

This example illustrates the fields and controls on Automated Expired Token page. You can find definitions for the fields and controls later on this page.

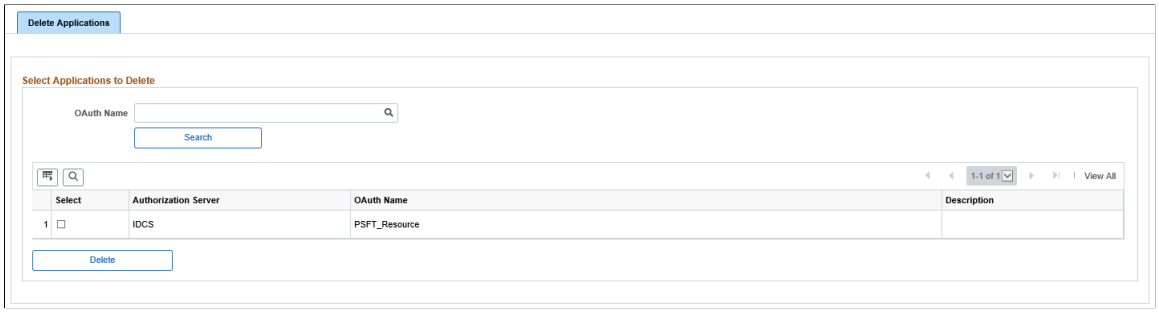


Field or Control	*Description*
Enable Expired Token Cleanup	Check box to turn on and off the process to clean up expired access tokens.
Expired Token Check	The time interval between checks for expired access tokens.
Refresh Token Cleanup	Removes refresh tokens if the last request to retrieve an access token is longer than the specified days.

Deleting Applications

Access the Delete Application page (select **PeopleTools** > **Security** > **OAuth2 Administration** > **Delete OAuth2 Service Apps**).

This example illustrates the fields and controls on Delete Applications page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
OAuth Name	Find the service application that needs to be deleted.

Chapter 18

Using Prompt Table Overrides

Understanding Prompt Table Overrides

PeopleSoft enables you to create an override prompt table for any record field that already has a prompt table defined for it. At runtime, if an override prompt table is assigned to the record field, the override table determines the result set instead of the defined prompt table.

Using prompt table override, a user has the flexibility to make changes in the result set without any customization. This is possible by defining a new dynamic view with SQL for limiting the OPRIDs to return values based on a role. For example, if you want an OPRID to view a different result set for a record field that has a prompt table defined, then you can use a prompt table override to show different results for different users.

Similar functionality is already available in PeopleSoft, which is enabled by including derived record on the page and defining the prompt table for the record field as a dynamic prompt (%<DERIVED RECORD FIELD NAME>). Prompt table override is required only when the record fields are not delivered with a dynamic prompt capability.

See “Setting Record Field Properties” (Application Designer Developer’s Guide).

View filtering based on runtime OPRID is already available view, the %OPRCLAUSE Meta-SQL. Use of this Meta-SQL is restricted to dynamic views.

Note: All override prompt tables must be dynamic views and must use %OPRCLAUSE.

Note: Override prompt table settings will not be a formal part of the RDM managed object. However, whenever a record field is loaded from DB, this information is cached as part of the RDM managed object to improve the performance.
See “%OPRCLAUSE” (PeopleCode Language Reference).

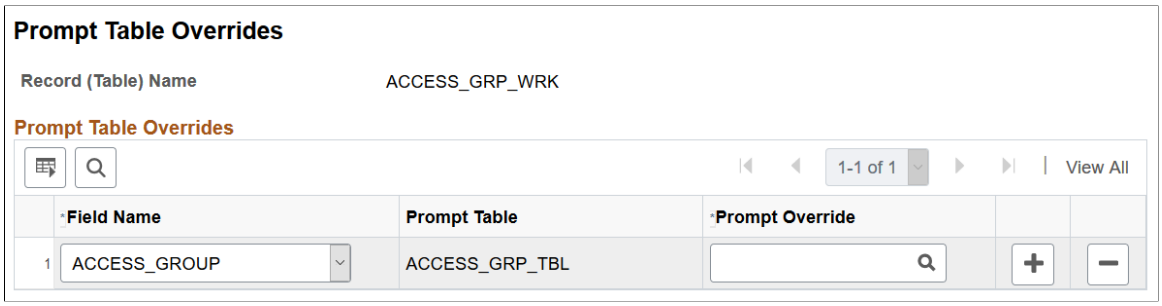
Using the Prompt Table Overrides Page

Use the Prompt Table Overrides page to set an override prompt table for a field that has a prompt table assigned to it. Using this functionality, a user gets the result set that is selected from the override prompt table; instead of the defined prompt table.

Access the Prompt Table Overrides page by selecting **PeopleTools** > **Security** > **Security Objects** > **Prompt Table Overrides**.

The search page shows only those records that have one or more fields with a prompt table defined for it.

This example illustrates the fields and controls on the Prompt Table Overrides page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Field Name	This is the component search record. It lists only the fields within the selected record that have a prompt table defined for it.
Prompt Table	The defined prompt table record for the selected field name.
Prompt Override	Select the appropriate dynamic view. The Prompt Override drop down remains disabled until a field is selected and is limited to records defined as dynamic views whose SQLtext contains %OPRCLAUSE. **Note:** Choosing a wrong or improperly constructed override prompt table may cause related page errors.

PeopleCode Considerations

The following Field Class properties are affected:

- PromptTableName

  If the defined prompt table is overridden, then this property returns the override prompt table name instead of the prompt table defined on the record field in Application Designer.

  See “PromptTableName” (PeopleCode API Reference).

- SQLText

  If the defined prompt table is overridden, then this property returns the SQL defined in the override prompt table. When this property is used to replace the SQL text being used at runtime, it continues to work as designed.

  See “SqlText” (PeopleCode API Reference).

Chapter 19

Implementing Query Security

Defining Query Profiles

Query takes advantage of user's security settings, row-level security, and primary permission list. Query Manager helps you build SQL queries to retrieve information from your application tables. For each Query Manager or Query Viewer user, you can specify the records they are allowed to access when building and running queries.

You do this by creating Query Access Groups in the Query Access Group Manager, and then you assign users to those groups with Query permissions. Keep in mind that Query permissions are enforced only when using Query; it doesn't control run-time *page* access to table data.

Building Query Access Group Trees

Trees are a graphical way of presenting hierarchical information. PeopleSoft Query uses *query access group trees* to control the access of the tables in the PeopleSoft database. You define a hierarchy of PeopleSoft record definitions, based on logical or functional groupings, and then give users access to one or more nodes of the tree. Users can retrieve information only from those tables whose record definitions to which they have access.

You create and update query access group trees using Query Access Manager. To get you started, we've included some sample query access group trees with the PeopleSoft applications. Which trees you have depend on which PeopleSoft applications you've installed. Each tree contains access groups and record definitions categorized by function.

Access groups mark and define a functional group of records or other access groups—in other words, they are descriptive placeholders used to categorize actual record definitions in a logical, hierarchical format. When you define users' security rights to a tree, you specify which access groups they are permitted to query.

This section explains how to create query access group trees. It assumes that you're familiar with the concept and terminology of PeopleSoft trees.

Query Access Group Tree Considerations

You should create query access group trees based on your organization's needs and on any customizations you've made. Remember that the sample trees we provide may be replaced when you upgrade to a subsequent PeopleSoft release, so if you modify the samples rather than create your own trees, you may lose your customizations.

Every record definition that you want users to be able to query must be in a query tree. However, they don't all have to be in the same query tree. One strategy is to use the sample query trees to provide access to the standard PeopleSoft record definitions, but create separate query trees for record definitions that

you add in the course of customizing the system. This way, you take advantage of the sample trees but avoid overwriting your changes during future upgrades.

How you organize the contents of the query tree depends on the needs of your organization and your users. For example, you might want to create small trees that are not intimidating to non-technical or casual users. The sample query trees provided in the PeopleSoft application are divided by functions, but to simplify the trees, you may want to create separate trees that contain subcategories of each function. For example, you could create separate trees for U.S. and Canadian record components to grant users in each region security access to only the record components they should use.

Note: You should consider adding record definitions to the query trees in a hierarchy that matches the parent/child relationship of records in your database. Though you don't have to organize records this way —Application Designer actually controls the parent/child hierarchy in your database—you'll probably find it helpful to keep the query trees consistent with your database structure.

Working with Query Trees

This section provides an overview of Query access group trees and discusses how to:

- Open Query access group trees.
- Define the Query tree.
- View and modify definitions.

Understanding Query Access Group Trees

If you have worked with Tree Manager or trees, take a moment to review the following information describing the differences between typical trees and the Query access group trees.

Nodes

Regarding nodes, consider the following points:

- Query access group trees contain two types of nodes: groups and records.
- Groups are a logical representation of a set of child groups or records, similar to folders in Microsoft Windows.
- Records represent a PeopleSoft record definition.

Structure

Regarding structure, consider the following points:

- Always use the ACCESS_GROUP Tree Structure.
- Do not use SetID or UKV/BU.
- Do not have Details.
- Do not use Levels.

- Do not use Branches.

Requirements

Regarding requirements, consider the following points:

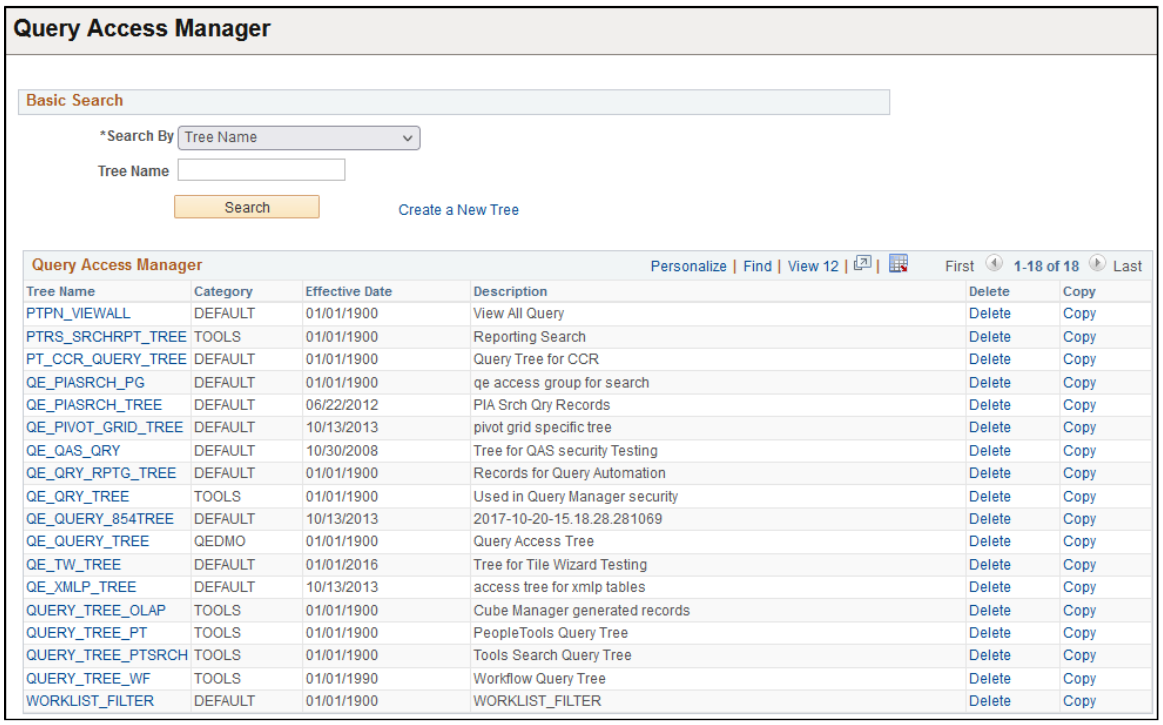
- The Root Node is always a group.
- Groups must be unique in a given tree while records definitions can be repeated.
- Groups and records could have Child Groups and Child Records.
- Each record needs a unique fully qualified path in the tree.

  You can't add the same record under the same parent node (group or record).

Opening Query Access Group Trees

Access the Query Access Manager page (**PeopleTools** > **Security** > **Query Security** > **Query Access Manager**).

This example illustrates the fields and controls on the Query Access Manager - Search page.



Before you can view and modify a Query access group tree definition, you need to locate the correct tree definition.

To open a query tree definition:

1. On the Basic Search page select your search criteria.

   You can search by Tree Name, Tree Category, Tree Description, Group Name used in a Tree, or Record Name used in a Tree.

2. Click **Search.**

   After clicking Search, a list appears containing the definitions that meet your criteria.

3. Click the tree name link.

   The search page also enables you to delete or copy a tree. Click the Delete or Copy link to perform the desired task. If you click **Delete**, the system prompts you to confirm the action, and if you click **Copy,** the system displays the Copy Tree page where you can enter the name for the copied tree.

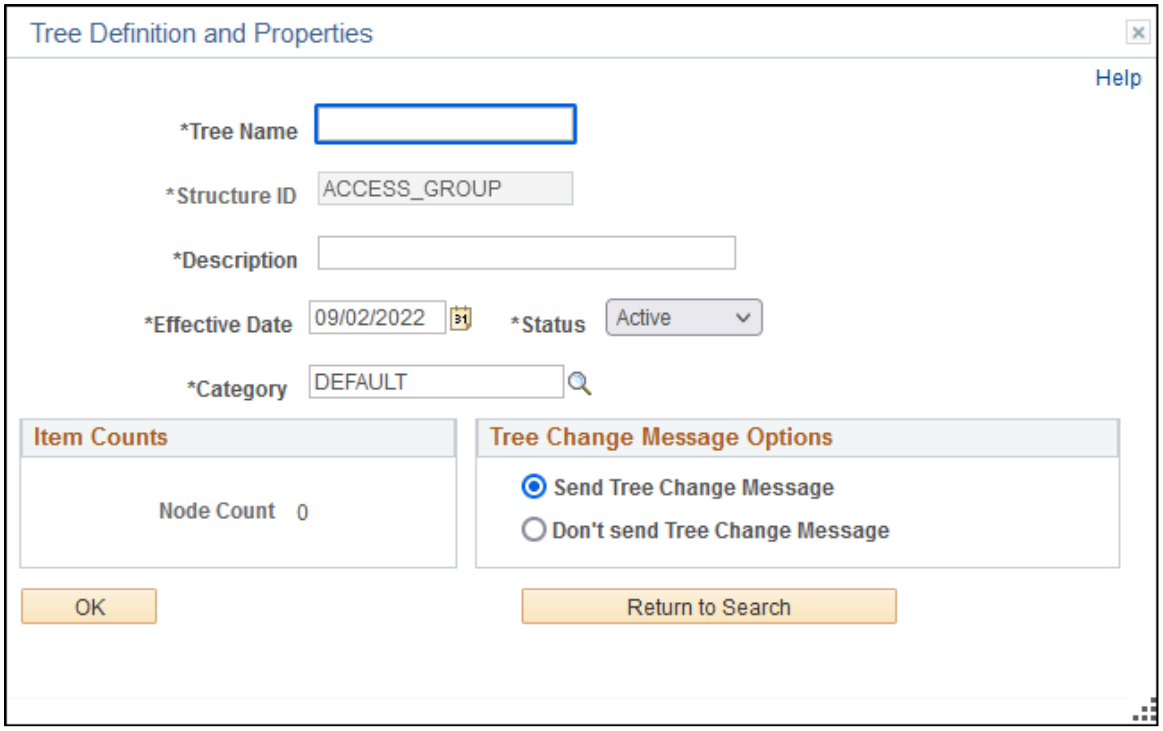
   Some of the trees in the grid may appear without Copy and Delete buttons visible. In this situation, Definition Security settings are such that you have only read-only access to these trees.

Defining the Query Tree

Access the Tree Definition and Properties page (click the **Create a New Tree** link on the Basic Search page).

Before you can insert nodes for access groups and record components, you must first define a number of important characteristics for the tree.

This example illustrates the fields and controls on the Tree Definition and Properties page.



Field or Control	*Description*
Tree Name	For the tree name, we recommend that you start the name with QRY_ so that you can easily identify the tree as a custom query tree. The standard query trees we deliver with the system start with QUERY_.

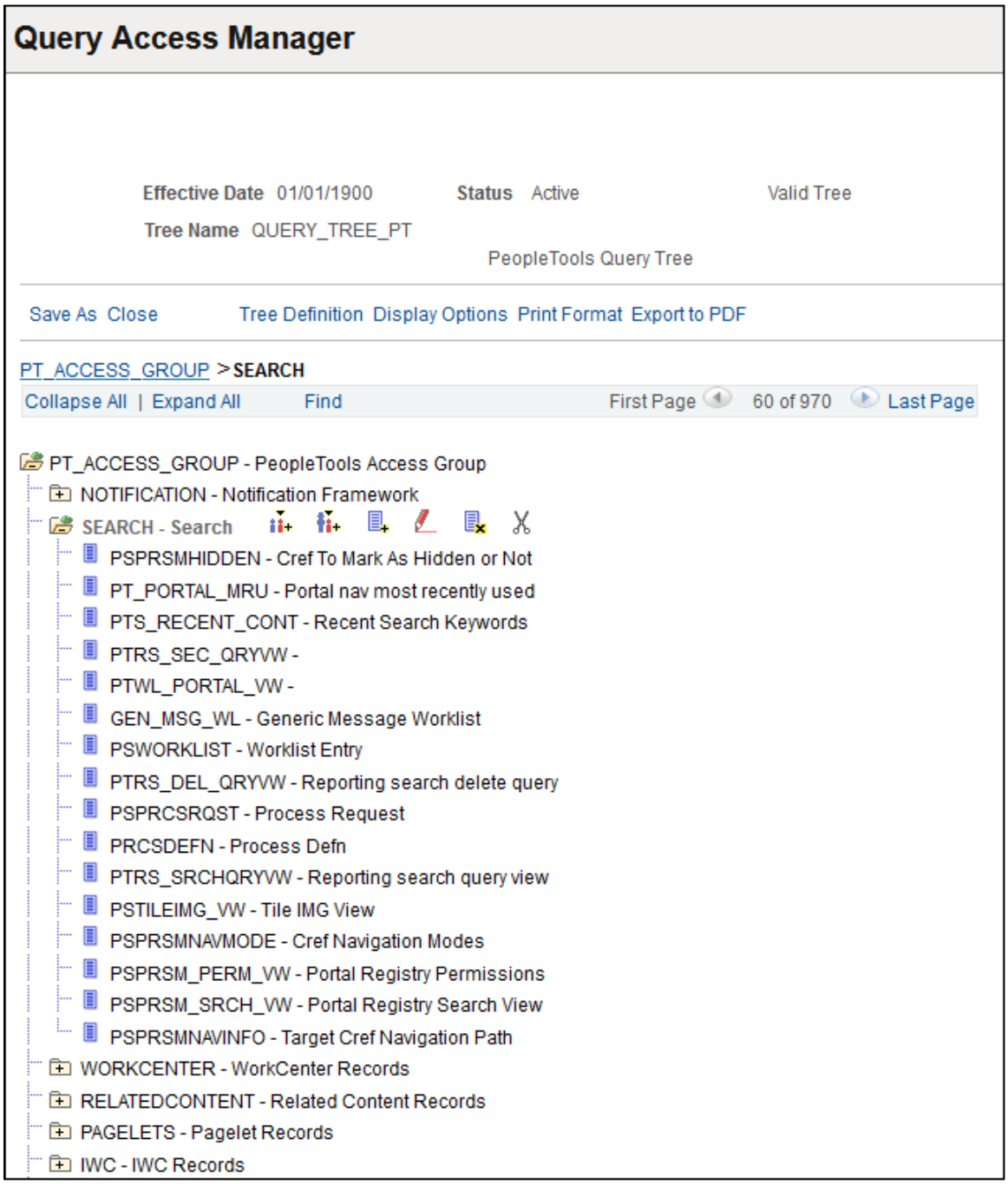
Field or Control	*Description*
Structure ID	The Structure ID is read only and always reads ACCESS_ GROUPS for Query access trees.
Description	The description appears with the name and effective date in the list box when you select from a list of trees.
Effective Date	The status default is set to Active. Query trees are available immediately if the effective date is active; you don’t need to run an SQR utility like you do for organizational security trees.
Category	If necessary add a category, which are groupings of the definitions.
Item Counts	Item Counts shows the number of nodes within the access group.

Once you've completed the tree definition, click OK. On the Enter Root Node for Tree page, select an existing Access Group using the Lookup Access Group control, or create a new one.

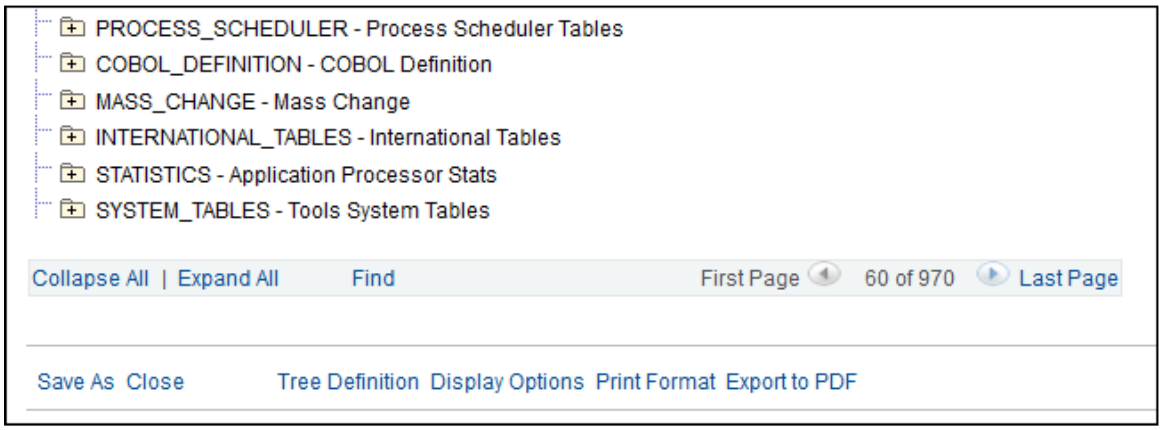
Viewing and Modifying Definitions

This section describes the controls you use to modify Query Access Group Trees after you have opened one from the search page.

This example illustrates the fields and controls on the top portion of the Query Access Manager page, showing details for a tree.



This example illustrates the fields and controls on the bottom portion of the Query Access Manager page, showing details for a tree.



Note: Not all fields and controls are visible on this page. Depending on the page arrangement, you see additional elements by scrolling either vertically or horizontally.

Field or Control	***Description***
Effective Date	Shows the current effective date.
Status	Shows either Active or Inactive.
Tree Name	Shows the name of the current tree.
Save, Save As	These are the two save options. Each option appears only if it relates to the current activity. Save enables you to save your changes to the database. Save As enables you to clone tree definitions at save time.
Close	Closes the definition and returns you to the search page.
Tree Definition	Shows the Tree Definition and Properties page that you modified when you created the definition.
Display Options	Shows the Configure User Options page where you can adjust the presentation of the trees. For example, you can choose whether the Node ID appears and how many lines of the definition appear at a time. Most of these don't apply for Query Access Trees so they're disabled.
Print Format	Displays a print preview of the tree definition.
Bread Crumbs	Once you have drilled down into a definition, a "bread crumb" view appears just above the Collapse/Expand All controls to provide orientation, especially within large trees.

Field or Control	*Description*
Collapse All	Collapses all nodes of the tree into their parent groups so that you see only the root node and the first layer of child groups.
Expand All	Expands all nodes of the tree so that each child object is visible.
Find	If you are looking for a specific access group or a record you can use the Find Value page rather than drilling down into the tree. You specify an access group or a record or its description. You can select a case sensitive search and specify that an exact match must be found. You can use pattern search option by deselecting the Exact Matching check box. This performs platform independent search for the Record/Group starting from the specified pattern. If you want to perform pattern search not starting from the beginning of Record/Group name, specify a platform dependent wildcard character at the beginning of the pattern. For example, to find all occurrences of 'TBL' in the Records, you specify *%TBL* as a search condition (for Microsoft SQL Server database). If you specify both Group and Record search conditions the search is performed on Group condition. If you specify both Group/Record ID (name) and Description conditions the search is performed on ID/name condition. **Note:** Always save modifications to the tree prior to using the Find feature.

Node/Record Controls

When you have a node or record selected, the actions you perform are controlled by the icons that appear to the left and right of the definition. The descriptions of the actions are below. You can pass the mouse pointer over an icon to reveal its label.

Field or Control	*Description*
	When a node folder is open, click the **Collapse Node** icon to collapse the node.
	When a node folder is closed, click the **Expand Node** icon to expand the node.
	The **Insert Sibling Group** icon inserts an access group node at the same level as the currently selected node.

Field or Control	*Description*
	The **Insert Child Group** icon inserts an access group node at the next level lower than the currently selected node.
	The **Insert Child Record** icon inserts a record definition within an access group node.
	For access groups, click the **Edit Data** icon to edit the Description and the Definition (long description) on the Access Group Table.
	Click the **Delete** icon to delete both access groups and records. You can't delete the root node.
 and 	You can cut and paste access groups and records to move them within the tree. Once you click the **Cut** icon, the **Paste as Child** icon becomes enabled. You can't cut the root node. **Note:** After you perform the cut function, only navigation and search features are available until you use the paste function. This protects the node in the clipboard.

Defining Row-Level Security and Query Security Records

By default, when you give Query users access to a record definition, they have access to all the rows of data in the table built using the associated record definition. In some cases, though, you want to restrict users from seeing some of those data rows. For example, you might not want your human resources staff to have access to compensation data for vice presidents or above. In other words, you want to enforce *row-level security*, (also called data permission security) which is offered by many PeopleSoft applications.

This section describes the relationship between row-level security and Query security record definitions.

Row-Level Security

With row-level security , users can have access to a table without having access to all rows on that table. This type of security is typically applied to tables that hold sensitive data. For example, you might want users to be able to review personal data for employees in their own department, but not for people in other departments. You would give everyone access to the PERSONAL_DATA table, but would enforce row-level security so that they could only see rows where the DEPTID matches their own.

PeopleSoft applications implement row-level security by using a SQL view that joins the data table with an authorization table. When a user searches for data in the data table, the system performs a related record join between the view and the base table rather than searching the table directly. The view adds a security check to the search, based on the criteria you've set up for row-level security. For example, to

restrict users to seeing data from their own department, the view would select from the underlying table just those rows where the DEPTID matches the user's DEPTID.

Query Security Record Definitions

You implement row-level security by having Query search for data using a query security record definition. The query security record definition adds a security check to the search.

Query security record definitions serve the same purpose as search record definitions do for panels. Just as a panel's search record definition determines what data the user can display in the panel, the query security record definition determines what data the user can display with Query.

To get Query to retrieve data by joining a security record definition to the base table, you specify the appropriate Query Security Record when you create the base table's record definition.

To apply row level security:

1. In Application Designer, open the record on which you want to apply row-level security.
2. With the record definition open in Application Designer, click the **Properties** button, and select the **Use** tab from the **Record Properties** dialog box.

   Note: You use this dialog box to set a number of different aspects of the record definition. The only item related to Query security is **Query Security Record** list box.

   ---

3. Select the security record definition (usually a view) in the Query Security Record list box.

   Each PeopleSoft product line comes with a set of views for implementing its standard row-level security options. See the product documentation for details.

   Note: The **Parent Record** list box is also relevant to Query. It identifies a record definition that is the current definition's parent, meaning that it holds related data and that its keys are a subset of the current record definition's keys. If you designate a parent record, Query automatically knows what fields to use when you join these two tables for a query.

   Typically, the Query Security Record definition you'll want to select is the same one you use as the search record definition for the panel that manages this table. If you're enforcing one of the standard row-level security options from a PeopleSoft application, select the PeopleSoft-supplied security view for that option. See the application documentation for a list of the available views. If you've designed your own security scheme, select a record definition that appropriately restricts the rows a query will return.

4. Once you've set the query security record definition, click **OK** to close the Record Properties dialog box, then save the record definition.

   If you've already used SQL Create to build the table or view from this record definition, you don't need to rebuild it.

Row-Level (Data Permission) Security Views

Using PeopleSoft row-level security views enables you to restrict users from seeing certain rows of data. You can restrict data by:

- User, by using the OPRID field.

- Primary permission list, by using the OPRCLASS field.
- Row security permission list, by using the ROWSECCLASS field.

To implement row-level security through a security view:

1. In Application Designer, insert one of the three row-level security fields (OPRID, OPRCLASS, ROWSECCLASS) into the record definition.
2. Configure the field as a Key, but not a List Box Item.
3. Save the record and build the view.
4. Use the record as the search record or query security record.

Now, when the user searches, the system dynamically adds a WHERE clause — that incorporates the security field — to the search SELECT statement. The value of the security field is based on the current user.

Chapter 20

Understanding Definition Security

Understanding Definition Security

This topic provides an overview of definition security.

Definition Security

Definition security enables you to secure record definitions, menu definitions, page definitions, and other definitions that make up your applications.

Just as you use security to control who can access the PeopleSoft pages in your system, you use definition security to control who can access and update PeopleTools definitions.

You can implement definition security in the browser client or in the Microsoft Windows client. Browser client definition security was introduced in PeopleTools 8.54 and provides additional functionality than in the legacy Windows client application. While Windows client definition security may still function, any new enhancements and functionality will be made to the browser client definition security.

To access browser client definition security navigate to **PeopleTools** > **Security** > **Definition Security.**

You can access definition security on the Windows client two ways:

- Run the PSOSE.exe file in the Windows client.
- In PeopleSoft Application Designer, select **Go** > **Definition Security.**

For information about the features and functionality of browser client definition security and Windows client definition security, see Comparing Browser Client and Windows Client Definition Security.

Definition Groups and Permission Lists

To implement definition security, you define definition groups and then link them to permission lists that you've created in security.

A definition group is a collection of one or more definitions that form a logical group for security purposes. For example, you've created a permission list for analysts who support the PeopleSoft Payroll module, and you call it PAYROLL_DEV. The analysts are allowed to update only payroll definitions. Using Definition Security, you create a definition group containing only payroll definitions, and give it a name, such as PAYROLL_OBJ. Finally, you link PAYROLL_OBJ to PAYROLL_DEV.

You can assign multiple definition groups to a single permission list. And you can assign a single definition group to multiple permission lists.

Definition groups must be assigned to the primary permission list defined for a user profile. The primary permission list is defined on the User Profile – General page, in the Permission List section of the page in the **Primary** field.

You can't declare directly that a particular permission list can modify a specific definition type. You do so indirectly by creating a definition group that consists solely of the desired definition type. Also, remember that you can assign a definition to multiple groups as needed. To ensure total definition security, assign every definition to at least one definition group.

Note: PeopleTools databases are delivered with a predefined definition group called PEOPLETOOLS that contains all the PeopleTools definitions. Until you create definition groups of your own, the PEOPLETOOLS definitions are the only definitions that you can secure.

Definition Security Authorization Rules

To set up definition security properly, it's helpful to understand how the system interprets definition security settings. The system applies the following rules to determine whether a user is authorized to update a definition:

Rule	*Description*
1	Is the definition type assigned to any definition group? If not, then anyone has update access to it. For this reason, you should add all definition types to at least one definition group.
2	Is the definition type a part of a definition group assigned to the user's *primary permission list*? If not, the system denies access and displays a message, such as "*definition_name* is not a definition that you are authorized to access."
3	Do all the definition groups of which the definition type is a member have the display-only option enabled? If so, then the system displays the message "*definition_name* is not a definition that you are authorized to update." The definition type appears with the Save command disabled.

If the definition passes these system checks, the user is allowed to access and update it—unless it's a PeopleSoft Application Designer definition, in which case several other security checks are performed first. PeopleSoft Application Designer definitions are also controlled by the PeopleTools in permission lists.

Important! A user gets definition security permissions through the primary permission list, not through roles. Access to definition types is granted through roles.

Comparing Browser Client and Windows Client Definition Security

PeopleTools enables you to implement definition security in the browser client or in the Windows client. Both enable you to implement basic definition security. However, browser client definition

security enables you to work with several more definition types and offers additional definition security functionality.

Definition Security Features

This table lists definition security features available in the browser client and in the Windows client:

Definition Security Feature	*Browser Client Definition Security*	*Windows Client Definition Security*
Create definition groups of one or more metadata definitions.	X	X
Link definition groups to a predefined permission lists.	X	X
Copy and delete definition groups.	X	X
Populate definition groups with SQL-like selection criteria.	X	NA
Create inclusion rules so that new objects added to the database that meet an inclusion rule are added to a definition group. This includes the ability to: • Save selection criteria used to initially populate a definition group as an inclusion rule. • Re-process inclusion rules to ensure any new definitions that meet or no longer meet the original selection criteria are automatically added/ deleted to/from the group.	X	NA
Assign row-level security.	X	NA
Enable or disable Secure by Default for all objects covered by Definition Security.	X	NA
View unsecured definitions/definitions that don’t belong to a group.	X	NA

Definition Security Definition Types

The following tables lists the definition types with which you can work in browser client definition security and Windows client definition security:

Definition Type	*Browser Client Definition Security*	*Windows Client Security*	*Associated Designer Tool*
Activities	X	X	PeopleSoft Application Designer
Analytic Model	X	NA	PeopleSoft Application Designer
Analytic Types	X	X	PeopleSoft Application Designer
App Engine Programs	X	X	PeopleSoft Application Designer
Application Packages	X	X	PeopleSoft Application Designer
Approval Rule Sets	X	X	PeopleSoft Application Designer
Business Interlinks	X	X	PeopleSoft Application Designer
Business Processes	X	X	PeopleSoft Application Designer
Component Interfaces	X	X	PeopleSoft Application Designer
Components	X	X	PeopleSoft Application Designer
Field Formats	X	X	PeopleSoft Application Designer
Fields	X	X	PeopleSoft Application Designer
File Layouts	X	X	PeopleSoft Application Designer
File Type Codes	X	X	PeopleSoft Application Designer
HTML	X	X	PeopleSoft Application Designer
Images	X	X	PeopleSoft Application Designer
Menus	X	X	PeopleSoft Application Designer
Message Channel(s)	X	X	PeopleSoft Application Designer
Messages	X	X	PeopleSoft Application Designer

Definition Type	*Browser Client Definition Security*	*Windows Client Security*	*Associated Designer Tool*
Mobile Pages **Important!** PeopleSoft Mobile Agent is a desupported product. These features exist for backward compatibility only.	X	X	PeopleSoft Application Designer
Optimization Models	X	NA	PeopleSoft Application Designer
Pages	X	X	PeopleSoft Application Designer
Projects	X	X	PeopleSoft Application Designer
Queries	X	X	PeopleSoft Query
Records	X	X	PeopleSoft Application Designer
SQL	X	X	PeopleSoft Application Designer
Style Sheets	X	X	PeopleSoft Application Designer
Styles	X	X	PeopleSoft Application Designer
Translate Tables	X	X	PeopleSoft Application Designer
Tree Structures	X	X	PeopleSoft Tree Manager
Trees	X	X	PeopleSoft Tree Manager

You can restrict access to an entire definition type, such as records or pages, using the Permission Lists – PeopleTools page (**PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists** and click the PeopleTools tab). This works by controlling access to the PeopleSoft Application Designer functionality that works with a particular definition type. For example, if you don't want developers to use application engine programs, don't allow them to access PeopleSoft Application Engine.

See Defining Permissions

Chapter 21

Implementing Definition Security (Browser Client)

Understanding Definition Security (Browser Client)

This topic provides a feature overview of definition security in the browser client and performance considerations.

Feature Overview

Like definition security in the Windows client, definition security in the browser client enables you to create, copy and delete definition groups, as well as grant permission list access to definition groups in the PeopleSoft Pure Internet Architecture (PIA).

However, definition security in the browser client provides additional functionality that is not available in the Windows client application, including the ability to:

- Populate definition groups based on SQL-like selection criteria.
- Populate definition groups based on an Application Designer project.
- Create inclusion rules so that new objects added to the database and that meet an inclusion rule are added to a definition group. This includes the ability to:
    - Save selection criteria used to initially populate a definition group as an inclusion rule.
    - Re-process inclusion rules to ensure any new definitions that meet the original selection criteria are automatically added to the group, or that any new definitions that no longer meet the original selection criteria are automatically deleted from the group.
- Define row-level security.
- Enable or disable Secure by Default for all objects covered by Definition Security.
- View unsecured definitions or definitions that don't belong to a definition group.

See Comparing Browser Client and Windows Client Definition Security for a comparison of features between definition security in the Windows client and definition security in browser client modes. This topic also lists the definition types with which you can work in the different environments.

Development Considerations

Consider the following when developing and managing definition groups:

- Note the following about the search mechanism when searching for definitions to insert into definition groups:

- Search results do not include definitions that are already defined in a group.
- The system processes the operator value *like* as *contains*.

- Inclusion rules are applied only during definition group set up or via Process Scheduler.

Performance Considerations

Definition types with a large number of definitions may take longer to load in the browser client than the PIA timeout setting. Consider the following suggestions as you work:

- When creating definition groups, use filters to narrow your search criteria and thereby your returned results. Doing so can improve data loading time into PIA.
- The Unsecured Definitions page provides access to viewing definitions that do not belong to a definition group. No search filters are provided on this page, so by default all unsecured definitions covered by definition security load.

  Use this page as a monitoring tool after you have secured the definitions in the database.

Navigating Definition Security (Browser Client)

This topic lists the pages of Definition Security in three-tier mode. Use the pages to:

- View and add definition groups.
- Manage definition groups.
- Apply security to definition types.
- Dynamically update definition groups.
- View unsecured definitions.

Common Elements Used to Create and Manage Definition Groups (Browser Client)

These fields and controls appear frequently on pages used to create and manage definition groups in browser client definition security:

Term	***Definition***
Definition Group	Displays the name of the definition group with which you are currently working.
Delete	Click the button to delete selected items from a grid.
Object Name	Displays the name of the object type.

Term	*Definition*
Notify	Click the button to access the Send Notification page to send an email notification to an individual or group when a relevant event or update has occurred.
Return to Search	Click the link to return to the Definition Group Search page.
Save	Click the button to save any changes on a page. The button is enabled only if changes have been made to a page.
Save As	Click the button to access the Definition Group Save As page to copy a definition group.
Select All	Click the button to select all items in a grid.
Un-select All	Click the button to de-select items in a grid.

These navigation links appear frequently at the bottom of pages used to create and manage definition groups:

Term	*Definition*
Definition Inclusion Rules	Click the link to access the Definition Inclusion Rules page to view, update or inclusion rules for a definition group.
Group Content Detail	Click the link to access the Group Content Detail page to view, insert or delete definitions from a definition group.
Group Content Summary	Click the link to access the Group Content Summary page to view definition types in a group, the number of definitions of each type in a group, and the number of definitions that have be dynamically included in the group through inclusion rules.
Group Permissions	Click the link to access the Group Permissions page to view the permission lists associated with a definition group.
Group Users	Click the link to access the Group Users page to view the users with access to a definition group and the permission list to which they are assigned that is enabling access.

Viewing and Adding Definition Groups

The following table lists and briefly describes the pages for viewing and adding definition groups in browser client definition security. These pages are located in the PTDEFSECSRCH component.

Page	Object ID	Description	Navigation
Definition Group Search	PTDEFSECSRCH	• Search for and view a definition group. • Access the Add New Definition Groups page to add a definition group. • Access the Add New Definition Group from Project page to add a definition group from a project. • Delete a definition group.	Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**
Add New Definition Groups	PTDEFSECSRCH_ADD	Add a new definition group.	1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.** 2. Click the Add New Definition Group link.
Add New Definition Group from Project	PTDEFSECSRCH_ADD_P	Add a new definition group from a PeopleSoft Application Designer projects.	1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.** 2. Click the Add New Definition Group from Project link.

Managing Definition Groups

The following table lists and briefly describes the pages for managing definition groups in browser client definition security. These pages are located in the PTDEFSEC component.

Page	Object ID	Description	Navigation
Group Content Summary	PTDEFSECCNT	For a selected definition group, view the following: • Definition types in the group. • Number of definitions of each type. • Number of definitions that have been dynamically included in a group (through inclusion rules).	1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.** 2. In the Search Results grid, click the name of a definition group.

Page	*Object ID*	*Description*	*Navigation*
Group Content Detail	PTDEFSECGRP	• View all definitions in a definition group that are of a specific type. • Access the Insert Definitions page to insert definitions into a group. • Delete definitions from a group.	1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.** 2. In the Search Results grid, click the name of a definition group. The Group Content Summary page appears. 3. Do one of the following: • Click the Group Content Detail tab. • In the Definition Type Counts grid, click an object name.
Insert Definitions	PTDEFSECINSRT	Insert definitions into a definition group.	1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.** 2. In the Search Results grid, click the name of a definition group. The Group Content Summary page appears. 3. Do one of the following: • Click the Group Content Detail tab. • In the Definition Type Counts grid, click an object name. The Group Content Detail page appears. 4. From the Object drop-down list, select the definition type of the definition(s) to insert. 5. Click the **Insert Definitions** button.

Page	*Object ID*	*Description*	*Navigation*
Definition Inclusion Rules	PTDEFSECINRL	View and manage rules to dynamically add definitions to a definition group.	1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.** 2. In the Search Results grid, click the name of a definition group. The Group Content Summary page appears. 3. Click the Definition Inclusion Rules tab.
Group Permissions	PTDEFSECPERM	• View permission lists defined for a group. • Define permission list access for a definition group.	1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.** 2. In the Search Results grid, click the name of a definition group. The Group Content Summary page appears. 3. Click the Group Permissions tab.
Group Users	PTDEFSECGRPUSERS	View the user IDs and their associated permission lists that can access the definitions in a definition group.	1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.** 2. In the Search Results grid, click the name of a definition group. The Group Content Summary page appears. 3. Click the Group Users tab.

Applying Security to Definition Types

The following table lists and briefly describes the page for securing definition types within definition groups in browser client definition security. The page is located in the PTDEFTYP component.

Page	Object ID	Description	Navigation
Definition Types	PTDEFSECTYP	• View definition types that you can secure. • Apply "Secure by Default" security to definition types. • Apply row-level security to definition types.	Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Types.**

Dynamically Updating Definition Groups

The following table lists and briefly describes the page for dynamically updating definition groups in browser client definition security. The page is located in the PTDEFSECINRL_P component.

Page	Object ID	Description	Navigation
Inclusion Processing	PTDEFSECINRL_P	Run the delivered application engine program, *Inclusion_ Processing,* to run inclusion rules defined for a group and dynamically update a group with definitions added to the database that meet at defined inclusion rules for a definition group.	Select **PeopleTools** > **Security** > **Definition Security** > **Run Security Inclusion Rules.**

Viewing Unsecured Definitions

The following table lists and briefly describes the pages for viewing unsecured definitions in browser client definition security. The pages are located in the PTDEFSEC_UNSECURED component.

Page	Object ID	Description	Navigation
Unsecured Definition Counts	PTDEFSECUSCNT	For a selected definition group, view the following: • Unsecured definition types in a definition group. • Number of definitions of each type.	Select **PeopleTools** > **Security** > **Definition Security** > **View Unsecured Definitions.**

Page	Object ID	Description	Navigation
Unsecured Definitions	PTDEFSEC_UNSECURED	View unsecured definitions by object.	1. Select **PeopleTools** > **Security** > **Definition Security** > **View Unsecured Definitions.** 2. Click the Unsecured Definitions tab.

Accessing Definition Groups (Browser Client)

This topic discusses how to:

- Use the Definition Group Search page.
- Search for a definition group.

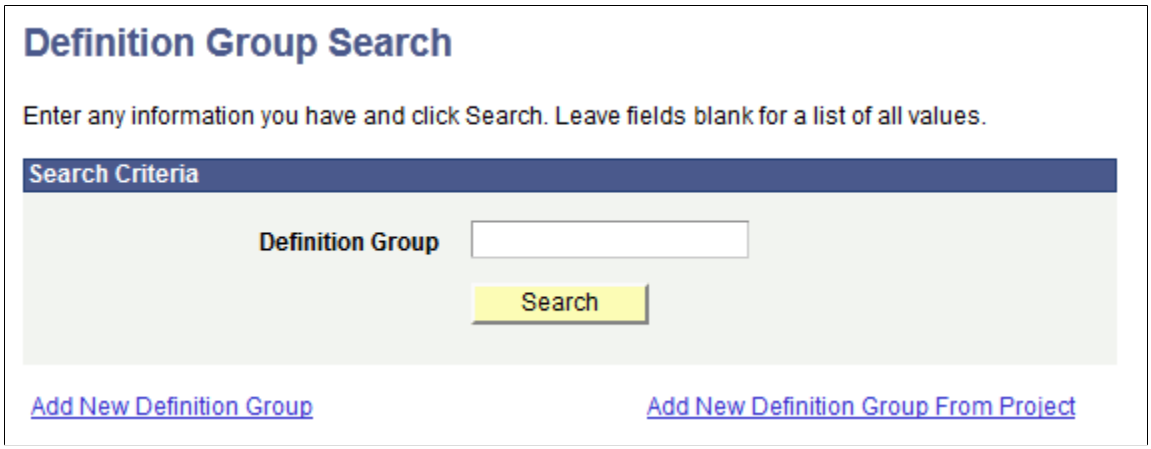
Using the Definition Group Search Page

Use the Definition Group Search page (PTDEFSECSRCH) to:

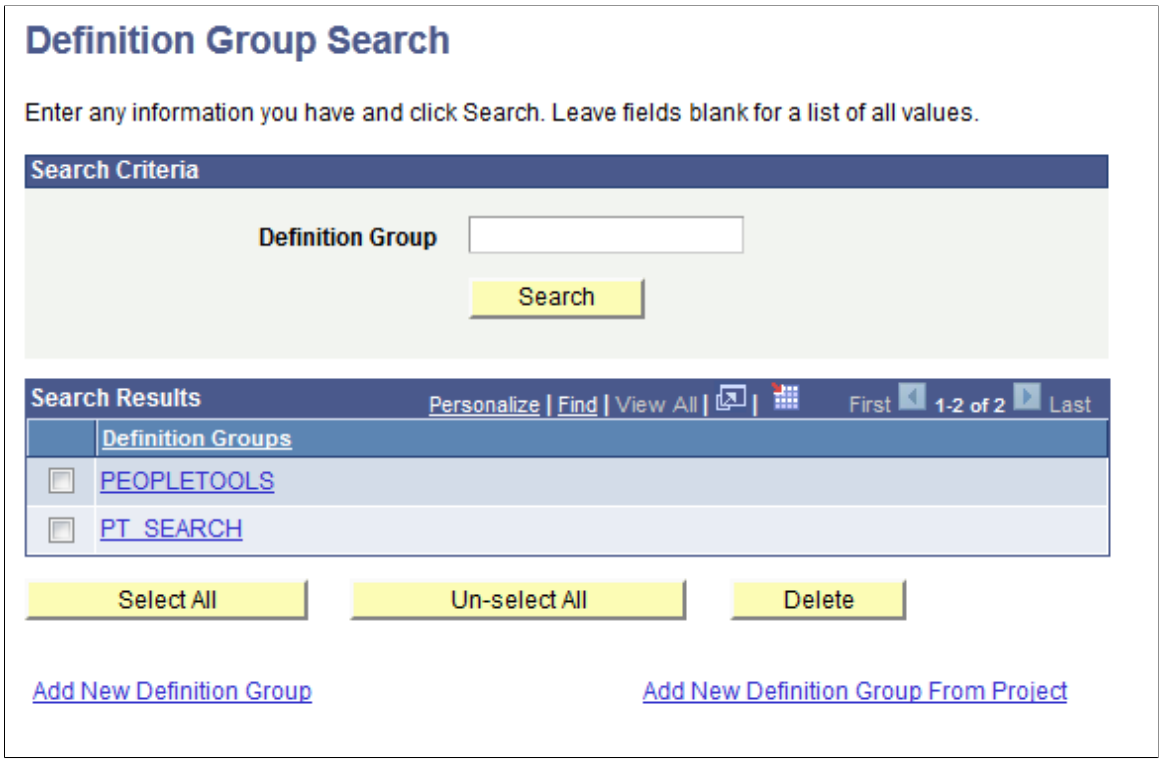
- Search for a definition group.
- Access the Add New Definition Groups page to add a definition group.
- Access the Add New Definition Group from Project page to add a definition group from a project.
- Delete a definition group.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**

This example illustrates the fields and controls on the Definition Group Search page. You can find definitions for the fields and controls later on this page.



This example illustrates the Definition Group Search page showing the results of a search of the database for all definition groups.



Field or Control	***Description***
Definition Group	Enter the name of the definition group to search. Leave the field empty to search the database for all definition groups in the system.
Search	Click the **Search** button to perform the search.
Search Results grid	Displays the results of the search.
Add New Definition Group	Click the link to access the Add New Definition Group page to add a new definition group to the system.
Add New Definition Group from Project	Click the link to access the Add New Definition Group From Project page to add a new definition group from an PeopleSoft Application Designer project.

Related Links

Accessing Definition Groups (Browser Client)

Searching for Definition Groups

To search for a definition group:

1. Access the Definition Group Search page (**PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**
2. In the Definition Group field enter the name of the definition group or leave the field blank to search the database for all values.
3. Click the **Search** button.

   The results appear in the Search Results grid.
4. Click the name of a definition group.

   The Definition Groups – Group Content Summary page appears.

Viewing Definition Groups (Browser Client)

This topic describes how to:

- Use the Group Content Summary page.
- Use the Group Content Details page.

Understanding Viewing Definition Groups

Use the Group Content Summary and Group Content Details page to view definition types and definitions defined in a definition group.

Related Links

Common Elements Used to Create and Manage Definition Groups (Browser Client)

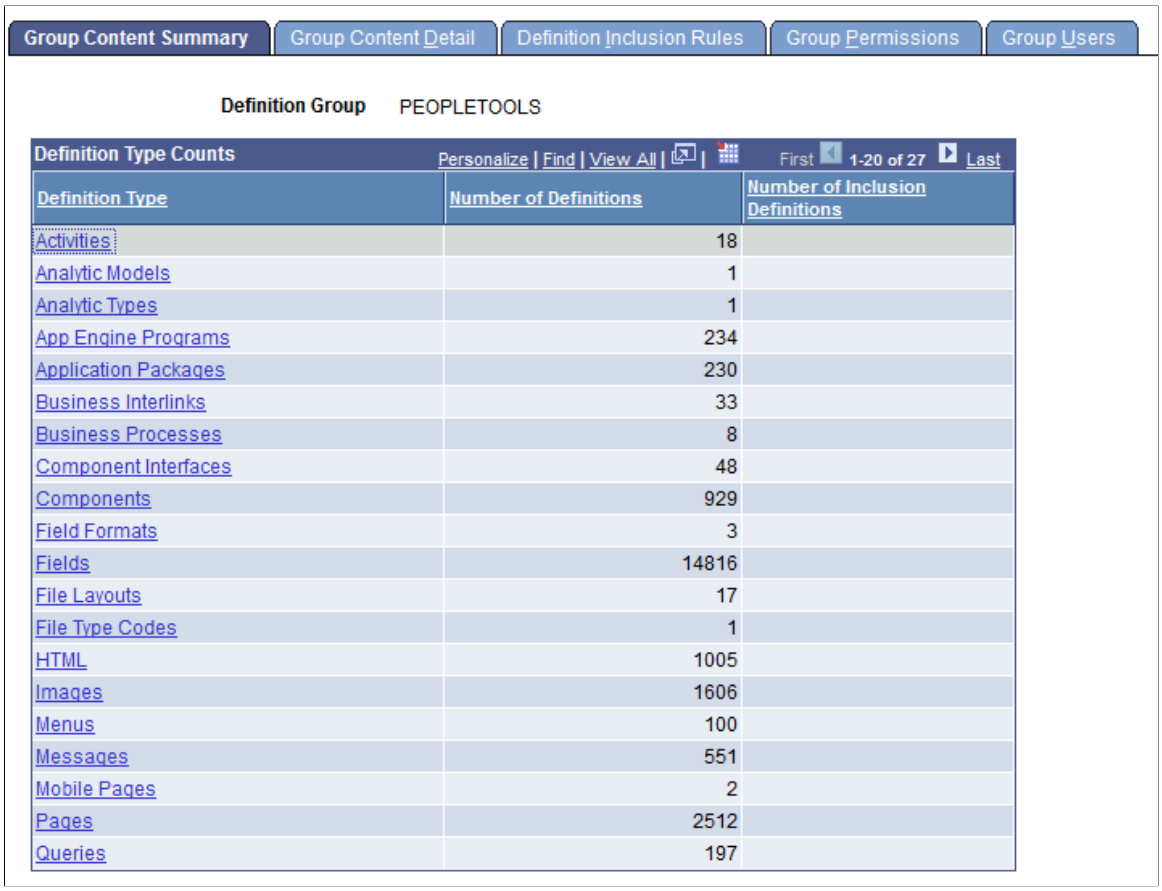
Using the Group Content Summary Page

Use the Group Content Summary page (PTDEFSECCNT) to view the following information for a selected definition group:

- Definition types in the group.
- Number of definitions of each type in the group.
- Number of definitions that have been dynamically included in the group through inclusion rules.

To access the page select Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups** and search for and select a definition group.

This example illustrates the fields and controls on the Group Content Summary page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Definition Type	Displays the object or definition type name. Click the link to access the Group Content Detail page to view a list of specific definitions included in the definition group.
Number of Definitions	Displays the number of definitions in a definition group for a specific definition type.
Number of Inclusion Definitions	Displays the number of definitions in a definition group added using inclusion rules.

Using the Group Content Detail Page

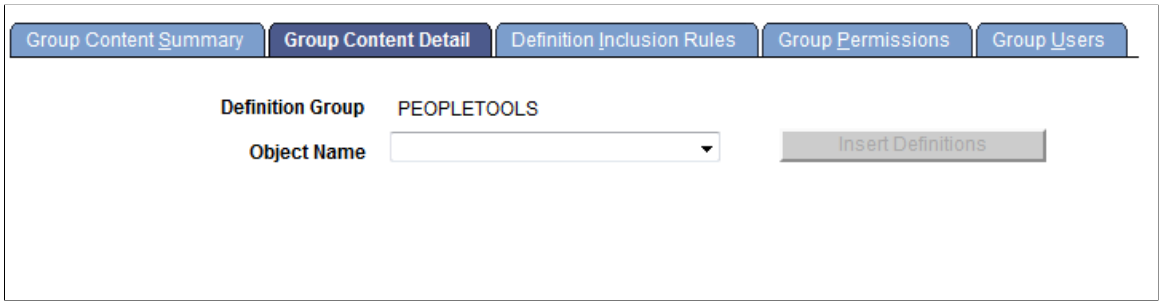
Use the Group Content Detail page (PTDEFSECGRP) to:

- View all definitions in a definition group that are of a specific type.
- Access the Insert Definitions page to insert definitions into a group.

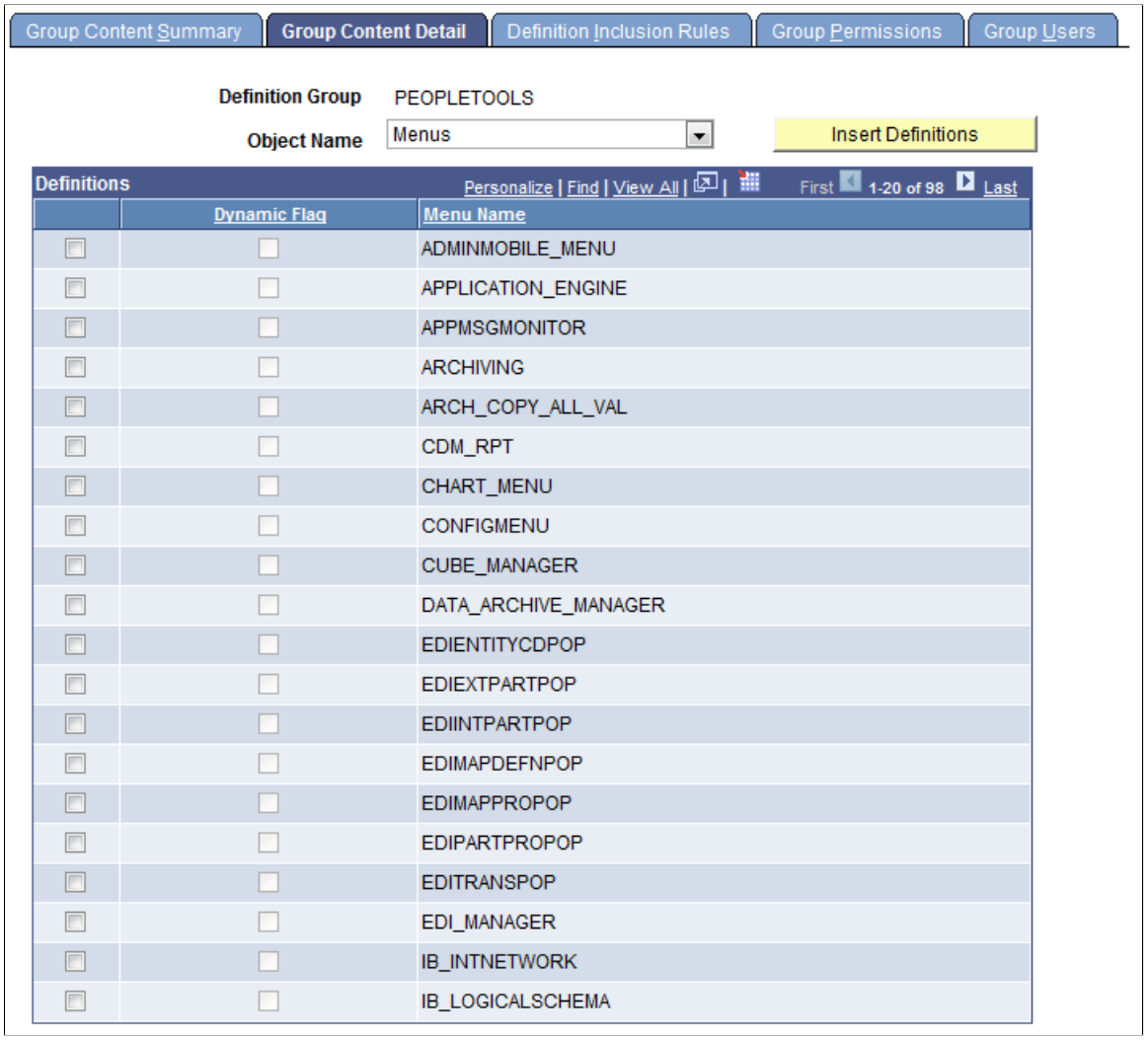
- Delete definitions from a group.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups**. Search for and select an existing definition group, or add a definition group to the system. Select the Group Content Detail tab.

This example illustrates the fields and controls on the Group Content Detail page for a new definition group. You can find definitions for the fields and controls later on this page.



This example illustrates the fields and controls on the Group Content Detail page for an existing definition group. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Object Name	From the drop-down list select the definition object with which to work.
Insert Definitions	Click the button to access the Insert Definitions page to insert definitions of the selected object type into the definition group and to create inclusion rules.
Dynamic Flag	The read-only field is selected when a definition has been added to a group as the result of an inclusion rule.
<Definition> Name	Displays the definitions in the definition group for the object selected in the **Object Name** field.

Adding Definition Groups (Browser Client)

This topic discusses how to:

- Add a definition group.
- Add a definition group from a project.

Related Links

Common Elements Used to Create and Manage Definition Groups (Browser Client)

Adding a Definition Group

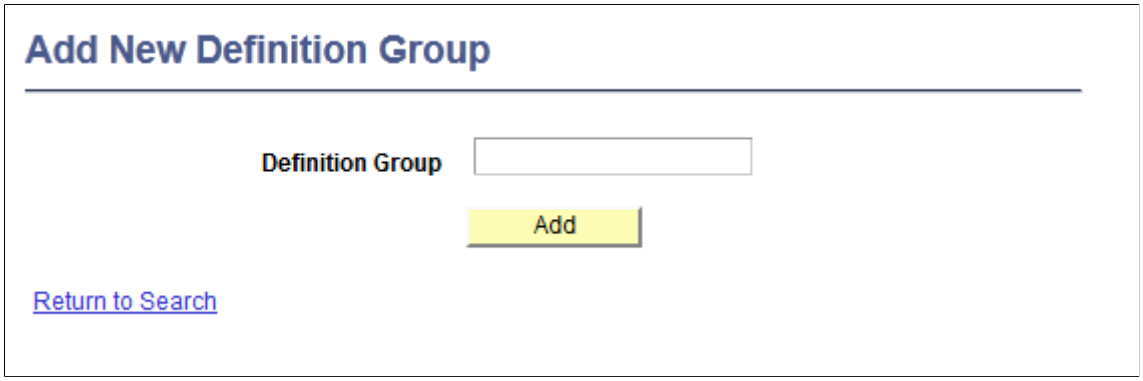
This section describes how to use the Add New Definition Group page and how to use the page to add a definition group to the database.

Using the Add New Definition Group Page

Use the Add Definition Group page (PTDEFSECSRCH_ADD) to add a definition group to the database.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups** and click the **Add New Definition Group** link.

This example illustrates the fields and controls on the Add New Definition Group page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Definition Group	Enter the name of the new definition group.
Add	Click the button to add the new definition group to the database. After you click the **Add** button the new definition group appears in the Definition Groups – Group Content Summary page.

Adding a Definition Group

To add a definition group:

1. Access the Add Definition Group page (select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups** and click the **Add New Definition Group from Project** link.)
2. In the **Definition Group** field, enter the name for the definition group.
3. Click the **Add** button.

The Definition Groups – Content Summary page appears.

Adding a Definition Group from a Project

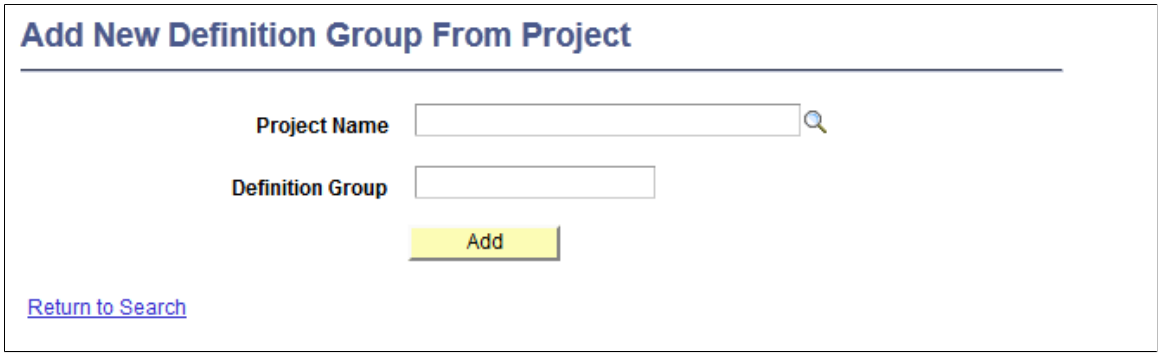
This section describes how use the Add New Definition from Project page and how to use the page to add a definition group from a PeopleSoft Application Designer project.

Using the Add New Definition Group from Project Page

Use the Add Definition Group from Project page (PTDEFSECSRCH_ADD_P) to add a definition group to the database from a PeopleSoft Application Designer project.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups** and click the **Add New Definition Group from Project** link.

This example illustrates the fields and controls on the Add New Definition Group From Project. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Project Name	Enter the name of the project or click the Lookup button to select one.
Definition Group	Define the name for the definition group. When you enter or select a project name the system populates this field with the project name by default. You can accept the default value or enter a different name.
Add	Click the button to add the definition group. After you click the button the new definition group appears in the Definition Groups – Group Content Summary page.

Adding a Definition Group from a Project

To add a definition group from a PeopleSoft Application Designer project:

1. Access the Add Definition Group from Project page (select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups** and click the **Add New Definition Group from Project** link).
2. In the **Project Name** field:
   - Enter the name of the project that contains the definition group to add to the database, or
   - Click the **Lookup** button to search the database for the project that contains the definition group to add to the database.
3. In the **Definition Group** field, enter the name for the definition group.

   By default the system populates the field with the project name.
4. Click the **Add** button.

The Definition Groups – Content Summary page appears.

Inserting Definitions into Definition Groups (Browser Client)

This topic describes how to:

- Use the Insert Definitions page.
- Insert definitions into definition groups.

Understanding Inserting Definitions into Definition Groups

To insert definitions into a definition group you define SQL-like statements to search for the definitions to add.

By default the key field for the record is displayed as a search field. You can add additional fields to include in your search criteria by adding a row to the Search Criteria grid. When you add a row to the grid a drop-down list appears in the Field Name area, and you choose any of the record fields to define search criteria.

You can save any search statement as an inclusion rule. When you save a search statement as an inclusion rule, any definitions that are not already part of the definition group are added to the group. You can then view, edit, or delete the rule using the Definition Inclusion Rules page.

Using the Insert Definitions Page

Use the Insert Definitions page (PTDEFSECINSRT) to insert definitions into definition groups and save searches as inclusion rules.

- Insert definitions into definition groups.
- Save searches as inclusion rules.

To access the page:

1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**

   The Definition Groups Search page appears.

2. Access a definition group or add a definition group.

   See Accessing Definition Groups (Browser Client) or Adding Definition Groups (Browser Client).

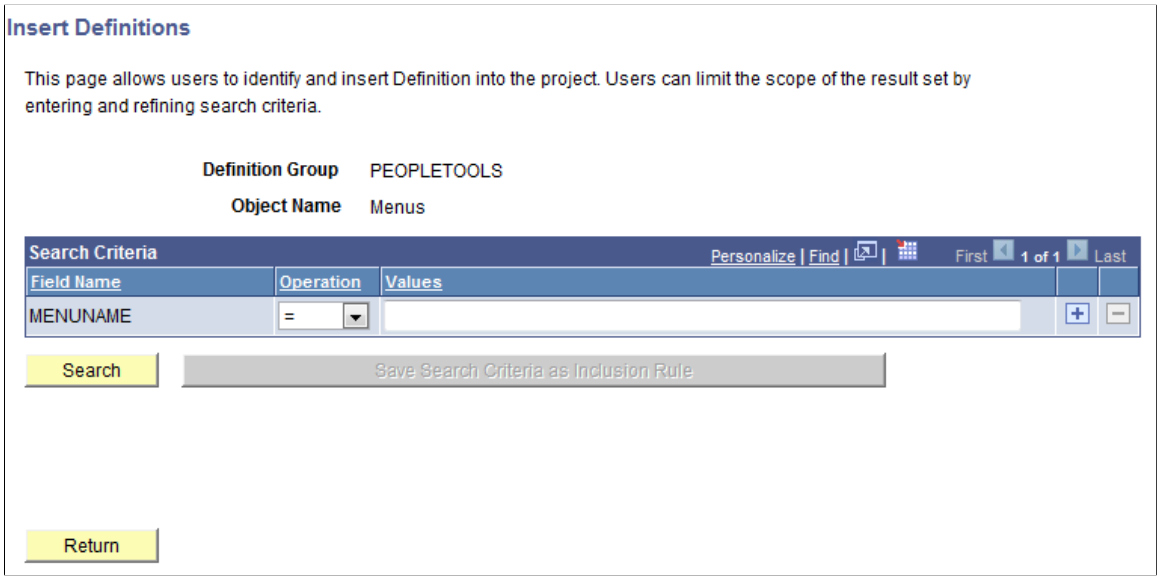
   The Group Content Summary page appears.

3. Click the Group Content Detail page.

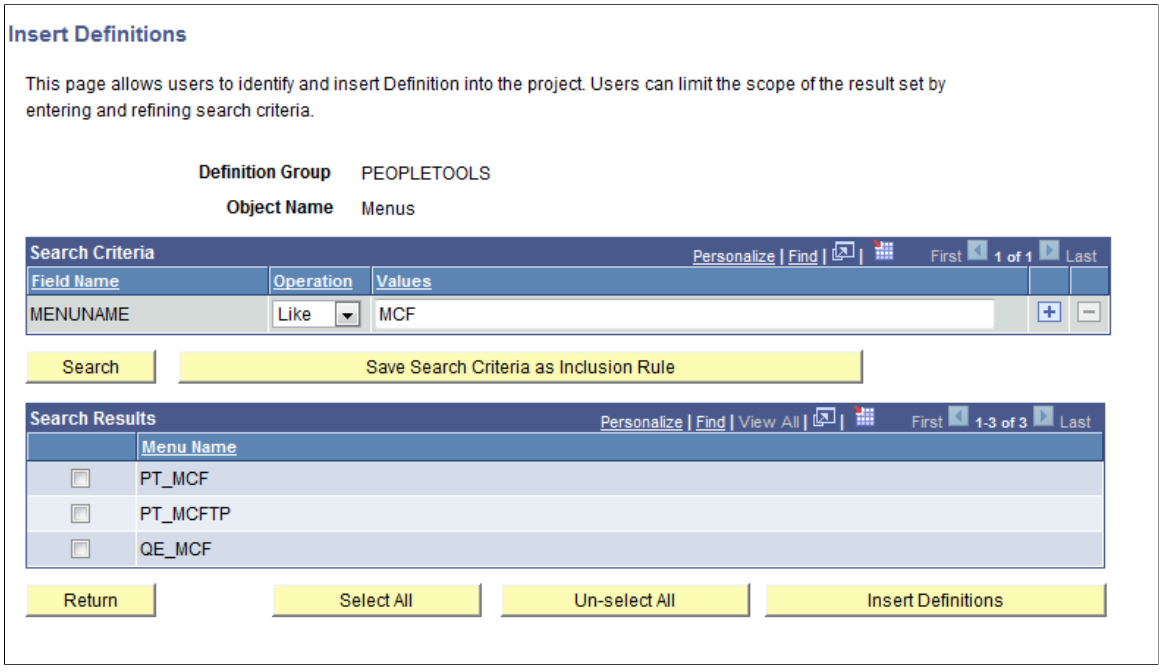
   In the **Object Name** field select an object type.

4. Click the **Insert Definitions** button.

This example illustrates the fields and controls on the Insert Definitions page when no search criteria has been defined and no search results are displayed. You can find definitions for the fields and controls later on this page.

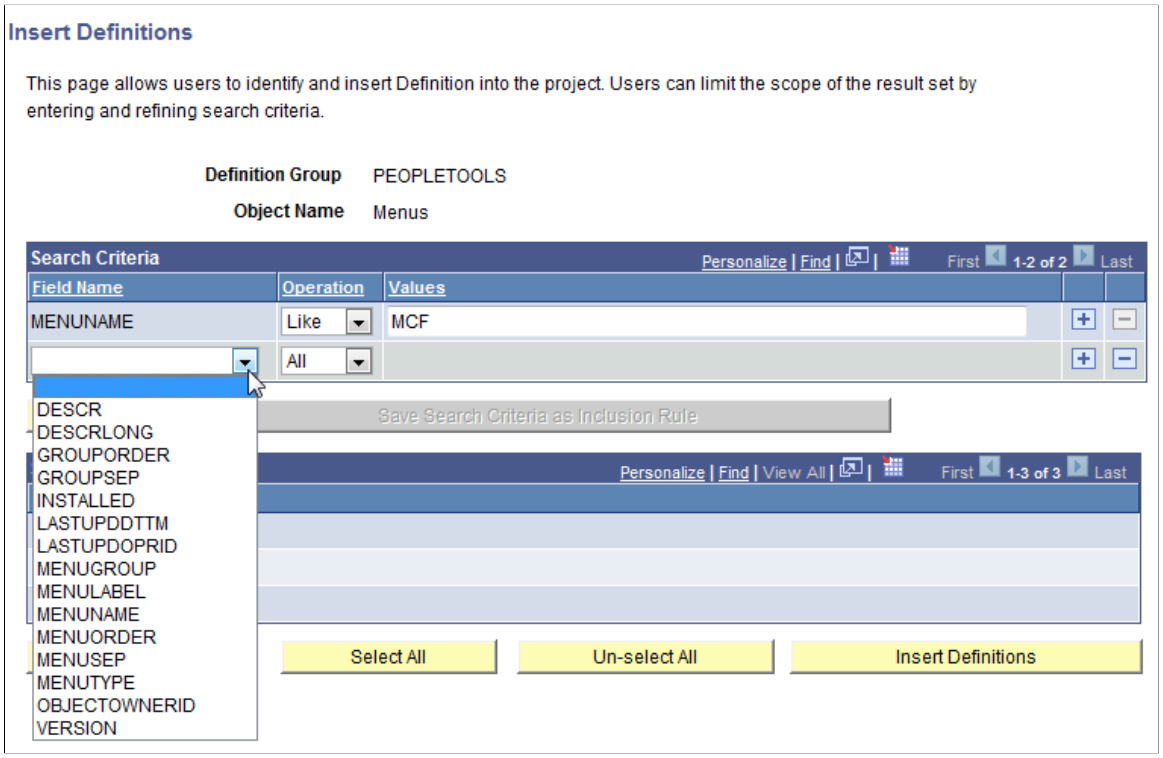


This example illustrates the fields and controls on the Insert Definitions page with search criteria defined and search results displayed. You can find definitions for the fields and controls later on this page.



Note: The system returns values of only those definitions that are not defined in the definition group.

This example shows a second row added to the Search Criteria grid for defining a second search statement. A drop-down list appears displaying the record field names you can use in the search statement.



Field or Control	***Description***
Field Name	Displays the field name from the definition. The definition name defaults in the first row of a search. You can insert additional rows to define additional search criteria using definition fields. As an example, if working with the menu definition type, you can define search criteria based on the menu group, menu label, menu type, and so on.

Field or Control	*Description*
Operation	From the drop-down list select an operator for the search. The operator is applied to the value defined in the Value field. The options are: • *!*= Not equal to. • < Less than. • <= Less than or equal to. • = Equal to. • > Greater than. • >= Greater than or equal to. • *All* Returns all definitions of the definition type not already defined in the group. • *Betwn* Between. Search for a value between a range of values. Use *AND* to separate range values. • *Like* Contains.
Value	Enter a value based on the operator selected from the **Operation** drop-down list.
Search	Click the **Search** button to perform a search.
Save Search Criteria as Inclusion Rule	Click the button to save a search as an inclusion rule. When you save a search as an inclusion rule, you can view the rule on the Definition Groups – Inclusion Rules page.
<Definition> Name	Displays the definition names returned by the search.
Return	Click the button to return to the Group Content Detail page.
Insert Definitions	Click the button to insert selected definitions in the Search Results grid into the definition group.

Inserting Definitions into Definition Groups

To insert a definition into a definition group:

1. Access the Insert Definitions page.
   a. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**

      The Definition Groups Search page appears.

b. Access a definition group or add a definition group.

See Accessing Definition Groups (Browser Client) or Adding Definition Groups (Browser Client).

The Group Content Summary page appears.

c. Click the Group Content Detail page.

In the **Object Name** field select an object type.

d. Click the **Insert Definitions** button.

2. Define a search statement.

a. In the Search Criteria grid, from the **Field Name** drop-down list, select a record field.

Note that for the first row the key record field appears by default and no drop-down list is available. To use a different field, click the **Insert Row** button (+).

b. From the **Operation** drop-down list, select an operator.

The operators that appear in the list are described in the previous section.

c. In the **Values** field, enter a search value.

The value to enter depends on the operator selected.

3. (Optional) To add additional search statements:

a. Add another row to the Search Criteria grid by clicking the **Insert Row** button (+).

b. Repeat Step 2.

4. Click the **Search** button.

The results appear in the Search Results grid.

5. Choose the definitions to insert into the definition group:

- Select individual definitions by selecting the box next to each definition to add, or
- Click **Select All** to choose all definitions in the grid.

6. Click the **Insert Definitions** button.

7. (Optional.) To save the search as an inclusion rule, click the **Save Search Criteria as Inclusion Rule** button.

8. Click **Return** to go back to the Group Content Detail page.

Related Links

Dynamically Adding Definitions to Definition Groups (Browser Client)

Dynamically Adding Definitions to Definition Groups (Browser Client)

This topic provides an overview of dynamically adding definitions to definition groups and discusses how to:

- Use the Definition Inclusion Rules page.
- Use the Inclusion Processing page.
- Use the Process Scheduler Request page.
- Create inclusion rules.
- Modify inclusion rules.
- Delete inclusion rules.

Understanding Dynamically Adding Definitions to Definition Groups

In three-tier definition security you can create inclusion rules to dynamically add definitions to definition groups.

This table describes the pages used to create, manage and run inclusion rules:

Page Name	***Object ID***	***Description***
Insert Definitions	PTDEFSECINSRT	Use the page to search for definitions to add to a definition group by creating SQL-like search statements. The system provides an option to save these statements as inclusion rules.
Definition Inclusion Rules	PTDEFSECINRL	Use the page to view, modify, and delete inclusion rules.
Inclusion Processing	PTDEFSECINRL_P	Use the page to launch inclusion rule processing on the database.
Process Scheduler Request	PRCSRQSTDLG	Use the page to process inclusion rules on the database using the PeopleSoft-delivered application engine program, *PTDEFSECINRL*. The *PTDEFSECINRL* process dynamically adds any definitions that have been added to the database that meet the inclusion rule(s).

Using the Definition Inclusion Rules Page

Use the Definition Inclusion Rules page (PTDEFSECINRL) to manage inclusion rules, including viewing, modifying, and deleting them.

To access the page:

1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**

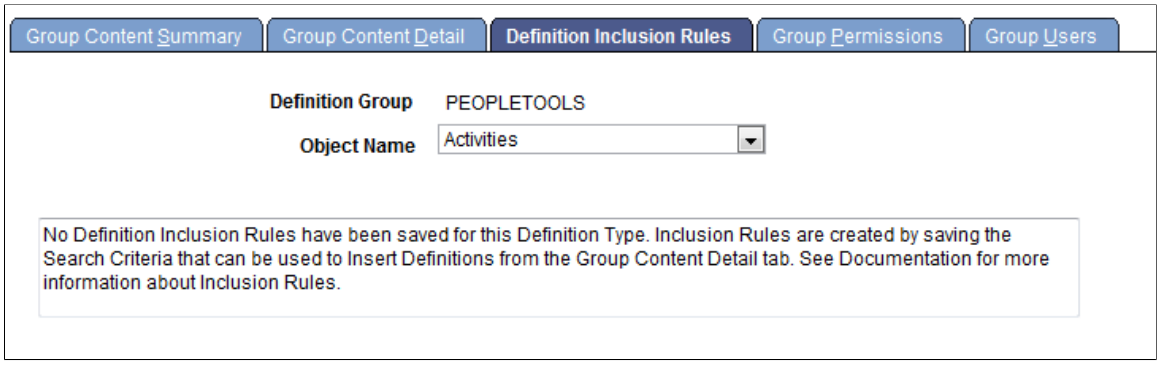
   The Definition Groups Search page appears.

2. Search for a definition group or add one to the system.

   The Group Content Summary page appears.

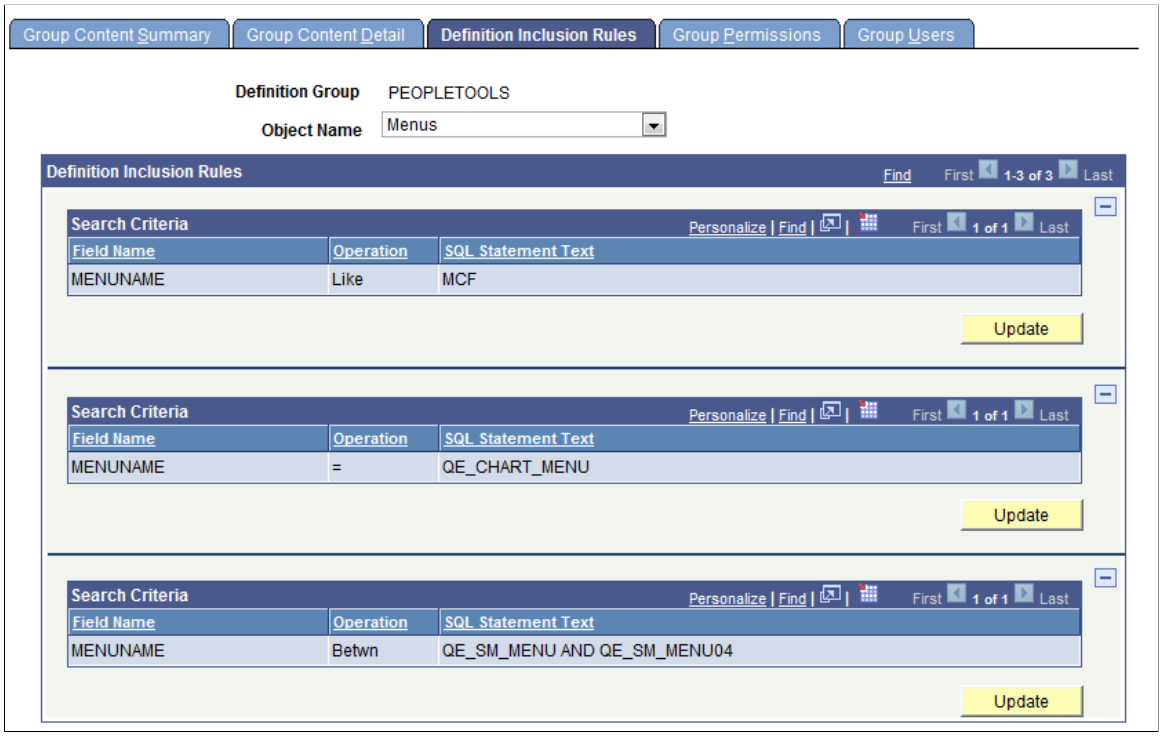
3. Click the Definition Inclusion Rules tab.

This example illustrates the fields and controls on the Definition Inclusion Rules page for a definition object for which no inclusion rules have been defined. You can find definitions for the fields and controls later on this page.



The previous example shows the Definition Inclusion Rules page when no inclusion rules have been defined for an object name (definition type).

This example illustrates the fields and controls on the Definition Inclusion Rules page when inclusion rules are defined for a definition object. You can find definitions for the fields and controls later on this page.



The previous example shows the Definition Inclusion Rules page when inclusion rules are defined for a definition type.

Field or Control	***Description***
Object Name	From the drop-down list, select the definition object type for which you want to display inclusion rules.
Field Name	Displays the field name from the definition.
Operation	Displays the operator used in the rule.
SQL Statement Text	Displays the rule statement.
	Click the button to delete an inclusion rule.
Update	Click the button to access the Insert Definitions page to modify an inclusion rule.

Using the Inclusion Processing Page

Use the Inclusion Processing page (PTDEFSECINRL_P) to launch inclusion rule processing on the database.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **Run Security Inclusion Rules.** Enter a run control ID or select an existing one.

This example illustrates the fields and controls on the Inclusion Processing page. You can find definitions for the fields and controls later on this page.

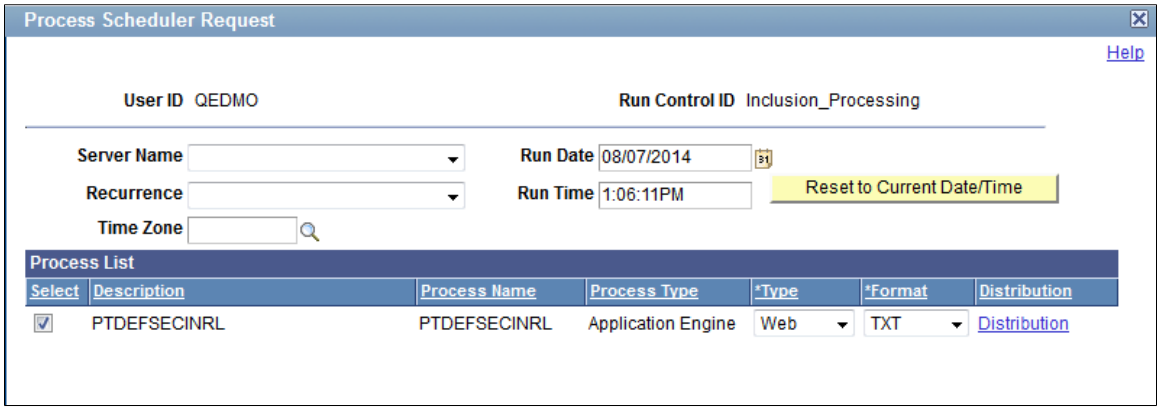
Field or Control	*Description*
Run Control ID	This field displays the Run Control ID from the Search Dialog.
Report Manager	Click the link to access Report Manager to view report information after you run the process.
Process Monitor	Click the link to access Process Monitor to monitor the status of the process.
Run	Click the button to access the Process Scheduler Request page to start inclusion processing.

Using the Process Scheduler Request Page

Use the Process Scheduler Request page (PRCSRQSTDLG) to initiate definition security inclusion rule processing.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **Run Security Inclusion Rules** and click the **Run** button.

This example illustrates the fields and controls on the Process Scheduler Request page.



The system auto-populates the Process List grid with the PeopleSoft-delivered application engine program, *PTDEFSECINRL,* used for all inclusion rule processing.

Information about the fields and controls that appear on this page are documented elsewhere in the product documentation. See “Scheduling Process Requests” (Process Scheduler).

Creating Inclusion Rules

You create inclusion rules using the Insert Definitions page when you define criteria for searching for definitions to insert into definition groups. The page features a **Save Search Criteria as Inclusion Rule** button. After you define a search statement, click the **Save Search Criteria as Inclusion Rule** button to save the criteria as an inclusion rule.

You can view inclusion rules in the Insert Definitions page and on the Definition Inclusion Rules page.

See Inserting Definitions into Definition Groups (Browser Client) and Using the Definition Inclusion Rules Page

Modifying Inclusion Rules

To modify an inclusion rule:

1. Access the Definition Inclusion Rules page.

   See Using the Definition Inclusion Rules Page for information on how to access the page.

2. From the **Object Name** drop-down list, select the definition type of the inclusion rule.

   The inclusion rules defined for the definition type appear in the Definition Inclusion Rules grid.

3. Locate the inclusion rule to modify and click the Update button.

   The Insert Definition page appears.

4. Modify the inclusion rule as necessary.

5. Click the **Return** button.

   The Definition Inclusion Rules page appears.

6. Click the **Save** button.

Deleting Inclusion Rules

To delete an inclusion rule:

1. Access the Definition Inclusion Rules page.

   See Using the Definition Inclusion Rules Page for information on how to access the page.

2. From the **Object Name** drop-down list, select the definition type of the inclusion rule.

   The inclusion rules defined for the definition type appear in the Definition Inclusion Rules grid.

3. Locate the inclusion rule to delete and click the **Delete Row** button.

4. Click the **Save** button.

Managing Definition Group Security (Browser Client)

This topic provides an overview of managing definition group security and discusses how to:

- Use the Definition Types page.
- Use the Group Users page.
- Use the Group Permission page.
- Enable Secure by Default for a definition type.
- Define component row-level security for a definition type.
- Define permission list access to a definition group.

Understanding Managing Definition Group Security

There are three mechanisms for securing definition types and definition groups: secure by default, component row-level security, and permission lists.

Term	***Definition***
Secure by Default (Definition Type)	When you enable “secure by default” for a definition type, definitions of the type are only accessible by the OPRID that last updated it (creator), or if it is associated with the primary permission list to which the OPRID belongs. **Note:** When you enable secure by default for a definition type you must explicitly grant permission list access to users. Use the Definition Types page described later in the topic to enable Secure by Default for definition types.

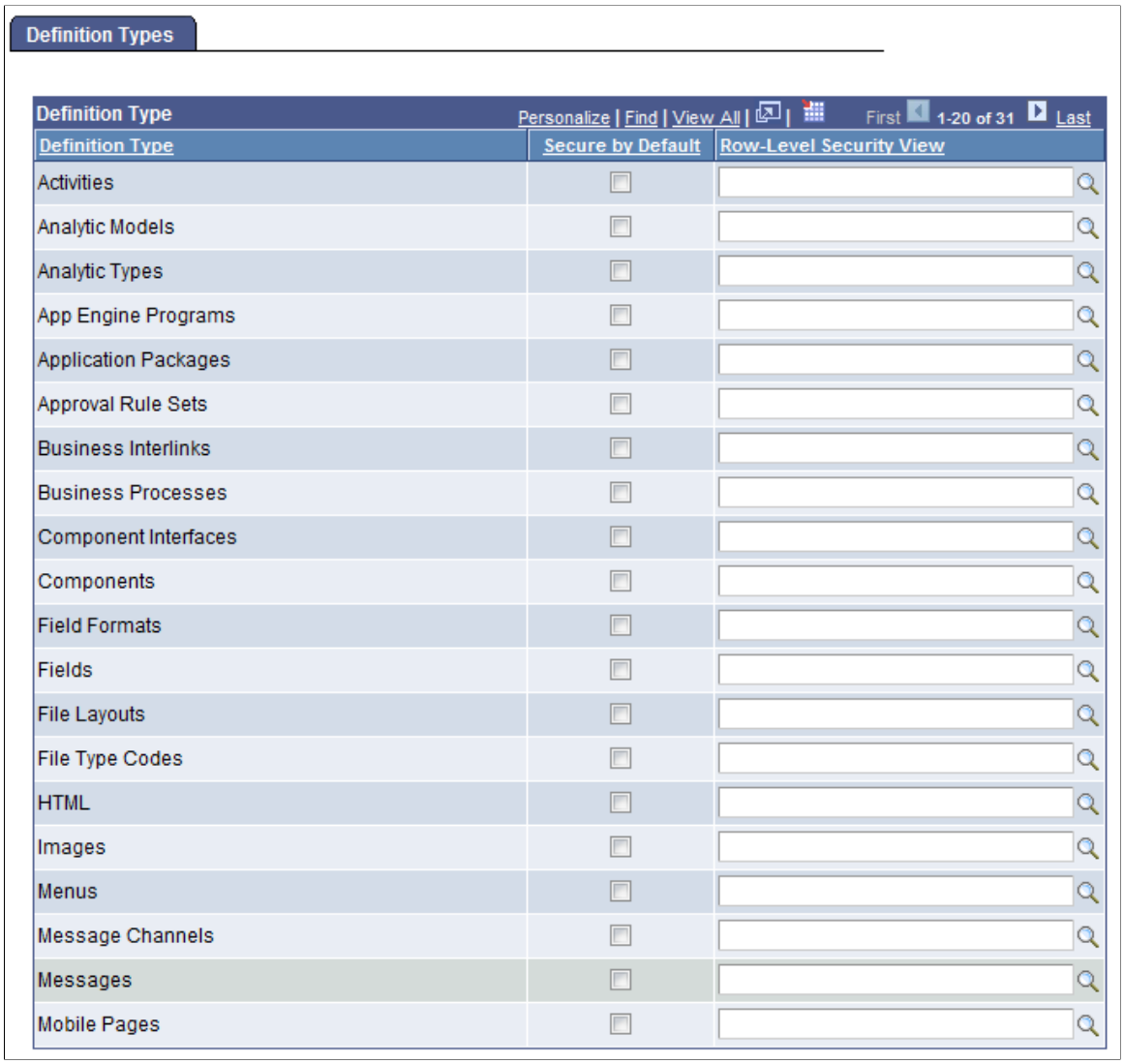
Term	*Definition*
Component Row-Level Security (Definition Type)	Associate component row-level security with definition types to limit access to data. You can specify dynamic views for a record for a definition type to control access. Use the Definition Types page described later in the topic to specify dynamic views for definition types.
Permission Lists (Definition Group)	You can assign permission list access to definition groups, providing users assigned to a permission list full or read-only access to a definition group. Use the Group Users page described later in this topic to view the permission lists and their associated users with access to a definition group. Use the Group Permission page described later in this topic to define this access.

Using the Definition Types Page

Use the Definition Types page (PTDEFSECTYP) to enable Secure by Default and set row-level security for definition types.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Types.**

This example illustrates the fields and controls on the Definition Types page. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Definition Type	Displays the name of the definition type.
Secure by Default	Select the box to enable Secure by Default for the definition type. When you select this control the definitions of the definition type are accessible only to those given access.
Row-Level Security View	From the drop-down list, select the record to which to apply row-level security. Note that only dynamic view records are available to select.

Using the Group Permissions Page

Use the Group Permissions page (PTDEFSECPERM) to view and manage permission list access to definition groups.

Full access is the default for a permission list defined for a definition group.

To access the page:

1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**

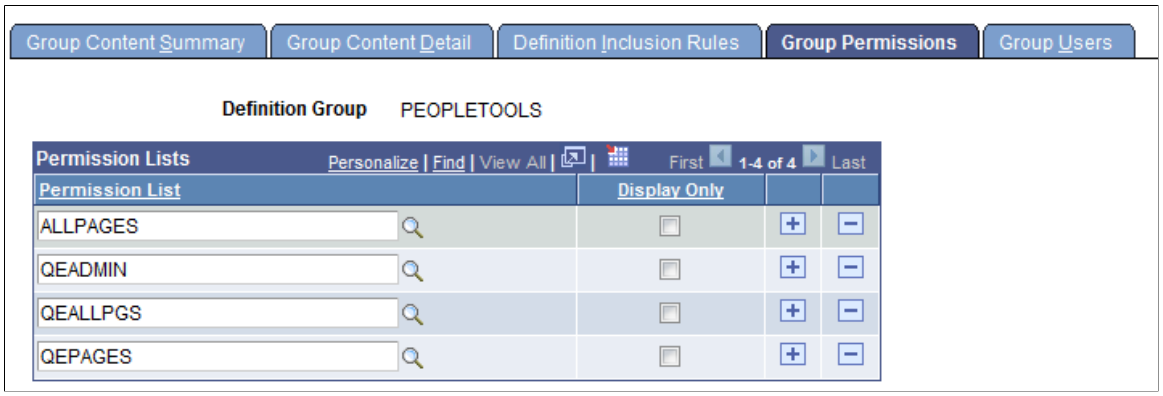
   The Definition Groups Search page appears.

2. Search for a definition group or add one to the system.

   The Group Content Summary page appears.

3. Click the Group Permissions tab.

This example illustrates the fields and controls on the Group Permissions page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Permission List	Click the Lookup button to search for a permission list to assign to the definition group.
Display Only	Select the box to allow read-only access to the definition group for users belonging to the permission list.

Using the Group Users Page

Use the Group Users page (PTDEFSECGRPUSERS) to view the user IDs with access to a definition group and the associated permission list with which they have gained access.

To access the page:

1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**

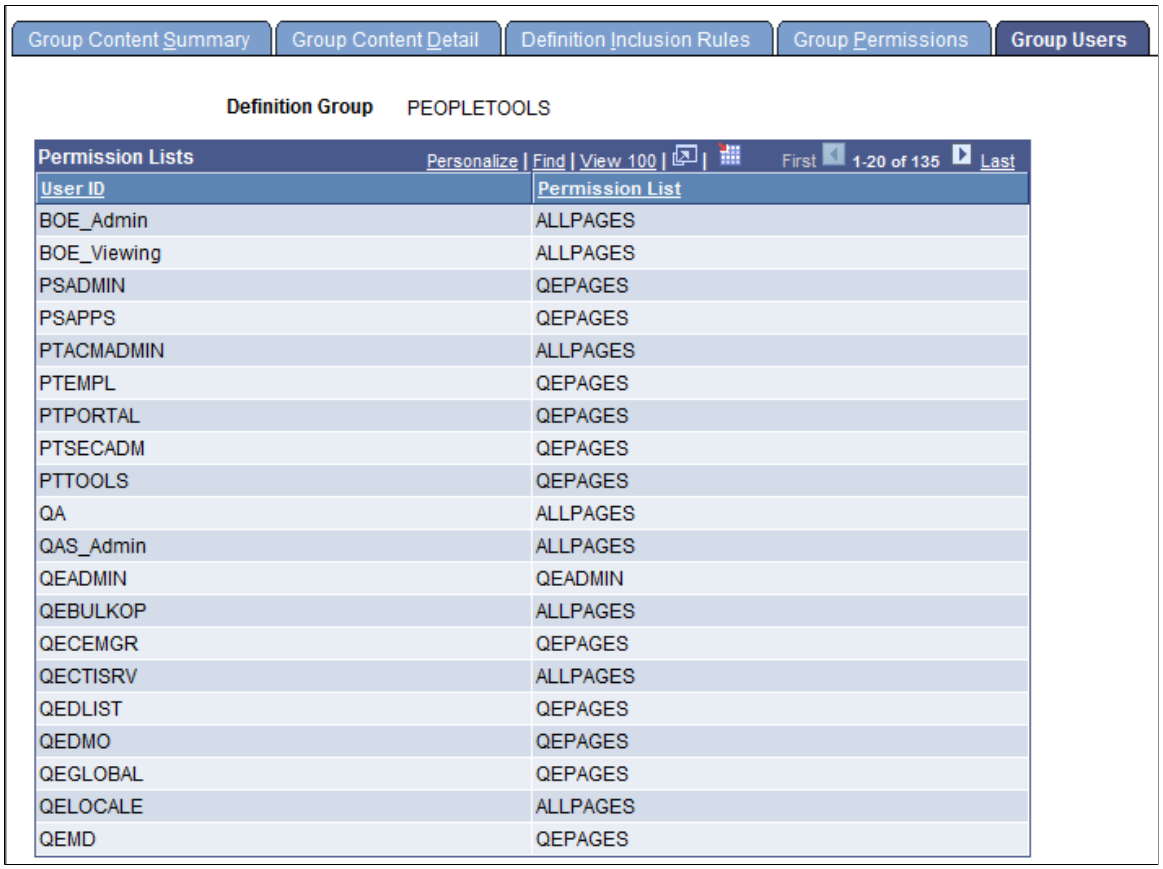
   The Definition Groups Search page appears.

2. Search for a definition group or add one to the system.

   The Group Content Summary page appears.

3. Click the Group Users tab.

This example illustrates the fields and controls on the Group Users page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
User ID	The user ID of a user with access to the definitions in the definition group.
Permission List	The name of the permission list to which the user belongs and with which the user is gaining access to the definition group.

Enabling Secure by Default for Definition Types

Remember that if you enable secure by default for a definition type, you must explicitly set permission list access to the definition type for users to be able to access the definitions of that type.

To enable Secure by Default for a definition group:

1. Access the Definition Types page.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Types.**

2. Select the Secure by Default box for each definition type to apply secure by default.
3. Click the **Save** button.

Setting Component Row-Level Security for Definition Types

To set component row-level security for definition types:

1. Access the Definition Types page.

   To access the page select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Types.**

2. In the **Row Level Security View** field, enter the name of a dynamic view record defined for component row-level security, or click the **Lookup** button to search for one.
3. Click the **Save** button.

Defining Permission List Access to Definition Groups

To define permission list access to a definition group:

1. Access the Group Permission page:
   a. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**

      The Definition Groups Search page appears.

   b. Search for a definition group or add one to the system.

      The Group Content Summary page appears.

   c. Click the Group Permissions tab.

2. In the Permission List field, select a permission list to assign to the definition group.
3. (Optional) Select the **Display Only** box to limit access to read-only.
4. Click the **Save** button.

To view permission lists and associated users with access to a definition group, use the Group Users page.

Viewing Unsecured Definitions (Browser Client)

This topic discusses how to:

- Use the Unsecured Definition Counts page.

- Use the Unsecured Definitions page.

Understanding Viewing Unsecured Definitions

View the number of unsecured definitions for each definition type, as well as the names of the unsecured definitions.

Using the Unsecured Definition Counts Page

Use the Unsecured Definition Counts page (PTDEFSECUSCNT) to view the following:

- Unsecured definition types in the database.
- Number of definitions of each type.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **View Unsecured Definitions.**

This example illustrates the fields and controls on the Unsecured Definition Counts page. You can find definitions for the fields and controls later on this page.

Field or Control	*Description*
Activities	Displays the definition type name. Click the link to access the Unsecured Definitions page to view a list of unsecured definitions for the type.
Number of Rows	Displays the number of unsecured definitions for the definition type.

Using the Unsecured Definitions Page

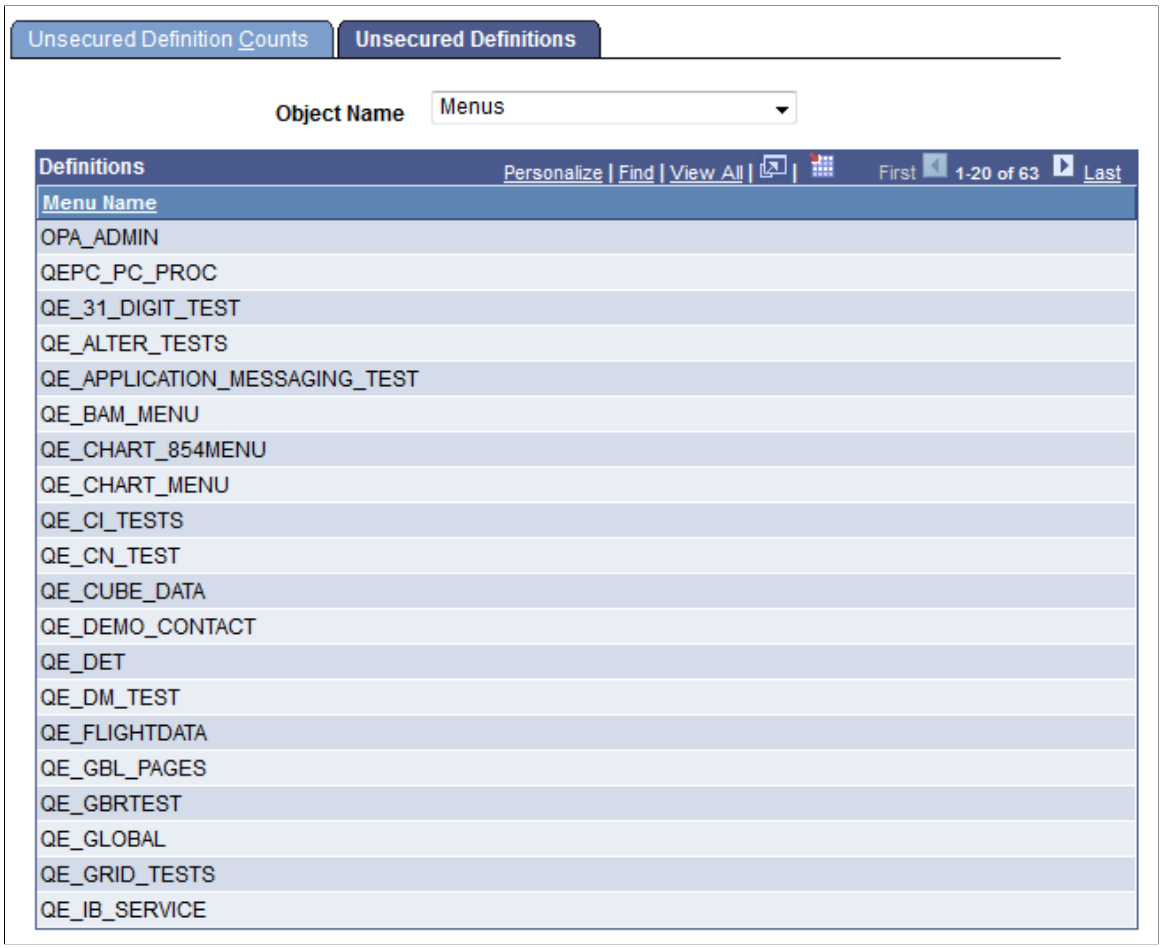
Use the Unsecured Definitions page (PTDEFSEC_UNSECURED) to view unsecured definitions for a definition type.

To access the page select **PeopleTools** > **Security** > **Definition Security** > **View Unsecured Definitions** and click the Unsecured Definitions tab.

This example illustrates the fields and controls on the default view of the Unsecured Definitions page. You can find definitions for the fields and controls later on this page.

The previous example shows the default view of the Unsecured Definitions page. By default, no definition type or object name is selected and no unsecured definitions display on the page.

This example illustrates the fields and controls on the Unsecured Definitions page when an object name or definition type is selected. You can find definitions for the fields and controls later on this page.



Field or Control	***Description***
Object Name	From the drop-down list, select a definition type.
<Definition> Name	Displays definitions in the database that are not secured by definition security.

Copying Definition Groups (Browser Client)

This topic discusses how to use the Definition Group Save As page to copy definition groups.

Understanding Copying Definition Groups

You can copy/clone a definition group and save it under a different name.

When you copy a definition group, all definition types, inclusion rules, and permissions are copied.

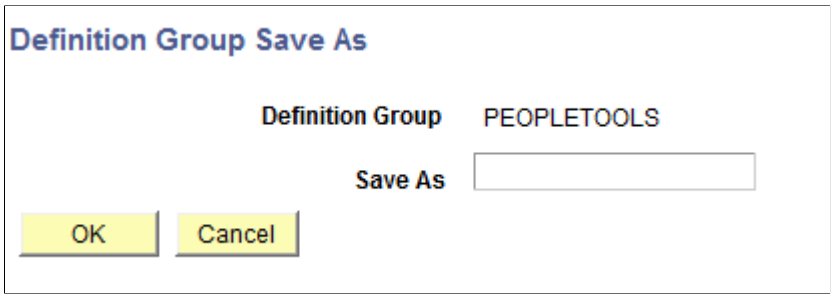
Using the Definition Group Save As Page

Use the Definition Group Save As page (PTDEFSEC_SAVEAS) to copy a definition group and define the copied group with a new name.

To access the page click the **Save As** button at the bottom of any of the pages listed in the following table:

Page	*Object ID*	*Source for Additional Information*
Group Content Summary page	PTDEFSECCNT	See Using the Group Content Summary Page.
Group Content Detail page	PTDEFSECGRP	See Using the Group Content Detail Page.
Definition Inclusion Rules page	PTDEFSECINRL	See Using the Definition Inclusion Rules Page.
Group Permission page	PTDEFSECPERM	See Using the Group Permissions Page.
Group Users page	PTDEFSECGRPUSERS	See Using the Group Users Page.

This example illustrates the fields and controls on the Definition Group Save As page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Save As	Enter a name for the new definition group.
OK	Click the button to copy the group.
Cancel	Click the button to exit the page without saving any changes or copying the definition group.

Deleting Definition Groups (Browser Client)

Use the Definition Group Search page to delete a definition group.

To delete a definition group:

1. Select **PeopleTools** > **Security** > **Definition Security** > **Security Definition Groups.**
2. Click the **Search** button to search for the definition group to delete.
3. In the Search Results grid, select the box next to the definition group you want to delete.
4. Click the **Delete** button.

   A message box appears asking you to confirm the deletion.
5. Click the **OK** button.

Chapter 22

Implementing Definition Security (Windows Client)

Understanding Definition Security (Windows Client)

This section provides an overview of definition security in the Microsoft Windows client.

You can restrict developer access to the record definitions, menu definitions, page definitions, and others that make up your applications. Just as you use security to control who can access the PeopleSoft pages in your system, you use definition security to control who can access and update PeopleTools definitions.

There are two tasks involved with definition security in the Windows client:

- Creating definition groups.
- Linking definition groups to predefined permission lists.

Note: Implementing Definition Security in the browser client offers additional features and enables you to secure additional definition types.

See Comparing Browser Client and Windows Client Definition Security for a comparison of features between Definition Security in two-tier and Definition Security in three-tier mode. This topic also lists the definition types with which you can work in the different modes.

Accessing Definition Security (Windows Client)

Access definition security in the Windows client using either of these actions:

- In PeopleSoft Application Designer, select **Go** > **Definition Security.**
- Run the PSOSE.exe file located in the following path:

  ```
  <PS_HOME>\bin\client\winx86\psos.exe
  ```

Working With Definition Groups (Windows Client)

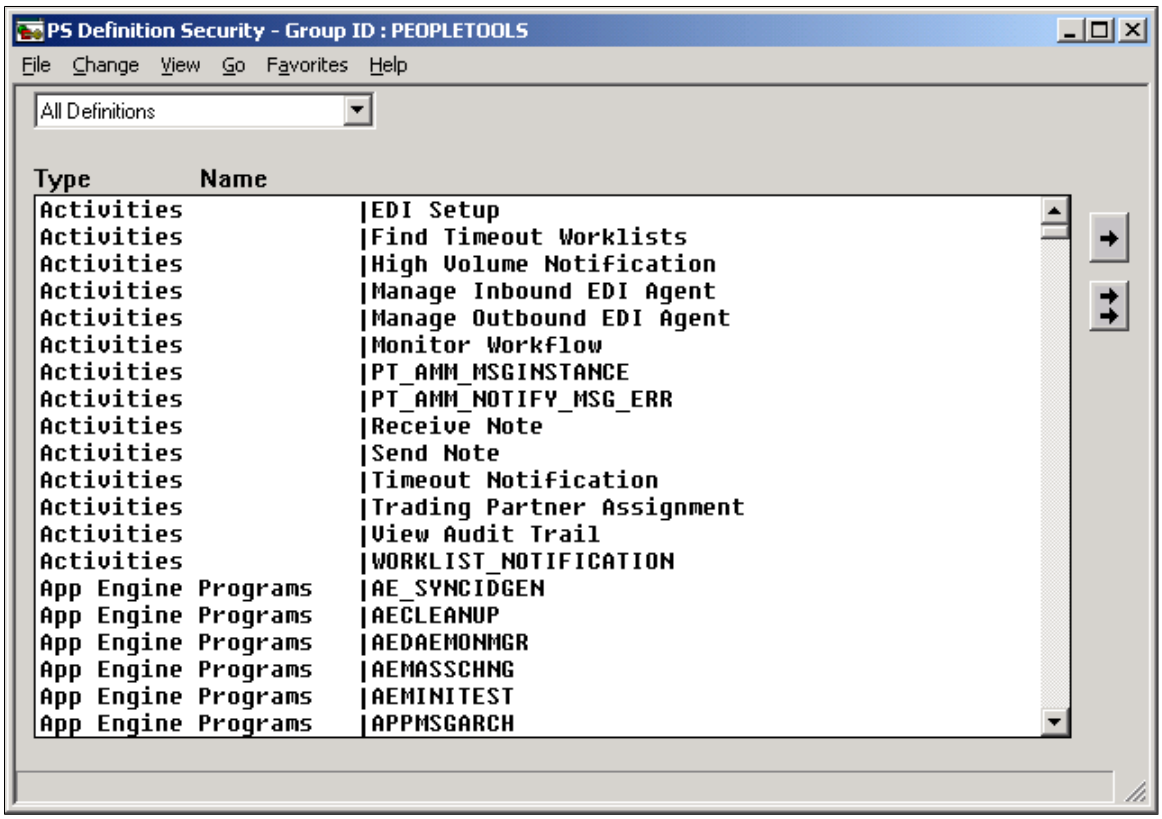
This topic describes how to:

- Open existing definition groups.
- Create definition groups.

- Clone definition groups.
- Rename definition groups.
- Delete definitions groups.

For information about how to access the definition security application in the Windows client, see Accessing Definition Security (Windows Client)

This example shows PS Definition Security displaying all definitions.



To open an existing definition group:

1. Select File, Open, Group.

   The Definition Security Open dialog box appears.

2. Select a group ID.
3. Click **OK.**

To create a new definition group:

1. Select File, New Group.
2. Add definitions to the group.
3. Save the group and give it a name in the Save Group As dialog box.

To clone a definition group:

1. Open the definition group you want to clone.
2. Select File, Save As.

   The Save Group As dialog appears.
3. Enter a group ID and click **OK.**

To rename a definition group:

1. Select File, Rename.

   The Rename Group ID dialog box appears.
2. From the **Rename** list, select the group that you want to rename.
3. Enter a new group ID in the **To** edit box.
4. Click **OK.**

To delete a definition group:

1. Select File, Delete.

   The Definition Security Delete dialog box appears.
2. Select the group ID for the group you want to delete.
3. Click **OK.**

   A confirmation prompt appears.

Viewing Definition Groups (Windows Client)

This topic discusses viewing definition groups in Definition Security in the Microsoft Windows client. This section discusses how to:

- Select a view.
- View all definitions.
- View definitions of a specific type.

Selecting a View

You can select how you view a definition group by using the View menu, or by selecting an item from the drop-down list box that appears at the top of the application window when you have a definition group open.

Viewing All Definitions

To see the entire definition group, select View, All Definitions.

You see every definition, regardless of type, assigned to the definition group. There are two columns: **Type** and **Name.**

- Type identifies the definition type, as in page, query, and so on.
- Name refers to the name given to the definition when it was created.

Viewing Definitions of a Specific Type

To view definitions of a particular type that belong to a definition group, select View, Pages.

The view window is split vertically into two list boxes. The box on the left contains a list of definitions that belong to the definition group and are of the selected type.

The list box on the right is the Excluded *definition_type* list. The label for the definition type changes according to the definition type you are viewing. For example, when you view pages, the label is Excluded Pages, and when you view menus, the label reads Excluded Menus, and so on. The Excluded *definition_type* list box displays the names of all the definitions of the selected type that are not included in the current definition group.

Adding and Removing Definitions (Windows Client)

This top discusses how to add and remove definitions for Definition Security in the Windows client. This topic discusses how to:

- Add and remove definitions.
- Remove definitions from a definition group.

Adding and Removing Definitions

To add definition types to a definition group, you need to view by the type of definition that you want to add. To add pages to a definition group, select View, Pages.

To add definitions to a definition group:

1. Open the definition group.
2. Select the definition type to view by.

   Use the View menu or the drop-down list box at the top of the application window.
3. Select the definitions to be added.

   In the Excluded *definition_type* list box, select the definitions to add to the active definition group.

   To select multiple definitions, use **Ctrl** or **Shift** keys as you click.
4. Click a left-arrow button to move the definitions into the group.

   To move just the selected definitions, use the single left arrow. To move all excluded definitions into the group, use the double left arrow.

Removing Definitions From a Definition Group

To remove definitions from a definition group:

1. Open the definition group.
2. Select the definition type to view by.

   Use the View menu or the drop-down list box at the top of the application window.
3. Select the definitions to be removed in the list box on the left.

   To select multiple definitions, press **Ctrl** key while you click.
4. Click one of the right-arrow buttons to move the definitions out of the group.

   To move just the selected definitions, use the single right arrow. To remove all definitions from the group, use the double right arrow.

Assigning Definition Groups to Permission Lists (Windows Client)

This topic discusses using Definition Security in the Windows client to assign definition groups to permission lists.

To link a definition group to a permission list, the permission list must already exist.

To link definition groups to a permission list:

1. Select **File** > **Open** > **Permission List.**

   The Definition Security Open dialog box appears.
2. Select a permission list and click the **OK** button.

   The window displays two list boxes, similar to what you see when adding or removing definitions.

   The list box on the right shows the existing definition groups that are not currently linked to the active permission list. The list box on the left shows the group IDs that the permission list is currently authorized to access. The group ID is the name that you specified when saving a definition group.
3. Specify the included and excluded groups.

   To enable access to a definition group, select it in the Excluded Group ID list box on the right and move it into the list box on the left. To restrict access to a group, select it on the left and move it into the Excluded Group ID list box on the right. To move just the selected groups, use the single arrows. To move all groups, use the double arrows

   The All Definitions group includes all system definitions. Use it to grant unrestricted access to all databases.
4. Select File, Save to save your changes

Enabling Display-Only Mode (Windows Client)

Enabling display-only access to a definition group means the definitions in that group can be viewed but not modified. You need to link the definition group to the permission list before you specify a display-only value.

For the All Definitions group, display-only mode applies only to the definition groups in the Excluded Group ID list.

The following example shows a permission list (INVPANLS) with access to all definitions, or All Definitions status. Notice that display only is activated. However, it only applies to those groups in the Excluded Group ID list: the NEWGROUP, ONEMENU, and PEOPLETOOLS groups. This means that the INVPANLS permission list has read and write access to all definitions in the system except for those that appear in the Excluded Group ID list. For those definitions, INVPANLS only has read access.

To enable or disable display-only access:

1. Select Change, Display Only.

   The Definition Security List dialog box appears.

   This dialog box lists all the definition groups assigned to the current permission list.

2. Select the groups in the list that you want to make display-only.

   You can use the **All** button to select all the groups in the list.

3. Click **OK.**

Viewing Definition Access by User and Permission List (Windows Client)

To view reports that detail *specific secured definitions* by user or by permission list, access the Common Queries - Definition Security Queries page (PeopleTools, Security, Review Security Information-click the Definition Security Queries link).

You can also view reports that detail access to *definition types* by user or by permission list from the User Profiles and Permission Lists components.

See Running Permission List Queries.

See Running User ID Queries.

Chapter 23

Managing System Personalizations

Understanding System Personalizations

PeopleSoft offers a variety of options that enable end users to personalize their workspaces to complete business transactions in a more efficient manner. These options improve users' navigation through the system, display content and options specific to their business needs, and display data in preferred formats.

PeopleSoft allows many levels of personalization. This section focuses on how administrators manage system personalizations and the relation to end-user preferences.

My Preferences Framework

The My Preferences framework provides a WorkCenter-like interface for end users to view and configure their system-level preference items, such as time format, date format, default dictionary language, and so on. These end-user preference options appear on the My Preferences page in the PeopleSoft Pure Internet Architecture.

As a system administrator or developer, you define, customize, and select the preference options that are available for end users on the My Preferences page, using the pages in the Personalization component.

By default, PeopleTools delivers a set of preferences that are categorized as General Settings. They include user preference options such as turning on accessibility features, defining the time and data format, managing pop-up notifications, and others.

When managing and administering multi-system environments, you can synchronize these preferences on all systems.

Understanding the My Preferences User Interface

This topic discusses:

- Accessing the My Preferences page.
- Default My Preferences - General Settings page.
- Custom navigation panel items.
- My Preferences in the PeopleSoft Fluid User Interface.
- My Preferences in clustered environments.

Accessing the My Preferences Page

End users access the My Preferences page using any of the methods described in this section.

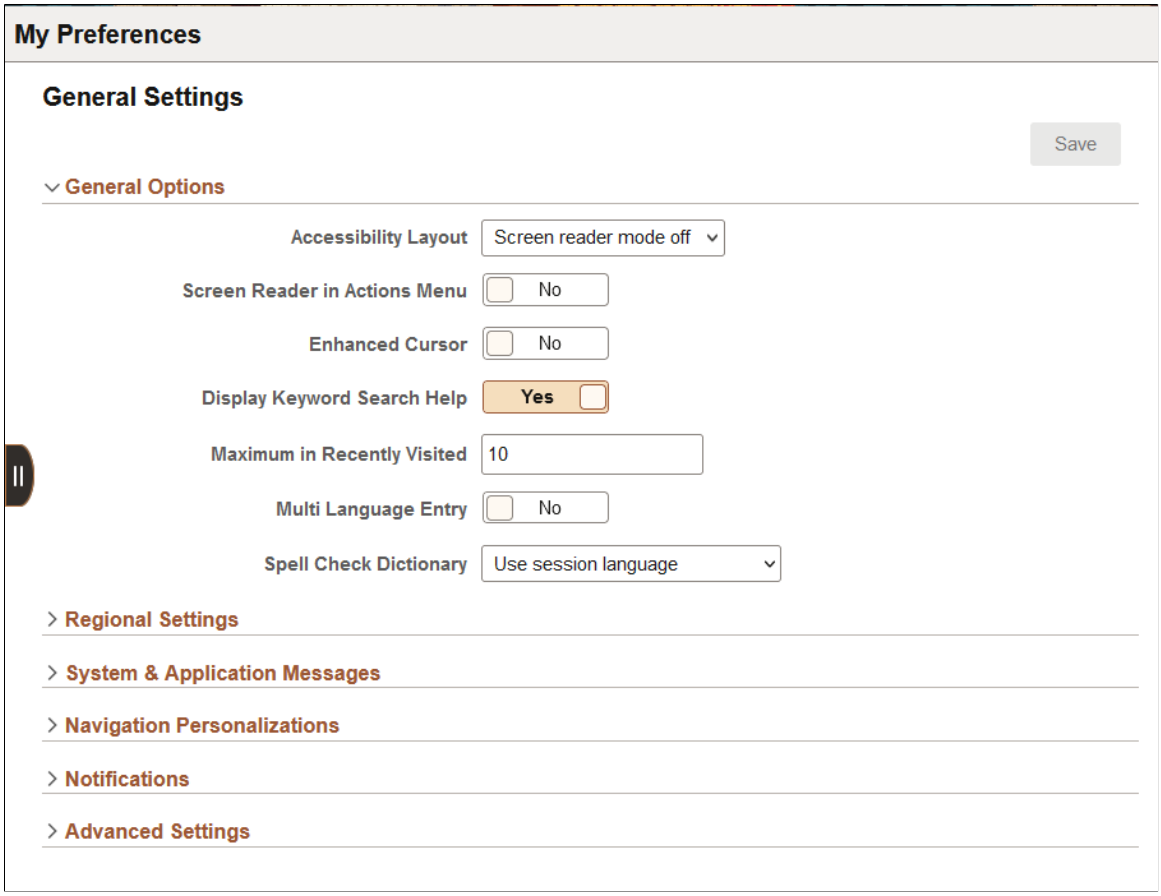
In the PeopleSoft Fluid User Interface or Classic User Interface, from the Actions menu in the banner, select **My Preferences.**

PeopleTools also delivers a My Preferences tile that you can add to fluid homepages that provides access to the My Preferences page. Note that the tile is not added to any fluid homepage by default. See My Preferences and the Fluid User Interface for more information.

Default My Preferences - General Settings Page

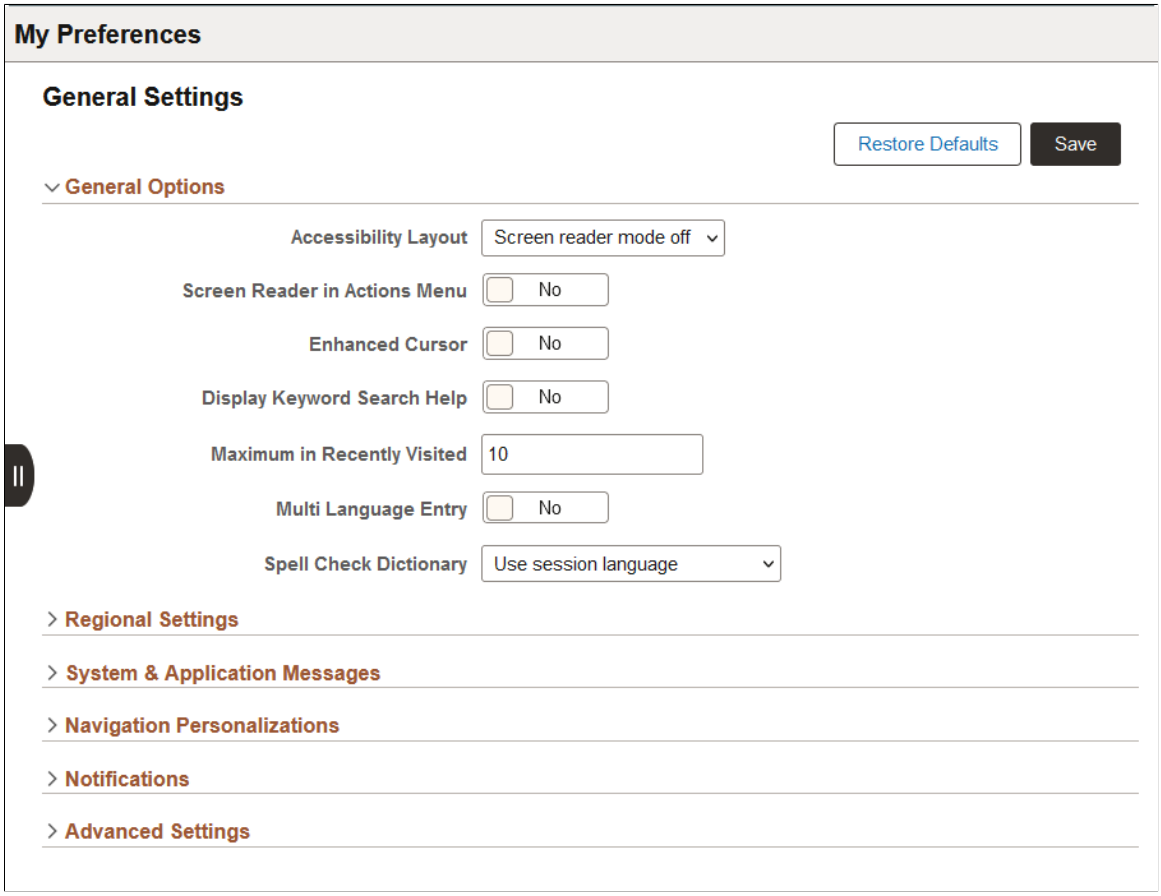
As described previously, PeopleTools delivers a default set of end-user system preferences. The default preferences appear in the My Preferences - General Settings page.

This example illustrates the default fields and controls on the PeopleTools-delivered My Preferences — General Settings page.



When a user modifies any of the default preferences and saves the changes, a Restore Defaults button appears on the page.

This example illustrates the fields and controls on the My Preferences – General Settings page when a user changes any of the default settings. When a user makes any change to the default preferences, a Restore Defaults button appears on the page. The user can click the button to restore all preferences to the default settings.



Each of the collapsible sections on the My Preferences page correspond to a personalization category. Each personalization category contains a group of preferences that are personalization options defined for the personalization category. Only those personalization options that a user is allowed to change via permission list access display for the user at runtime.

As described previously, system administrators and developers use the pages in the Personalization component to manage the preference options and settings that appear in the My Preferences - General Settings page. The preference categories that appear on this page are described in detail in the Working with System Personalization Categories topic.

The option codes, descriptions, default values, and any default permission list to which they belong are described in the Working with System Personalization Options topic.

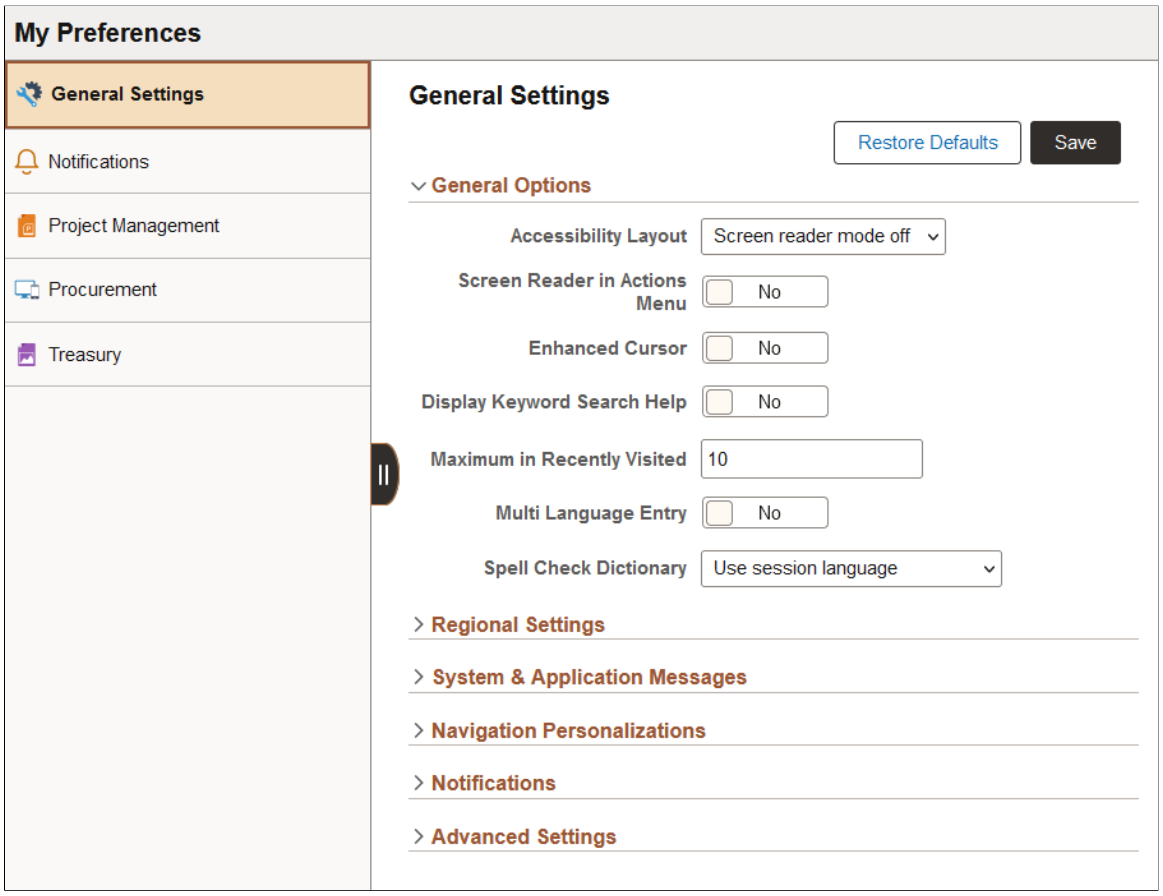
End-user documentation for using the My Preferences page is provided in the “Defining User Preferences” (Applications User’s Guide) topic.

Custom Navigation Panel Items

You can add custom navigation panel items to the My Preferences component.

When you add custom preference items they appear in the navigation panel on the left side of the My Preferences page, as illustrated by the following example.

This example illustrates the My Preferences page with the default General Settings navigation panel item and several custom items added to the page.



In the example, the list in the left navigation panel includes navigation panel items defined in the Fluid Structure Content > My Preferences folder of the current portal, and its first-level sub-folders in all systems in the same cluster.

The list is sorted by the sequence number of corresponding content reference objects or folder reference objects across the systems. You may see leaf items and folders in mixed order, depending on how the sequence numbers are defined.

My Preferences and the Fluid User Interface

This section provides information about using the My Preferences framework with the PeopleSoft Fluid User Interface.

Accessing My Preferences from a Fluid Page

When the My Preferences component is accessed through the Actions menu on a Fluid page, the framework checks if the Fluid page has an associated preference item in the same system. When there is one found, that preference item is displayed on the navigation panel of the My Preferences page by

default. If there are no preference items associated with the Fluid page, the default PeopleTools General Settings preference item appears on the navigation panel.

My Preferences Tile

A My Preferences tile is delivered by PeopleTools, but it is not added to the Fluid homepage or NavBar by default.

This example illustrates the My Preferences tile delivered with PeopleTools.

Note the following points about the My Preferences tile:

- The tile is cluster-aware and always opens the My Preferences component on the portal system.
- The tile does not support context sensitivity and therefore always displays the General Settings preference items described previously in this section.
- The tile features a custom pre-load image and does not have any live tile content.

To add a content reference to the My Preferences tile:

1. Select **PeopleTools** > **Portal** > **Structure and Content.**

   The Structure and Content page appears.

2. In the Folders grid, click the Fluid Structure Content link.

   The Structure and Content for Fluid Structure and Content page appears.

3. In the Folders grid, click the Fluid Homepages link.

   The Content and Structure page for Fluid Homepages page appears.

4. In the Content References grid, locate the homepage to which you want to add the tile, and click the Edit link.

   The Content Reference Administration page appears.

5. Click the Tile Content tab.

   The Tab Content page appears.

6. In the PeopleSoft Applications group of controls:
    a. Select the **My Preferences** box.
    b. From the drop-down list, select the tile behavior.
7. Click the **Save** button.

My Preferences in Clustered Environments

In clustered environments note the following behavior of the My Preferences page:

- The My Preferences link is cluster-aware. As a result, when you click the link it always shows the portal system's My Preferences component.
- Changes made to the General Settings preferences are synchronized to all systems in the same cluster automatically.
- The My Preferences page displays preference items from all systems in the same cluster and allows users to change settings in those systems from within the same My Preferences page.

Clusters and portal systems are defined in "Definition of Terms" (Portal Technology).

Pages Used to Manage System Personalizations and My Preferences

This table describes the pages used to manage system personalizations and the My Preferences page.

Page	*Navigation*	*Action*	*Description*
My Preferences – Structure and Content	**PeopleTools** > **Portal** > **Structure and Content** > **Fluid Structure Content** > **My Preferences**	**Note:** These actions pertain to creating custom My Preference items. • Create preference item labels. • Create the folder structure for preference items. • Define content references to preference pages.	The preference item label is the name that appears at the top of the My Preferences page when the preference item is selected. When multiple preference items are defined, this is the label that appears in the left navigation panel on the My Preferences page. You can create a nested folder structure for preferences, defining a label for each level in the structure. You also must create content references to the pages that contain the options for users to set. See Developing Custom Navigation Panel Items.
Categories	**PeopleTools** > **Personalization** > **Personalization Categories**	View or create option category names.	These are the drop-down categories that the end-user sees on the My Preferences – General Settings page. A category is a holder of related preferences. For example, in the default General Settings preferences delivered with PeopleTools, Regional Settings is a category that contains default personalization options such as calendar type, date format, time format, and so on. See Working with System Personalization Categories.

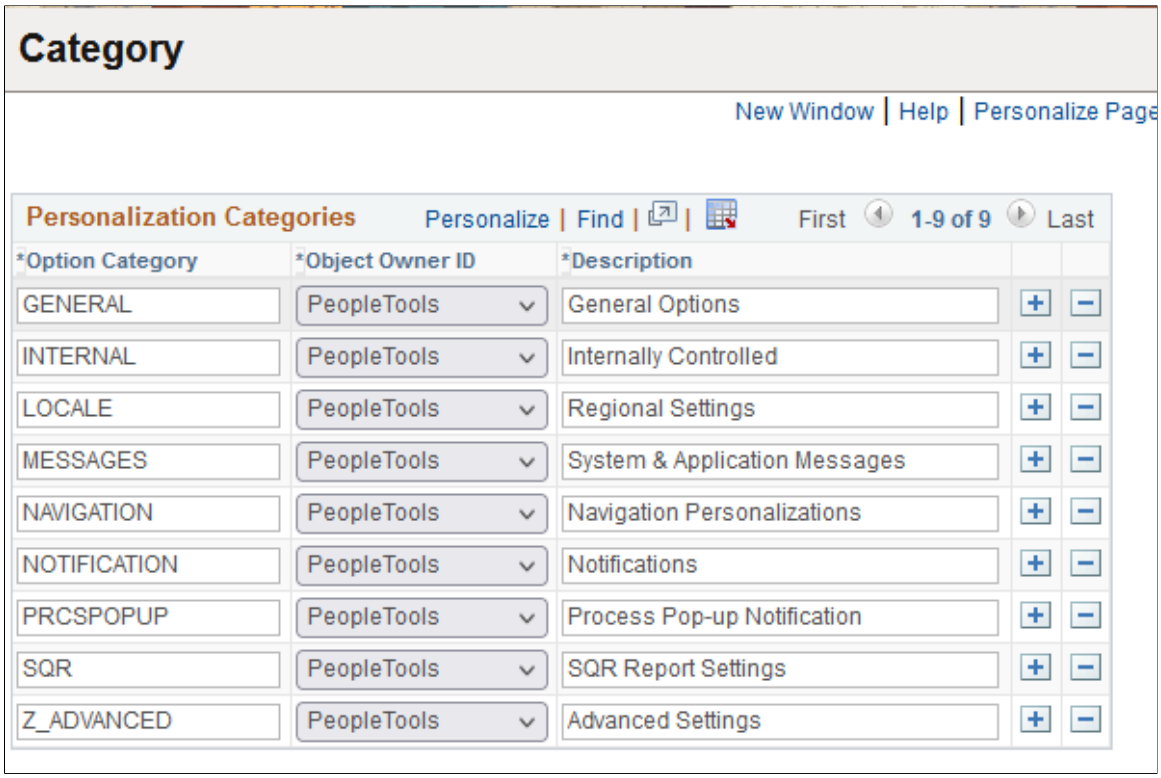
Page	*Navigation*	*Action*	*Description*
Personalization Options	**PeopleTools** > **Personalization** > **Define System Personalizations**	View, define and manage the My Preferences options that users set.	Personalization options are the individual preferences that users set on the My Preferences – General Settings page. When you define a personalization option, you define the category in which to place it. For example, PeopleTools delivers Calendar, Date Format, and Local Time Zone personalization options. These personalization options are assigned to the Regional Settings category by default, and are grouped together on the My Preferences page. See Working with System Personalization Options.
Category Groups	**PeopleTools** > **Personalization** > **Personalization Groups**	Create category groups for applying permission list access to personalization options.	The category groups that you define on this page appear on the Define Personalizations page. On the Define Personalizations page you specify a category group for each preference. As an example, the Calendar, Date Format, and Local Time Zone personalization options are defined in the PS Internet Architecture category group by default. When you define permission list access to personalizations on the My Preferences – General Settings page, you do so by category group. See Working with System Personalization Category Groups.

Working with System Personalization Categories

Categories are the way that you group and present personalization options to your end users.

Access the Category page (**PeopleTools** > **Personalization** > **Personalization Categories**).

This example illustrates the fields and controls on the Category page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
Option Category	Shows the name of the category in which options are displayed on the My Preferences - General Settings page.
Object owner identifier	Displays the name of the group responsible for the maintenance of the category group.
Description	Provides a description of the category for identification purposes. This field has a 30-character limit. **Important!** This is the text that appears on the My Preferences - General Settings page. If you add custom categories make sure the text is meaningful for end users.

Working with System Personalization Options

Before you begin defining and deploying system personalization options, you need to be familiar with the default categories delivered with PeopleSoft software, and the pages used to view and modify them. This section discusses:

- General options.
- Regional settings.
- System and application messages.
- Navigation personalizations.
- Process pop-up notifications.
- Advanced settings.
- SQR report settings.
- Internally controlled options.

Note: PeopleSoft Mobile applications use the standard personalizations.
PeopleSoft Mobile Agent is a desupported product. Mobile personalization features exist for backward compatibility only.

Accessing System Personalization Options

To access the system personalization options delivered with PeopleTools and described in this topic:

1. Select **PeopleTools** > **Personalization** > **Define System Personalizations.**

   The Define Personalizations search page appears.

2. Click the **Search** button.

   The Search Results grid appears.

3. In the Optional Category column, click the **PPTL** link.

   The personalization options delivered for PeopleTools appear in the Define Personalizations page.

To view and sort personalization options by category, in the Define Personalizations grid click the Option Category column header.

Enabling System Personalization Options

System administrators use the Personalizations page (**PeopleTools** > **Security** > **Permissions and Roles** > **Permission Lists**) to control which system personalization options a user can change (personalize).

An end user can personalize a setting if *both* of the following conditions are met:

- An entry for the system personalization option exists in a permission list that is assigned to the user.

  See Setting Personalization Permissions

- Allow User Option is selected for the entry within that permission list, which enables the system personalization option.

If both conditions are not met, the system personalization option does not appear in the user's My Preferences page.

See “Defining User Preferences” (Applications User’s Guide)

For example, the delivered PTPT1000 permission list, PeopleSoft User, is assigned to most user accounts, and includes personalization settings of general interest. The section “Defining User Preferences” (Applications User’s Guide) lists the settings that are available to most end users, and which are part of PTPT1000. If using PTPT1000 is not appropriate for a user, the system administrator can add and enable the necessary system personalization option in other permission lists.

Understanding General Options

The following table presents the delivered general options. Users see the options allowed by permission lists.

Option Code	***Description***	***Default Value***
ACCESS (Accessibility Layout)	Specify accessibility features. This option provides better support for assistive technologies. Select from the following values: • *Screen reader mode on* — For use with screen readers. Page elements (fields, links, buttons, and so on) are presented in linear fashion to assistive software. • *Screen reader mode off* — This disables accessibility features.	*Screen reader mode off*
ACCESSMENU (Screen Reader in Actions Menu)	Specify whether to display the Enable Screen Reader Mode link in the Actions menu. See “Setting System Defaults for Accessibility Personalizations” (Accessibility Guide).	No

Option Code	*Description*	*Default Value*
CUSTOMPGSET (Customize Page Settings)	Indicate that the **Customize Page** pagebar link should appear at the top of pages at runtime. Users can use this control to define, share, and copy page personalizations. ***Warning!*** When this option is disabled, all existing page personalizations for the user are deleted. Grid personalizations aren't affected. **Note:** You can prevent the **Customize Page** pagebar link from appearing in a given component, regardless of whether users have access to this option, by clearing the **Customize Page Link** check box in the Internet properties of the component definition. See “Setting Component Properties” (Application Designer Developer’s Guide).	Yes
ENHANCEDCURS (Enhanced Cursor)	Enables the enhanced focus indicator.	No
KEYWORDSRCH (Display Keyword Search Help)	Displays keyword search Help if the component is indexed.	Yes
MAXMRU (Maximum in Recently Visited)	Specify the maximum number of entries shown under Recently Visited. You see these entries when you select the Recently Visited option on the NavBar or on the quick access bar on fluid homepages. The maximum number of entries that you can specify is 30. See “Working with Recently Visited Items” (Applications User’s Guide).	10
METAXP (Time page held in cache)	Enable browser caching for the navigation pages that remain relatively static. This option specifies the time, in minutes, that portal homepage and navigation pages are held in the cache. You can specify a value between *0* (no caching) and *525600* minutes (one year).	900

Option Code	*Description*	*Default Value*
MLTLNG (Multi Language Entry)	Enable data entry in multiple languages. On a page where the **Data Language** drop-down list box is available, users can select a preferred language for data entry on that page. When this option is disabled, the **Data Language** drop-down list box has no effect.	No
SCLANG (Spell Check Dictionary)	Specify the language to use for the spell check dictionary. Users can select from a wide range of supported languages, or use their session language.	Use session language

Related Links

"Understanding Portal Caching" (Portal Technology)

Understanding Regional Settings

The following table presents the delivered regional settings. Users see the options allowed by permission lists.

Option Code	*Description*	*Default Value*
ADES (Afternoon designator (PM, pm))	(Locale-based) Specify the afternoon designator string to use to indicate PM on a 12 hour display, such as *PM* or *pm*. This value has a 5-character limit.	PM
AUTOGREGCAL (Auto-recognize Gregorian dates)	Specify whether the system automatically recognizes and converts date values to Gregorian calendar dates.	Yes

Option Code	*Description*	*Default Value*
CALENDAR (Calendar)	(Locale-based) Specify the calendar type to use. Select from these values: • *Gregorian* • *Hijri (UmmA l-Qura)* • *Thai* **Note:** If auto-recognize Gregorian dates is set to *Yes* and the calendar is set to non-Gregorian, any dates entered in date fields that fall in the range of the Gregorian calendar will be assumed to be Gregorian and will be converted to specified calendar dates.	Gregorian
DCSP (Decimal Separator)	(Locale-based) Specify the decimal separator character for values with decimals, such as *1.00* or *1,00*. You can enter any single character.	. (period)
DFRMT (Date Format)	(Locale-based) Specify the format for displaying the date. Select from the following values: • *DDMMYY* (day first) • *MMDDYY* (month first) • *YYMMDD* (year first)	MMDDYY
DTSP (Date Separator)	(Locale-based) Specify a date separator character used to separate the month, day, and year in a date. For example, if you specify a hyphen (-), the date appears as 01-01-2001. If you specify a slash (/), the date appears as 01/01/2001. You can enter any single character.	/ (slash)
LTZONE (Local Time Zone)	Select the local time zone, such as *Moscow Time*, *Greenwich Mean Time*, or *Japan Standard Time*. **Note:** This setting alters the *display* of the time for the end user, but does not affect the Base Time Zone setting on the PeopleTools Options page.	Pacific Time (US), Tijuana

Option Code	*Description*	*Default Value*
MDES (Morning designator (AM, am))	(Locale-based) Specify the morning designator string to use to indicate AM on a 12 hour display, such as *AM* or *am*. This value has a 5-character limit.	AM
TFRMT (Time Format)	(Locale-based) Specify the time format for display. Select from the following values: • *12 hour clock* (01:05:00 PM) • *24 hour clock* (13:05:00) **Note:** Whether microseconds appear is not a personalization option.	12 hour clock
TMSP (Time Separator)	Specify the time separator character to separate hours, minutes, and seconds, such as (:) or (.). You can enter any single character.	: (colon)
TSEP (Digit Group Separator)	(Locale-based) Specify the digit group separator character for displaying numerical values over 999 — such as a comma (1,000) or a period (1.000). To specify a space, enter the space between single quotes (' '). You can enter any single character.	, (comma)
TZONE (Use Local Timezone)	Indicate that transactions are to use the local time zone of the client machine. If you select *No*, transactions use the local time zone of the server, where the server may in turn be set to a corporate time zone.	No
WEEKFIRSTDAY (First day of week)	(Locale-based) Specify which day begins the week.	Sunday

Locale-Based Regional Settings

Some of the regional settings, as noted in the table, are locale-based. Their values can be determined based on the locale setting of the user's browser. Because this is one of three sources that can determine which value applies, it's important to understand which source takes precedence:

- In the Define Personalizations component (PSUSEROPTNDEFINE), you can specify default values for locale-based settings, which apply in the absence of any overriding setting.

- The user's browser locale setting is used by the PeopleSoft system to invoke the default values of regional settings for that locale, which you can configure on the Locale Defaults page. Each setting for which you configure a value overrides any default value that's specified for that setting in the Define Personalizations component.
- If a user specifies a value for a locale-based setting in the My Preferences - General Settings page, that value overrides any value configured for that setting for the user's browser locale on the Locale Defaults page. That value also overrides any default value that's specified for that setting in the Define Personalizations component.

Related Links

Working with Locale-Based System Personalizations

Understanding System and Application Messages

System and applications messages are those that the system displays for the user when certain events occur, such as a save or a request to view another page. The following table presents the options for system messages. Users see the options allowed by permission lists.

Option Code	*Description*	*Default Setting*
CFRMSIDM (Disable AutoClose Confirmation)	This flag determines whether a confirmation message automatically closes (dismisses) after a specific period of time. If disabling the auto-closure is set, the user must manually dismiss the message.	No
SCNFRM (Save Confirmation)	Display a brief message confirming each save action.	Yes
SWARN (Save Warning)	Display a warning when the user makes a change and attempts to leave the transaction without saving.	Yes
WARNTRANS (Pagelet Transfer Warning)	Warn users when being transferred from a pagelet to a new page. Applies only in homepages and dashboards in Screen Reader mode.	No

Understanding Navigation Personalizations

The following table presents the delivered navigation personalization options. Users see the options allowed by permission lists.

Option Code	*Description*	*Default Value*
ACEGRDCOLS (Max Col/View All Analytic Grid)	Specify the maximum number of columns that are displayed in an analytic grid when the user selects **Show All Columns**. You can specify up to *100*.	40
ACEGRDROWS (Max Row/View All-Analytic grid)	Specify the maximum number of rows that are displayed in an analytic grid when the user selects **View All**.	100
ADBTN (Tab over Add/Del Buttons (+/-))	Enable tabbing over the **Add** (+) and **Delete** (-) buttons within grids and scrolls.	No
ANAVSORT (Drop down Menu Sort Order)	Enable top navigation sort.	Yes
AUTOMENU (Automatic Menu Collapse)	Enable the menu to automatically collapse when a transaction is selected. The user can expand the menu either by pressing **Ctrl-Y** or clicking the **Show Menu** icon.	No
BADDRESSBAR (Show browser address location)	Enable the display of the browser's address bar. **Note:** This option takes effect only after a new browser instance is launched.	Yes
BBUTTONS (Show browser navigation bar)	Enable the display of the browser's navigation bar, which usually contains the **Back, Forward, Home,** and **Refresh** buttons, among others depending on the browser in use. **Note:** This option takes effect only after a new browser instance is launched.	Yes
BGLYPHTAB (Tab over Glyph Icon)	Enable tabbing over the red glyphs, which indicate a field-level related content contextual menu.	Yes
BLINKS (Show browser links)	Enable the display of the browser's personal links toolbar. **Note:** This option takes effect only after a new browser instance is launched.	Yes

Option Code	*Description*	*Default Value*
BMENU (Show browser menu)	Enable the display of the browser's menu bar. **Note:** This option takes effect only after a new browser instance is launched.	Yes
BMOPOPUP (Mouse over popup event)	Enable mouse over pop up pages, which appear over the main page when you hover over certain text fields.	Yes
BNEWWIN (Open new browser window)	Override the browser setting that causes new windows to appear in browser tabs and instead force all new windows to open in a separate browser window. **Note:** No status bar appears at the bottom of new windows.	No
CALBTN (Tab over Calendar Button)	Enable tabbing over the calendar controls, which appear as buttons on the page.	No
EXPERT (Default Expert Entry On)	Enable expert entry.	Yes
GRDRWS (Max rows for View All)	Specify the maximum number of rows that are displayed in a grid or scroll area when the user selects **View All**.	100
GRDTAB (Tab over Grid Tabs)	Enable tabbing over the tabs or headings within grids.	No
HPPC (PC Homepage)	Sets the default homepage type for all users accessing the system from laptop or desktop computers. The valid options are: • *Fluid.* • *Classic.*	Fluid

Option Code	*Description*	*Default Value*
HPTABLET (Tablet Homepage)	Sets the default homepage type for all users accessing the system from a tablet. The valid options are: • *Fluid.* • *Classic.* **Note:** Homepages always appear in Fluid for smartphones and other small form factor devices.	Fluid
LKPBTN (Tab over Lookup Button)	Enable tabbing over the lookup buttons to the right of edit boxes that have an associated list of valid values.	No
NBAR (Tab over Navigation Bar)	Enable tabbing over navigation bars, which appear at the top of grids and scroll areas to control the appearance of rows and columns.	No
NONPS (Tab over Browser Elements)	Restrict tabbing to include only the PeopleSoft elements of the page, and tab over non-PeopleSoft elements.	No
PGLNK (Tab over Page Links)	Enable tabbing over links to other pages in the same component.	No
POPUP (Tab over Related Page Links)	Enable tabbing over the pop-up menu icon that opens a page of associated menu items.	No
TBAR (Tab over Toolbar)	Enable tabbing over the toolbar at the bottom of a page. Toolbar items include buttons that control standard operations on the page, such as **Save** and **Return to Search**.	No
TYPEAHD (Autocomplete)	Enable autocomplete on prompt edit boxes. The system performs a prompt lookup as you type, suggesting appropriate values.	Yes

Understanding Notifications Settings

The following table presents the delivered notifications setting options. Users see the options allowed by permission lists.

Option Code	Description	Default Value
TEXT (Phone number for texts)	Enter a phone number to receive text messages. Enter your phone number in international phone number format (known as E.164), which is a number of up to fifteen digits in length starting with a plus sign (+). Do not include any other non-numeric characters such as spaces, dashes, periods, or parenthesis.	+14155551234

Understanding Process Pop-Up Notification Settings

The following table presents the delivered process pop-up notification setting options. Users see the options allowed by permission lists.

Option Code	Description	Default Value
0POPUPWIN (Enable Popup Notification)	Enables pop-up notification.	No
1AUTO (Auto Dismiss)	The value determines the duration that the pop-up window is shown.	10 (secs)
2POPUPWIN (Queued State)	If this flag is enabled, the user will receive pop-up notification when the process is in *Queued* state.	No
3POPUPWIN (Processing State)	If this flag is enabled, the user will receive pop-up notification when the process is in *Processing* state.	No
4POPUPWIN (Success State)	If this flag is enabled, the user will receive pop-up notification when the process is in *Success* state.	No
5POPUPWIN (No Success State)	If this flag is enabled, the user will receive pop-up notification when the process is in *No Success* state.	No
6POPUPWIN (Posted State)	If this flag is enabled, the user will receive pop-up notification when the process is in *Posted* state.	No

Understanding SQR Report Settings

The following table presents the delivered SQR report setting options. Users see the options allowed by permission lists.

Option Code	*Description*	*Default Value*
PSIZE (Paper Size for SQR reports)	215	Letter
SQR_PRZ (Personalise SQR Reports)	214	No

Understanding Advanced Settings

The following table presents the delivered advanced setting options. Users see the options allowed by permission lists.

Option Code	*Description*	*Default Value*
METAXP (Time page held in cache)	Meta tag value (in minutes) that is used to determine how long a page is stored in the browser cache. This value is set once for the entire system. The maximum value is *525600* (1 year). The minimum value is *0* (no caching)	*900* (minutes)
TILETIMER (Grouplet Live Update)	Grouplets may be configured to automatically update at an interval you define in on the Fluid Attributes page. You can disable this automatic update.	Yes

Understanding Internally Controlled Options

Internally controlled personalization options are different from the other personalization option categories. Although they're defined in the Define Personalizations component (PSUSEROPTNDEFINE), they never appear in My Preferences - General Settings, even if you assign them to a permission list.

Instead of accessing these options in My Preferences - General Settings, users access and configure them at other locations; the location depends on the individual option. These options are always enabled and can't be disabled, but you can specify their default settings in the Define Personalizations component.

Query Preferences

You specify the default values of the Query preference options in the Define Personalizations component, and individual users can modify those values in Query preferences. The following personalization options are used by PeopleSoft Query:

Term	*Definition*
AUTOJOIN (Enable Auto Join)	This option appears as the **Enable Auto Join** check box on the Query Preferences page. It's selected by default.
NAMESTYLE (Display of query names)	This option appears as the **Name Style** setting on the Query Preferences page. Its default value is *Name and Description*.
DICTIONARY (Display of records)	This option is not used in the current release.
SORTBY (Ways to sort)	This option is not used in the current release.

See “Specifying Query Preferences” (Query).

PS Internet Architecture

The following personalizations are used by PeopleTools Internet Architecture:

Term	*Definition*
ENABLEQAB (Enable Quick Access Bar(QAB))	This personalization setting enables a system administrator to control whether the Quick Access Bar (QAB) is displayed on all pages (fluid or classic). The default is Yes. **Note:** This option is *not* available to end users. When ENABLEQAB is set to No, the QAB is globally disabled, and HPQABPNL is ignored. QAB is never displayed when ENABLEQAB is set to No. See “Using the Quick Access Bar” (Applications User’s Guide).
HPNOTIFYPNL	This personalization setting appears as **Show Notifications Panel** only in Personalize Homepage to enable end users to control the display of the notification panel. The system default is Yes, that is, display the notification panel on the homepage. See “Managing Fluid Homepages” (Applications User’s Guide).

Term	*Definition*
HPQABPNL (QAB as Panel on Homepage)	This personalization setting enables a system administrator to control the presentation of the Quick Access Bar (QAB) on homepages.The default is No, meaning that the QAB is displayed on the banner at the top of the page. If the system administrator sets it to Yes, the QAB is displayed as a panel either on the side or bottom of the homepage (depending on the browser size). The small form factor always displays the QAB as a panel on the bottom or side of the homepage (depending on the device size and orientation). System administrators set the value in the Define Personalizations component. **Note:** This option is *not* available to end users.
NAVMENUSORT	This personalization setting appears as Menu Order on Personalize NavBar and enables end users to change the menu sort order. Options for menu sort order: • Alphabetical • Standard **Note:** The default menu sort order is alphabetical. See “Personalizing the NavBar” (Applications User’s Guide).

Portal Preference

The following personalization option is used by PeopleTools portal technologies:

Term	*Definition*
PAGEHDRCACHE (Time header held in cache)	**Note:** This option is not available to end users. The default value that you set for it in the Define Personalizations component is the only value used, and it applies globally to all users. Use PAGEHDRCACHE to configure caching for the PeopleSoft portal navigation header. This option specifies the time, in minutes, that portal headers are held in the cache. The delivered initial value of this option is *480* minutes.

Tree Manager Preference

The following personalization option is used by PeopleSoft Tree Manager:

Term	*Definition*
TMLINES (Display Lines Per Page)	This option appears as the **Display Lines Per Page** setting on the Configure User Options page of PeopleSoft Tree Manager. Its default value is *60* lines. See “Setting Display Options” (Tree Manager).

Defining System Personalization Options

This section provides an overview of the option category levels and discusses how to:

- Use the Define Personalizations – Definition page.
- Use the Define Personalizations – Format page.
- Use the Define Personalizations – Explanation page.

Note: Adding personalization options involves setting up your options in the Personalizations component, implementing the behavior using PeopleCode, and adding the appropriate permissions through PeopleTools Security. Adding a row to the table using the following interface is only one part of the process.

Understanding Option Category Levels

You add and modify the delivered personalization options using the Define Personalizations page in the Personalization component.

To access the personalization definition pages, select **PeopleTools** > **Personalization** > **Define System Personalizations**.

On the search page, you can choose to search by Option Category Level or Description. If you select Option Category Level and click Search, the following result set appears:

- Customer Relationship Management (CRM).
- Custom (CSTM).
- Enterprise Performance Management (EPM).
- Financials (FIN).
- Human Resources (HCM).
- Learning Solutions (LS).

- PeopleTools (PPTL).
- Supply Chain Management (SCM).

These are the default option category levels delivered with PeopleTools. To add other category levels, add translation values to the field.

Note: These are the only available option category levels. You can't add custom option category levels.

This list corresponds directly to the collection of PeopleSoft applications. In addition, there is a custom category where you store any personalization options you create for applications you have built using PeopleTools. You can also add, or extend, the personalizations for each category. For example, if you wanted to add a new personalization to the HCM category, you add it to the list and define it.

This high-level separation of the personalization options enables you to take a modular approach in deploying the options to your user base. It also helps you to avoid collisions by separating equivalent personalization options by application. For example, you can assign different default values for the same personalization for your Human Resources and Financials applications.

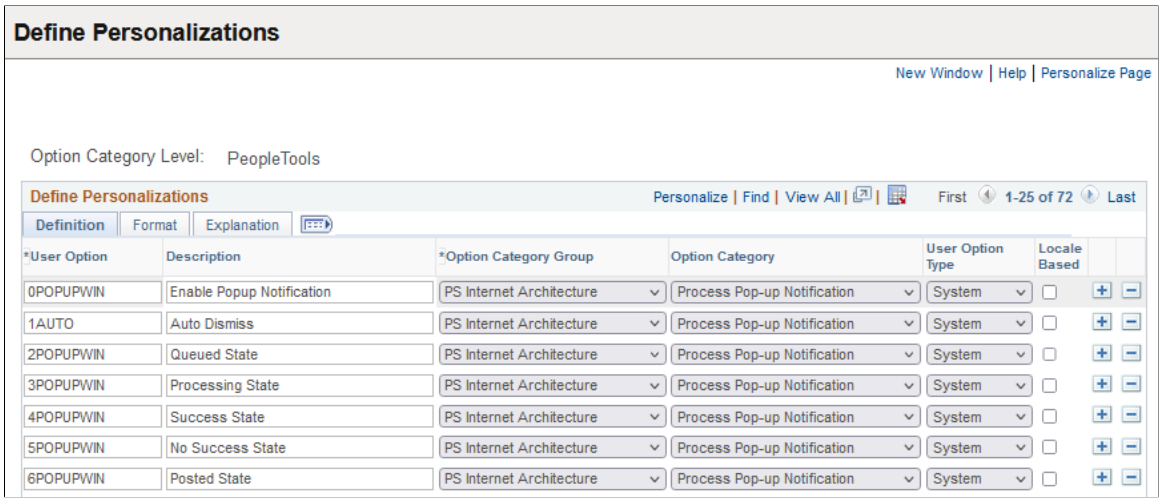
Before adding or modifying personalizations, you select the appropriate category. For example, for CRM personalizations, select the CRM category.

Note: Whether you have installed all of the applications listed in the option category level options, the same category levels appear. Ignore any category levels that do not apply to your site.

Using the Define Personalizations – Definition Page

To access the Define Personalizations – Definition page, select **PeopleTools** > **Personalizations** > **Define System Personalizations** and click the Definition tab.

This example illustrates the fields and controls on the Define Personalizations – Definition page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
User Option	Displays the code associated with the user option. This is the code that the system (PeopleCode) recognizes at run time.
Description	This is the description of the option that the end user sees on the My Preferences - General Settings interface. The description should be unique within the same category. When adding custom personalizations, special attention needs to be paid to this field. Make sure the description is meaningful to end users.
Option Category Group	Specify the product or functional groupings of options. This value acts as an administrative attribute providing ownership for maintenance purposes. It further divides the Option Category Level.
Option Category	Categorizes and encompasses a set of options for the end user. The category you choose from the drop-down list determines the navigation panel item the end user selects to view and modify the option. You add new Categories using the Category page.
User Option Type	Enables you to set where an option is exposed to the end user for override purposes. There are two types: • *Functional:* Options that users set within an application or tool, such as the Application Designer preferences. Functional personalizations are not exposed to the end user through the personalizations pages. If the users have access to the tool or component, then they are able to override the settings. • *System:* Options that are exposed directly to the user through the personalization pages. A user can override default values if permission lists grant them authority.
Locale Based	Indicates that the option derives the default values based on the Locale of the browser.

To add an option, click the insert row (+) button. To delete an option, click the delete row (-) button.

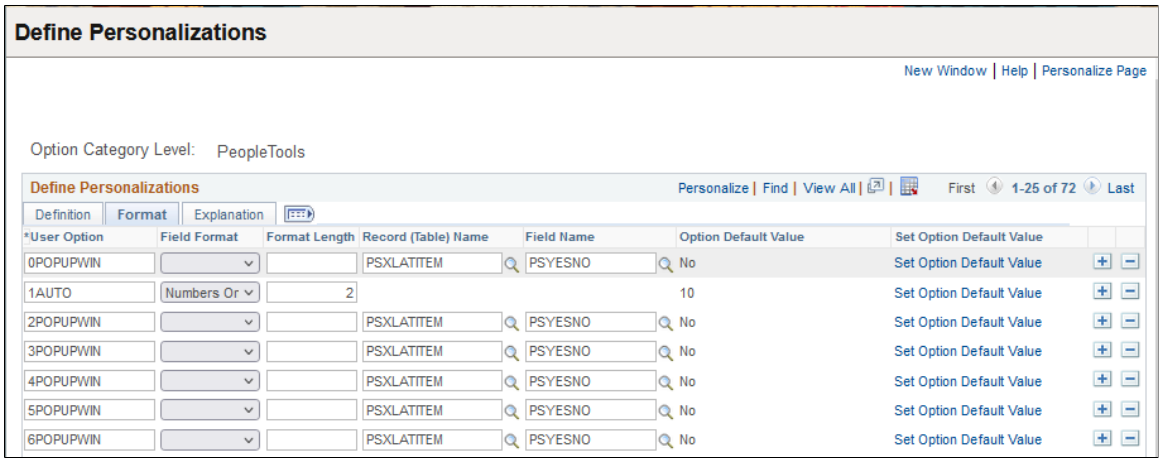
Note: If you add any custom values for these fields, complete all the appropriate planning beforehand. There is no built-in mechanism to prevent collisions.

Note: In the My Preferences - General Settings interface, end users see only options that possess the following attributes: the User Option Type is set to *System* and permission to override that option is granted by one of the users' assigned permission lists.

Using the Define Personalizations – Format Page

To access the Define Personalizations – Format page, select **PeopleTools** > **Personalizations** > **Define System Personalizations** and click the Format tab.

This example illustrates the fields and controls on the Define Personalizations – Format page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
User Option	Shows the code associated with the option.
Field Format and Field Format Length	Specify the field characteristic of the option. Used for the Option Default Value for options that are not validated against the database.
Record (Table) Name	Specifies the lookup table that holds the personalization options values.
Field Name	Specifies the field on the lookup table containing the valid option values.
Option Default Value	Shows the current default for the option. To change the value, select the Set Option Default Value link.
Set Option Default Value	This is a link to the secondary page used to set Option Default Values (discussed in the following section).

Set Option Default Value

The following items appear on the Set Option Default Value page:

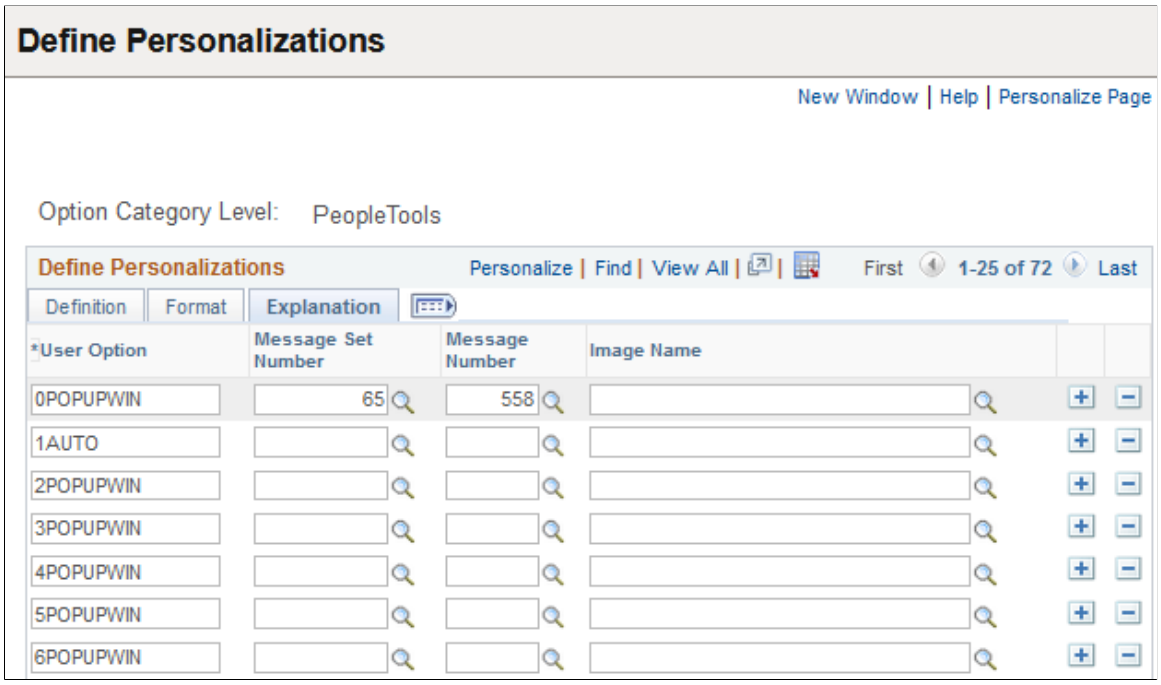
Field or Control	*Description*
Option Category Level	Shows the high-level category to which the option belongs, such as PeopleTools or HCM.
User Option	Displays the code associated with the option.
Description	Displays the description of the option.
Current Default Value	Displays the current default value
Option Default Value	Select the appropriate value from the drop-down list, or add the appropriate option manually. For options that derive default values from a prompt table, the system displays a drop-down list. Otherwise, the system displays an edit box.

Using the Define Personalizations – Explanation Page

Note: The message catalog values that are specified on the Define Personalization – Explanation page are no longer used as of PeopleTools 8.55. No personalization messages are displayed to users in the fluid version of My Preferences that was introduced in PeopleTools 8.55.

To access the Define Personalizations – Explanation page, select **PeopleTools** > **Personalizations** > **Define System Personalizations** and click the Explanation tab.

This example illustrates the fields and controls on the Define Personalizations – Explanation page. You can find definitions for the fields and controls later on this page.



Field or Control	*Description*
User Option	Displays the code associated with an option.
Message Set Number	Specify the message set containing the message that contains the explain text.
Message Number	Specify the message number of the message containing the explain text.
Image Name	Points to the image that the system presents to the end user to provide clarification and context for the personalization. For example, for the "Tab over add button" option, the image of the add button is included so the user can recognize the object.

Working with System Personalization Category Groups

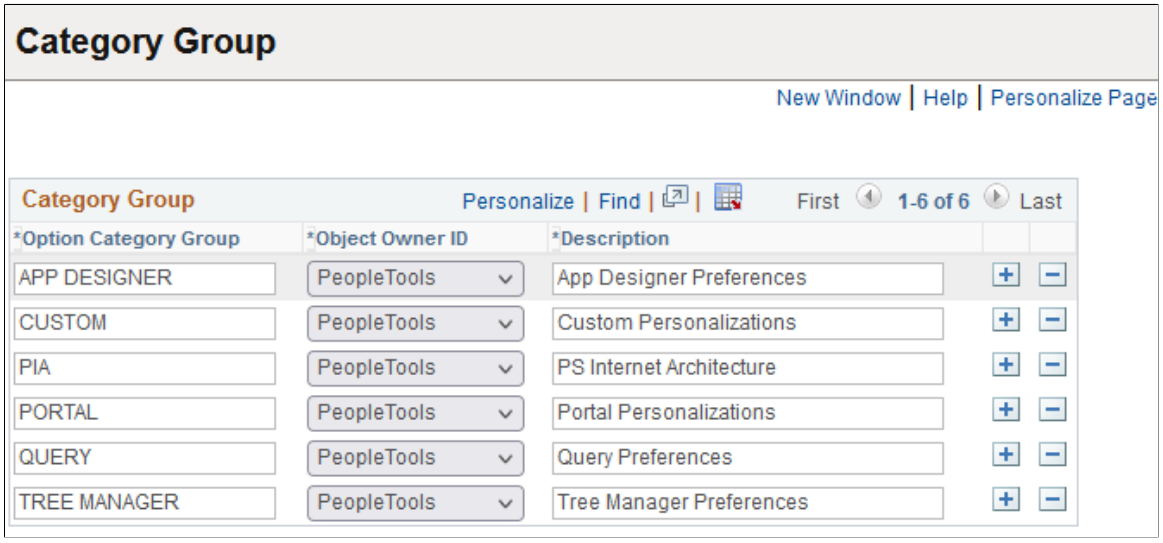
Category groups can represent products, such as Query or Tree Manager, or functional groupings. A category group is an attribute that enables you to designate ownership of personalizations for administrative duties, such as maintenance.

Category groups are also used when assigning system personalizations to permission lists. When you assign system personalizations to permission lists, you specify the option category level (for example, PeopleTools, Campus Solutions, Supply Chain Managements, and so on) and the category group.

Note: By default, all options created within the category level of Custom appear in the Custom category group.

Access the Category Group page (**PeopleTools** > **Personalization** > **Personalization Groups**).

This example illustrates the fields and controls on the Category Group page.



Field or Control	*Description*
Option Category Group	Displays the name of the category group.
Object owner identifier	Displays the name of the group responsible for the maintenance of the category group.
Description	Provides a description of the category group for identification purposes. This field has a 30-character limit.

Working with Locale-Based System Personalizations

Locale-based personalizations enable you to handle settings for globalization. Locale-based personalizations are treated separately than the other personalizations.

You use the following pages to manage these personalization options:

- Locale Definition.

  To access the page select **PeopleTools** > **Personalization** > **Personalization Locales.**

- Locale Defaults.

  To access the page select **PeopleTools** > **Personalization** > **Personalization Local Defaults.**

The system derives the locale information based on the locale specified in the browser. PeopleSoft provides these pages populated with the codes that represent the current browser locales.

This topic is discussed in more detail in the Global Technology product documentation.

Related Links

“Setting Up Locale-Based Formatting for the PeopleSoft Pure Internet Architecture” (Global Technology)

Adding System Personalizations to Permission Lists

You assign personalizations to users by way of permission lists in PeopleTools Security. Before doing so, make sure you have added or modified all the necessary personalizations in the Define Personalizations pages. PeopleTools Security only recognizes personalizations that have been defined in the Define Personalizations interface. This topic is covered in the PeopleTools Security documentation.

Related Links

Setting Personalization Permissions

Creating Custom My Preferences Options

Creating custom options for end users on the My Preferences - General Settings page involves the following steps:

1. Define the option using the Define Personalization interface.

   See Defining System Personalization Options.

2. Implement the behavior using PeopleCode personalization functions (discussed in the following section).

3. To enable users to control the custom option, you need to make the option accessible on the appropriate permission list through PeopleTools Security.

   See Setting Personalization Permissions.

4. Create new category groups to group personalizations for applying to permission lists.

   See Working with System Personalization Category Groups.

Personalization PeopleCode Functions

There are two PeopleCode functions related to personalizations. These functions are:

- `GetUserOption`.
- `SetUserOption`.

If you intend to modify or create custom personalizations, you may need to employ the use of these functions. Refer to the PeopleCode documentation for use and syntax.

Related Links

"GetUserOption" (PeopleCode Language Reference)
"SetUserOption" (PeopleCode Language Reference)

Developing Custom Navigation Panel Items

This topic discusses developing custom items for the My Preferences navigation panel.

Understanding Developing Custom Navigation Panel Items

You can create custom preference options to add to the General Settings preference item category using the pages in the Personalization component. You can also add custom items to the navigation panel on the My Preferences page by creating Fluid components and defining content references in the My Preferences folder in the portal registry.

As with custom navigation panel items, the General Settings navigation panel item is registered in the portal registry. As a result, you can control the order in which navigation panel items appear in the navigation panel by folder and content reference sequence number.

Development Considerations for Custom Navigation Panel Items

This section provides an overview of developing custom items for the My Preferences navigation panel, defining content references for those items, and specifying content reference attributes.

Development Overview

This section lists the general steps for creating custom navigation panel items on the My Preferences page.

1. Create a PeopleSoft Fluid User Interface component.

    - Use the *PSL_USER_PREF* layout page. Do not delete any items on the layout page, with the exception of restore defaults.
    - Load the *PTGP_USER_PERS_FL* stylesheet in either the page Activate event or in a component post build event.

    As a general guideline the component should be:

    - Self-contained with a Save button.
    - Possess functionality that enables users to restore default preference values, for example through a **Restore Defaults** control.

    The component should *not* feature:

    - A left panel.
    - An application header.

- Custom items in the Actions menu in the banner.

2. Add the navigation panel item to the portal registry, including:
   - Add and define a content reference.
   - Define optional content reference attributes.

   Additional information for adding and defining content references and content reference attributes is provided in the next sections.

Content References for Custom Navigation Panel Items

To define a content reference for custom navigation panel items, you must add and define a content reference in the Fluid Structure and Content > My Preferences folder. To navigate to the folder:

1. Select **PeopleTools** > **Portal** > **Structure and Content**.

   The Structure and Content page appears.

2. In the Folders grid, click the **Fluid Structure Content** link.

   The Structure and Content page for Fluid Structure Content appears.

3. In the Folders grid, click the **My Preferences** link.

   The Structure and Content page for My Preferences appears.

4. Below the Content References grid, click the **Add Content Reference** link.

   The Content Ref Administration page for My Preferences appears.

5. Define the content reference.

   See the following section for information about optional content reference attributes to define in the content reference definition.

Content Reference Attributes for Custom Navigation Panel Items

Depending on your business requirements there are several attributes to consider setting when defining content references for custom navigation panel items.

Term	***Definition***
CLUSTER_AWARE	When a custom navigation panel item is defined on multiple systems in a clustered environment, use the CLUSTER_AWARE attribute so that only the navigation panel item defined on the local system appears on the end-user My Preferences page. If you define this attribute, set the attribute value equal to *true.*

Term	*Definition*
PTPP_IMAGE	Use this attribute to add an image to the navigation panel item label. Images are typically added only to root-level folder items. If you define this attribute, specify the image object name in the attribute value field.
CONTEXT_KEYS	Use this attribute to associate the navigation panel item with a comma-separated list of components. Add the comma-separated list of components to the attribute value field.

Related Links

"Administering Folders" (Portal Technology)
"Administering Content References" (Portal Technology)

Using Other PeopleSoft Personalizations

This table lists other types of PeopleSoft personalizations.

Action	*Description*	*Components and Pages*	*User Navigation*	*Additional Information*
Personalize your home page. (Classic)	The fields and controls on the Personalized Home Page page enable end users to personalize their home page, including selecting the pagelets and menus to display on their home page.	Personalized Home Page page.	From the Classic home page: • Click the Content link to manage the pagelets and menus that appear on the homepage. • Click the Layout link to manage the layout of the homepage.	See "Personalizing Classic Homepages and Dashboards" (Applications User's Guide)
Personalize your home page. (Fluid)	The fields and controls on the Personalize page enable you to add or remove home pages, add and delete tiles from your homepage, and more.	Personalize page.	From the Action menu, select **Personalize.**	See "Working with Fluid Homepages and Dashboards" (Applications User's Guide)

Action	*Description*	*Components and Pages*	*User Navigation*	*Additional Information*
Personalize NavBar. (Fluid)	The fields and controls on the Personalize NavBar page enable you to add and remove items on the Fluid NavBar.	Personalize NavBar page	Click the Gear icon at the top of the NavBar pane.	See “Setting Component Properties for Fluid Components” (Fluid User Interface Developer’s Guide) and “Applying Styles” (Fluid User Interface Developer’s Guide)
Personalize pagelets. (Classic)	Some pagelets provide options to personalize pagelet content, such as display charts, show/hide prompts, and more. **Note:** Not all pagelets allow personalization.	NA	Click the Gear icon in top right corner of the pagelet to display any personalizations available for the content. Personalization options can appear in a drop-down list or in secondary pages.	See “Personalizing Pagelets” (Applications User’s Guide)
Personalize related content. (Fluid)	Users can show or hide specific items on embedded related content on a page or items displayed in the Related Content frame at the right of the page.	Personalize Related Information page.	Click the Gear icon in the top right corner of a Related Content frame.	See “Working with Embedded Related Content” (Fluid User Interface Developer’s Guide)

Chapter 24

Using Virus Scanning

Enabling Virus Scanning for Web Servers

Virus scanning on web servers can be enabled for inbound IMAP and POP3 MCF Email, and for attachments. To enable virus scanning, configure the VirusScan.xml file located on the web server.

This section discusses:

- Scanning attachments for viruses.
- Configuring VirusScan.xml.
- Logging virus scans.
- Virus scan errors and return codes.

Scanning Attachments for Viruses

Virus scanning can be performed on all files uploaded with the AddAttachment, InsertImage, and MAddAttachment functions.

Note: If the HTML sanitizer is also configured on this web server, virus scanning is performed on the file before the HTML sanitizer is run.

Another topic covers scanning attachments on the application server.

Related Links

"Using the HTML Sanitizer" (PeopleCode Developer's Guide)
Enabling Virus Scanning for Application Servers
"AddAttachment" (PeopleCode Language Reference)
"InsertImage" (PeopleCode Language Reference)
"MAddAttachment" (PeopleCode Language Reference)

Configuring the VirusScan.xml File

To enable the virus scanning feature:

1. Locate VirusScan.xml on the web server.

   The location of this file on your WebLogic web server is:

   <*PS_CFG_HOME*>/webserv/<*domain_name*>/applications/peoplesoft/PSIGW.war/WEB-INF/classes/psft/pt8/virusscan

2. Open VirusScan.xml for editing.

```
<?xml version="1.0" encoding="UTF-8"?>
<Providers disableAll="True" logFile="./servers/PIA/logs/VirusScan%u.log">
    <!-- Sample Configuration for Symantec Engine
 <Provider>
       <name>Symantec</name>
     <class>psft.pt8.virusscan.provider.GenericVirusScanProviderImpl</class>
     <icapversion>ICAP/1.0</icapversion>
        <service-name>/SYMCScanResp-AV</service-name>
        <policycommand>?action=SCAN</policycommand>
        <address>192.0.2.44</address>
        <port>1344</port>
        <disable>false</disable>
   </Provider>-->

   <!-- Configure your own proivider -->
   <Provider>
   <!-- Provider Name of the Scan Engine -->
   <name></name>
   <!-- Provider Class of the Scan Engine.
        psft.pt8.virusscan.provider.GenericVirusScanProviderImpl is
        the default
        provider class.  -->
   <class>psft.pt8.virusscan.provider.GenericVirusScanProviderImpl</class>
   <!-- ICAP version -->
   <icapversion>ICAP/1.0</icapversion>
   <!-- ICAP ServiceName. The Service Name changes from Scan Engine to
  Scan Engine.
        This is the name Scan Engine Service is will be hosted with -->
   <service-name></service-name>
   <!-- RESPMOD extra commands, These are the RESPMOD commands
  (SEE ICAP Protocol).
        Usually these commands will be changing from Engine to Engine
   -->
   <policycommand></policycommand>
   <!-- IP Address of Scan Engine host> -->
   <address></address>
   <!-- IP Port of Scan Engine host -->
   <port></port>
   <!-- Disable scanning for this provider -->
   <disable></disable>
   <!--
        Default codes = 200 and 204 for clean, 201,403 for infected
        Use these tags to change the behaivior if needed
        <clean>200,204</clean>
        <infected>201,403</infected>
        -->
  <virusheader></virusheadercheck>
</Provider>
</Providers>
```

Note: A sample configuration for Symantec Engine is provided in the remarks.

3. In the Providers tag, set the attribute *disableAll* to "False".

Note: The default value is "True".

```
<Providers disableAll="False" logFile="./servers/PIA/logs/VirusScan%u.log">
```

4. Specify scan engines under the <Providers> tag.

   Multiple scan engines can be configured under <Providers>. Each <Provider> tag represents one scan engine. All configured scan engines will check for viruses. For each <Provider> tag enter values for the tags:

Tag	*Description*	*Example Value*
<name>	Provider name of the scan engine	Symantec
<class>	Provider class of the scan engine Default provider class is: psft.pt8.virusscan.pr⇒ ovider.GenericVirusSc⇒ anProviderImpl	⇒ psft.pt8.virusscan.pr⇒ ovider.GenericVirusSc⇒ anProviderImpl
<icapversion>	ICAP version	ICAP/1.0
<service-name>	Service name for the scan engine host	/SYMCScanResp-AV
<policycommand>	Policy command used by the scan engine. Only SCAN is supported.	?action=SCAN
<address>	IP address of the scan engine host.	IP address of the machine where the scan engine is running
<port>	IP port of the scan engine host.	Port where the scan engine is running
<disable>	Disable scanning for this provider.	false
<clean>	Default codes = 200 and 204 for clean. You can use this tag to change the behavior if needed.	200,204
<infected>	Default codes = 201 and 403 for infected You can use this tag to change the behavior if needed.	201,403

Tag	*Description*	*Example Value*
<virusheadercheck>	This tag contains a comma-separated pair of configurable header and error. This tag can be configured only for these two errors: INFECTED and SCANERROR. For example, <virusheadercheck>X-Violation::INFECTED, FileAttributeError::SCANERROR</virusheadercheck>. In this example, X-Violation will be checked in the response header from the scan engine and if found, INFECTED will be returned. If X-Violation is not found, FileAttributeError will be checked in the response header from the scan engine. If FileAttributeError is found, SCANERROR will be returned. • If the first header is found in the response header from the scan engine, the subsequent headers are not checked. • If error is not configured for X-Violation, INFECTED will be returned by default. • The <virusheadercheck> tag is not applicable when <clean> or <infected> is configured. • When <clean> or <virusheadercheck> is not configured, the default codes for clean (200, 204) and for infected (201, 403) will be checked in the response header and either CLEAN or INFECTED will be returned.	<virusheadercheck>X-Violation::INFECTED, FileAttributeError::SCANERROR</virusheadercheck>

Viewing Virus Scanning Logs

Virus scanning logs are the only interface with the scanning engine.

Virus Scanning Logs

The virus scanning logs are located in the path indicated by the *logFile* property in VirusScanning.xml.

```
<Providers disableAll="False" logFile="./servers/PIA/logs/VirusScan%u.log">
```

The following results are logged with the date and the file name that was scanned:

- CLEAN, INFECTED, and SCANERROR

The results for these statuses is logged in this form:

```
filename = result
```

For example:

```
finance.xls = INFECTED
```

- CONNECTERROR and CONFIGERROR

  The results for these statuses is logged in this form:

  ```
  Unable to connect to the Scan engine: REASON = result
  ```

  For example:

  ```
  Unable to connect to the Scan engine: REASON = CONFIGERROR
  ```

In addition, detailed logging is configured in the logging.properties file for WebLogic server:

```
PS_CFG_HOME\webserv\domain_name\piaconfig\properties\logging.properties
```

See “Debugging File Attachment Problems” (PeopleCode Developer’s Guide).

Viewing Virus Scanning Error Logs

If there are any errors during file processing the error codes listed in this table will be generated.

If there is a failure, the details will be logged in the location specified for the parameter *ig.errorLog.filename* in integrationGateway.properties, which is located in <*PS_CFG_HOME*>/webserv/<*domain_name*>/applications/peoplesoft/PSIGW.war/WEB-INF.

The return value when the virus scans for mail attachments is REPOSITORY_FAILURE = 8.

See “Error Messages Returned by MCFGetMail Class Methods” (PeopleCode API Reference).

If the file is uploaded successfully and no problems are found in the virus scan, the AddAttachment, InsertImage, or MAddAttachment function returns %Attachment_Succeeded.

If a problem is found, the PeopleCode function returns one of the following return codes:

Numeric Value	***Error Code***	***Description***
9	%Attachment_FileNotFound	Cannot locate file.
13	%Attachment_ViolationFound	File violation detected by the virus scan engine.
14	%Attachment_VirusScanError	Virus scan engine error.
15	%Attachment_VirusConfigError	Virus scan engine configuration error.
16	%Attachment_VirusConnectError	Virus Scan engine connection error.

Numeric Value	*Error Code*	*Description*
24	%Virusscan_Disabled	Virus scan is not enabled.

Enabling Virus Scanning for Application Servers

PeopleTools provides the capability to scan attachments for viruses before streaming the attachments from the application server to the PeopleSoft system.

The attachment is streamed to the PeopleSoft system if the attachment is clean, that is, it is not infected. For example, you can scan a file for viruses before downloading it from the Oracle Digital Assistant (ODA) server to the PeopleSoft repository.

Additionally, you may use the ScanFile built-in function to perform virus scanning. See “ScanFile” (PeopleCode Language Reference).

To enable virus scanning, PeopleTools provides a VirusScan.xml file on the application server, and its configuration is similar to the VirusScan.xml configuration file on the web server. The virus scan error return codes are also similar to the virus error return codes on the web server.

See Enabling Virus Scanning for Web Servers.

Follow these steps to configure the VirusScan.xml file:

- The VirusScan.xml file on your system for application server is at this location:

  PS_CFG_HOME\appserv*domain_name*\VirusScan.xml

- Open the VirusScan.xml file and set the value of disableAll to "False". By default, disableAll is "True".

  ```
  <Providers disableAll="False" logFile="%VIRUS_SCAN_LOG%">
  ```

- Replace %VIRUS_SCAN_LOG% with the name of the log file on the application server. The log file is created in the application server logs folder.

Note: Except for the configuration steps mentioned in this topic, VirusScan.xml tags, return codes and other settings are similar to the VirusScan.xml configuration file on the web server.
See Configuring the VirusScan.xml File, Viewing Virus Scanning Error Logs.

