ORACLE
PEOPLESOFT

Oracle's PeopleTools PeopleBook

PeopleTools 8.52: Security Administration

June 2013

ORACLE

PeopleTools 8.52: Security Administration
SKU pt8.52tsec-b0613

Trademark Notice

License Restrictions Warranty/Consequential Damages Disclaimer

Warranty Disclaimer

Restricted Rights Notice

Hazardous Applications Notice

Third Party Content, Products, and Services Disclaimer

Contents

Chapter 4

Chapter 5

Chapter 7

Chapter 8

Chapter 9

Chapter 14

Chapter 15

Chapter 16

Chapter 17

Security Administration Preface

This preface provides an overview of the content discussed in the Security Administration PeopleBook and discusses PeopleBooks and the online PeopleSoft library.

Security Administration

This PeopleBook covers a wide range of different tools and techniques for administering security on your PeopleSoft system, including:

- Permission lists.
- Roles
- User profiles.
- Lightweight Directory Access Protocol (LDAP).
- Single signon.
- Secure Sockets Layer/Transport Layer Security (SSL/TLS) and digital certificates.
- Web Service security.
- PeopleSoft Encryption Technology (PET).
- Query and definition security.
- Personalization features.

Note. Remember that your application documentation also contains security topics that are more specific to the applications you've purchased.

PeopleBooks and the PeopleSoft Online Library

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

- Understanding the PeopleSoft online library and related documentation.
- How to send PeopleSoft documentation comments and suggestions to Oracle.
- How to access hosted PeopleBooks, downloadable HTML PeopleBooks, and downloadable PDF PeopleBooks as well as documentation updates.
- Understanding PeopleBook structure.
- Typographical conventions and visual cues used in PeopleBooks.

- ISO country codes and currency codes.
- PeopleBooks that are common across multiple applications.
- Common elements used in PeopleBooks.
- Navigating the PeopleBooks interface and searching the PeopleSoft online library.
- Displaying and printing screen shots and graphics in PeopleBooks.
- How to manage the locally installed PeopleSoft online library, including web site folders.
- Understanding documentation integration and how to integrate customized documentation into the library.
- Application abbreviations found in application fields.

You can find *PeopleBooks and the PeopleSoft Online Library* in the online PeopleBooks Library for your PeopleTools release.

Chapter 1

Getting Started with Security Administration

This chapter provides overviews of PeopleSoft Enterprise security administration and security administration integrations and discusses security administration implementation.

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 Chapter 3, "Setting Up Permission Lists," page 25.

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 Chapter 4, "Setting Up Roles," page 71.

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 Chapter 5, "Administering User Profiles," page 91.

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 Chapter 7, "Employing LDAP Directory Services," page 143.

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 Chapter 8, "Employing Signon PeopleCode and User Exits," page 177; Chapter 9, "Implementing Single Signon," page 197 and Chapter 9, "Implementing Single Signon," Implementing Kerberos as the Desktop Single Signon Solution, page 221.

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 Chapter 14, "Securing Data with PeopleSoft Encryption Technology," page 279.

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.

See Chapter 15, "Implementing Query Security," page 309 and Chapter 16, "Implementing Definition Security," page 319.

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 Chapter 17, "Managing PeopleSoft Personalizations," page 329.

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), and 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.

See Also

PeopleTools 8.52: PeopleSoft Component Interfaces, "Understanding 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 deprecated 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 User Profiles component (USERMAINT), the User Profile Save As component, the My System Profile component (USERMAINT_SELF), the Distributed User Profile component (USERMAINT_DIST), the USER_PROFILE component interface, and the 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.

See Also

PeopleTools 8.52: PeopleSoft Integration Broker, "Managing Service Operations"

Application Engine Programs

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

DYNROLE_PUBL

The DYNROLE_PUBL Application Engine program is called when Dynamic Role Rules are executed 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.

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, 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.

LDAPSCHEMA

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

You run this program by selecting PeopleTools, Security, Directory, Cache 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, Authentication Map.

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

See Also

PeopleTools 8.52: Application Engine, "Understanding 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 installation guide 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 PeopleBooks.

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, 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.

Security Links - User page

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, 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. The Common Queries page enables you to run the following sets of 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.

Chapter 2

Understanding PeopleSoft Security

This chapter discusses:

- Security basics.
- PeopleSoft online security.
- PeopleSoft authorization IDs.
- PeopleSoft sign in.
- Implementation options.

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 definitions 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.

See Also

https://support.oracle.com

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.

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.

See Also

PeopleTools 8.52: PeopleSoft Integration Broker Administration, "Setting Up Secure Integration Environments"

PeopleSoft Applications Portal 9.1: Portal and Site Administration PeopleBook, Part 5: Integration

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 Also

Chapter 2, "Understanding PeopleSoft Security," PeopleSoft Sign In, page 20

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 PSACCESSPRFL, 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 PSACCESSPRFL 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, being sure to specify a new password. You should change the passwords of all delivered IDs as soon as possible.

Note. PeopleSoft-delivered IDs and passwords 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, PSACCESSPRFL, 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.

See Also

Chapter 7, "Employing LDAP Directory Services," page 143

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 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 portal, which aggregates content from several different applications and data sources into a single, integrated display.

See Also

Chapter 11, "Working with SSL/TLS and Digital Certificates," page 263

Chapter 9, "Implementing Single Signon," page 197

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.

See Also

http://www.ietf.org/

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 applicationos, 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

This chapter provides an overview of permission lists and discusses how to:

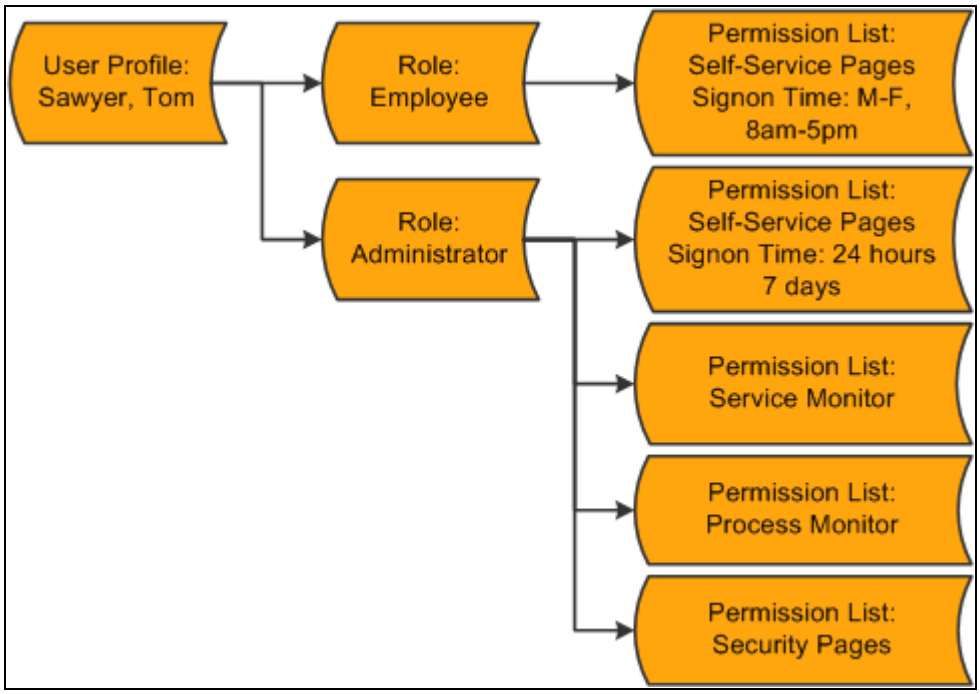
- Manage permission lists.
- Define permissions.

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.



Security definitions hierarchy showing how permissions flow to roles, which flow to user profiles

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 & Roles, Permission Lists.
2. On the search page, click Add a New Value.
3. In the Permission List edit box, enter the name of 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 & 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 To: 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 & 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 & Roles, Permission Lists, Pages to access the Pages page, and then click the Edit Components link to access the Component Permissions page.

See Chapter 3, "Setting Up Permission Lists," Granting Access to Components and Pages, page 37.

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.

See Also

PeopleTools 8.52: PeopleTools Portal Technologies, "Administering Portals," Administering Content References

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 personalization permissions.
- Set query permissions.
- Set mass change permissions.
- Display additional links.
- View when a permission list was last updated.
- Assign search group permissions.
- Set search group permissions.
- Run permission list queries.

Pages Used to Define Permission Lists

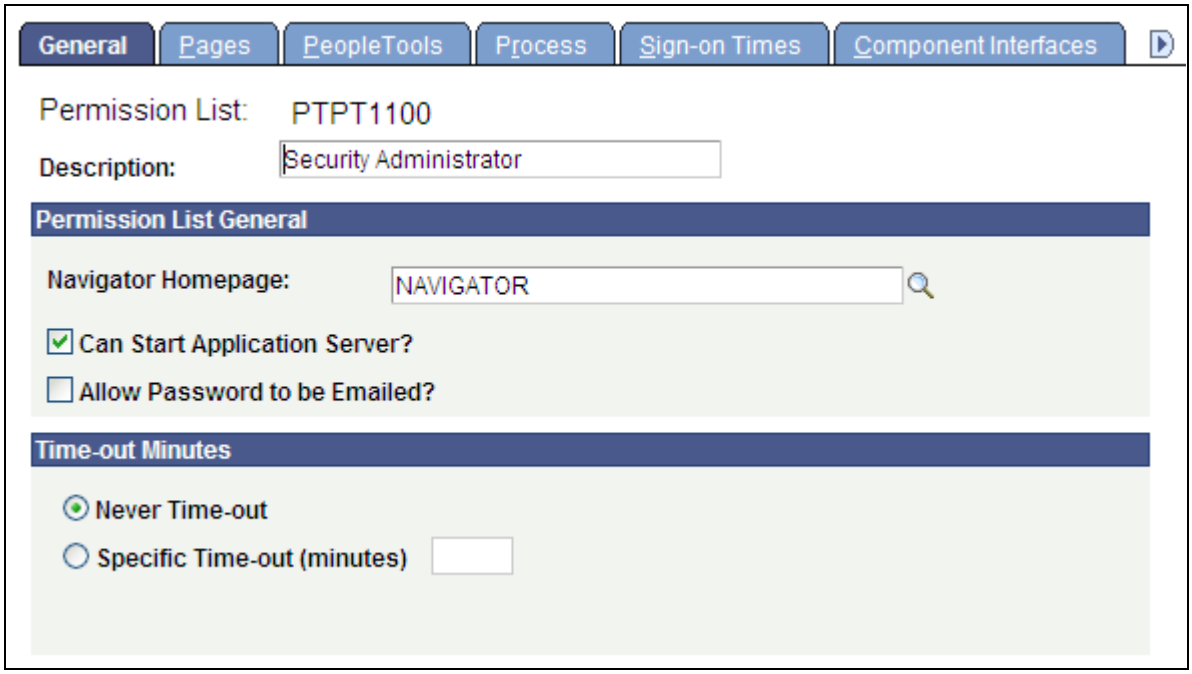
Page Name	*Definition Name*	*Navigation*	*Usage*
General	ACL_GENERAL	PeopleTools, Security, Permissions and Roles, Permission Lists, General	Set general or miscellaneous attributes and system defaults.

Page Name	*Definition Name*	*Navigation*	*Usage*
Pages	ACL_MENU2	PeopleTools, Security, Permissions and Roles, Permission Lists, Pages	Set page permissions.
PeopleTools	ACL_MISCTOOLS	PeopleTools, Security, Permissions and Roles, Permission Lists, PeopleTools	Grant access to PeopleTools applications, such as PeopleSoft Application Designer, and grant access for specific operations within PeopleTools.
Process	ACL_PROCESS	PeopleTools, Security, Permissions and Roles, Permission Lists, Process	Specify to what capacity a user or role can modify PeopleSoft Process Scheduler settings.
Sign-on Times	ACL_SIGNON2	PeopleTools, Security, Permissions and Roles, Permission Lists, Sign-on Times	Specify when users are authorized to sign in to the PeopleSoft system. If users are signed in to the system when the sign in time expires, they are automatically signed out.
Component Interface	ACL_COMP_INTERFACE	PeopleTools, Security, Permissions and Roles, Permission Lists, Component Interface	Grant access to any component interfaces that a user may need to use to complete business transactions.
Web Libraries	ACL_WEBLIBS	PeopleTools, Security, Permissions and Roles, Permission Lists, Web Libraries	Set web library permissions.
Web Services	ACL_WS_OPR	PeopleTools, Security, Permissions and Roles, Permission Lists, Web Services	Set web services permissions.
Personalizations	PLIST_OPTN	PeopleTools, Security, Permissions and Roles, Permission Lists, Personalizations	Specify which personalizations users can use and customize.
Query	PERMLIST_QUERY	PeopleTools, Security, Permissions and Roles, Permission Lists, Query	Control the query operations a user can perform and the data a user can access while using PeopleSoft Query.
Mass Change Operator Security	MC_OPR_SECURITY	PeopleTools, Security, Mass Change Operator Security	Set mass change security permissions.

Page Name	Definition Name	Navigation	Usage
Audit	PERMLIST_AUDIT	PeopleTools, Security, Permissions and Roles, Permission Lists, Audit	Inquire when a permission list was last updated and by whom.
Search Groups	PERMLIST_SRCHGRP	PeopleTools, Security, Permissions and Roles, Permission Lists, Search Groups	Assign search groups to permission lists.

Setting General Permissions

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



Permission Lists - General page

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.

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.

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).

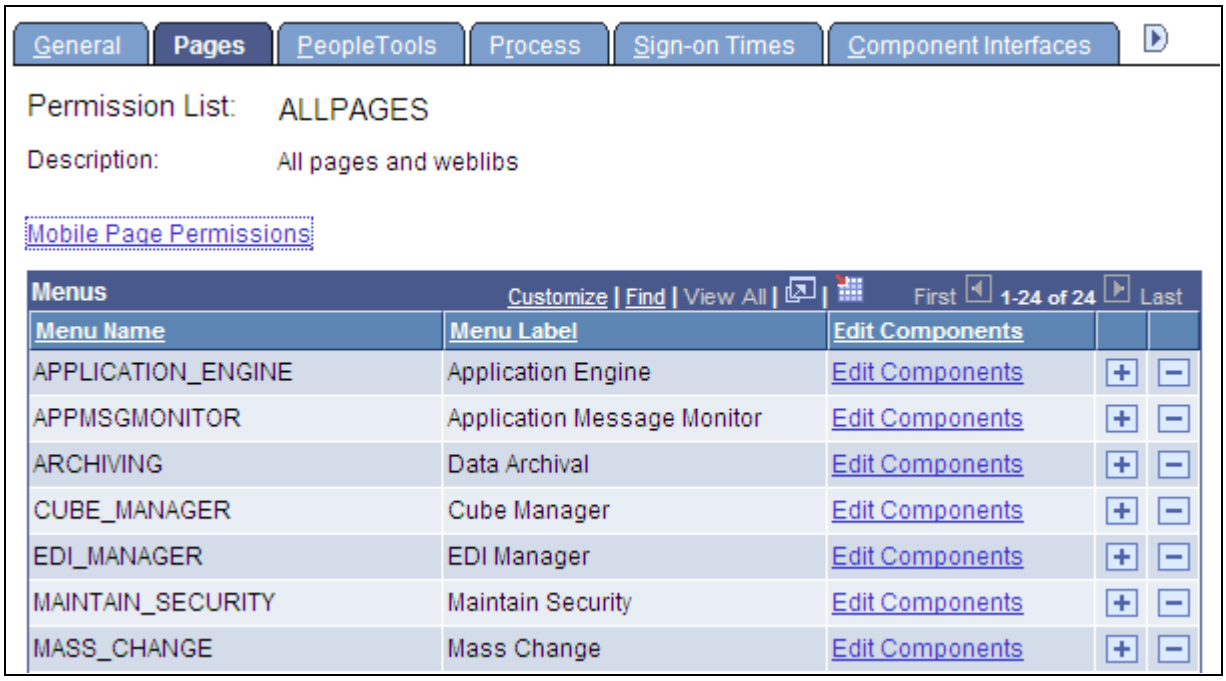
Entering a value of zero (0) is equivalent to selecting Never Time-out.

To comply with the Americans with Disabilities Act (ADA), you might set up most permission lists to time out in 20 minutes, but create a special ADA permission list for which timeout occurs after 60 minutes.

Note. Because timeout limits are also controlled at the web server level, you will need to change the web server timeout values also.

Setting Page Permissions

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



Permission Lists - Pages page

This table describes the fields on the Pages page.

Mobile Page Permissions Click to grant access to mobile application pages.

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

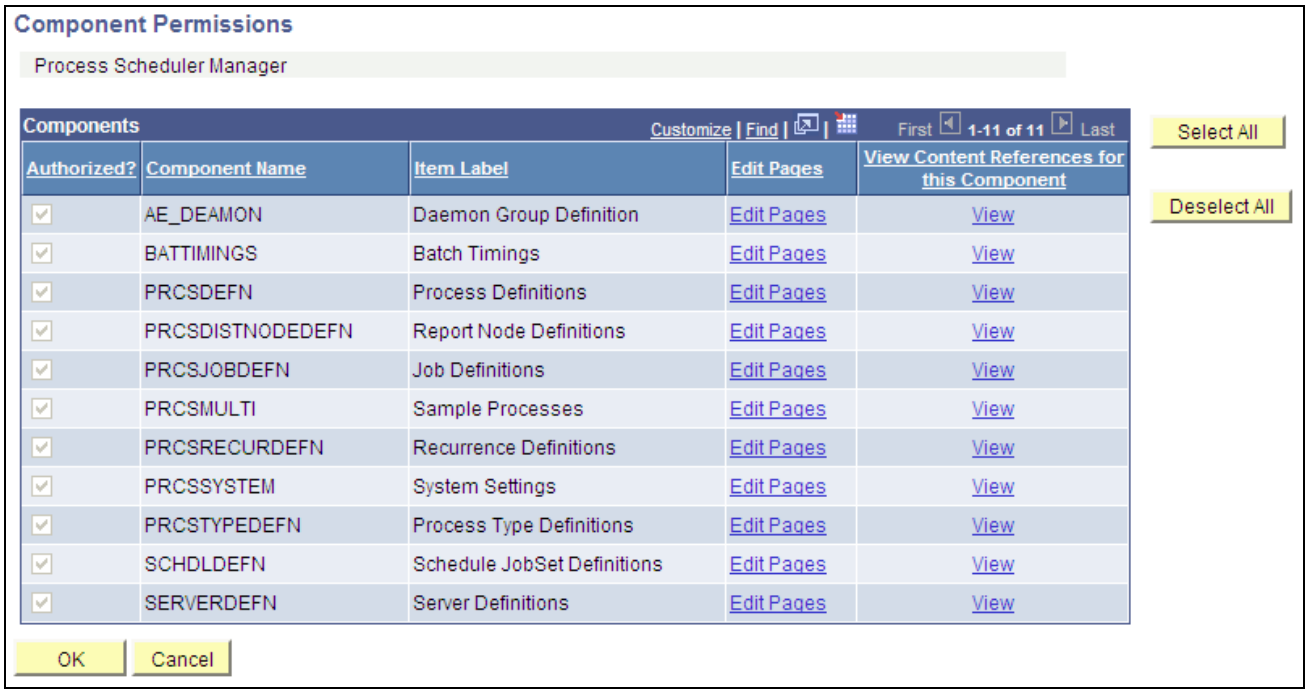
Menu Name Displays all menu names in the database. Add new rows to add more menu names. The name reflects the definition name in PeopleSoft Application Designer.

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:

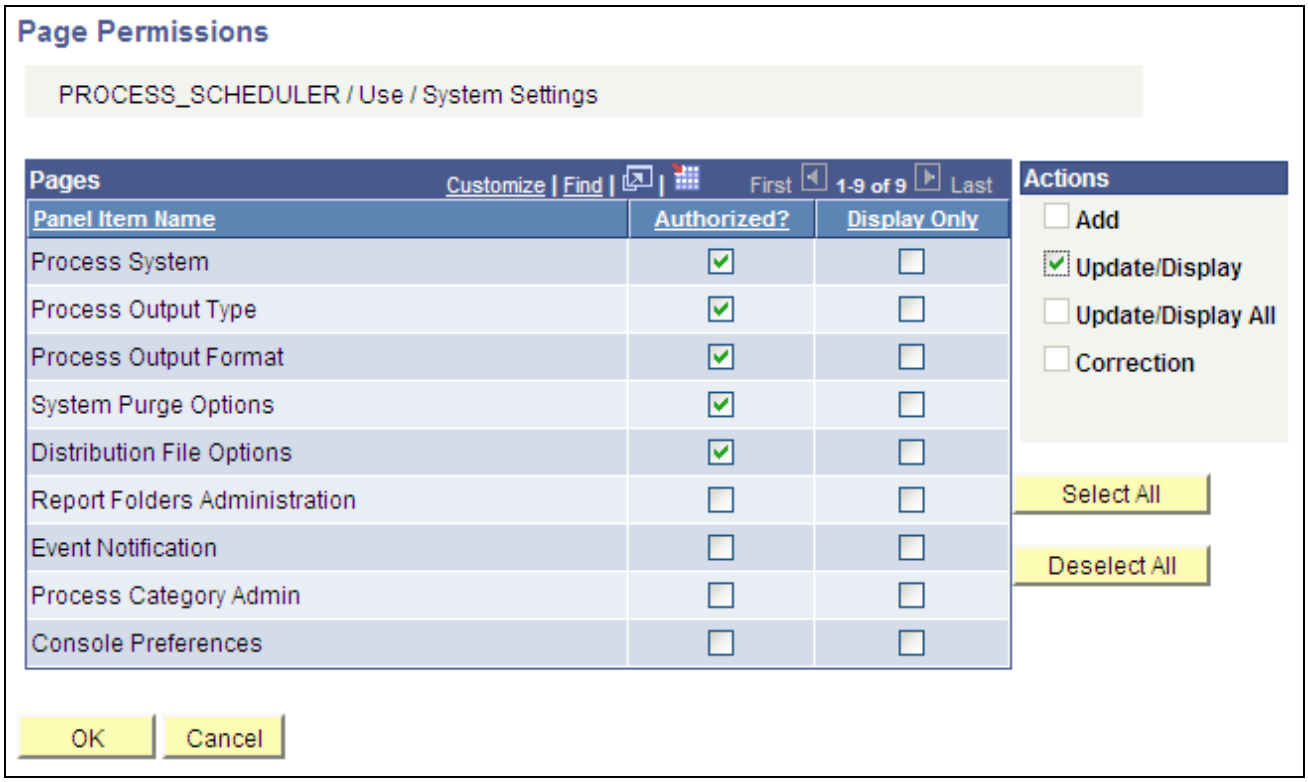


Component Permissions page

This table describes the fields on the Component Permissions page:

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 Components link, the Page Permissions page appears:



Page Permissions page

This table describes the fields on the Page Permissions page:

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 while accessing the page with the browser, 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 deprecated product. These features exist for backward compatibility only.

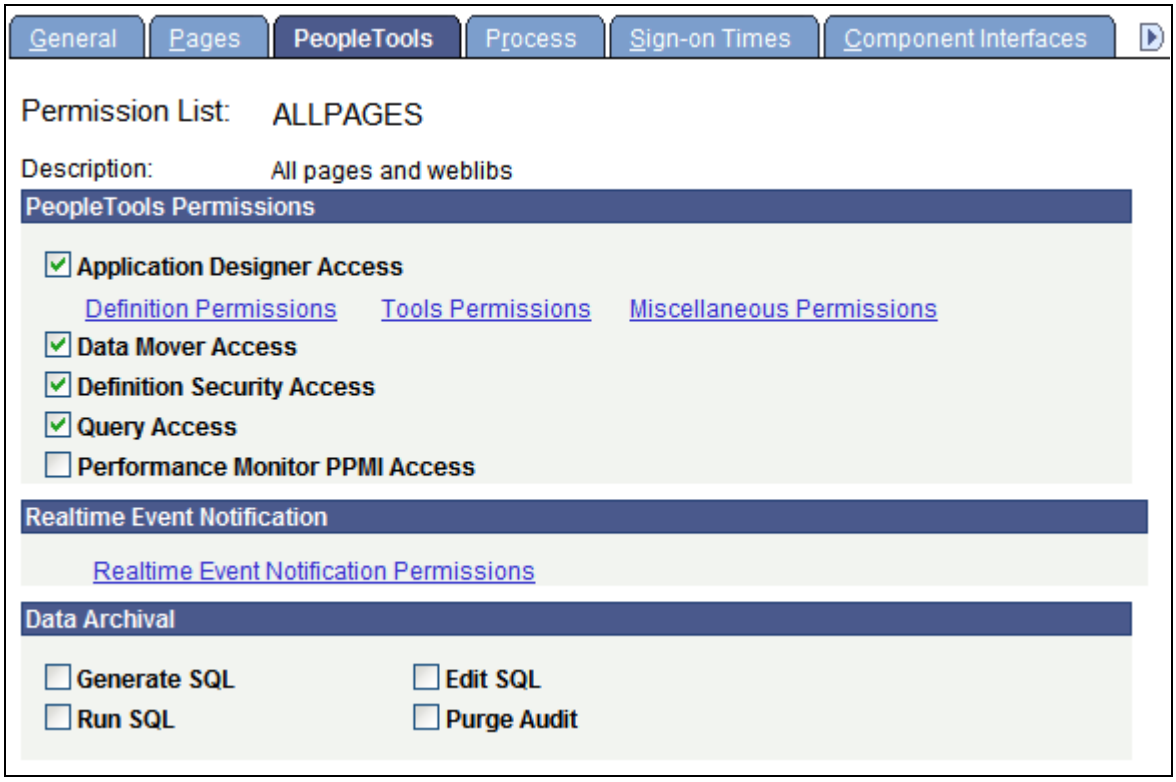
---

1. Select PeopleTools, Security, Permissions & 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 PeopleTools page (select PeopleTools, Security, Permissions and Roles, Permission Lists and click the PeopleTools tab).



Permission Lists - PeopleTools page

The PeopleTools Permissions section of this page applies to standalone PeopleTools applications. They are not 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.

See *PeopleTools 8.52: Performance Monitor*, "Setting Up the Performance Monitor."

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

Definition Permissions

Access the Definition Permissions page (click the Definition Permissions link on the PeopleTools page).

Definition Permissions page

Grant 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 access to *specific* definitions, such as PeopleSoft Payroll Application Engine programs, using Definition Security.

Access 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).

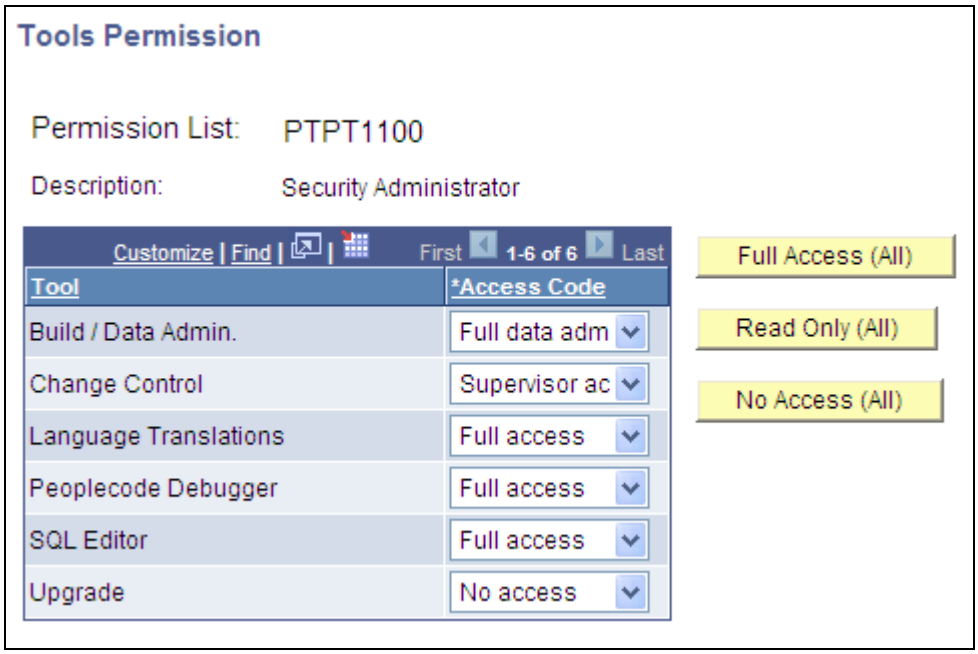
Full Access (ALL), Read Only (ALL), and **No Access (ALL)** Click to set all definition types in the list to the same access level.

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

See *PeopleTools 8.52: PeopleSoft Application Designer Developer's Guide*, "Using PeopleSoft Application Designer," Building and Maintaining Data and Chapter 16, "Implementing Definition Security," page 319.

Tools Permissions

Access the Definition Permissions page (click the Tools Permissions link on the PeopleTools page).



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:

- Build/Data Admin (select Build, Project and Tools, Data Administration).
- Change Control (select Tools, Change Control).
- 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 (ALL) buttons to set across the board settings. Remember that every button affects every access level for the tools.

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.

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.

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.

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.

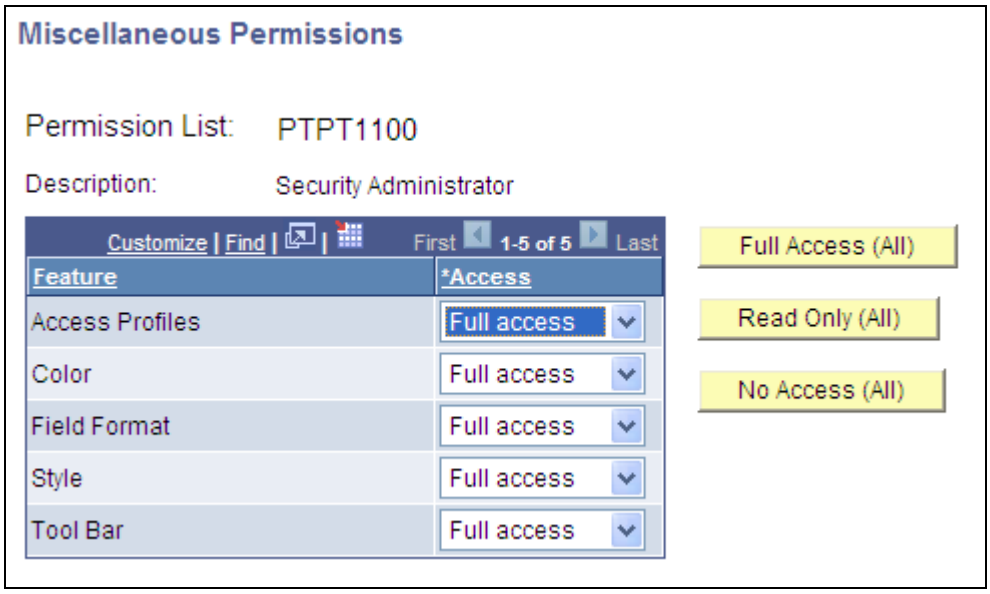
With *Read-only access,* users can run compare reports against the database, but they cannot copy objects into the database.

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 PeopleTools page).



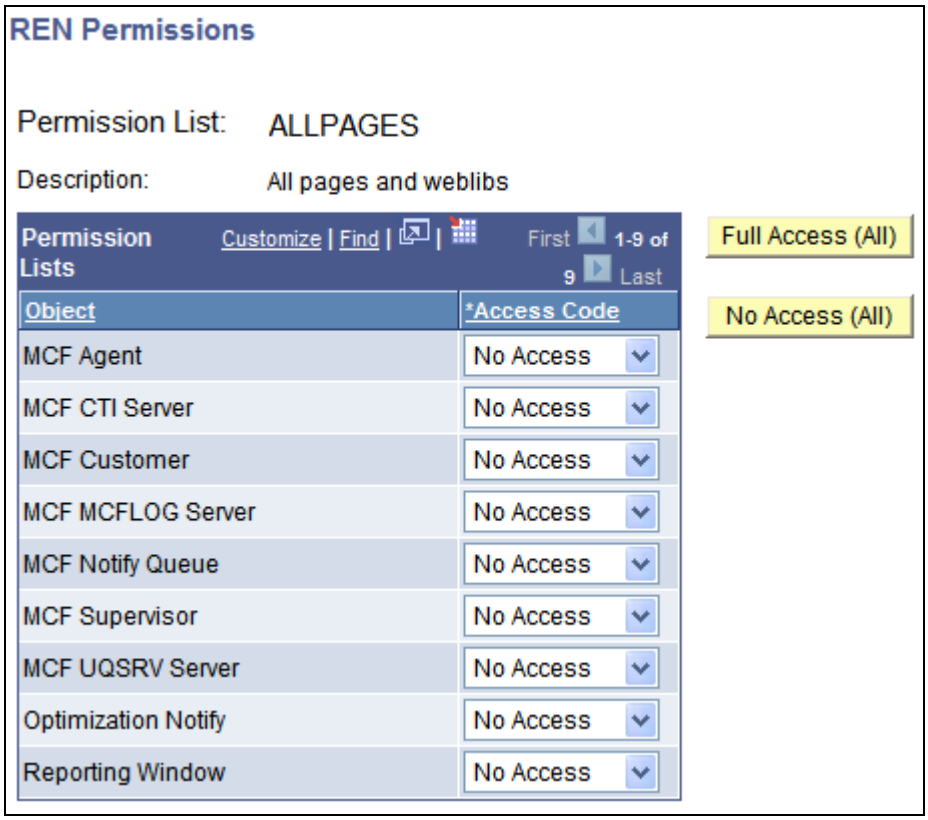
Miscellaneous Permissions page

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 (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 PeopleTools page).



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 *PeopleTools 8.52 : MultiChannel Framework*, "Configuring REN Servers," Configuring REN Server Security.

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.

See Also

PeopleTools 8.52: Data Management, "Using PeopleSoft Data Archive Manager"

Setting Process Permissions

Access the 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.

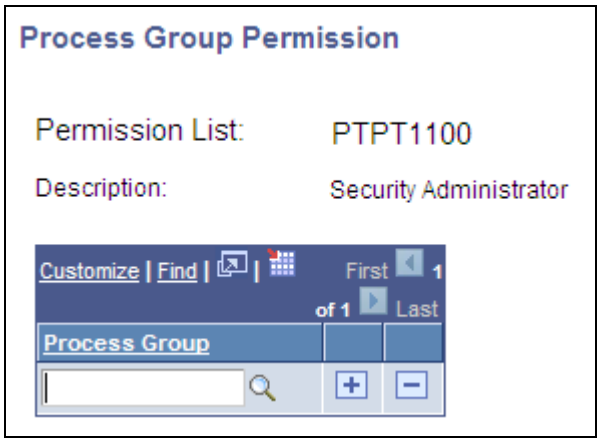
When you create a process permission list, you add the appropriate process groups so that a user belonging to a particular role can invoke the proper batch programs to complete their business transactions. You do this using the Process Group Permission page.

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 (select the Process Group Permissions link on the Process page).



Process Group Permission page

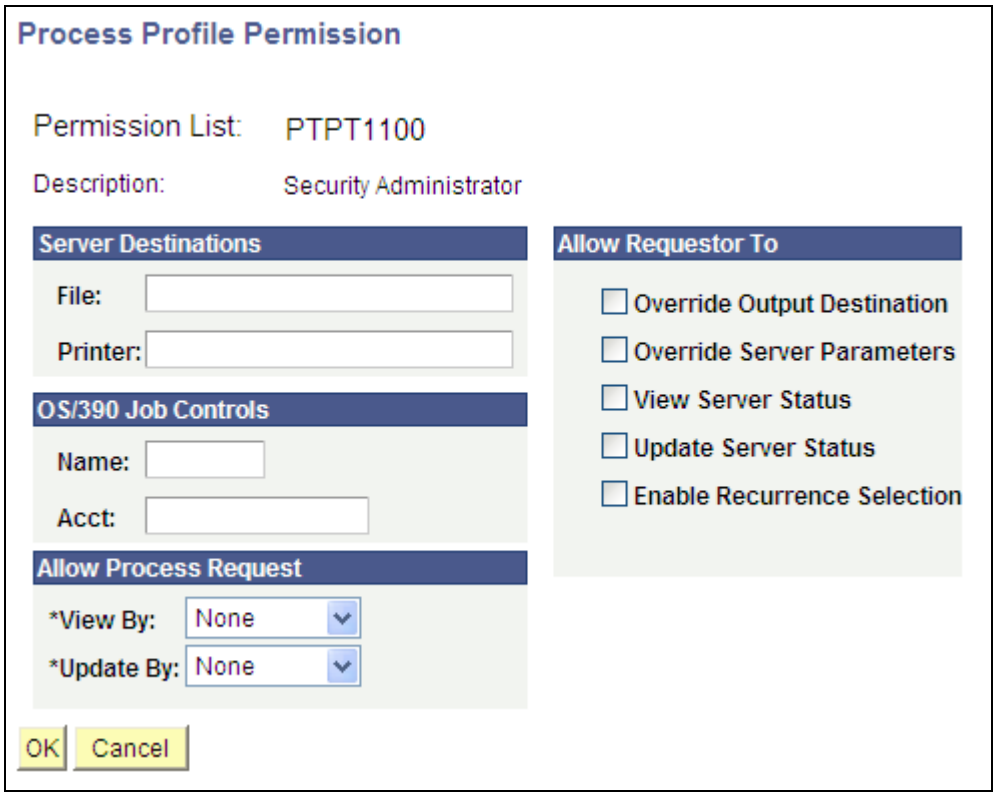
This page lists the process groups associated with a permission list. Process groups are collections of process definitions that you create using PeopleSoft Process Scheduler.

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 Process page).



Process Profile Permission page

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 group of options applies only to DB2 UDB for z/OS.

All PeopleSoft Process Scheduler shell JCLs use meta-strings to pass data stored in the database. PeopleSoft Process Scheduler takes advantage of meta-strings to generate the JCL job cards based on the user who initiated the request. For example, Job Name and Job Account can be passed by setting the Name and Account values, respectively, on the Process Profile page. For z/OS, you have the following options:

- Job:

  Enter *%JOBNAME%*.

- Account:

  Enter *%JOBACCT%*.

See your relational database management system documentation and the PeopleSoft installation guides for details about JCL meta-variables and strings.

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.

  - *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.

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.

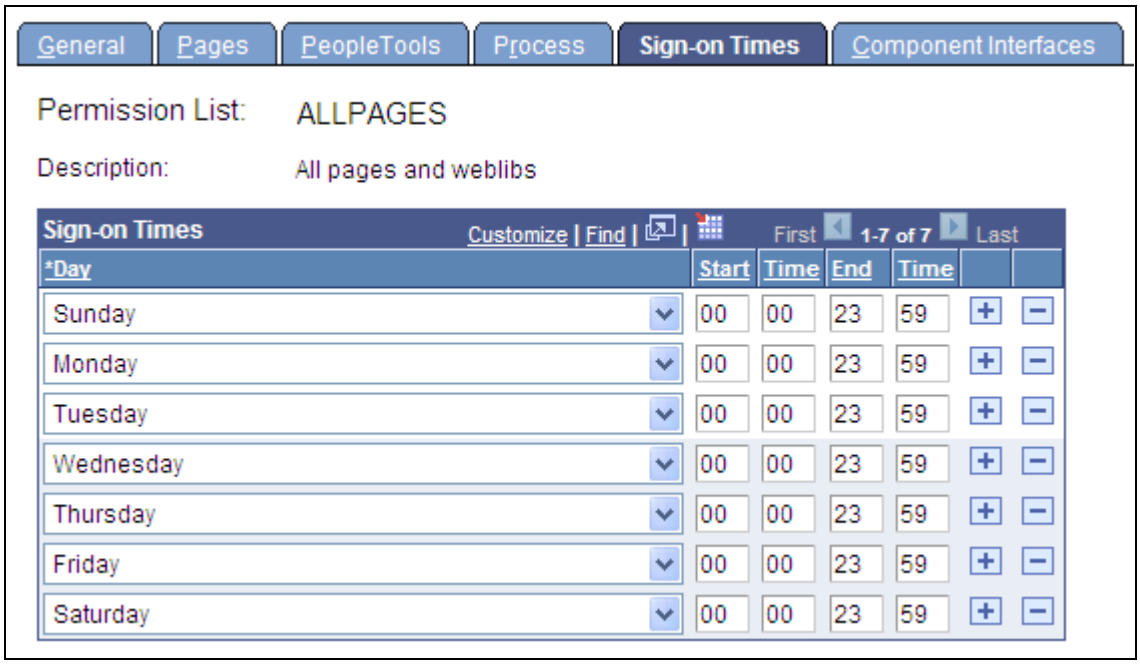
See Also

PeopleTools 8.52: PeopleSoft Process Scheduler, "Setting Up PeopleSoft Process Scheduler Security," Setting Up PeopleSoft Process Scheduler Privileges and Profiles

PeopleTools 8.52: PeopleSoft Process Scheduler, "Defining PeopleSoft Process Scheduler Support Information," Setting Process Definition Options

Setting Sign-on Time Permissions

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



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.

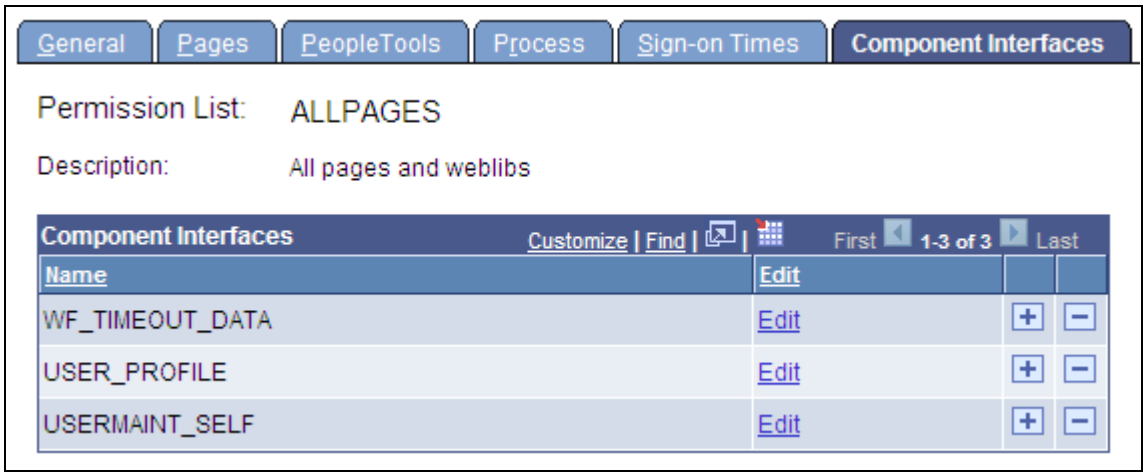
To create a sign-on time that spans multiple days, use adjoining sign-on times. For example, to create a sign-on time running from 8 p.m. Tuesday to 6 a.m. Wednesday, you need a Tuesday start time of 20:00 and end time of 23:59. Then you need to add a Wednesday sign-on time with a start time of 0:00 and an end time of 5:59.

By default, all start times are 0:00 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 Component Interfaces page (select PeopleTools, Security, Permissions and Roles, Permission Lists and click the Component Interfaces tab).

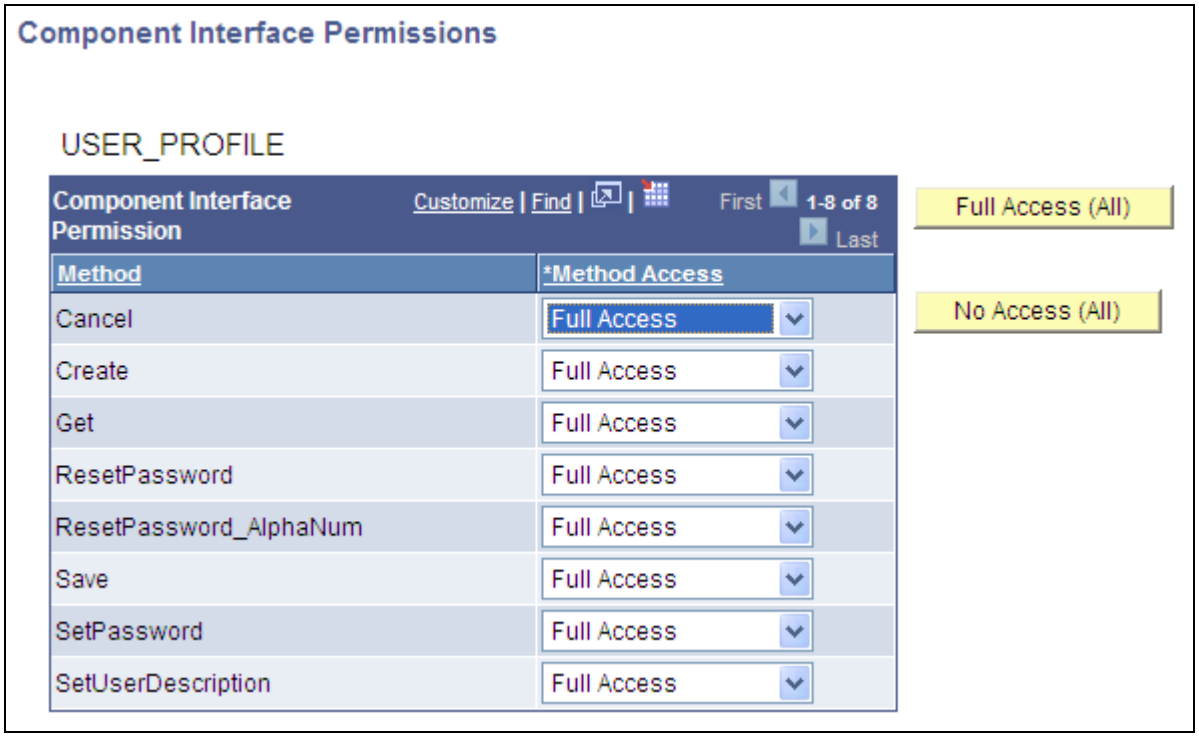


Permission Lists - Component Interfaces page

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:



Component Interface Permissions page

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.

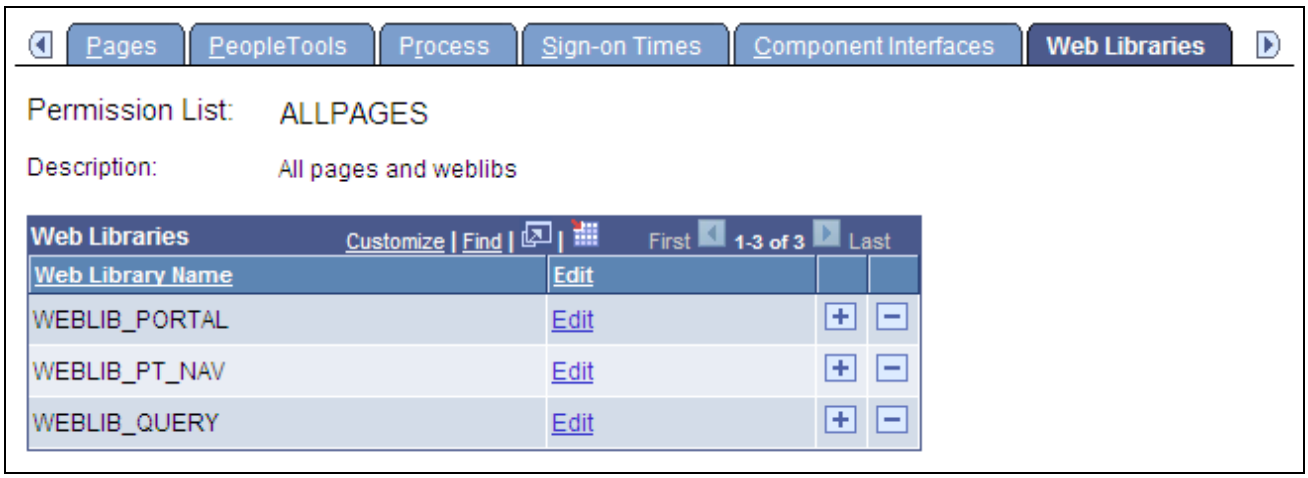
You grant access to component interfaces similarly to adding page access. Add a new row to insert a component interface into the definition list. You must also grant access to the component interface methods.

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 Web Libraries page (select PeopleTools, Security, Permissions and Roles, Permission Lists and click the Web Libraries tab).



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.

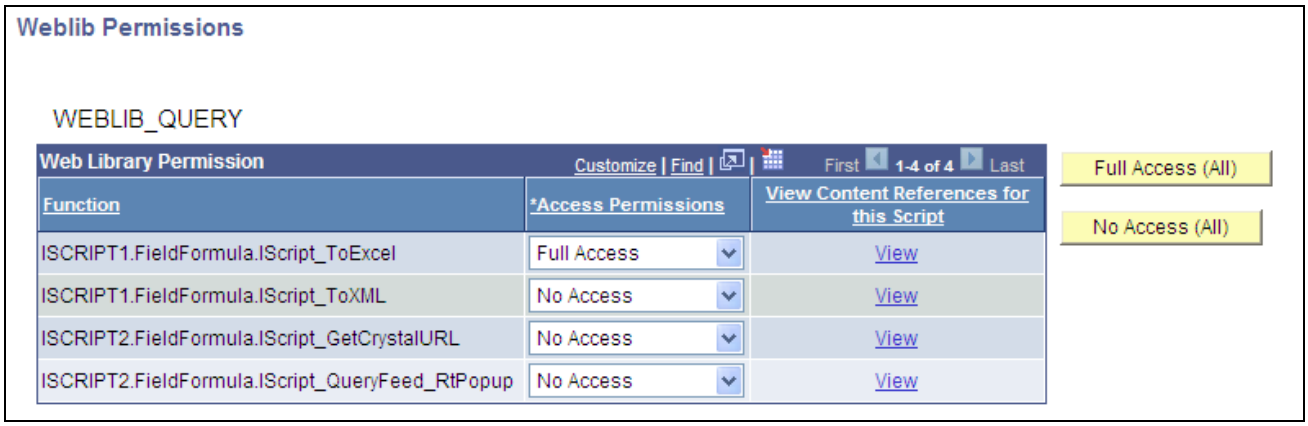
Web Library Name Displays the web libraries added to the permission list.

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:



Weblib Permissions page

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.

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.

See Also

PeopleTools 8.52: PeopleCode Developer's Guide, "PeopleCode and PeopleSoft Pure Internet Architecture," Using Internet Scripts

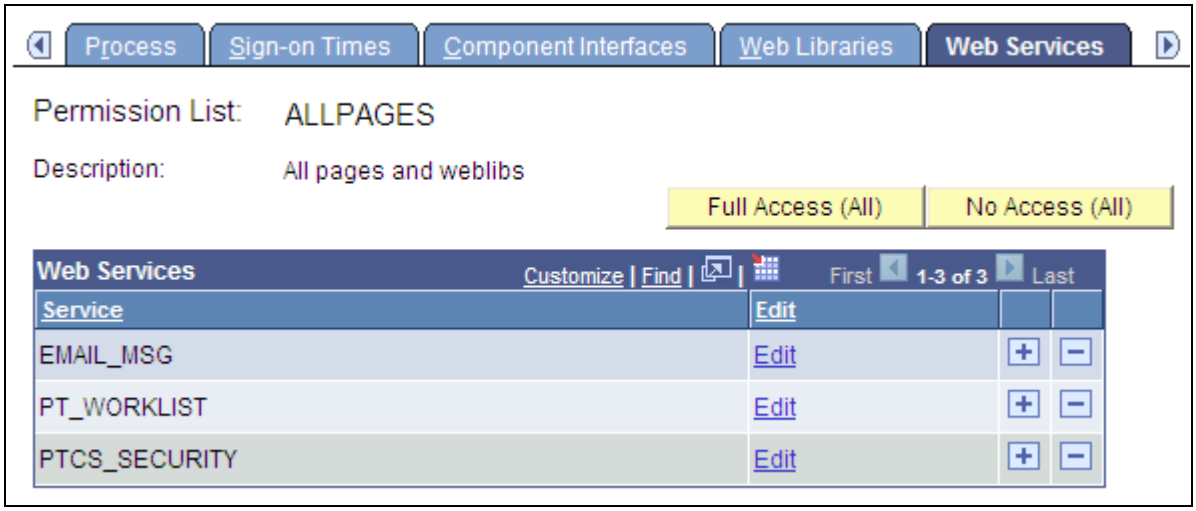
Setting Web Services Permissions

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, then 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 *PeopleTools 8.52: PeopleSoft Integration Broker PeopleBook.*

See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Setting Up Secure Integration Environments."

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



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

Service Displays the name of the web service defined in the PeopleSoft system.

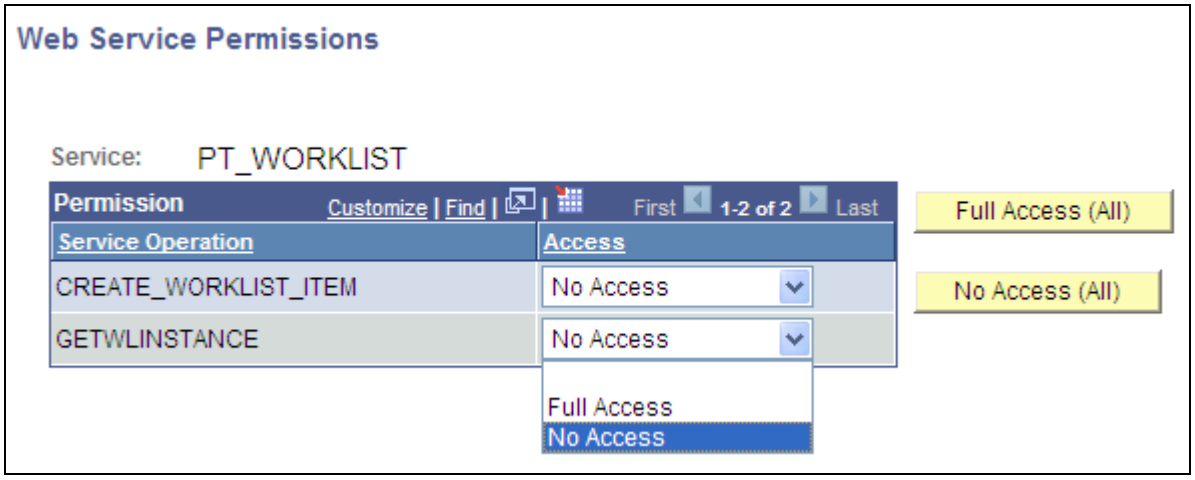
Edit Click to launch the Web Service Permissions page.

Full Access (All) Click to grant full access to all services listed on the page.

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).



Web Service Permissions page

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.

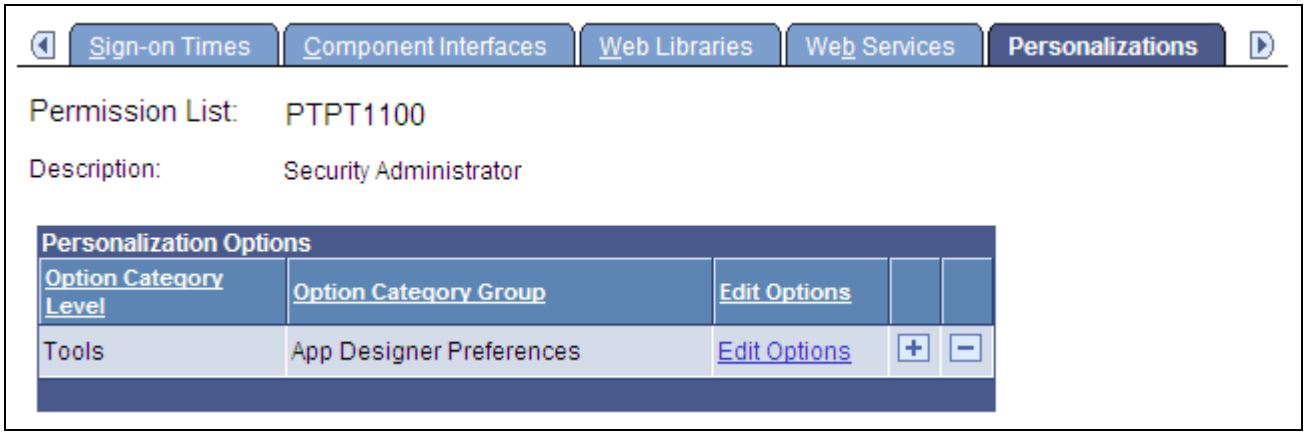
See Also

PeopleTools 8.52: PeopleSoft Integration Broker, "Managing Service Operations," Setting Permissions to Service Operations

PeopleTools 8.52: PeopleTools Portal Technologies, "Configuring WS-Security for WSRP Consumption and Production"

Setting Personalization Permissions

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



Permission Lists - Personalizations page

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

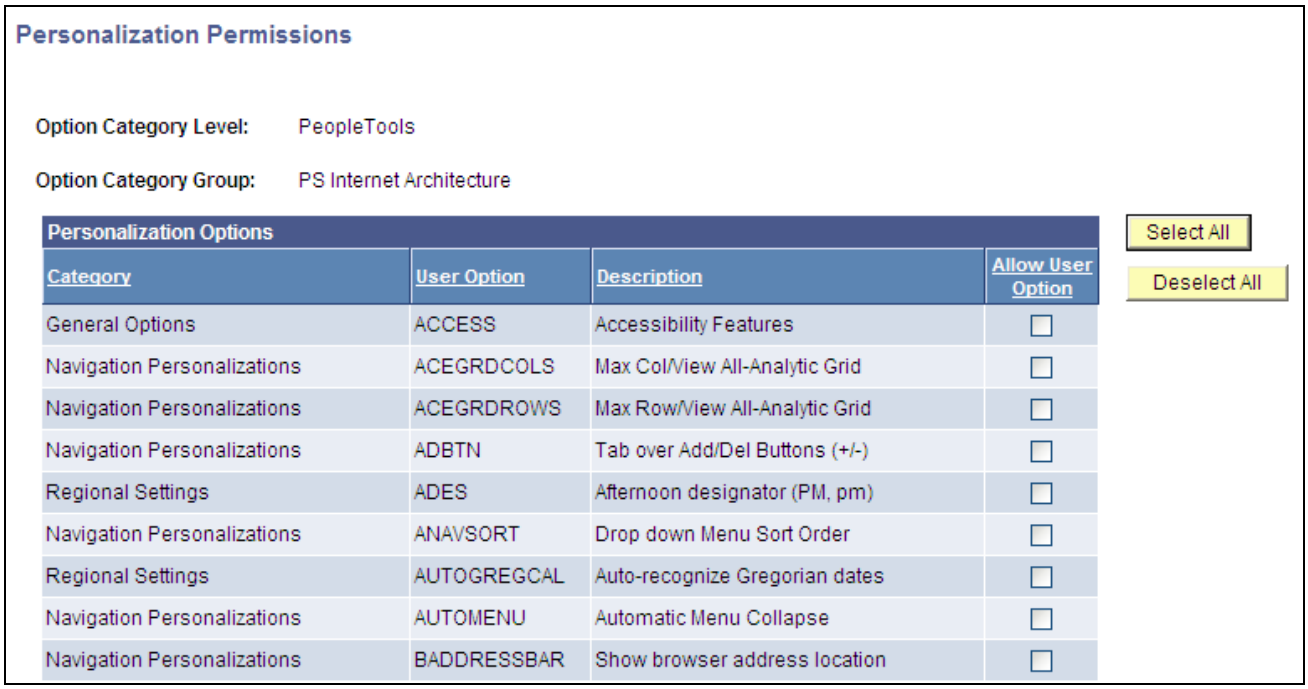
Option Category Level Displays the high-level grouping of personalizations.

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.



Personalization Permissions

Option Category Level: PeopleTools

Option Category Group: PS Internet Architecture

Personalization Options

Category	User Option	Description	Allow User Option
General Options	ACCESS	Accessibility Features	☐
Navigation Personalizations	ACEGRDCOLS	Max Col/View All-Analytic Grid	☐
Navigation Personalizations	ACEGRDROWS	Max Row/View All-Analytic Grid	☐
Navigation Personalizations	ADBTN	Tab over Add/Del Buttons (+/-)	☐
Regional Settings	ADES	Afternoon designator (PM, pm)	☐
Navigation Personalizations	ANAVSORT	Drop down Menu Sort Order	☐
Regional Settings	AUTOGREGCAL	Auto-recognize Gregorian dates	☐
Navigation Personalizations	AUTOMENU	Automatic Menu Collapse	☐
Navigation Personalizations	BADDRESSBAR	Show browser address location	☐

Select All

Deselect All

Personalization Permissions page

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. 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 deselect the Allow User Option check box for every row in the grid.

See Chapter 17, "Managing PeopleSoft Personalizations," page 329.

Setting Query Permissions

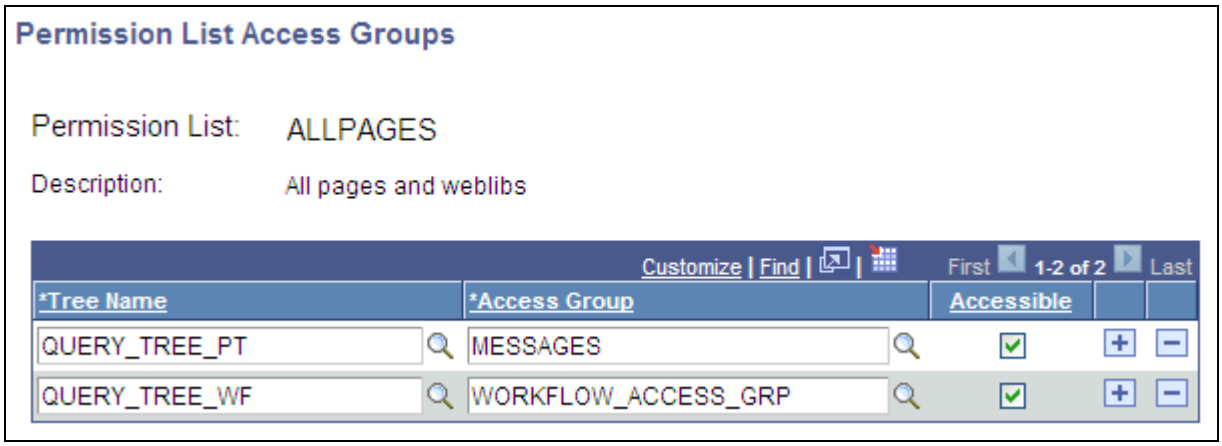
Access the Query page (select PeopleTools, Security, Permissions and Roles, Permission Lists and click the Query tab).

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 Query page).



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.

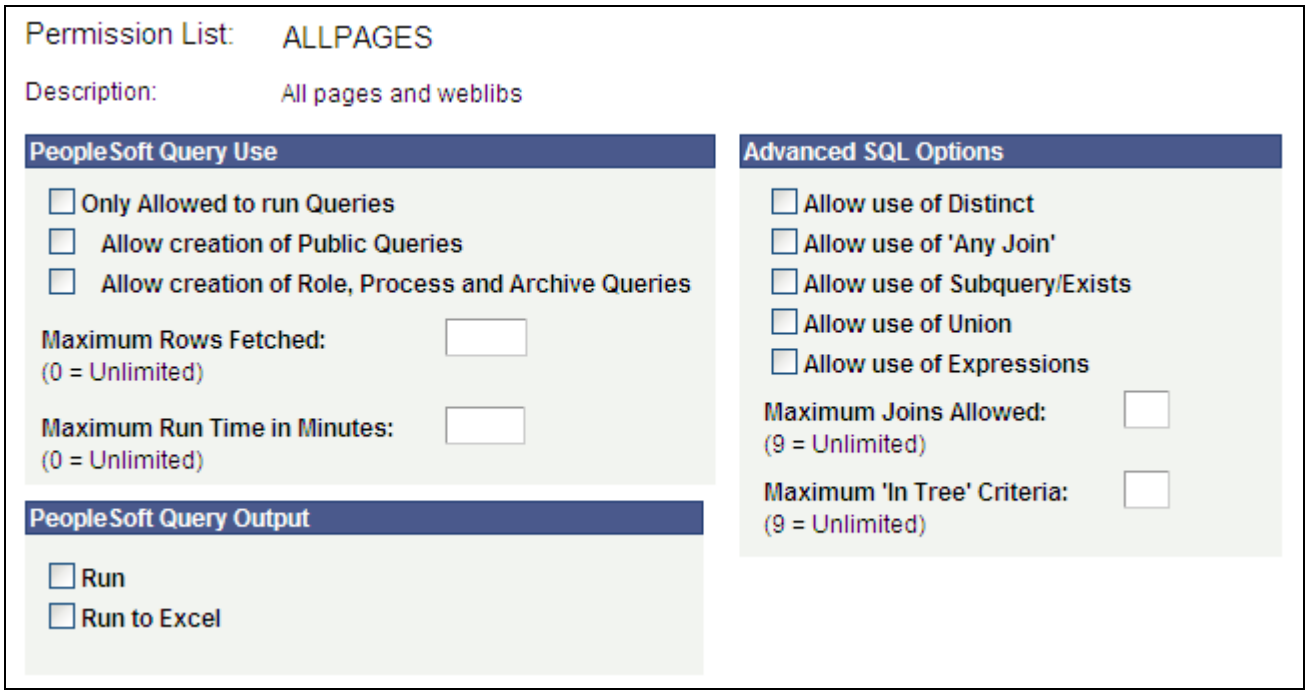
   Note. Because it hinders system performance, do not deselect the Accessible check box for lower-level access groups. To restrict access to record components on a particular branch of a tree, consider creating a new tree for those definitions. Attempting to expand an access group that is not accessible 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 Query page).



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.

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.

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.

Setting Mass Change Permissions

Access the Mass Change page (select PeopleTools, Security, Permissions and Roles, Permission Lists and click the Mass Change tab).

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.

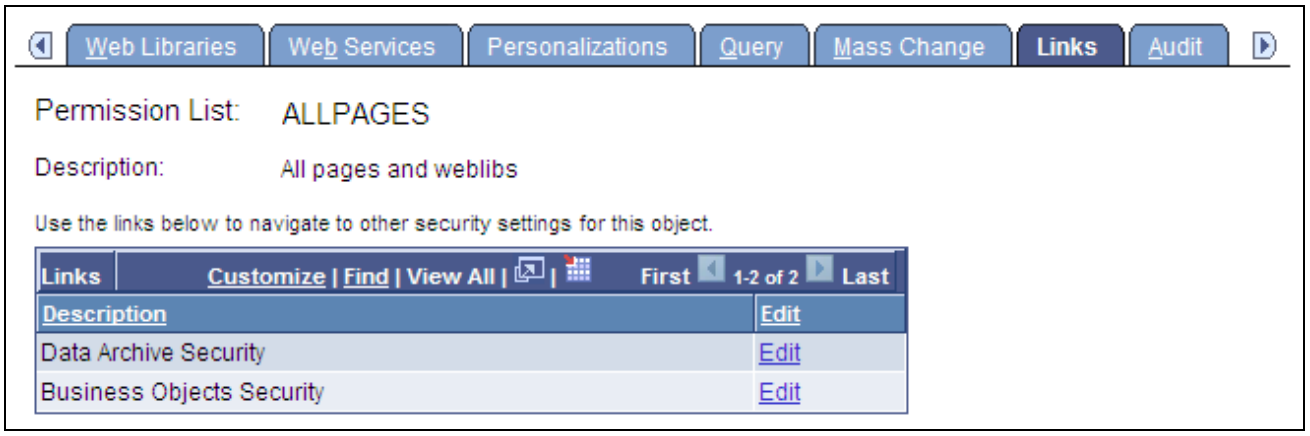
To modify mass change template permissions:

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 Links page (select PeopleTools, Security, Permissions & Roles, Permission Lists and click the Links tab).



Permission List - 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 (PeopleTools, Security, Security Objects, Security Links) component.

See Also

Chapter 1, "Getting Started with Security Administration," Administering Security from Applications, page 9

Viewing When a Permission List Was Last Updated

Access the Audit page (select PeopleTools, Security, Permissions & Roles, Permission Lists and click the Audit tab).

Permission List - Audit page

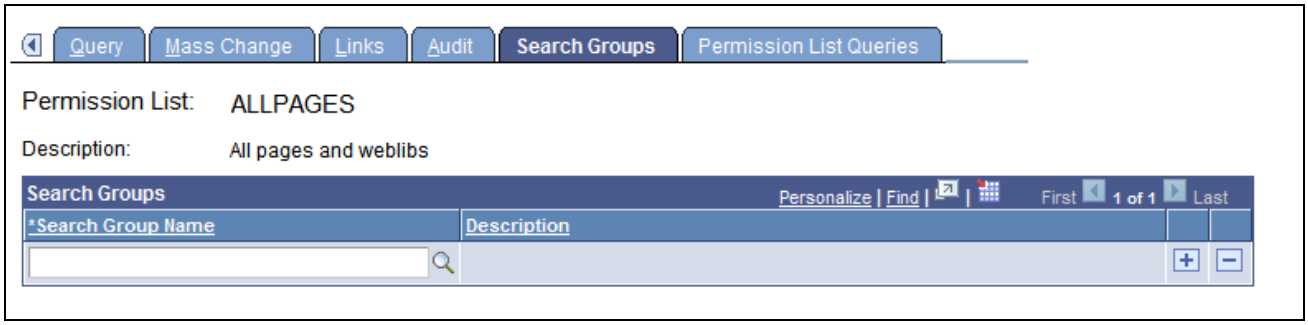
View when a permission list was last updated and by whom. You can also view who has made changes to security tables by using the Database Level Auditing feature.

See Also

PeopleTools 8.52: Data Management, "Employing Database Level Auditing," Understanding Database Level Auditing

Assigning Search Group Permissions

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



Permission Lists – Search Groups

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

To assign a search group to a permission list:

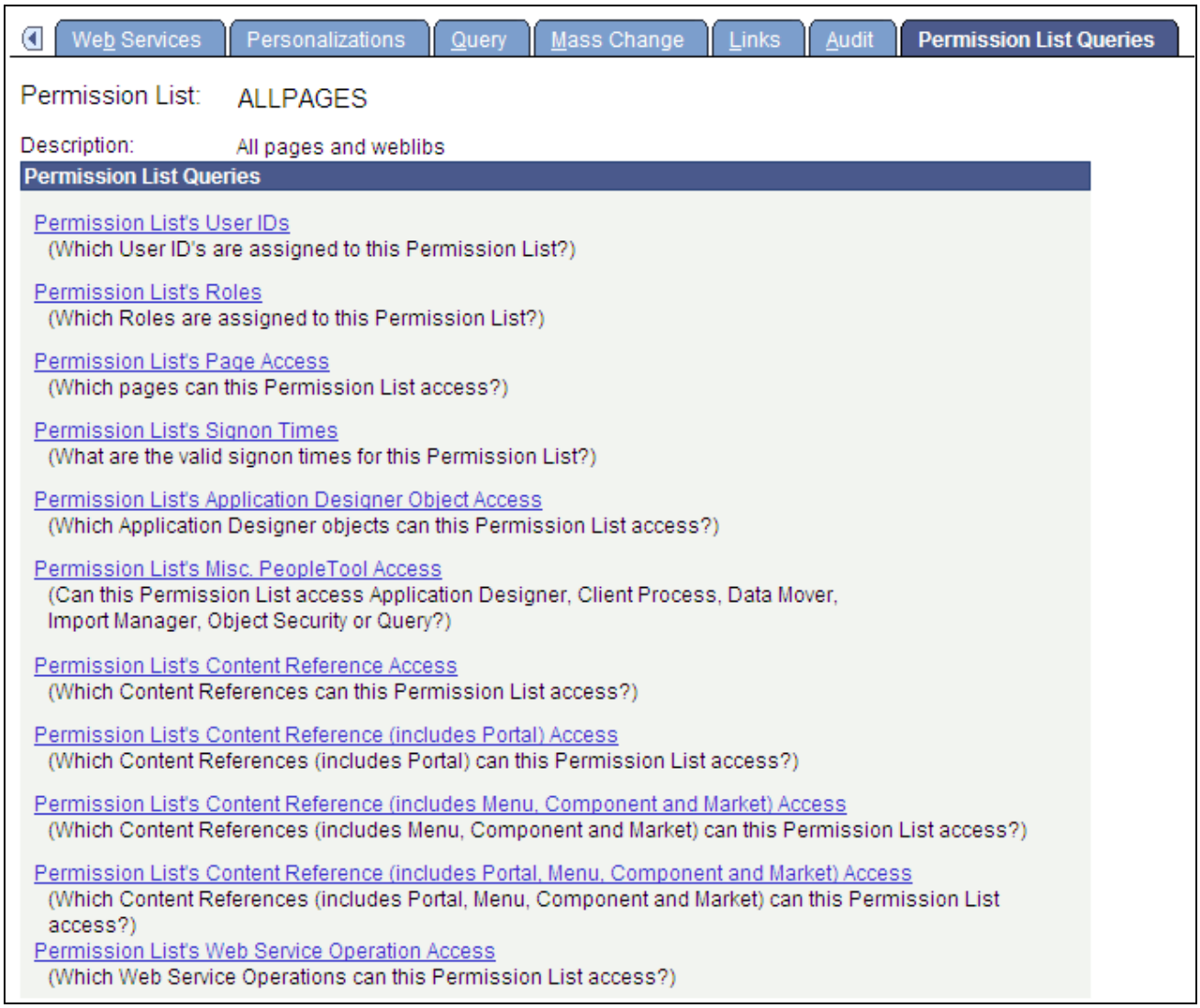
1. Access the Permission Lists - Search Groups page.
2. Click the Search Group Name lookup button to search for and select a search group to add to the permission list.
3. Click the Save button.

See Also

PeopleTools 8.52: PeopleSoft Search Technology, "Working with PeopleSoft Search Framework Security Features," Setting Up Role-Based Search Group Access

Running Permission List Queries

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



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 values (.csv) file.

- XML file.

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

Chapter 4

Setting Up Roles

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

- Manage roles.
- Define role options.
- Create a NEWUSER role.
- Execute dynamic role rules.
- Using the PeopleSoft Administrator role.

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 & 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 & 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.

See Also

Chapter 4, "Setting Up Roles," Displaying Dynamic Role Members, page 75

Defining Role Options

This section discusses how to:

- Assign permissions to roles.
- Display static role members.
- Display dynamic role members.
- Execute dynamic role rules.
- Set user routing options.
- Decentralize role administration.
- Display additional links for user profiles.
- Run role queries.
- View when a role was last updated.

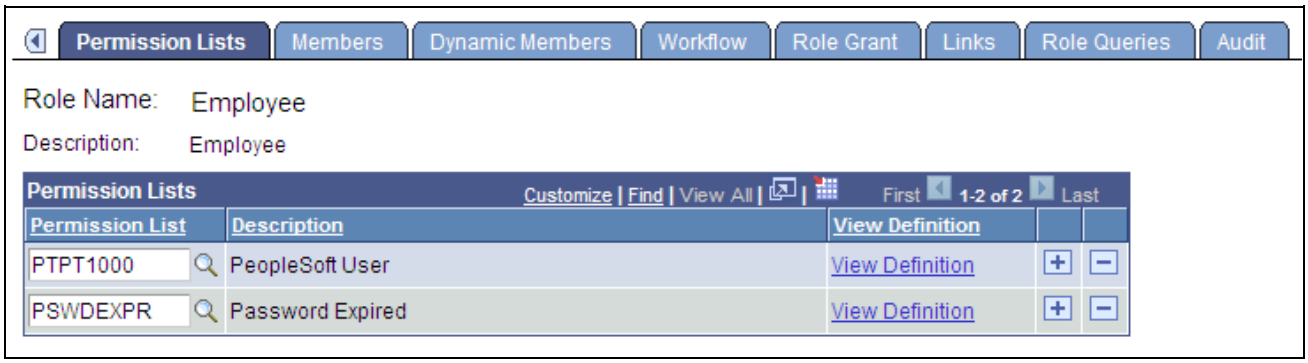
Pages Used to Define Role Options

Page Name	*Definition Name*	*Navigation*	*Usage*
General	ROLEDEFN	PeopleTools, Security, Permissions & Roles, Roles, General	Describe the role.
Permissions Lists	ROLE_CLASS	PeopleTools, Security, Permissions & Roles, Roles, Permission Lists	Grant permissions to roles.
Members	ROLE_MEMBER	PeopleTools, Security, Permissions & Roles, Roles, Members	View the current list of static role members.
Dynamic Members	ROLE_DYNMEMBER	PeopleTools, Security, Permissions & Roles, Roles, Dynamic Members	View the current list of dynamic role members. If you aren't using the dynamic roles, this list isn't populated.

Page Name	Definition Name	Navigation	Usage
Workflow	ROLEWRKFLOW	PeopleTools, Security, Permissions & Roles, Roles, Workflow	Set user routing options.
Role Grant	ROLE_GRANT	PeopleTools, Security, Permissions & Roles, Roles, Role Grant	Decentralize role administration.
Links	ROLE_OTHER	PeopleTools, Security, Permissions & Roles, Roles, Links	View additional links for user profiles.
Role Queries	ROLE_QUERY	PeopleTools, Security, Permissions & Roles, Roles, Role Queries	Run queries about a role.
Audit	ROLE_AUDIT	PeopleTools, Security, Permissions & Roles, Roles, Audit	View when a permission list was last updated.

Assigning Permissions to Roles

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



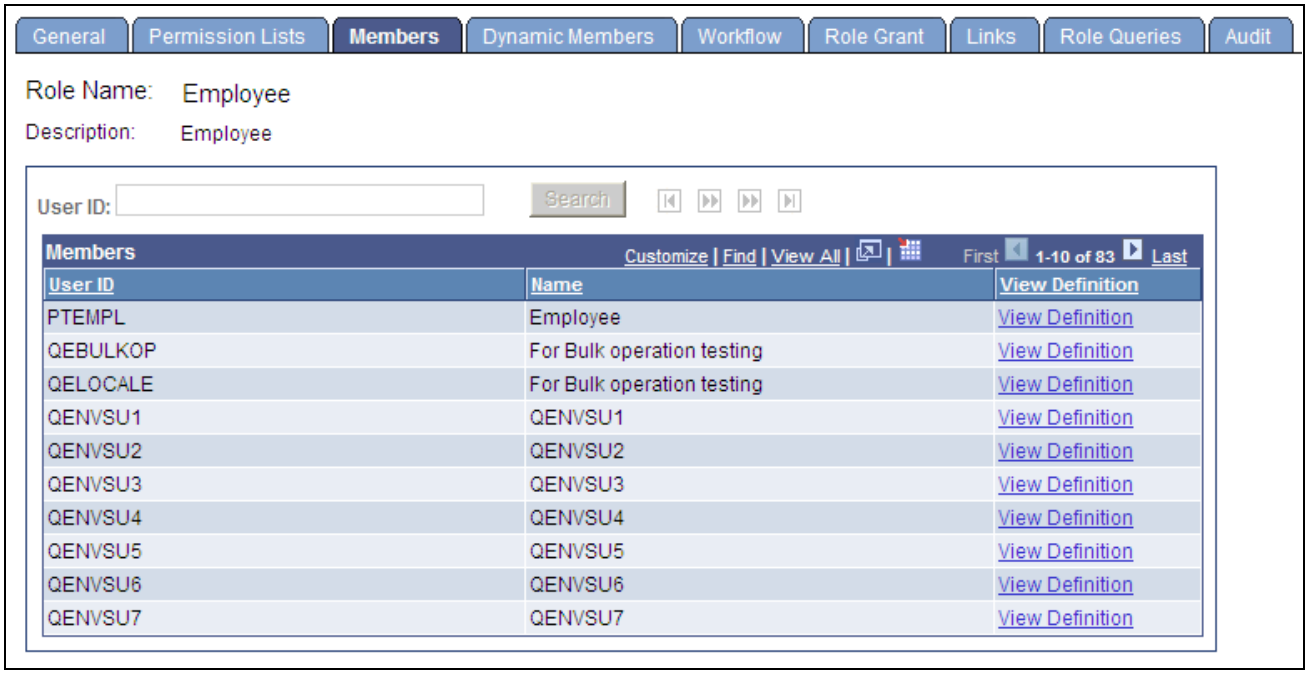
Roles - Permission Lists page

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.

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 & Roles, Roles and click the Members tab).



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.

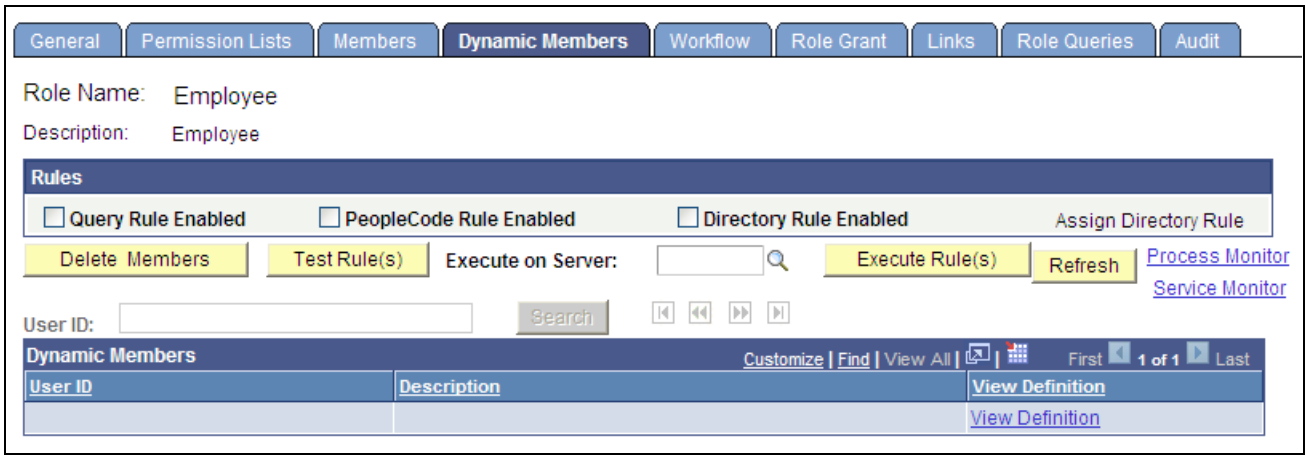
User ID Enter part or all of a role member user ID for which to search.

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 Dynamic Members page (select PeopleTools, Security, Permissions & Roles, Roles and click the Dynamic Members tab).



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.

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.

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 Chapter 4, "Setting Up Roles," Executing Dynamic Role Rules, page 87.

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.

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:

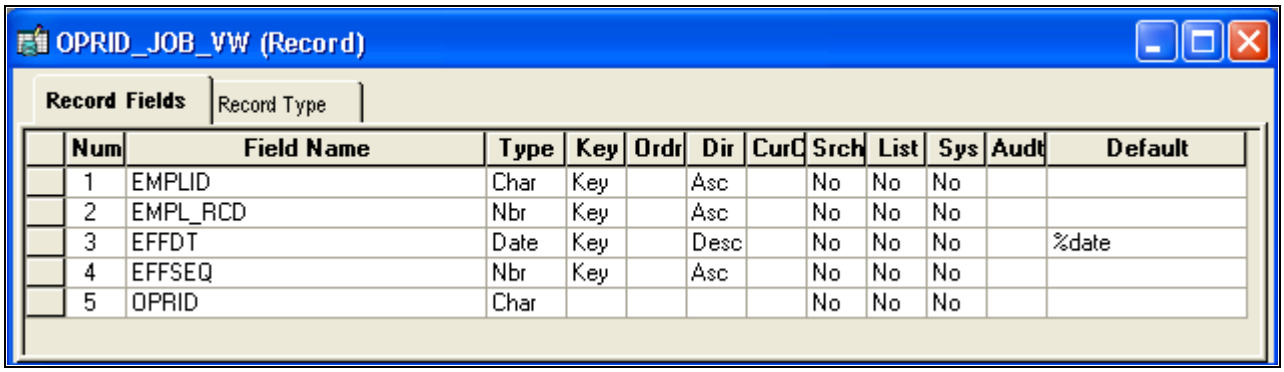
1. Create a view.
2. Create the query.
3. Run the dynamic rule.

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.

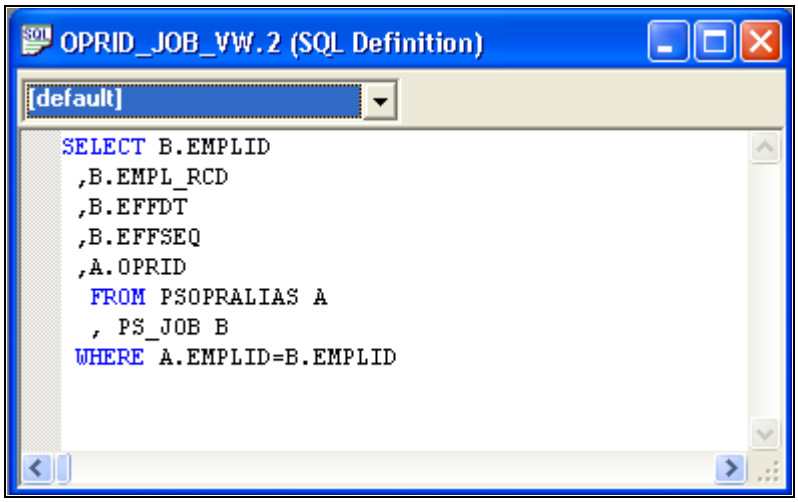
This example shows a possible view definition for the example role rule:



Num	Field Name	Type	Key	Ordr	Dir	CurD	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		

Example of dynamic role rule, SQL view

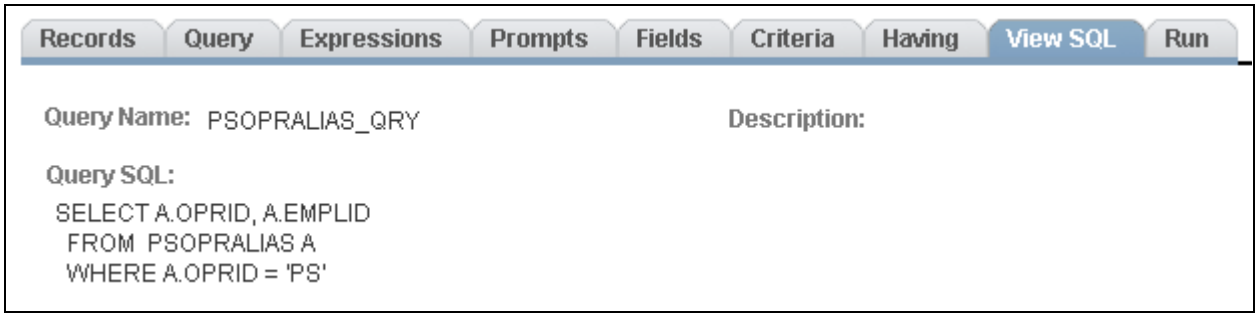
The associated SQL object is:



Dynamic role rule, SQL definition

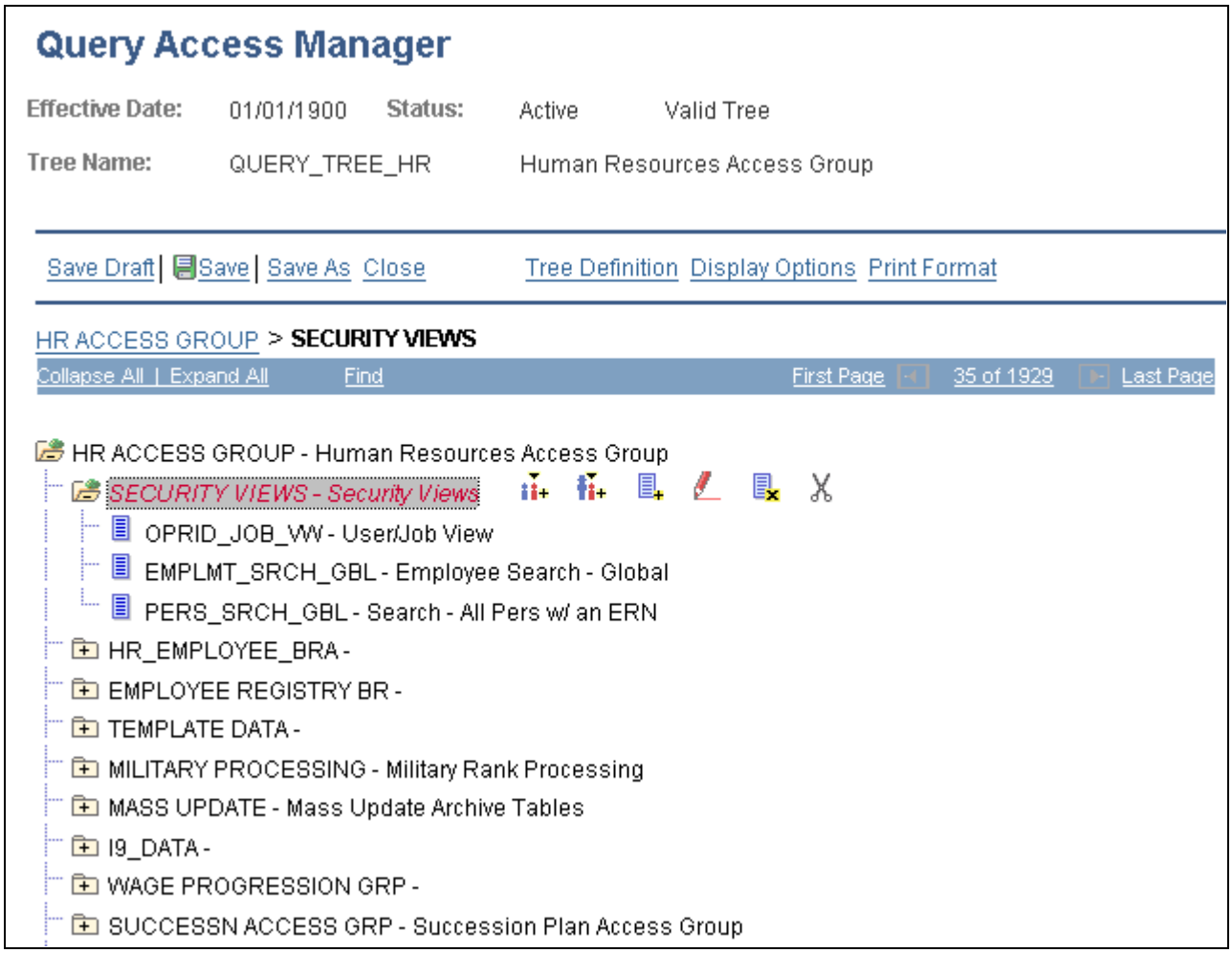
Note. The OPRID field must not be a key in this view because PeopleTools appends AND OPRID = "<CURRENT_USER_ID>" in PeopleSoft Query Manager. This action occurs if you use the record OPRALIAS directly in the query.

The SQL is:



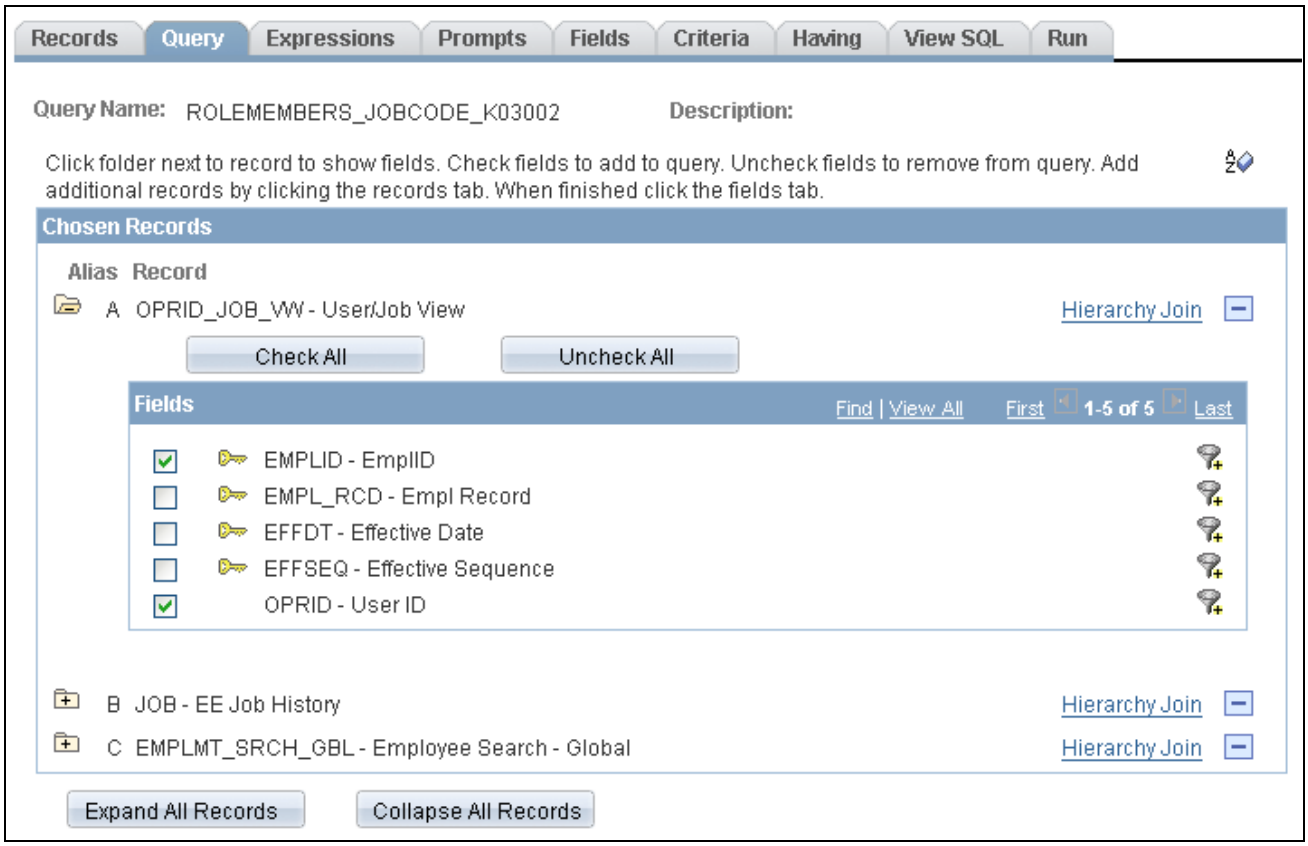
Query view SQL

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:



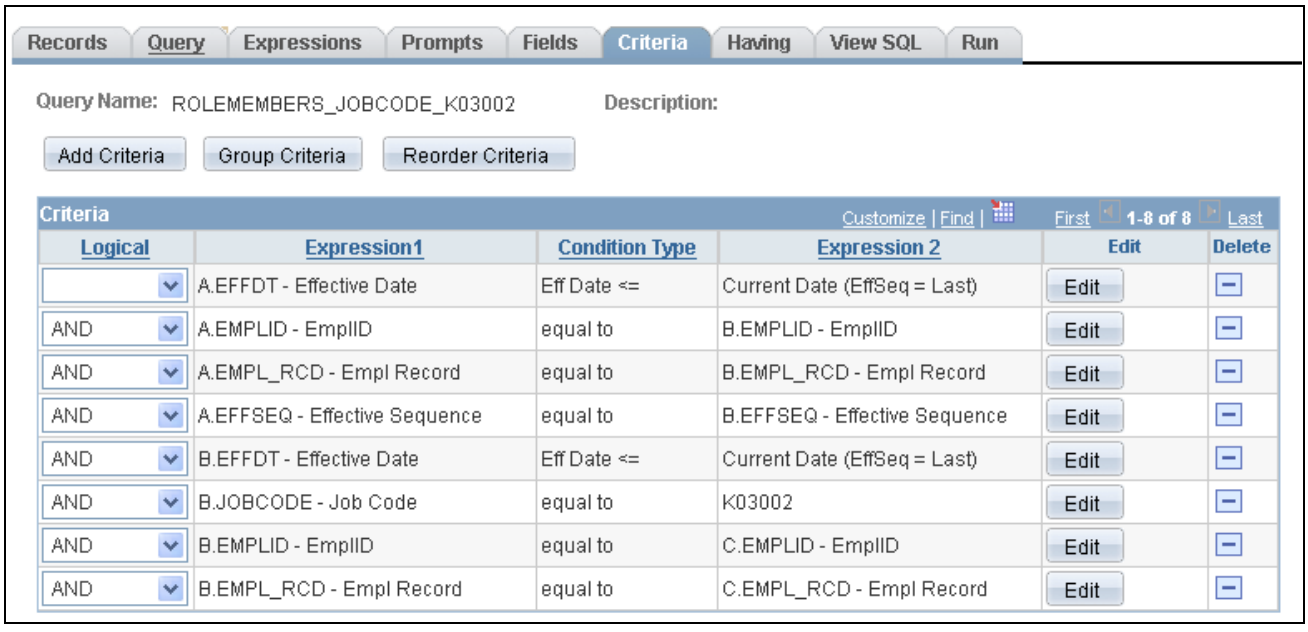
Adding the view to a query tree

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:



Example of Query page

The query contains the following criteria:



Example of Criteria page

The SQL for the query is:

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.`

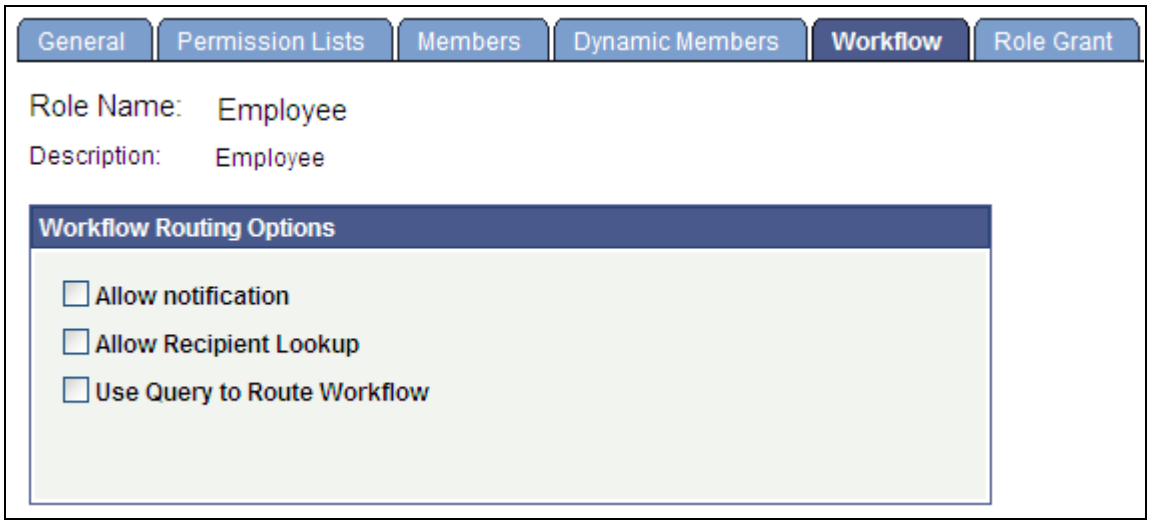
Note. When you save a query used for a dynamic role query, you should specify that it is a role query.

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 Workflow page (select PeopleTools, Security, Permissions and Roles, Roles and click the Workflow tab).



Roles - Workflow page

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 Select to determine workflow routings by a workflow query. This value depends on your workflow scheme.

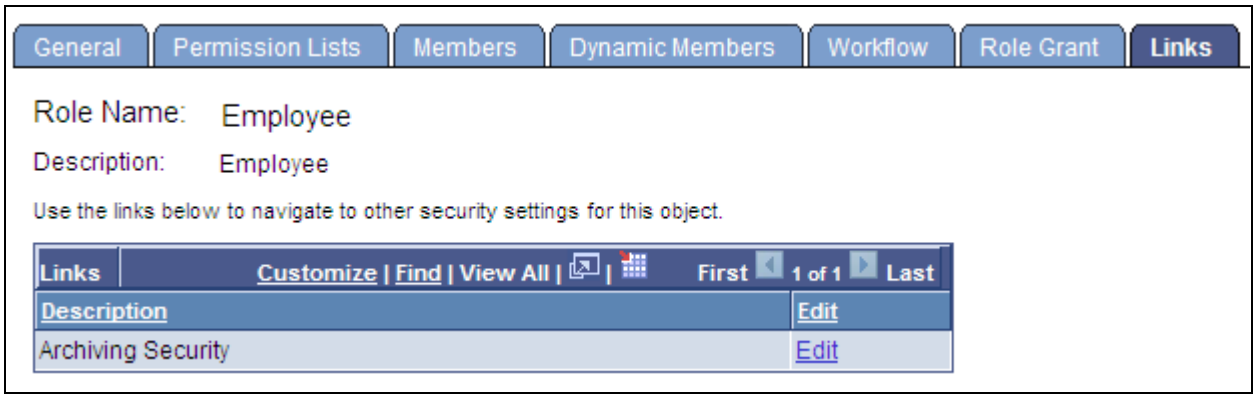
Decentralizing Role Administration

You use the 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 Chapter 5, "Administering User Profiles," Implementing Distributed User Profiles, page 122.

Displaying Additional Links

Access the Links page (select PeopleTools, Security, Permissions & Roles, Roles and click the Links tab).



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.

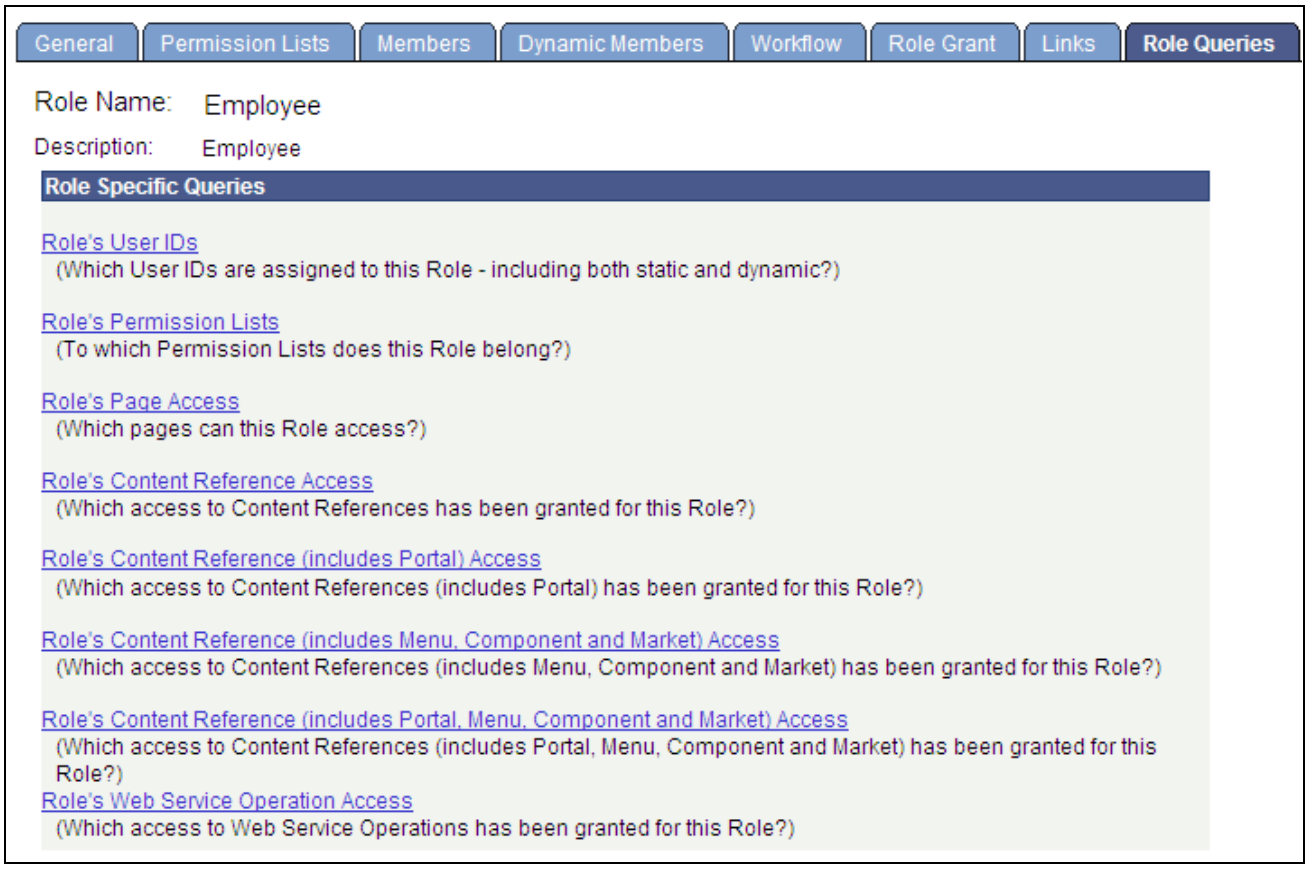
If you have added any links for roles in the Security Links component, they appear on the Links page.

See Also

Chapter 1, "Getting Started with Security Administration," Administering Security from Applications, page 9

Running Role Queries

Access the Links page (select PeopleTools, Security, Permissions & Roles, Roles and click the Role Queries tab).



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 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

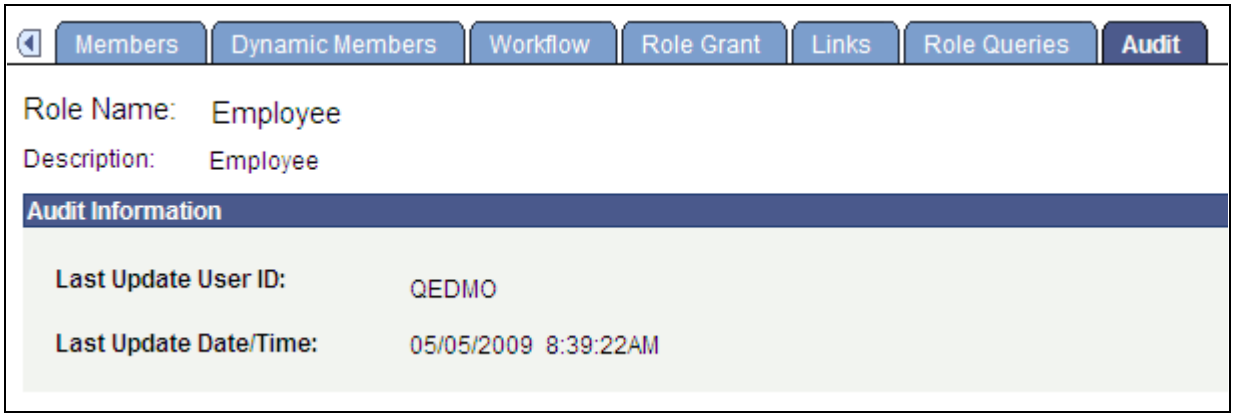
     Downloads the query results as a Microsoft Excel spreadsheet (.xls) file.

   - CSV text file

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

Viewing When a Role Was Last Updated

Access the Audit page (select PeopleTools, Security, Permissions & Roles, Roles and click the Audit tab).



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.

See Also

PeopleTools 8.52: Data Management, "Employing Database Level Auditing," Understanding Database Level Auditing

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 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.
- 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 & 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 on this page launches the DYNROLE_PUBL application engine program which executes the rule(s).

See Chapter 4, "Setting Up Roles," Displaying Dynamic Role Members, page 75.

Executing Dynamic Role Rules for a User Profile

To execute a dynamic role rule for a single 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 on this page launches the DYNROLE_PUBL application engine program which executes the rule(s).

See Chapter 5, "Administering User Profiles," Setting Roles, page 106.

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 & Roles, Execute Role Rules. The following example shows the page:

Use the Dynamic Role Rules page to execute role rules for all roles and user profiles.

The Dynamic Role Rules page features the following page controls:

Server Name	Enter the name of the process scheduler server to run the rule(s).
Execute Dynamic Role Rules	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.
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.

Using the PeopleSoft Administrator Role

The PeopleSoft Administrator role gives full access to all 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

Administering User Profiles

This chapter provides an overview of user profiles and discusses how to:

- Set up access profiles.
- Set up user profile types.
- Work with user profiles.
- Specify user profile attributes.
- Work with passwords.
- Implement distributed user profiles.
- Transfer users between databases.
- Track user sign in and sign out activity.
- Purge inactive user profiles .
- Preserve historical data.

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.

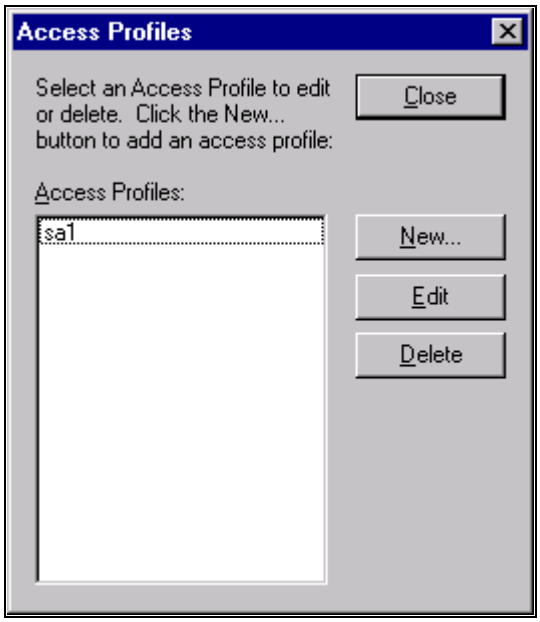
Access profiles enable you to minimize the number of users who need to know system administrator passwords. In fact, only one person needs to know these passwords. That person can create the required access profiles—by providing the necessary passwords when prompted—and all other security administrators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



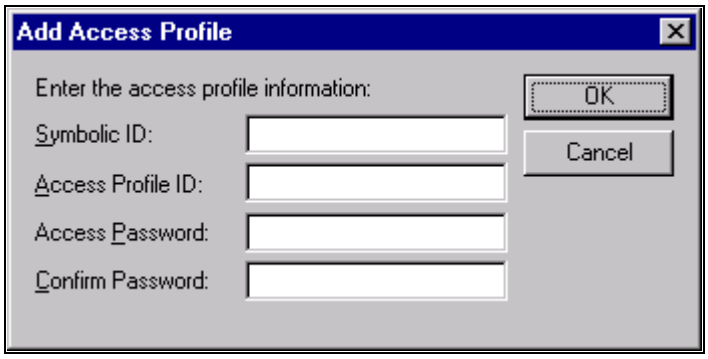
Access Profiles dialog box

Close	Click to exit this dialog box.
New	Click to create a new access profile definition.
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).



Add Access Profile dialog box

Symbolic ID Enter the Symbolic ID that is used to retrieve the encrypted ACCESSID and ACCESSPSWD from PSACCESSPRFL. For your initial installation, set it equal to the database name.

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 ACCESSID and ACCESSPSWD from PSACCESSPRFL.

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:

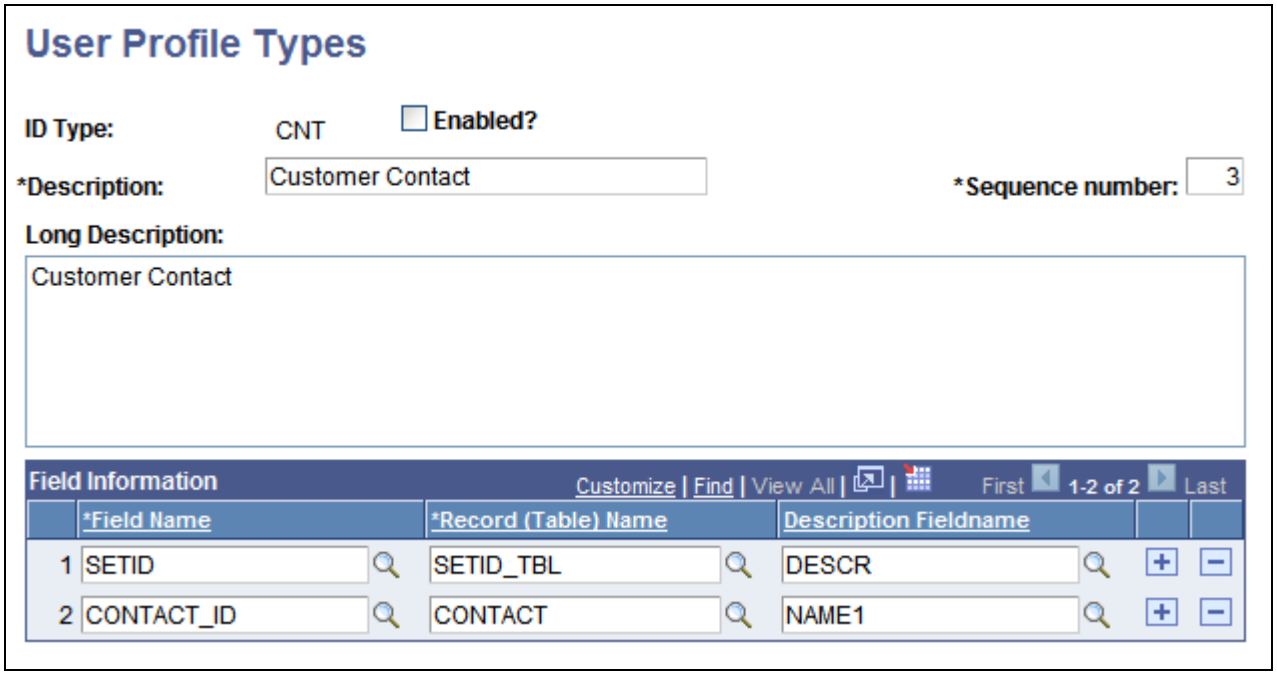
ID Type	*Description*
BID	Bidder
CNT	Customer Contact
CST	Customer
EJA	External Job Applicant
EMP	Employee
NON	None
ORG	Organization ID
PER	Person (CRM)
VND	Vendor
PTN	Partner

Page Used to Set Up User Profile Types

Page Name	Definition Name	Navigation	Usage
User Profile Types	PSOPRALIASTYPE	PeopleTools, Security, Security Objects, User Profile Types	Define user profile types.

Defining User Profile Types

Access the User Profile Types page (PeopleTools, Security, Security Objects, User Profile Types).



User Profile Types page

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.

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 recordfield 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.

Creating a New User Profile

To create a new user profile:

1. Select PeopleTools, Security, User Profiles, User Profiles to access the Find Existing Values 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 Find an Existing Value 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.

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. Make sure that you have selected the *correct* user profile.
3. Click Delete User Profile to remove information related to this particular user profile that appears 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.

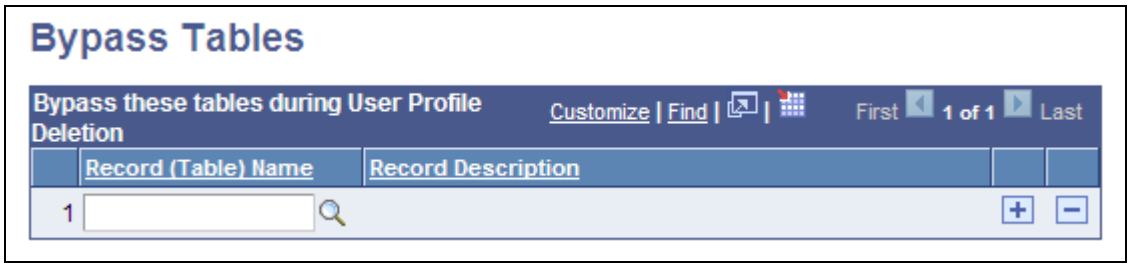
   To prevent user information in a specific table from being deleted, you can designate tables that the delete user process bypasses.

See Also

Chapter 1, "Getting Started with Security Administration," Component Interfaces, page 5

Bypassing Tables During the Delete User Profile Process

Access the Bypass Tables page (PeopleTools, Security, Security Objects, Tables to Skip).



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.

See Also

Chapter 5, "Administering User Profiles," Preserving Historical User Profile Data, page 129

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.

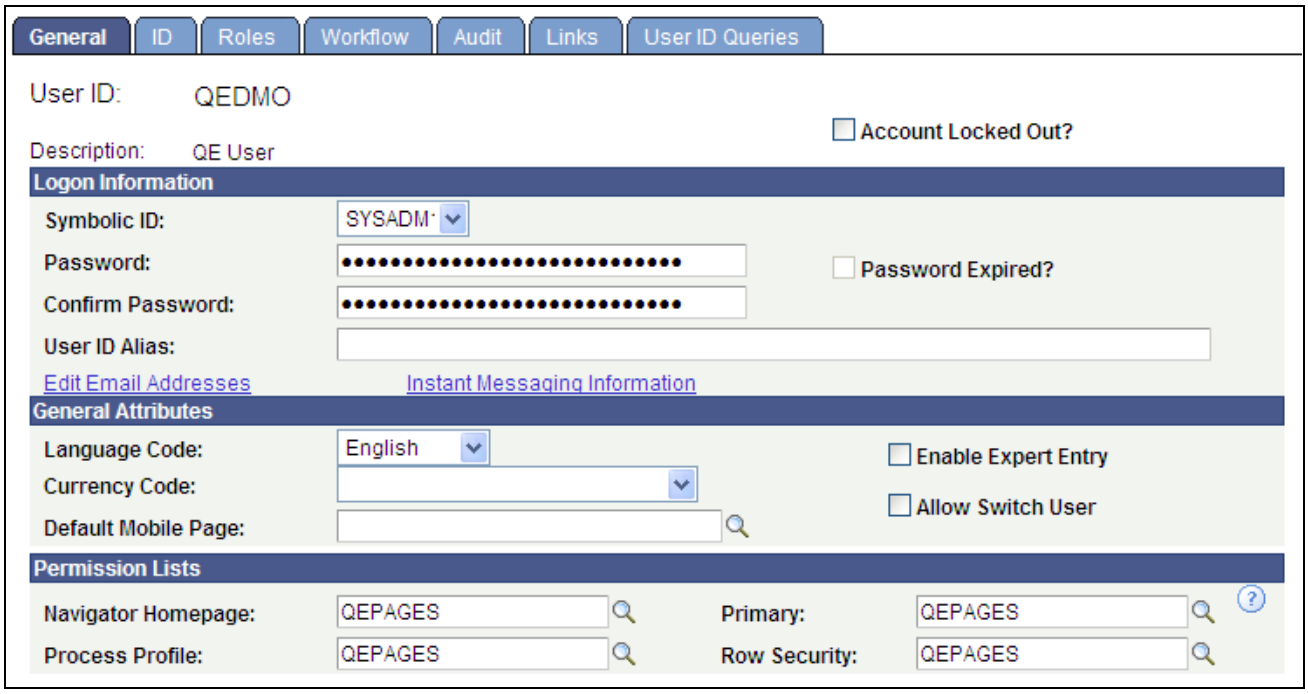
- View when a user profile was last updated.
- Display additional links.
- Run user ID queries.

Pages Used to Specify User Profile Attributes

Page Name	*Definition Name*	*Navigation*	*Usage*
General	USER_GENERAL	PeopleTools, Security, User Profiles, User Profiles, General	Set general user profile attributes.
ID	PSOPRALIAS	PeopleTools, Security, User Profiles, User Profiles, ID	Set ID type and attribute value.
Roles	USER_ROLES	PeopleTools, Security, User Profiles, User Profiles, Roles	Add roles to a user profile. This task defines user access in the PeopleSoft system. Through roles, the user inherits permission lists.
Workflow	USER_WORKFLOW	PeopleTools, Security, User Profiles, User Profiles, Workflow	Specify workflow settings for a user.
Audit	USER_AUDIT	PeopleTools, Security, User Profiles, User Profiles, Audit	Determine when and who last updated a profile.
Links	USER_OTHER	PeopleTools, Security, User Profiles, User Profiles, Links	Display any additional links added.
User ID Queries	USER_QUERY	PeopleTools, Security, User Profiles, User Profiles, User ID Queries	Run queries about a user profile.

Setting General User Profile Attributes

Access the General page (select PeopleTools, Security, User Profiles, User Profiles and click the General tab).



User Profiles - General page

Logon Information

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.

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). 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.

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 User Password field. The maximum password length is 32 characters.

Note. These values are required to sign in to the system, but you can save the profile without populating these fields.

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.

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.

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.

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.

General Attributes

Language Code

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.

Currency Code

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 deprecated 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 *PeopleTools 8.52: System and Server Administration*, "Using PeopleTools Utilities," General Options.

Permission Lists

Navigator Homepage

Enter a value associated with PeopleSoft Workflow.

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.

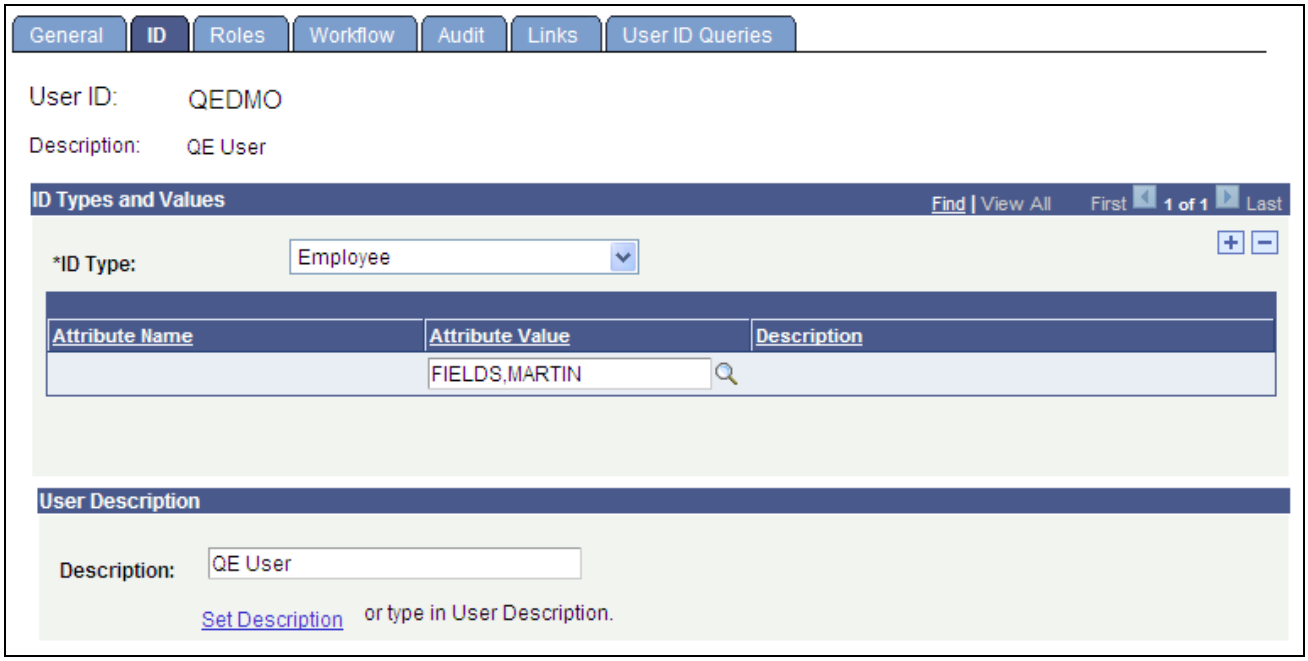
Note. Only the process profile comes from this permission list, not the list process groups.

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 ID page (select PeopleTools, Security, User Profiles, User Profiles and click the ID tab).



User Profiles - ID page

ID Types and Values

ID Typeand **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. In this case, the value reflects the employee number, but it could be a customer number or vendor number.

User Description

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

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.

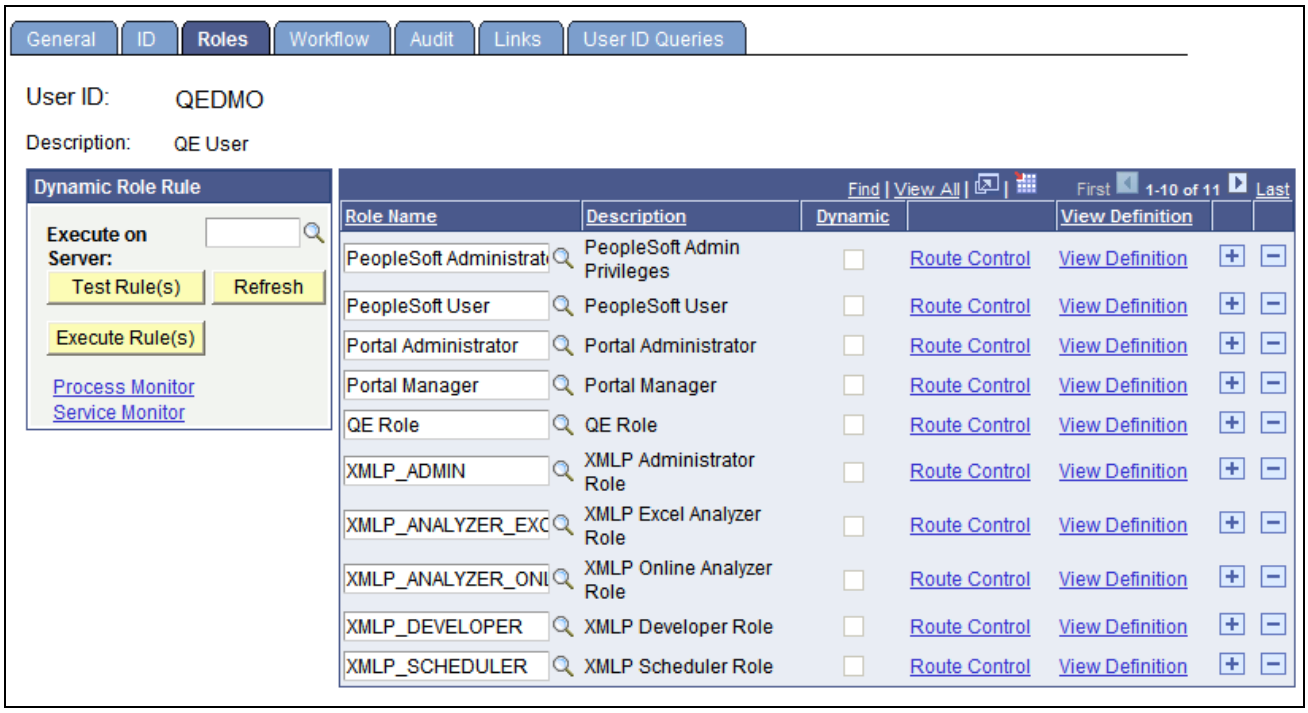
Note. Before you assign a user type to a user, you must create user types.

See Also

Chapter 5, "Administering User Profiles," Setting Up User Profile Types, page 95

Setting Roles

Access the Roles page (select PeopleTools, Security, User Profiles, User Profiles and click the Roles tab).



User Profiles - Roles page

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.

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 *PeopleTools 8.52: Workflow Technology*, "Using Additional Routing Options," Understanding Route Control Development.

See Chapter 4, "Setting Up Roles," Using the PeopleSoft Administrator Role, page 89.

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.

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.

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 Chapter 4, "Setting Up Roles," Executing Dynamic Role Rules, page 87.

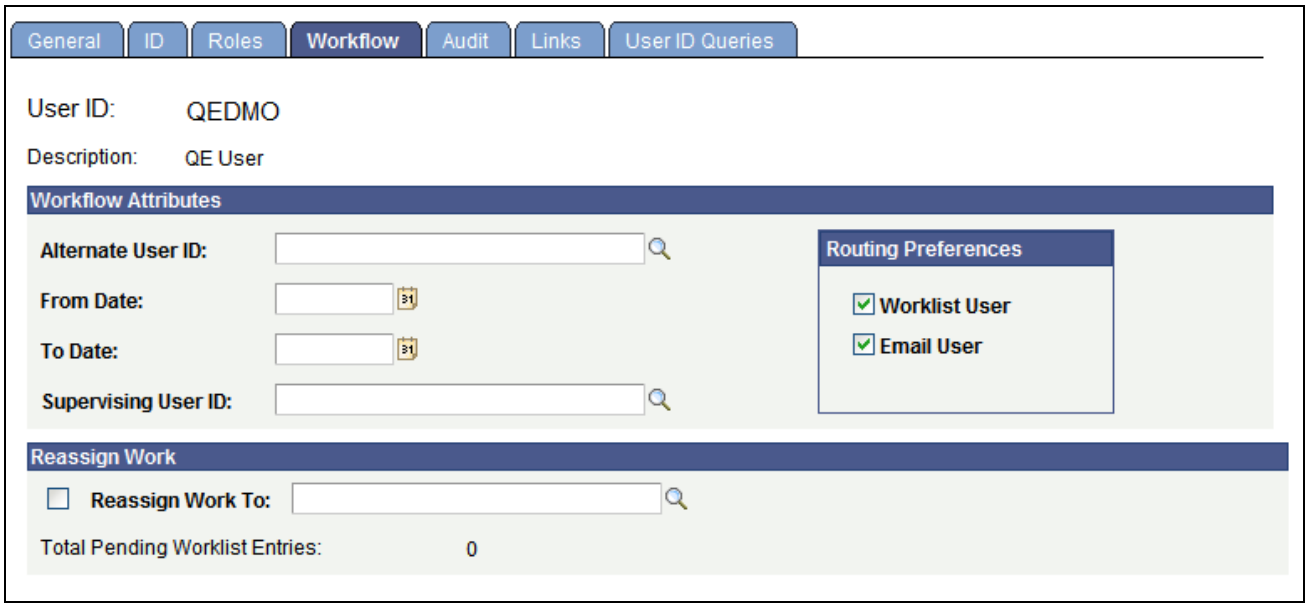
Process Monitor Click to view the status of the DYNROLE_PUBL application engine program in the Process Scheduler Monitor.

Service Monitor Click the link to check the status of the publication of the ROLESYNCHEXT_MSG message in the Service Operations Monitor.

See Chapter 4, "Setting Up Roles," Executing Dynamic Role Rules, page 87.

Specifying Workflow Settings

Access the Workflow page (select PeopleTools, Security, User Profiles, User Profiles and click the Workflow tab).



User Profiles - Workflow page

Workflow Attributes

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 is only meant to work 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 use 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.

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

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

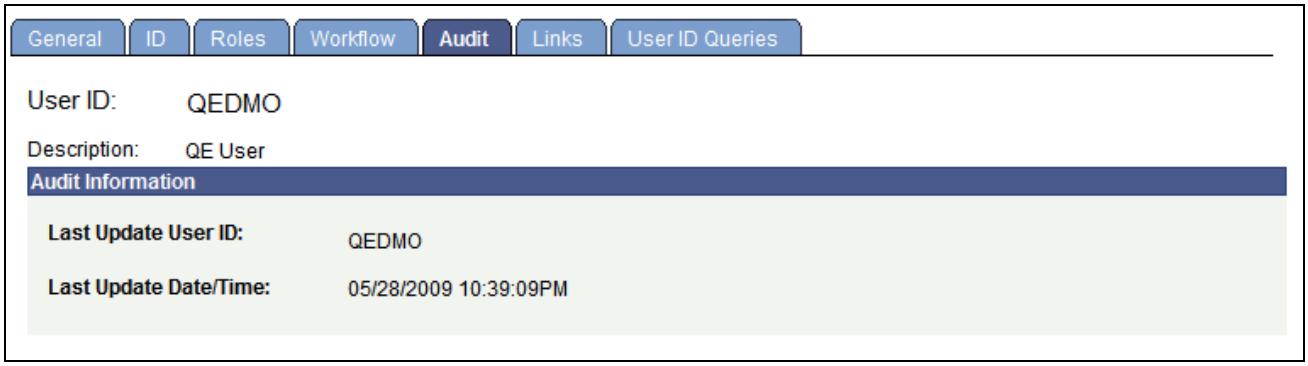
Displays worklist items that require a user's attention.

See Also

PeopleTools 8.52: Workflow Technology, "Administering PeopleSoft Workflow"

Viewing When a User Profile Was Last Updated

Access the Audit page (select PeopleTools, Security, User Profiles, User Profiles and click the Audit tab).



User Profiles - Audit page

The Audit page is a display-only page that enables you to determine:

- When a profile was last updated.
- Who updated the profile.

Displaying Additional Links

Access the Links page (select PeopleTools, Security, User Profiles, User Profiles and click the Links tab).

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 ID Queries page (select PeopleTools, Security, User Profiles, User Profiles and click the User ID Queries tab).

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.

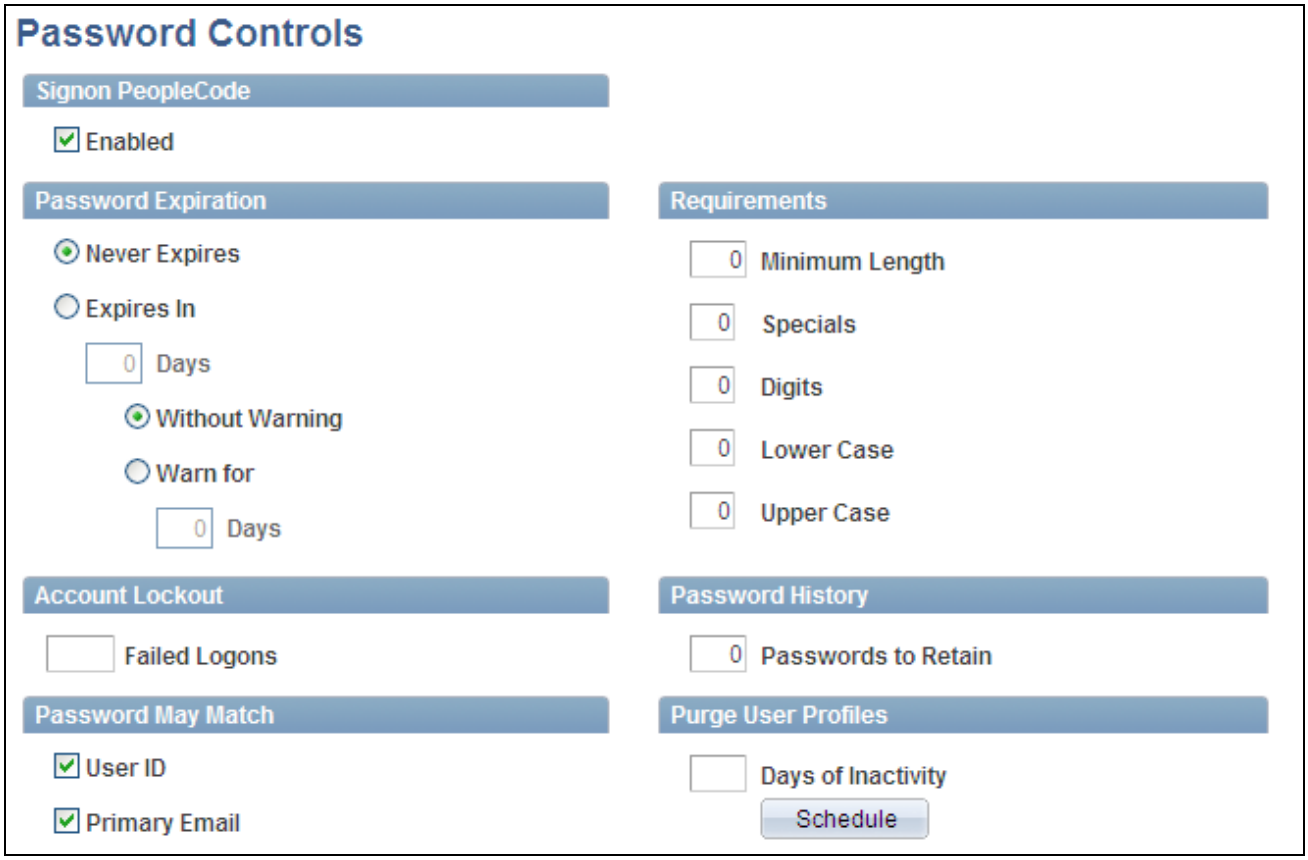
Working With Passwords

This section discusses how to:

- Set password controls.
- Change passwords.
- Create email text for forgotten passwords.
- Create hints for forgotten passwords.
- Delete hints for forgotten passwords.
- Set up the site for forgotten passwords.
- Request new passwords.

Setting Password Controls

Access the Password Controls page (PeopleTools, Security, Password Configuration, Password Controls).



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 applications, not within a directory server.

Signon PeopleCode

Enabled Select to enable the PeopleSoft password expiration and account lockout fields. The other password controls are not enabled by this box.

If you do not want these password controls, for example, you already have a third-party utility that performs equivalent features, then do not select this check box.

Note. If you change the status of the Enabled check box, you must restart the application server.

You can extend or customize the controls by modifying the PeopleCode.

Password Expiration

Never Expires Select to disable password expiration options for all users.

Expires in Select to enable password expiration options for all users.

You must enter a value between *1* (the default value) and *365* in the Days field to specify the number of days that a password is valid. Users signing on after a password expires must change their password to sign in.

You must select a warning option.

Without Warning 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.

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 the Change Password component (CHANGE_PASSWORD) only. 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.

Account Lockout

Failed Logons Enter the maximum number of failed sign in attempts to allow before the system disables the user profile. 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.

Password May Match

User ID Select to enable users to use their own user ID as a password.

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.

See Chapter 5, "Administering User Profiles," General Attributes, page 104.

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 32 characters.

Minimum Length Enter the value that determines the *fewest number of characters* that a user must enter when creating his password. 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.

The allowable special characters are:

```
! @ # $ % ^ & * ( ) - _ = + \ | [ ] { } ; : / ? . > <
```

Digits Enter the required number of integers, such as 1 or 2, that the password must include.

Lower Case Enter the required number of minuscule letters, such as 'q' or 'i,' that the password must include.

Upper Case Enter the required number of majuscule letters, such as 'Q' or 'I,' that the password must include.

Leading, intermediate, and trailing white spaces are not supported in PeopleSoft passwords. If you want to include 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 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.

Password History

Passwords to Retain Enter the number of user passwords to retain in the password history table (PSPSWDHISTORY). 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

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. 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.

Changing Passwords

Access the Change My Password page (from the homepage, click Change My Password). The PeopleSoft system enables users to change their passwords as needed.

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.

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.

Access the Forgot My Password Email Text page (PeopleTools, Security, Password Configuration, Forgot My Password Email Text).

Forgot My Password Email Text page

Add the following text string in the Email Text field:

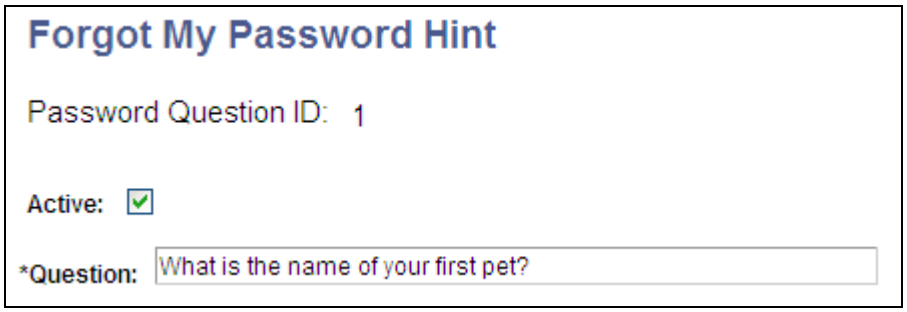
```
<<%PASSWORD>>
```

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.

Creating Hints for Forgotten Passwords

Access the Forgot My Password Hint page (PeopleTools, Security, Password Configuration, Forgotten Password Hint).



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. Enter your question to verify that the user is who he or she claims to be.
6. Click Save.

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.

Setting Up the Site for Forgotten Passwords

PeopleSoft recommends setting 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 site should have this format:
http://webserver/psp/sitename/portalname/localnodename/c/MAINTAIN_SECURITY
.EMAIL_PSWD.GBL?

Requesting New Passwords

To request a new password, access the hidden Forgot My Password page (EMAIL_PSWD2).. The system randomly generates a new password and emails it to the user.

Before the system can email the user a new password, complete these tasks:

- Create a forgotten password hint.
- Specify an email address in the user profile.
- Grant permission to have a new password emailed.

Note. The security administrator must select the Allow Password to be Emailed check box in at least one of the user's permission lists. 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 Chapter 3, "Setting Up Permission Lists," Setting General Permissions, page 31.

To request a new password:

1. Click the Forgotten Password link on the PeopleSoft signon page (or direct the user to the Forgotten Password link.)

2. On the Forgot My Password page, enter your user ID.

3. Click Continue.

4. On the Email New Password page, verify that the system is set to send the new password to the appropriate email address.

   If the appropriate email address does not appear, contact your system administrator. System administrators must make sure that the email address is correctly represented for each user who intends to use this feature.

Note. Use Application Designer to change any display properties of the fields on the EMAIL_PSWD2 page.

---

5. Respond to the user validation question.

   Note. The user must have set up the forgotten password help.

   See *PeopleTools 8.52: PeopleSoft Applications User's Guide*, "Setting User Preferences," Changing Your Password.

6. Click Email New Password.

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 Chapter 5, "Administering User Profiles," Defining User Profile Access for Remote Security Administrators, page 123.

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 Chapter 5, "Administering User Profiles," Defining Remote Security Administrator Role Grant Capability, page 123.

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.

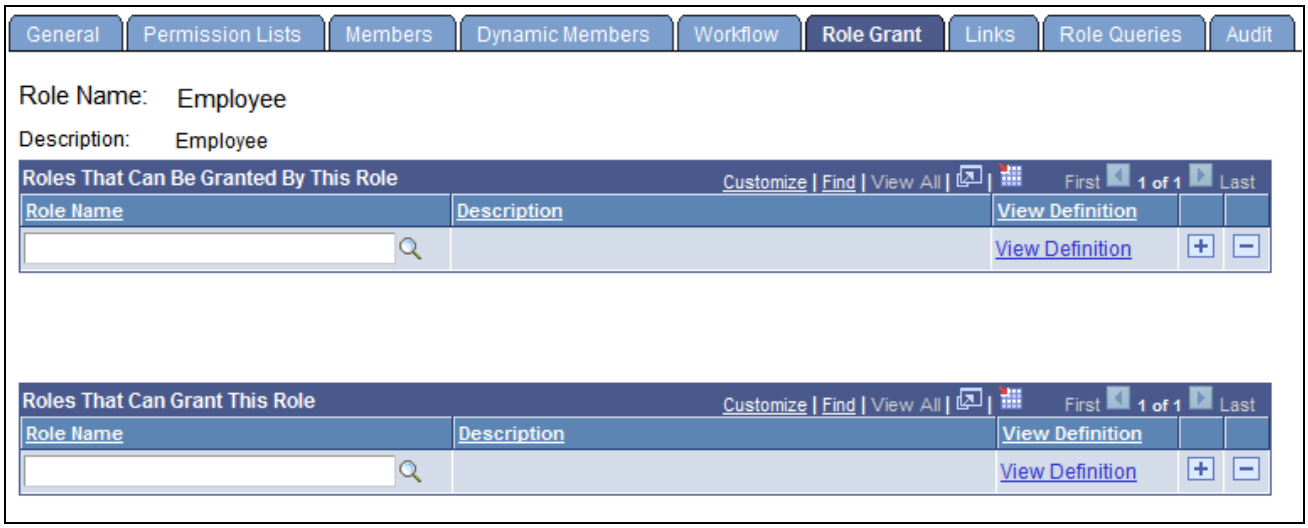
   When remote security administrators access the Distributed User Profiles component, this search record enforces row-level security to restrict the set of user IDs that they can select and administer.

See Also

PeopleTools 8.52: PeopleSoft Application Designer Developer's Guide, "Creating Component Definitions," Using Search Records

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.



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.

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.

See Also

Chapter 5, "Administering User Profiles," Specifying User Profile Attributes, page 100

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 (6.X to 7.X or 7.X to 8.X), 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.

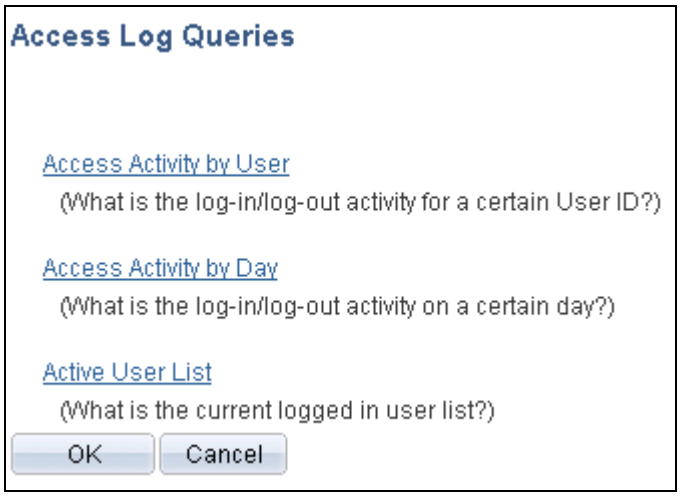
2. Using Data Mover, sign in to the target database and run USERIMPORT.DMS for user definitions.

   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 2.

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

Access the Access Log Queries page (select PeopleTools, Security, Common Queries and click the Access Log Queries link on the Review Security Information page).



Access Log Queries page

PeopleSoft Security provides three audit logs that track user sign in and sign out activity in PeopleSoft applications..

Select one of the following logs:

- Access Activity by User

  View a single user's sign in and sign out activity. This log includes the users' client IP addresses, sign in times, and sign out times.

- Access Activity by Day

  View one or more days of all user sign in and sign out activity. This log includes the users' IDs, client IP addresses, sign in times, and sign out times.

- Active User List

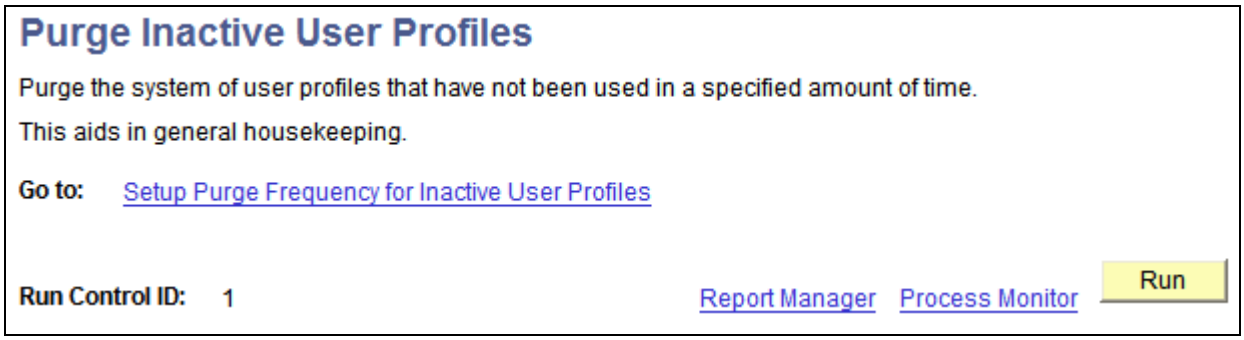
  View the users who are currently signed in to the application in the browser. This log includes users' IDs, client IP addresses, sign in times, and duration in minutes.

These logs are generated using data from the PSACCESSLOG table. If you are not interested 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 using the same exact column names and order as were in the previous PSACCESSLOG table: OPRID, LOGIPADDRESS, LOGINDTTM, LOGOUTDTTM. Use Application Designer to open the PSACCESSLOG record definition and create the table.

Purging Inactive User Profiles

Access the Purge Inactive User Profiles page (PeopleTools, Security, User Profiles, Purge Inactive User Profiles).



Purge Inactive User Profiles page

Note. Before accessing this page, you must enter a run control ID.

See *PeopleTools 8.52: PeopleSoft Process Scheduler*, "Submitting and Scheduling Process Requests," Understanding Run Control IDs.

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.

See Also

Chapter 5, "Administering User Profiles," Working With Passwords, page 114

Chapter 5, "Administering User Profiles," Bypassing Tables During the Delete User Profile Process, page 100

PeopleTools 8.52: Data Management, "Using PeopleSoft Data Archive Manager"

Chapter 1, "Getting Started with Security Administration," Component Interfaces, page 5

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 *PeopleTools 8.52: Data Management*, "Using PeopleSoft Data Archive Manager," History Tables.

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 Chapter 5, "Administering User Profiles," Bypassing Tables During the Delete User Profile Process, page 100.

Chapter 6

Working with User Profiles Across Multiple PeopleSoft Databases

This chapter provides an overview of user profile synchronization and discusses how to:

- Implement default user profile synchronization.
- Implement configurable user profile synchronization.
- Transfer users between 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.

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:

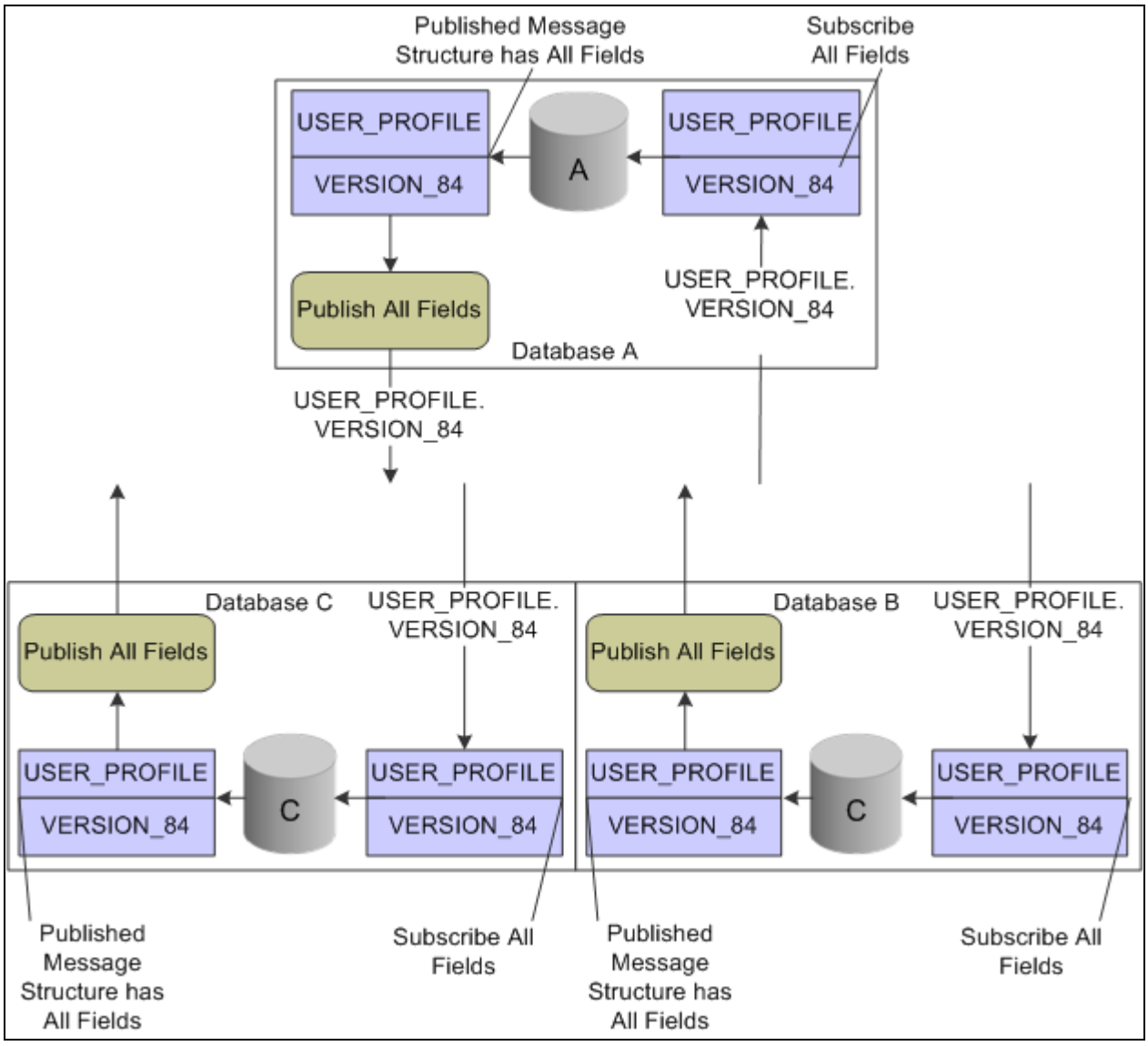


Diagram of the flow of user profile information as it uses standard synchronization among three PeopleSoft databases

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 *PeopleTools 8.52: System and Server Administration*, "Using the PSADMIN Utility," Using the Quick-Configure Menu.

2. Define the local gateway URL for the integration broker.

   See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Managing Integration Gateways," Defining Integration Gateways.

3. In each participating database, activate the domain in integration broker.

   See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Managing Pub/Sub Server Domains," Activating Pub/Sub Server Domains.

4. In each participating database, create and configure the remote nodes.

   See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Adding and Configuring Nodes."

5. In each participating database, configure single signon by setting up each subscribing database as a trusted node.

   See Chapter 9, "Implementing Single Signon," Defining Nodes for Single Signon, page 202.

6. In each participating database, define the gateway properties; include all PeopleSoft nodes.

   See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Managing Integration Gateways," Setting Oracle Jolt Connection String Properties.

7. In each participating database, activate the USER_PROFILE service operation.

   Note. The default setting is *Enabled.*

   See *PeopleTools 8.52: PeopleSoft Integration Broker*, "Managing Service Operations," Configuring Service Operation Definitions.

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 *PeopleTools 8.52: PeopleSoft Integration Broker*, "Managing Service Operation Routing Definitions."

   - 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 *PeopleTools 8.52: PeopleSoft Integration Broker*, "Managing Service Operation Routing Definitions."

9. For each subscribing database, grant permission list security for the USER_PROFILE service operations.

   See Chapter 3, "Setting Up Permission Lists," Setting Web Services Permissions, page 57.

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:

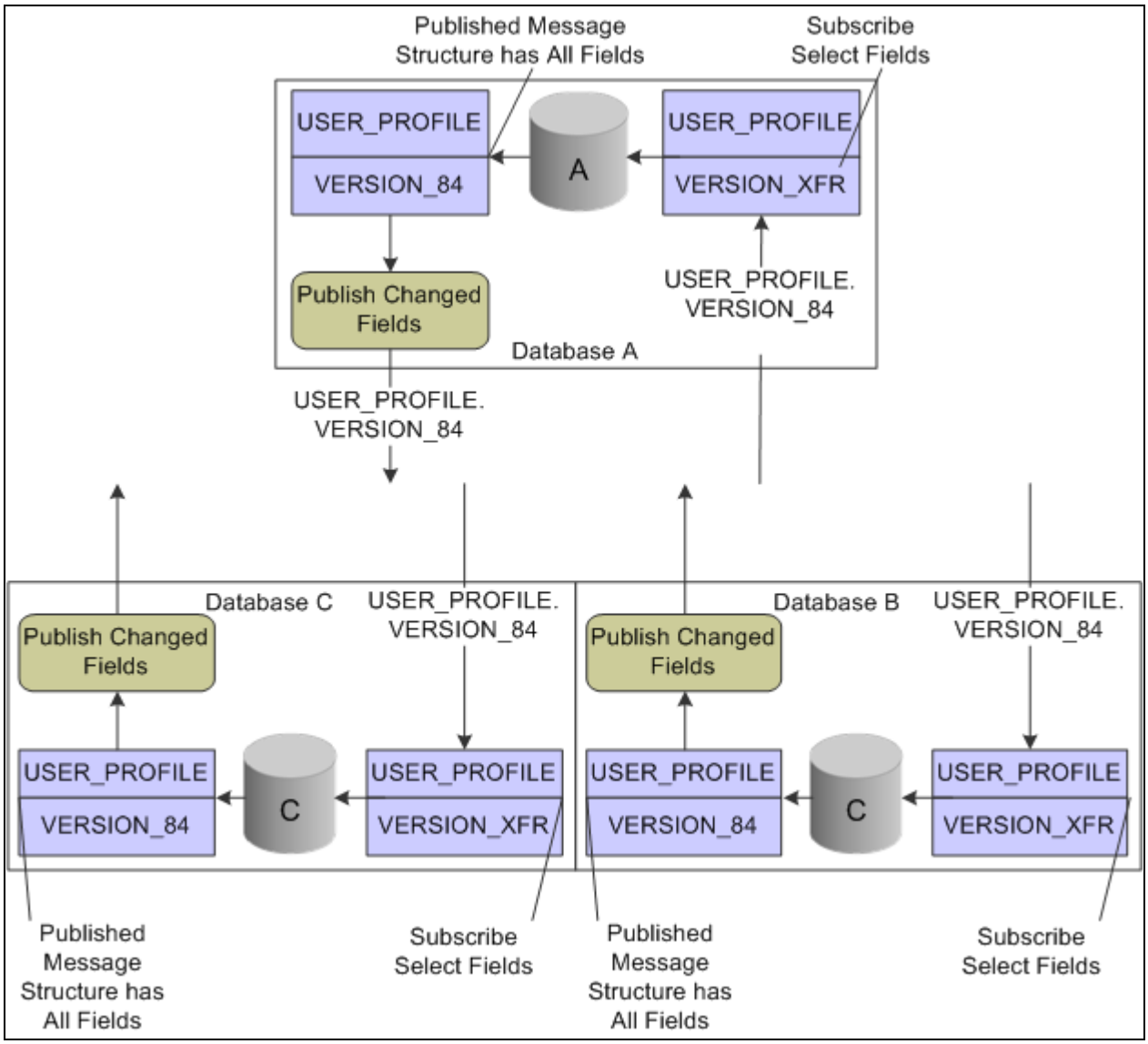
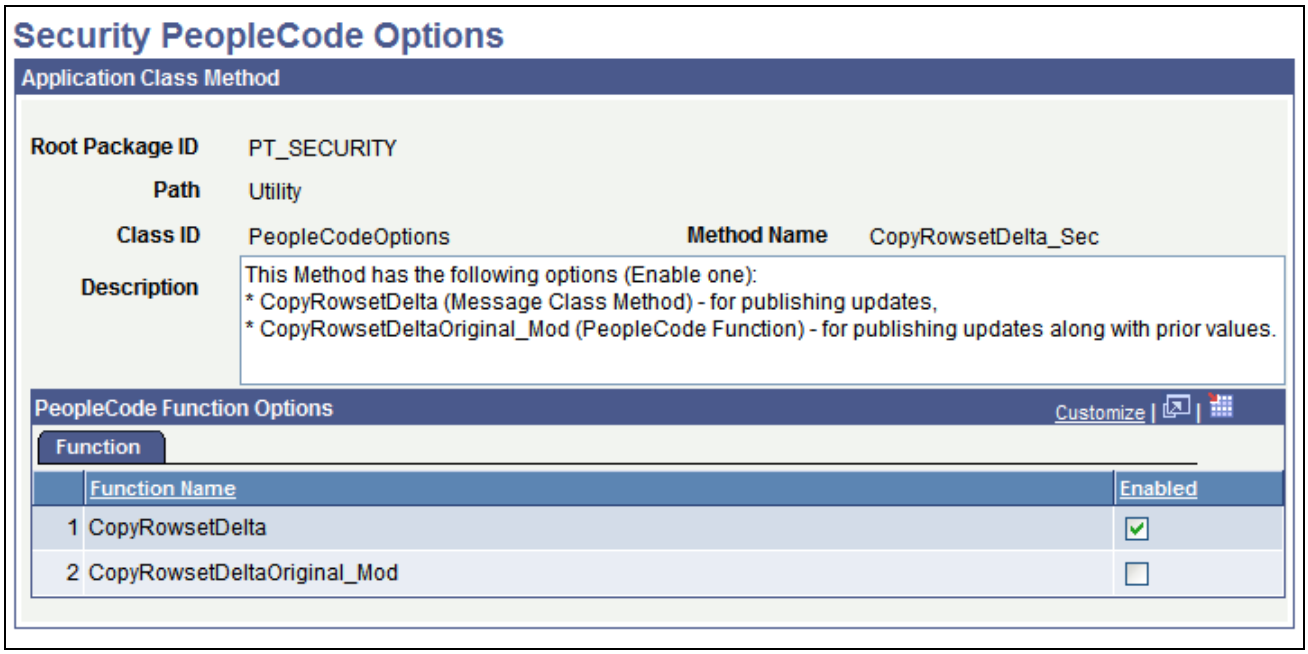


Diagram of the flow of user profile information as it uses configurable synchronization among three PeopleSoft databases

Enabling Security PeopleCode Options

Access the Security PeopleCode Options page (PeopleTools, Security, Security Objects, Security PeopleCode Options).



Security PeopleCode Options page

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.

See *PeopleTools 8.52: PeopleCode API Reference*, "Message Classes."

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 *PeopleTools 8.52: System and Server Administration*, "Using the PSADMIN Utility," Using the Quick-Configure Menu.

2. In each publishing database, access the Security PeopleCode Options page and enable the CopyRowsetDeltaOriginal_Mod PeopleCode function.

   See Chapter 6, "Working with User Profiles Across Multiple PeopleSoft Databases," Enabling Security PeopleCode Options, page 136.

3. Define the local gateway URL for Integration Broker.

   See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Managing Integration Gateways," Defining Integration Gateways.

4. In each participating database, activate the domain in Integration Broker.

   See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Managing Pub/Sub Server Domains," Activating Pub/Sub Server Domains.

5. In each participating database, create and configure the remote nodes.

   See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Adding and Configuring Nodes."

6. In each participating database, configure single signon by setting up each subscribing database as a trusted node.

   See Chapter 9, "Implementing Single Signon," Defining Nodes for Single Signon, page 202.

7. In each participating database, define the gateway properties; include all PeopleSoft nodes.

   See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Managing Integration Gateways," Setting Oracle Jolt Connection String Properties.

8. In each participating database, deactivate the *inbound* generated USER_PROFILE.VERSION_84 routing definition.

   See *PeopleTools 8.52: PeopleSoft Integration Broker*, "Managing Service Operation Routing Definitions," Activating and Inactivating Routing Definitions in the Routing Component.

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 *PeopleTools 8.52: PeopleSoft Integration Broker*, "Managing Service Operation Routing Definitions," Activating and Inactivating Routing Definitions in the Routing Component.

10. In each participating database, activate the USER_PROFILE_XFR service operation.

    See *PeopleTools 8.52: PeopleSoft Integration Broker*, "Managing Service Operations," Configuring Service Operation Definitions.

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.

      See *PeopleTools 8.52: PeopleSoft Integration Broker*, "Managing Service Operation Routing Definitions."

    - In each publishing database, change the external alias on the Parameters page to *USER_PROFILE.VERSION_84*.

      See *PeopleTools 8.52: PeopleSoft Integration Broker*, "Managing Service Operation Routing Definitions."

12. In each subscribing database, grant permission list security for the USER_PROFILE_XFR service operations.

    See Chapter 3, "Setting Up Permission Lists," Setting Web Services Permissions, page 57.

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.

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 (6.X to 7.X or 7.X to 8.X), 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 2.

3. After copying user and role definitions, run the PeopleTools audits, including DDDAUDIT and SYSAUDIT, to check the consistency of your database.

Chapter 7

Employing LDAP Directory Services

This chapter provides an overview of the PeopleSoft Lightweight Directory Access Protocol (LDAP) solution and discusses how to:

- Configure the LDAP directory.
- Cache the directory schema.
- Create authentication maps.
- Create user profile maps.
- Create role membership rules.
- Delete directory configurations.
- Enable Signon PeopleCode for LDAP authentication.
- Use LDAP over SSL (LDAPS).
- View SSL/TLS for LDAP transaction setup examples.

Note. PeopleTools uses JNDI libraries only. JNDI requires no added installation as it is part of the standard PeopleTools installation.

This chapter assumes you have a working knowledge of LDAP-enabled directory servers.

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 chapter 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. 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 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.

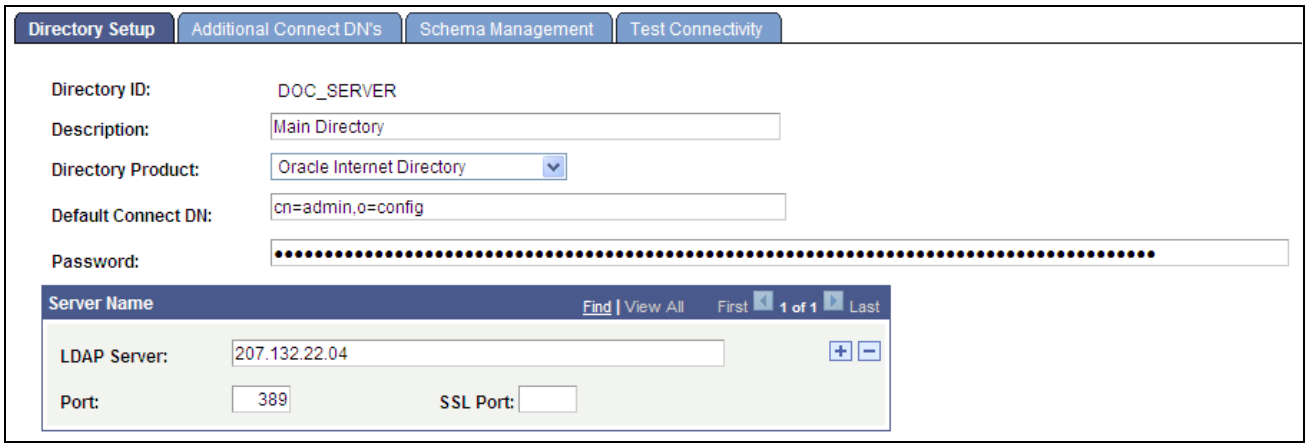
The PeopleSoft application server uses the User DN and password to connect to the LDAP server to retrieve user profile information about the specific user signing in to the system. The User DN must reflect a user with the appropriate LDAP browse rights.

Pages Used to Configure the Directory

Page Name	*Definition Name*	*Navigation*	*Usage*
Directory Setup	DSDIRSETUP	PeopleTools, Security, Directory, Configure Directory, Directory Setup	Specify the network information of your LDAP directory servers, such as user IDs and passwords.
Additional Connect DN's	DSSERVERID	PeopleTools, Directory, Configure Directory, Additional Connect DN's	Specify connect DNs in addition to the default connect DN specified on the Directory Setup page.
Schema Management	DSEXTINSTALL	PeopleTools, Security, Directory, Configure Directory, Schema Management	Install selected PeopleSoft-specific schema extensions into your directory.
Test Connectivity	DSSRCHRSLT	PeopleTools, Security, Directory, Configure Directory, Test Connectivity	Test the distinguished names and search criteria that you entered on the previous pages of the Configure Directory component and view the results. The system tests the connectivity when you access this page.

Specifying Network Information for LDAP

Access the Directory Setup page (PeopleTools, Security, Directory, Configure Directory. Click the Directory Setup tab).



Configure Directory - Directory Setup page

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) — 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.

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.

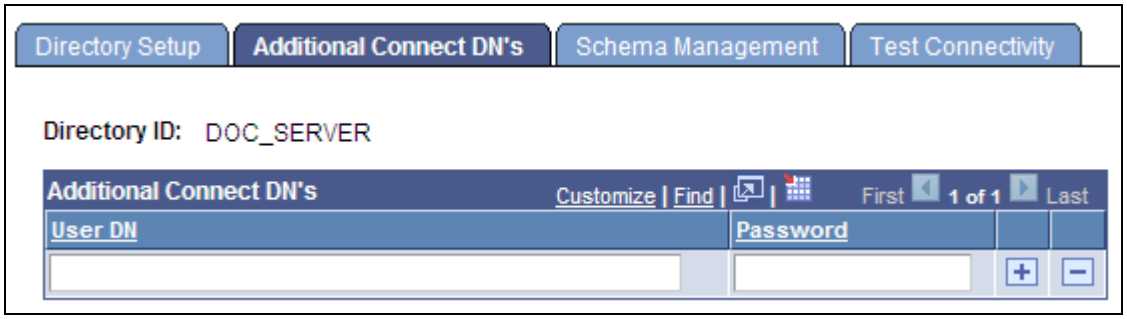
LDAP Server — Identify a specific LDAP server. You can use the DNS name or you can use IP address dotted notation. For example, either of the following formats is acceptable: ldap12.yourcompany.com or 192.201.185.90.

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, Directory, Configure Directory and click the Additional Connect DN's tab).



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.

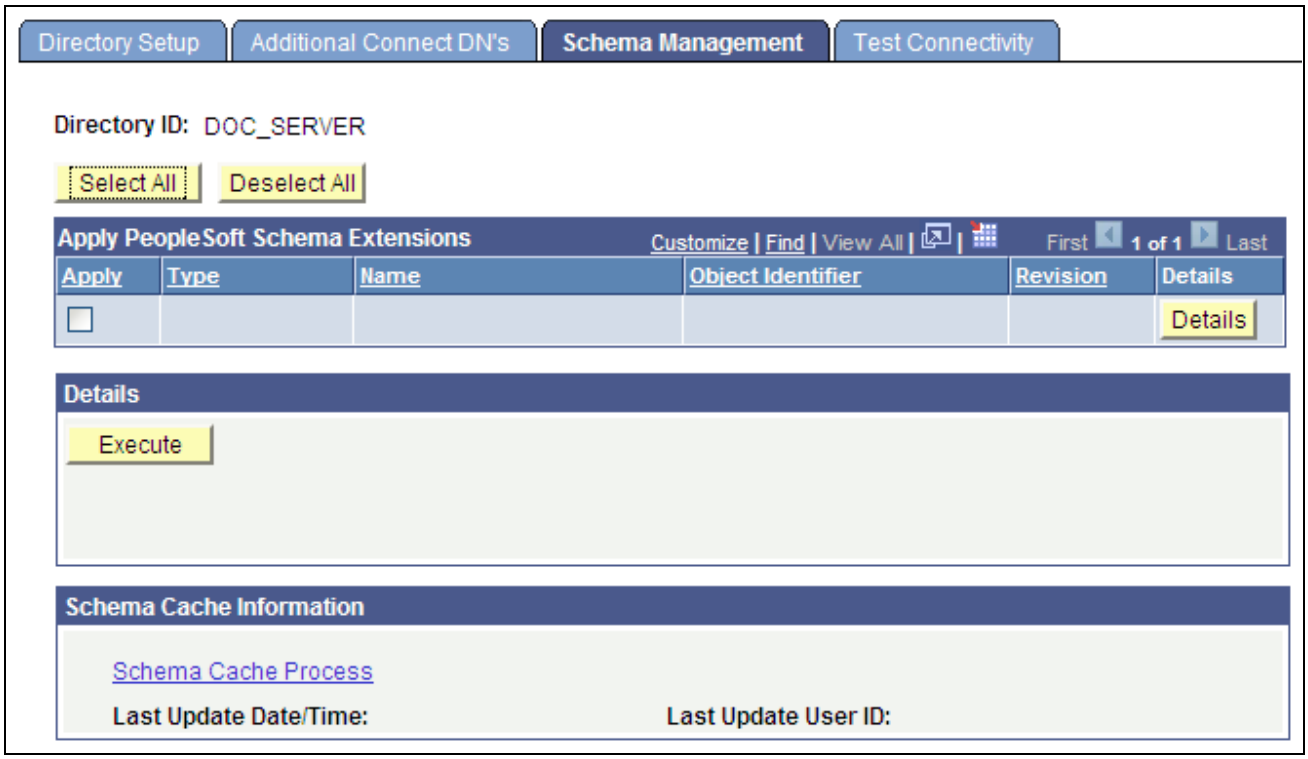
Note. You will not see any available schema extensions unless you have installed the PeopleSoft Directory Interface.

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 Directory and click the Schema Management tab).



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.

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.

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

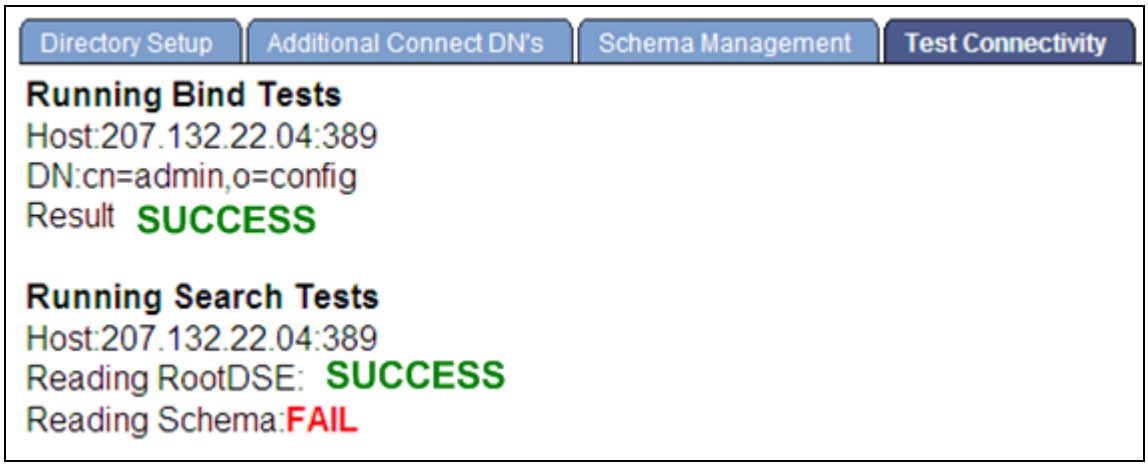
When you click a schema extension Details button, the system displays the details of that extension. In addition to the object identifier and name, you may also be interested in the Superiors detail, which indicates which extensions, if any, are above this one in the hierarchy. Also of interest is the 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 Directory and click the Test Connectivity tab).



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.

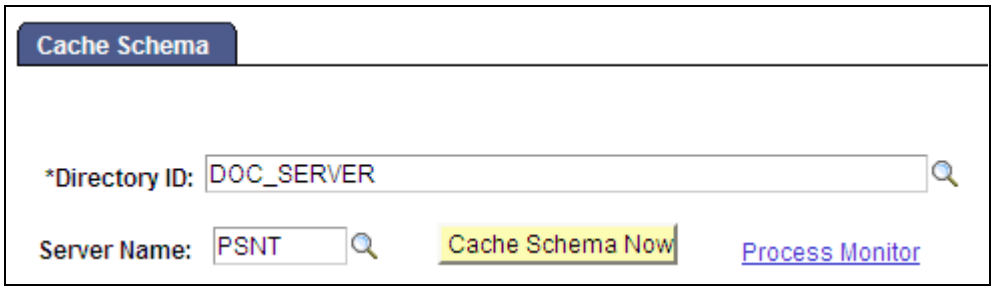
This section discusses how to create a cache of the directory schema.

Page Used to Cache the Directory Schema

Page Name	*Definition Name*	*Navigation*	*Usage*
Cache Schema	DSSCHEMACACHE	PeopleTools, Security, Directory, Cache Directory Schema	Specify a directory server and invoke an Application Engine program designed to create a cache in the PeopleSoft database of the directory schema. The cache of the LDAP schema is used to simplify creating maps for authentication and user profile maintenance.

Creating a Cache of the Directory Schema

Access the Cache Schema page (PeopleTools, Security, Directory, Cache Directory Schema).



Cache Schema page

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:

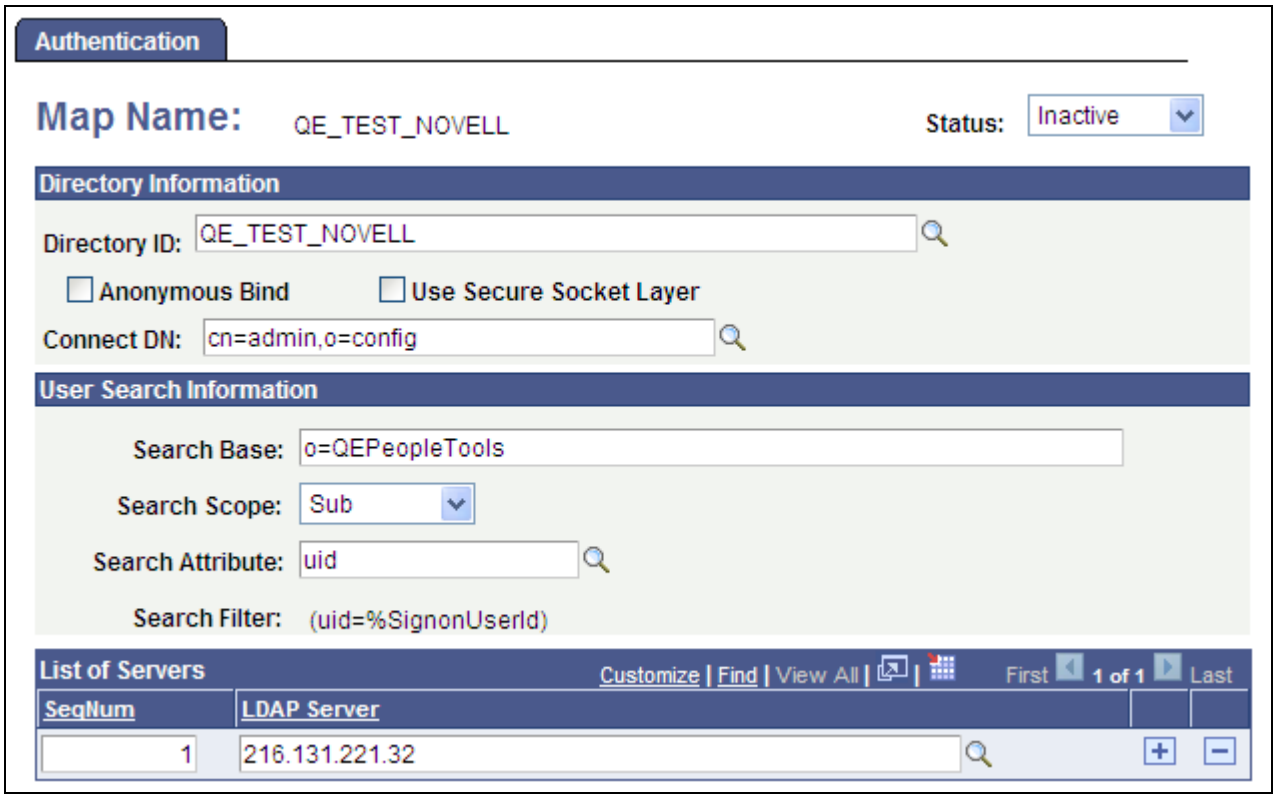
- Defining an authentication map.
- Use the Search Attribute field in authentication maps.

Page Used to Create Authentication Maps

Page Name	*Definition Name*	*Navigation*	*Usage*
Authentication	DSSECMAPMAIN	PeopleTools, Security, Directory, Authentication Map	Create a mapping to the directory that the PeopleSoft system relies on for authenticating users.

Defining an Authentication Map

Access the Authentication page (PeopleTools, Security, Directory, Authentication Map).



Authentication page

Status Activate the authentication map by selecting *Active*. To disable an authentication map, select *Inactive*.

Directory Information

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

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 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 Chapter 7, "Employing LDAP Directory Services," Using the Search Attribute Field in Authentication Maps, page 154.

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

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 *sramdass* in the LDAP directory.

```
dn: uid=sramdass, dc=peoplesoft, dc=com
cn: sramdass
uid: sramdass123
description: peoplesoft user
mail: sramdass@oracle.com
telephone: 12345678
objectclass: person
password: PASSWORD
```

If the user is to log in with *sramdass/PASSWORD,* then the Search Attribute prompt value should be *cn.* If the user wants to log in with *sramdass@oracle.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.

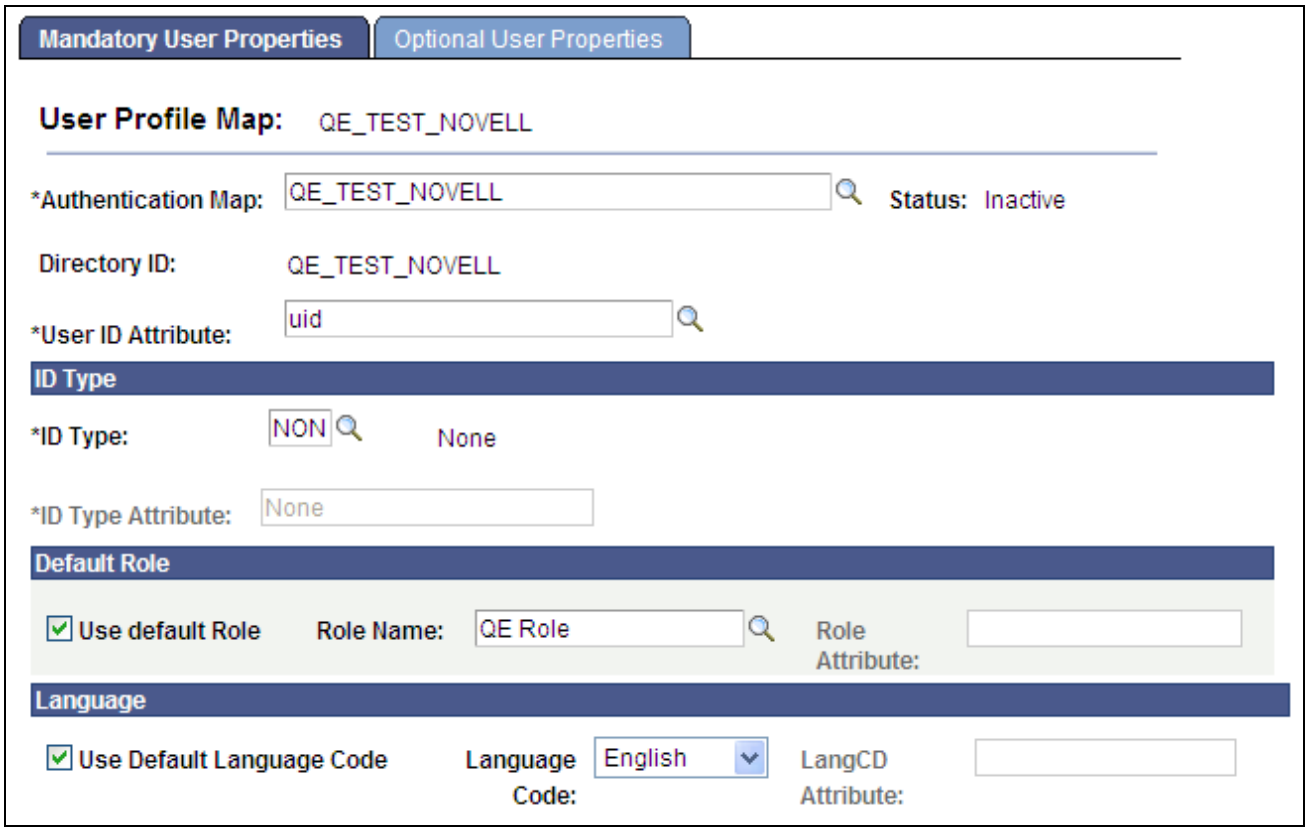
Note. You must supply user properties to Signon PeopleCode only if you intend to authenticate users with your LDAP directory.

Pages Used to Create User Profile Maps

Page Name	*Definition Name*	*Navigation*	*Usage*
Mandatory User Properties	DSUSRPRFLMANMAP	PeopleTools, Security, Directory, User Profile Map, Mandatory User Properties	Specify the attributes required for signon. You can select to have the system retrieve these mandatory values from the directory server, or you can enter default values.
Optional User Properties	DSUSRPRFLOPTMAP	PeopleTools, Security, Directory, User Profile Map, Optional User Properties	Specify optional user properties to retrieve from the directory.

Specifying Mandatory User Properties

Access the Mandatory User Properties page (select PeopleTools, Security, Directory, User Profile Map and click the Mandatory User Properties tab).



User Profile Map - Mandatory User Properties page

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.

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

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

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.

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

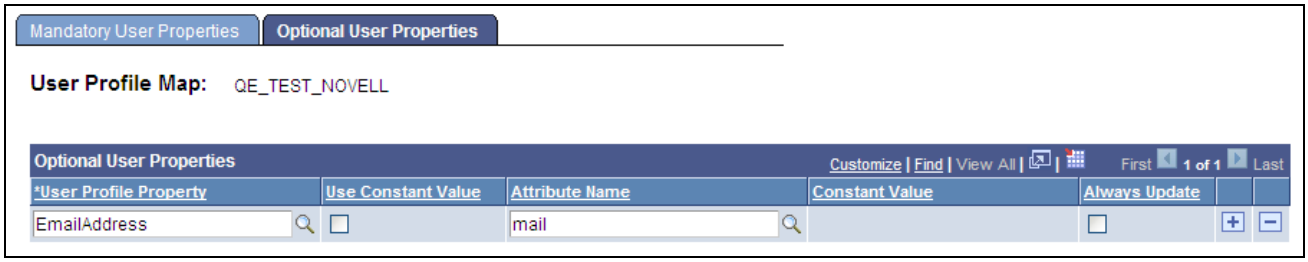
Use Default Language Code — Select if you do not maintain language codes in the directory.

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 (PeopleTools, Security, Directory, User Profile Map, Optional User Properties).



User Profile Map - Optional User Properties page

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:

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).
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 systems creates corresponds to the User Profile Map - User ID Attribute field, which contains an LDAP attribute name.

Consider an entry corresponding to the user *sramdass* in LDAP:

```
dn: uid=sramdass, dc=peoplesoft, dc=com
cn: sramdass
uid: sramdass123
description: peoplesoft user
mail: sramdass@oracle.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 *sramdass/PASSWORD,* while the system creates the user profile with the name *sramdass@oracle.com.*

Example 2

Authentication Map Search Attribute: *uid*

User Profile Map User ID Attribute:*telephone*

You must log in as *sramdass123/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.

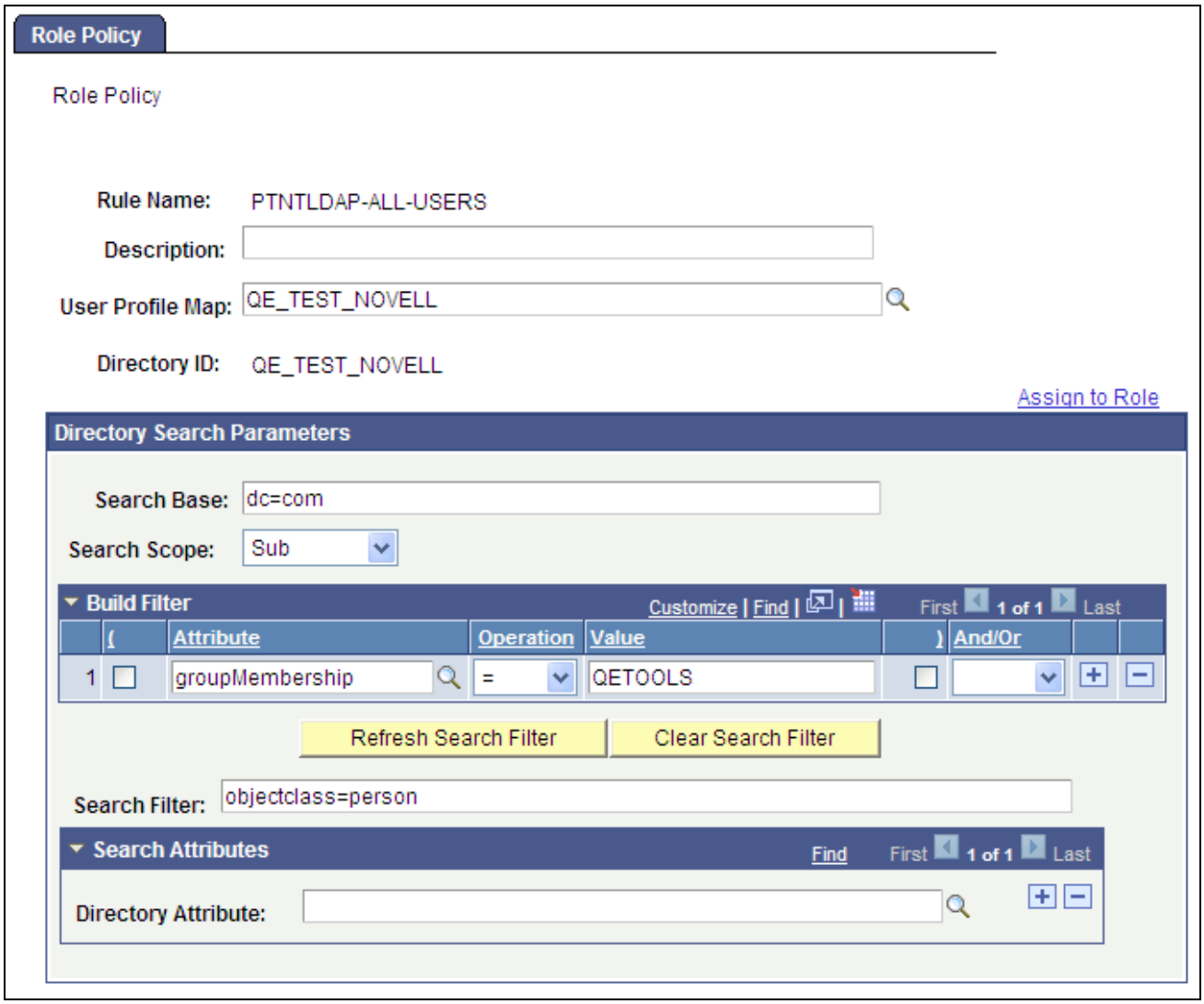
Important! You must keep the data within PeopleSoft consistent with any changes made to the structure or content of the external directory server, especially when you are dealing with security data. The Role Membership Rules page enables you to modify a PeopleSoft role based on directory criteria.

Page Used to Create Role Membership Rules

Page Name	*Definition Name*	*Navigation*	*Usage*
Role Policy	DSSECROLERULE	PeopleTools, Security, Directory, Role Membership Rules	Define the rules that are read by Dynamic Role Rule PeopleCode and populate PeopleSoft roles with members.

Defining Role Membership Rules

Access the Role Policy page (PeopleTools, Security, Directory, Role Membership Rules).



Role Policy page

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

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

( ) 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.

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

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:

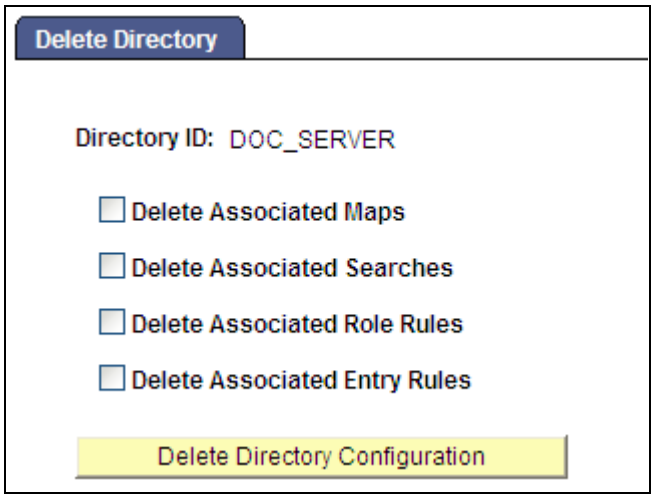
- Delete the directory configuration.
- Work with the workflow address book.

Page Used to Delete Directory Configurations

Page Name	*Definition Name*	*Navigation*	*Usage*
Delete Directory	DSPURGEDIRID	PeopleTools, Security, Directory, Delete Directory Configuration	Delete the entire directory configuration or just parts of it.

Deleting the Directory Configuration

Access the Delete Directory page (PeopleTools, Security, Directory, Delete Directory Configuration).

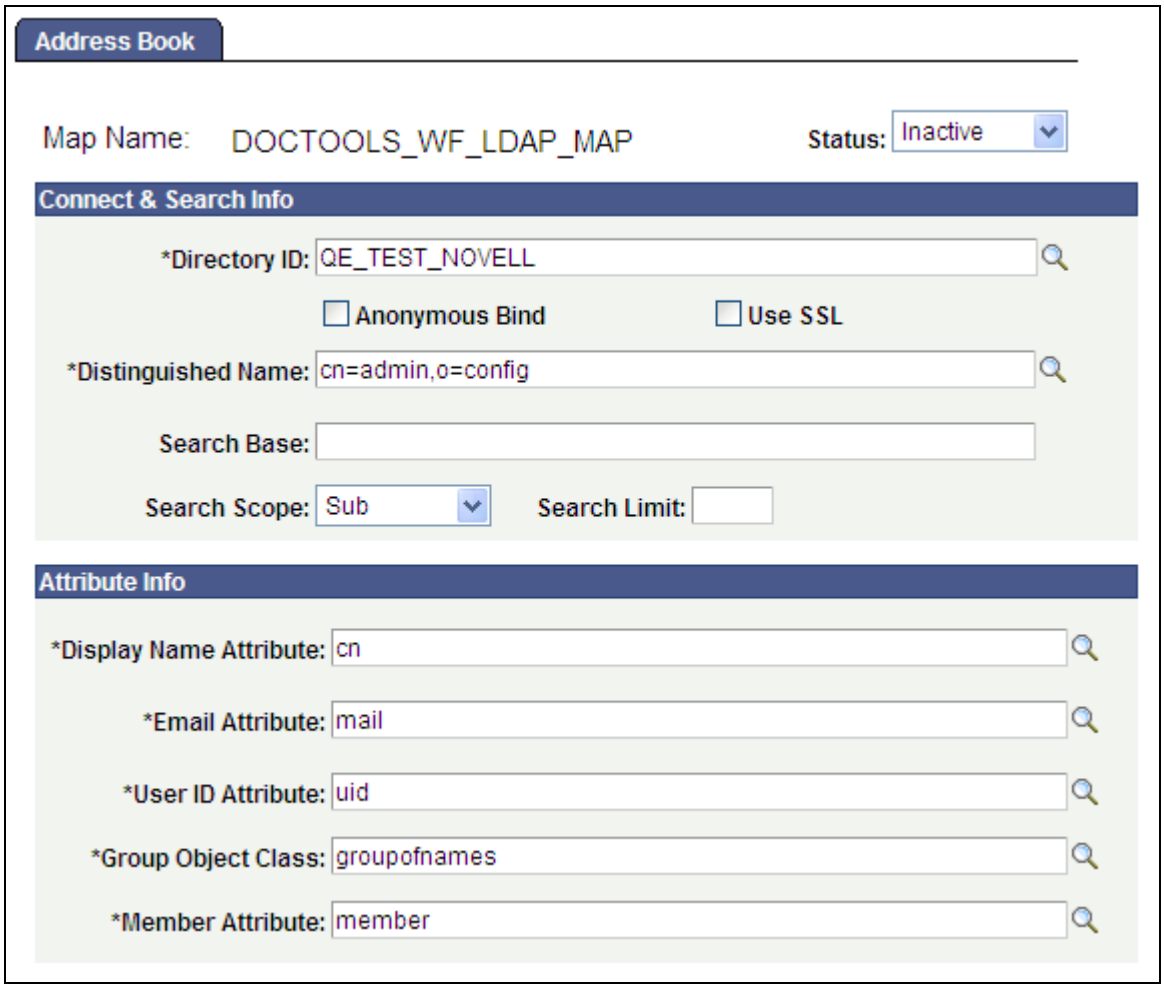


Delete Directory page

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 Address Book).



Directory - 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.

Map Name Displays the name of the workflow address book map.

Status Select *Active* or *Inactive.*

Connect & Search Info

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 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

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.
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.

See Also

PeopleTools 8.52: Workflow Technology, "Adding Events and Routings"

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.

   Note. The user ID entered, whether it is Invoke As user signing in or a default user, must be able to access the User Profiles component in a permission list.

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, 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, 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 Directory, 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, Authentication Map to access the Authentication Map page, and select the Use Secure Sockets Layer check box.

6. Enable the LDAP_AUTHENTICATION Signon PeopleCode.

   See Chapter 8, "Employing Signon PeopleCode and User Exits," Enabling Signon PeopleCode, page 184.

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=<instance⇒
    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=<instance⇒
    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:

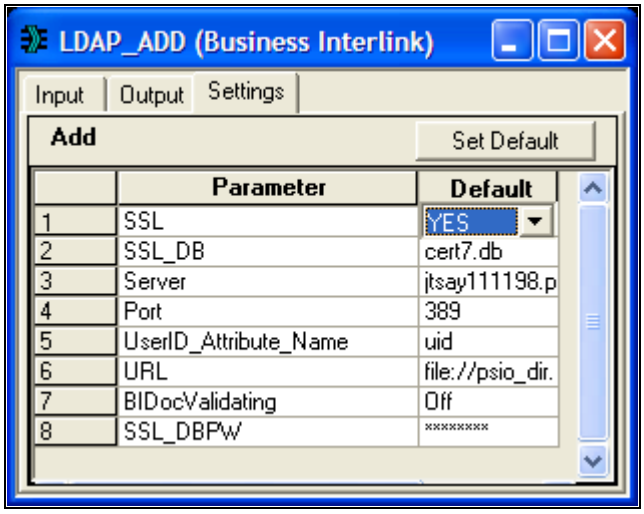
- Manually in Application Designer.
- Programmatically through PeopleCode.

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:



Example LDAP Business Interlink SSL parameter

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 helper
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 parameters.
*/
&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 8

Employing Signon PeopleCode and User Exits

This chapter provides an overview of the delivered external authentication solutions and discusses how to:

- Use Signon PeopleCode.
- Use the web server security exit.
- Use the Windows security exit.

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.
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 Chapter 8, "Employing Signon PeopleCode and User Exits," LDAP_ProfileSynch Considerations, page 181.

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 Directory.
- 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.

See Also

PeopleTools 8.52: PeopleCode Language Reference, "PeopleCode Built-in Functions," RevalidatePassword

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 In 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 is logging 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.

Page Used to Develop Signon PeopleCode

Page Name	*Definition Name*	*Navigation*	*Usage*
Signon PeopleCode	SIGNONPPC_PAGE	PeopleTools, Security, Security Objects, Signon PeopleCode	Enable Signon PeopleCode programs.

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 PeopleBooks:

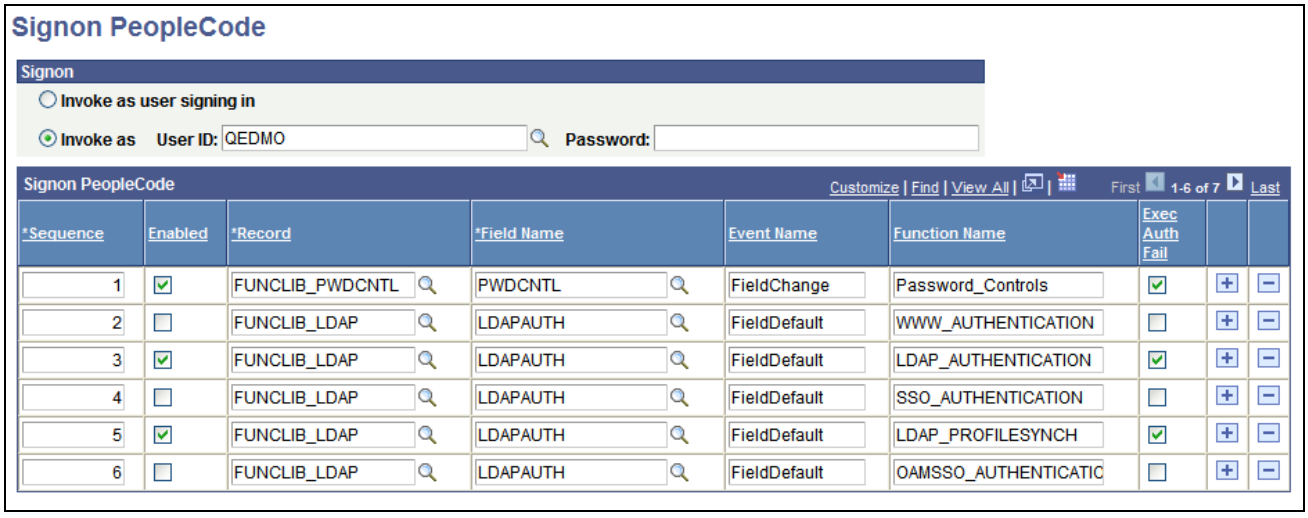
PeopleCode Function	*Description*
See *PeopleTools 8.52: PeopleCode Language Reference*, "System Variables," %PSAuthResult.	Returns the result (boolean) of PeopleSoft authentication.

PeopleCode Function	*Description*
See *PeopleTools 8.52: PeopleCode Language Reference*, "PeopleCode Built-in Functions," SetAuthenticationResult.	Verifies customers who log on to the system even if the PeopleSoft authentication fails.
See *PeopleTools 8.52: PeopleCode Language Reference*, "System Variables," %SignonUserId.	User ID value entered by the user on the Signon page. This applies to Pure Internet Architecture and Windows signon.
See *PeopleTools 8.52: PeopleCode Language Reference*, "System Variables," %SignOnUserPswd.	User password value the user entered at the Signon page. This value is encrypted. This applies to Pure Internet Architecture and Windows signon.
See *PeopleTools 8.52: PeopleCode Language Reference*, "System Variables," %Request.	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).



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.

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=PeopleTools, 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 the a user profile with these parameters:

- User ID: PUBUSER
- Password: ekdJl3838**&^^%kdjflsdkjfJHJIK

See Chapter 5, "Administering User Profiles," Working With User Profiles, page 98.

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 *ekdJl3838**&^^%kdjflsdkjfJHJIK*.

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*.

See Also

PeopleTools 8.52: PeopleTools Portal Technologies, "Configuring the Portal Environment," Configuring Web Profiles

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 server */
   /*  retrieve the USERID from the HTTP request⇒
 and pass it to SetAuthentication Result */

   &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.
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.

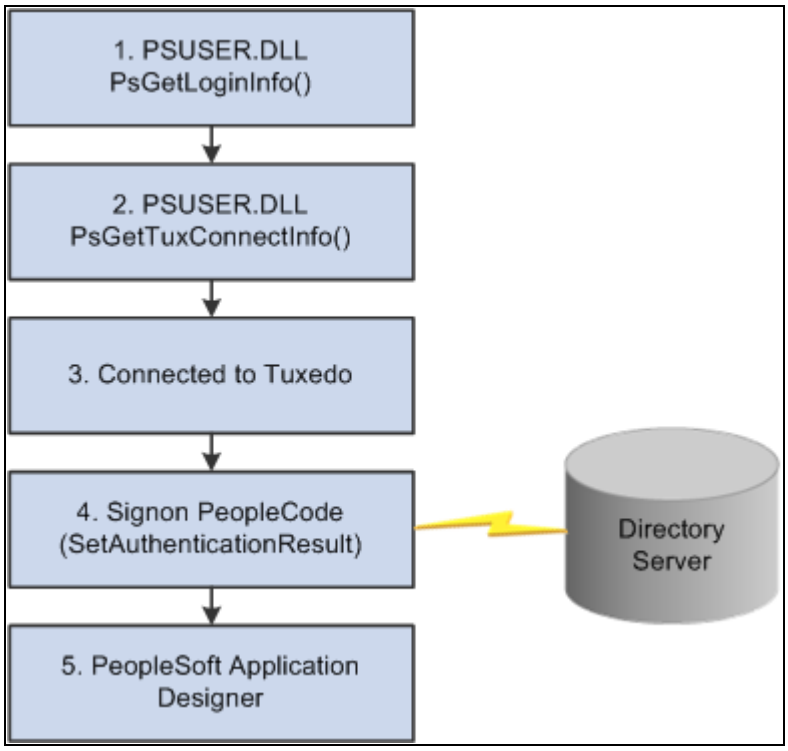
The Windows client-side exits are:

- 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 those authorizations:



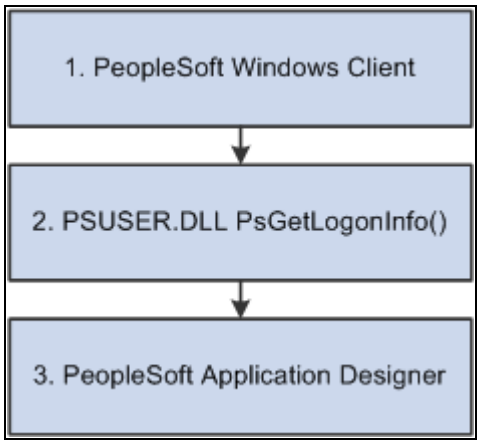
Microsoft Windows Client three-tier signon exits

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:



Two-tier Microsoft Windows Client signon using PsGetLogonInfo

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. DB2UNIX: DB2 UNIX. INFORMIX: Informix. MICROSFT: Microsoft SQL Server. ORACLE: Oracle Server. SYBASE: Sybase SQL Server. APPSERV: Application Server.
szDBName	Database name or application server name.
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 pshome\src\psuser.

   The resulting file, PSUSER.DLL, is used by PeopleTools (PSTOOLS.EXE), and the Windows COBOL interfaces. For Windows NT, 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 9

Implementing Single Signon

This chapter provides an overview of single signon and discusses how to:

- Implement PeopleSoft-only single signon.
- Implement Oracle Access Manager as the PeopleSoft single signon solution.
- Implement Kerberos as the desktop single signon solution.

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 documentation for creating the public access user profile in this PeopleBook. You find documentation for specifying the identity of the public access user in *PeopleTools 8.52: PeopleTools Portal Technologies PeopleBook.*

See Also

Chapter 5, "Administering User Profiles," Working With User Profiles, page 98

Chapter 8, "Employing Signon PeopleCode and User Exits," Creating a Public Access User, page 187

PeopleTools 8.52: PeopleTools Portal Technologies, "Configuring the Portal Environment," Public Users

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 Chapter 9, "Implementing Single Signon," Implementing PeopleSoft-Only Single Signon, page 200.
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 or IBM WebSphere. This option applies to all previous PeopleTools 8.x versions. For example, if you intend to incorporate applications running on Enterprise PeopleTools 8.46, you can implement this option. See Chapter 9, "Implementing Single Signon," Implementing Oracle Access Manager as the PeopleSoft Single Signon Solution, page 219.
PeopleSoft and Kerberos	Kerberos is a network protocol that uses secret key cryptography to provide authentication between clients and servers. It is the desktop solution for single signon. See Chapter 9, "Implementing Single Signon," Implementing Kerberos as the Desktop Single Signon Solution, page 221.

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.

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.
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.

Field	*Description*
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 Node 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:

- Working with the Single Signon page.
- Defining nodes 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 or IBM WebSphere. You can use passwords or digital certificates for single signon authentication.

Understanding PeopleSoft-Only Single Signon

PeopleSoft applications supports single signon among 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.

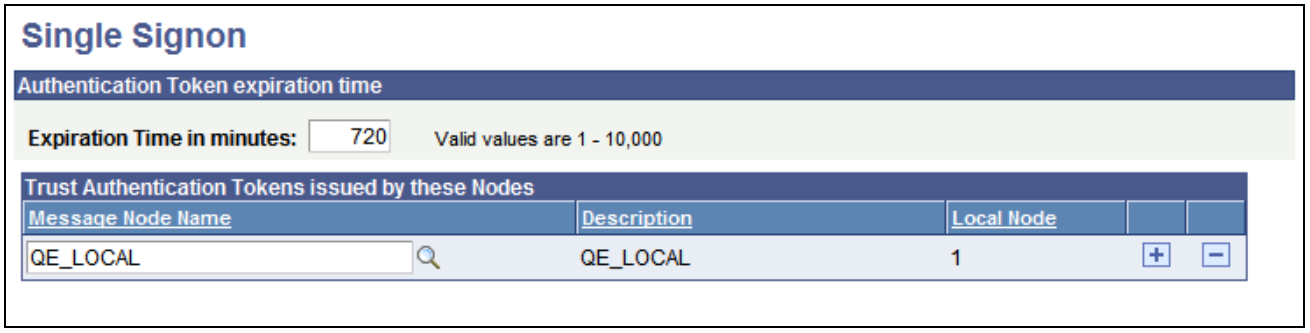
After the first application server/node authenticates a user, the system delivers a web browser cookie containing an authentication token (PS_TOKEN). PeopleSoft uses web browser cookies to store a unique access token for each user after they are authenticated initially. When the user connects to another PeopleSoft application server/node, the second application server uses the token in the browser cookie to re-authenticate users automatically so they don't have to sign in repeatedly.

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.

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.

Working with the Single Signon Page

Access the Single Signon page (PeopleTools, Security, Security Objects, Single Signon).



Single Signon Page

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, Node Definitions.

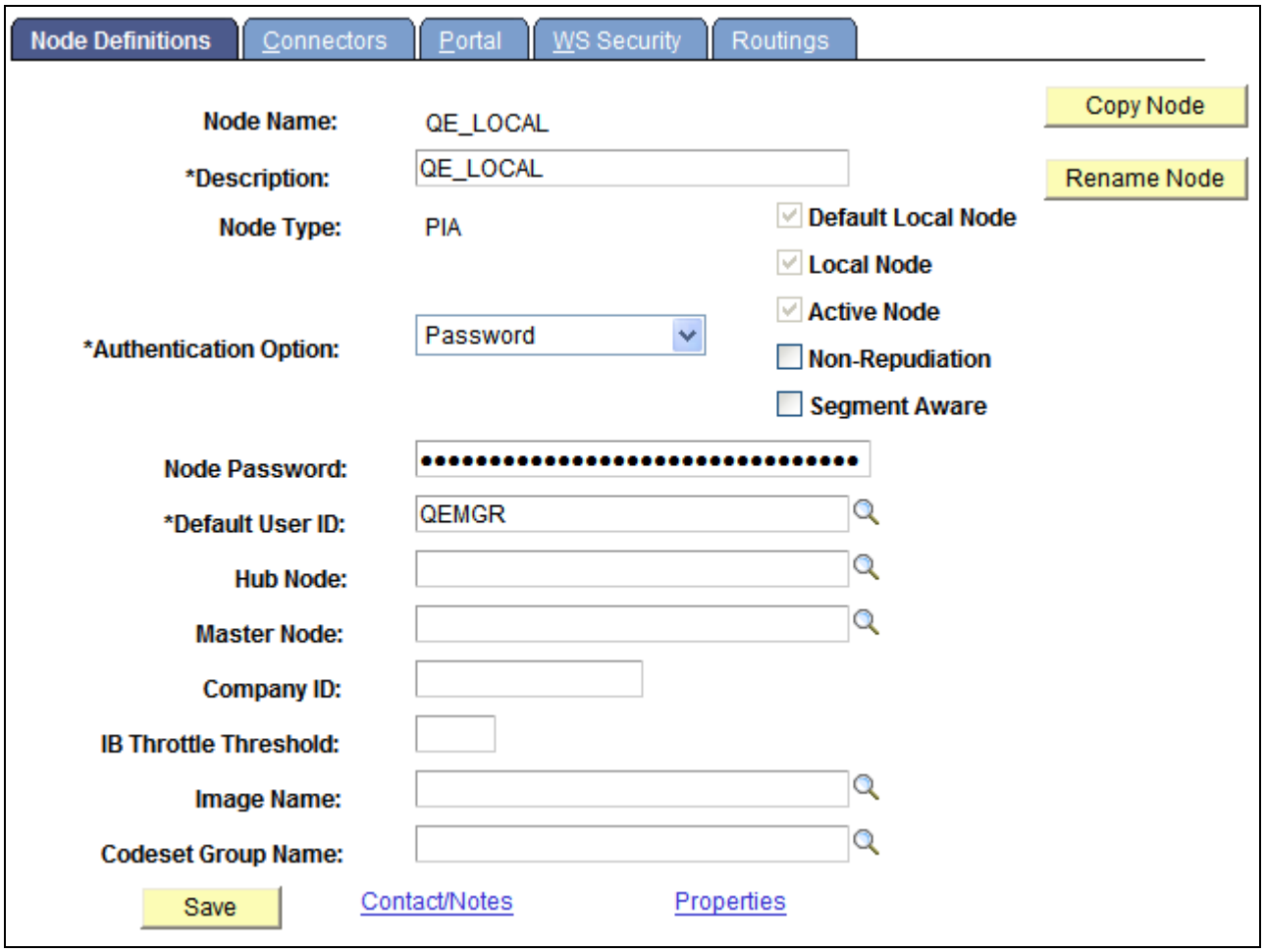
Local Node

Indicates whether the node is local or not.

Note. After you update the list of trusted nodes, the system automatically recognizes the new list. Rebooting the application server is not required.

Defining Nodes for Single Signon

Access the Node Definitions page (PeopleTools, Portal, Node Definitions).



Node Definitions page

The two options related to single signon are:

Authentication Option

Determines how nodes in a single signon configuration authenticate other nodes in the same configuration. You have the following options:

None: Specifies no authentication between nodes.

Note. This option conflicts with PeopleSoft Integration Broker. If you select None, PeopleSoft Integration Broker messaging will fail, as will single signon.

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.
- 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.

Default Local Node

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.

See Also

PeopleTools 8.52: PeopleSoft Integration Broker Administration, "Adding and Configuring Nodes"

PeopleTools 8.52: PeopleSoft Integration Broker Administration, "Setting Up Secure Integration Environments," Implementing Nonrepudiation

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, 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, Single Signonand 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.peoplesoft.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

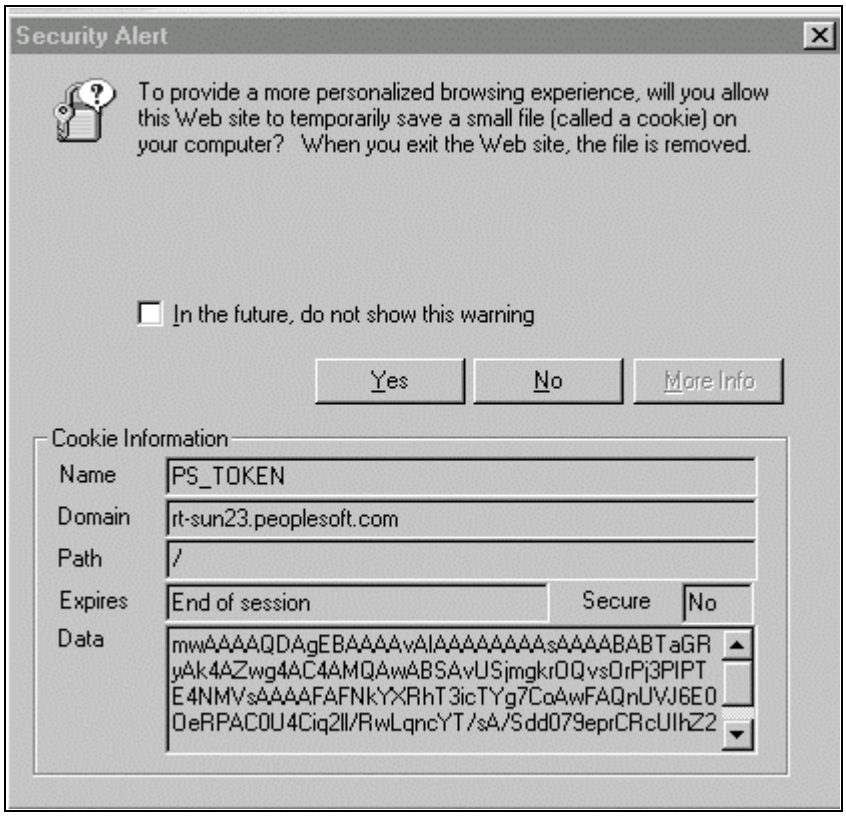
The following occurs:

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.



Message alerting user about the cookie

The cookie that the web server distributes for PeopleSoft single signon is named PS_TOKEN. In this case the domain rt-sun23.peoplesoft.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 Objects, 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 *.peoplesoft8.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.peoplesoft.com to signon to the web server a.peoplesoft.com, and then access b.peoplesoft.com using single signon. In this situation, you would need to update the hosts file on a.peoplesoft.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:
#
#      102.54.94.97     rhino.acme.com          # source server
#       38.25.63.10     x.acme.com              # x client host

127.0.0.1       localhost
216.131.221.88  a.peoplesoft.com
66.122.220.101  b.peoplesoft.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.peoplesoft.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 serverX.peoplesoft.com/servlets, not serverX/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.PSFT_ecenter.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, 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, 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.peoplesoft.com sets a cookie after the user is authenticated, the browser (by default) only sends the cookie to a.peoplesoft.com. By default, the browser would not send the cookie to b.peoplesoft.com. To make the browser send the single signon cookie to all servers at in a domain (peoplesoft.com), access the Web Profile Configuration - General page and set a value of *.peoplesoft.com* for the Authentication Domain property.

   Note. You need the leading period (.) before the domain. It should appear as ".peoplesoft.com," not "peoplesoft.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, 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, 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.peoplesoft.com sets a cookie after the user is authenticated, the browser (by default) only sends the cookie to a.peoplesoft.com. By default, the browser would not send the cookie to b.peoplesoft.com. To make the browser send the single signon cookie to all servers at in a domain (peoplesoft.com), modify the authentication domain as follows.

See Chapter 11, "Working with SSL/TLS and Digital Certificates," page 263 and Chapter 7, "Employing LDAP Directory Services," page 143.

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 Chapter 8, "Employing Signon PeopleCode and User Exits," Creating a Public Access User, page 187.

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=.peoplesoft.com; path=/; MaxAge=1";
document.cookie = cookie;
location="https://hcm.oraclepeoplesoft.com/psp/hcmprod/EMPLOYEE/HCM/c/ROLE_⇒
EMPLOYEE.TIME_OFF.GBL?FolderPath=PORTAL_ROOT_OBJECT.EE_SELF_SERVE.EE_TIMEOFF_⇒
GBL&IsFolder=false&IgnoreParamTempl=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 Enterprise Portal. 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 executed 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 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.peoplesoft.com/servlets/psp/ps/hrdb/?cmd=logout"
   height=0 width=0 border=0>
   <IMG src="http://fin.peoplesoft.com/servlets/psp/ps/hrdb/?cmd=logout"
   height=0 width=0 border=0>
   <IMG src="http://htmlaccess.peoplesoft.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 Manager as the PeopleSoft single signon solution:

1. Install and configure Oracle Access Manager.

   See Oracle Access Manager Installation Guide

2. In the PeopleSoft application, create *OAMPSFT* as a new user profile and associate a low security role such as PeopleSoft User.

   See Chapter 5, "Administering User Profiles," Creating a New User Profile, page 99.

3. In the user profile, access the ID page and select *NONE* as the ID type.

   See Chapter 5, "Administering User Profiles," Defining User Profile Types, page 97.

4. Save the user profile.

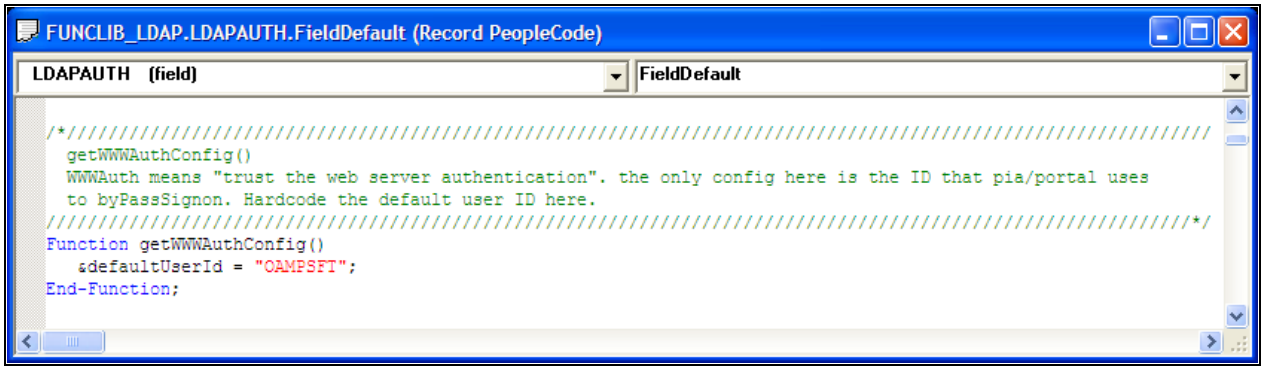
5. Access the web profile and enter *OAMPSFT*as the public access user ID.

   See *PeopleTools 8.52: PeopleTools Portal Technologies*, "Configuring the Portal Environment," Configuring Web Profiles.

6. Using PeopleSoft Application Designer, open the FUNCLIB_LDAP record.

7. Right-click the LDAPAUTH field and select View PeopleCode.

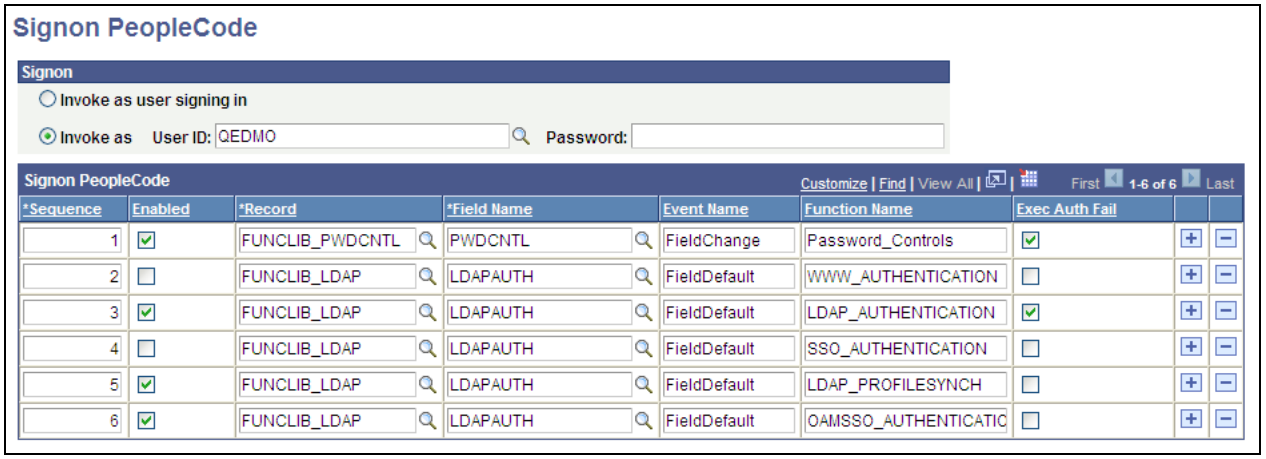
8. Find the `getWWWAuthConfig()` function and replace the value that is assigned to the `&defaultUserId` with *OAMPSFT*.



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.



Signon PeopleCode page showing OAMSSO_AUTHENTICATION function enabled

11. Save the page.

12. WebLogic users must disable basic authentication.

    Access <PIA_HOME>\webserv\peoplesoft\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/security⇒
/xacml">
    <name>peoplesoft</name>
    <realm>myrealm</realm>

.......

    <credential-encrypted>{3DES}dOa1fqoTbX1GUq7RQPhDNDgkWkIZhzWVlEXkmSMbt9Uuf1Ff⇒
VZIrJC</credential-encrypted>
    <enforce-valid-basic-auth-credentials>false</enforce-valid-basic-auth-⇒
credentials>

  </security-configuration>
```

13. Stop and restart the application server, web server, and HTTP server.

Implementing Kerberos as the Desktop Single Signon Solution

This section presents an overview of a Kerberos-PeopleSoft integration, presents considerations for securing Kerberos tokens across the enterprise, and then discusses how to:

- Configure the directory server to act as the Key Distribution Center (KDC).
- Set up Kerberos authentication on the web server.
- Set up Kerberos authentication on the application server.
- Write Signon PeopleCode for Kerberos authentication.
- Configure the PeopleSoft application for Kerberos authentication.
- Modify Signon PeopleCode to include Kerberos authentication.

Understanding Implementing Kerberos Authentication in PeopleSoft Systems

PeopleTools delivers a Kerberos Software Development Kit (SDK) that enables you to configure your PeopleSoft web and application servers to accept Kerberos authentication from Microsoft Active Directory, an LDAP version 3 compliant directory server. The SDK includes a servlet filter and also provides Java source code, which makes it possible for you to create more extensive custom single-signon features, such as support for multi-factor authentication or to consume custom or other authentication environment tokens.

An appendix in this PeopleBook describes how to set up Microsoft Active Directory.

The Kerberos SDK is a proof of concept that was developed and tested in the Microsoft Windows environment. To implement Kerberos authentication in non-Windows environments, you must customize the Kerberos SDK source code.

Understanding Kerberos-PeopleSoft Integration

Kerberos is a network protocol that uses secret key cryptography to provide authentication between clients and servers. The Kerberos authentication protocol enables mutual authentication between clients and servers before secure network connections are established. Kerberos establishes trust between a client and server by using a third party intermediary called the Key Distribution Center (KDC). The KDC stores a secret key (password) for every client and server on the network. Clients and servers authenticate each other by exchanging tickets, which represent the clients' network credentials. When a client wants to access a server, the client first authenticates against the KDC and requests a client-to-server ticket for that server. The client then presents that ticket to the server, which validates the credentials and then processes the client's request.

In Kerberos-PeopleSoft single signon integrations, the Kerberos server handles HTTP requests. While Kerberos was not specifically designed to operate over HTTP, tokens can nonetheless be wrapped within HTTP headers according to the Simple Protected Negotiate (SPNEGO) protocol. In a Kerberos-PeopleSoft implementation, you must configure the web server to:

1. Accept the HTTP-wrapped Kerberos tokens that are sent by the client.

2. Forward the user credentials on to the application server for signon.

Note. The Kerberos protocol assumes that transactions between clients and servers take place on an open network where most clients and many servers are not physically secure, and packets traveling along the network can be monitored and modified freely. In such a situation, Kerberos authentication prevents hackers, who can appear to be either a client or a server, from accessing , viewing , and manipulating communications between legitimate clients and servers.

When implementing Kerberos desktop single signon, you configure your directory server to act as the KDC. All clients and servers must authenticate against the KDC before they can communicate securely with each other. Clients authenticate when a user signs in using network credentials from a laptop or desktop machine. Servers, on the other hand, generally authenticate as a different user than the one running the server process. Because of this, you must create a server user account in your directory.

A user's credentials must pass to the server machine; you do this by using a keytab file. The keytab file stores the name and encrypted password of the server user account. The use of the keytab file enables a server to automatically authenticate against the KDC without prompting for the password or storing it as plaintext. You also must map the server user's credentials to a Service Principal Name (SPN) that matches the website that the server hosts. For example, if a browser requests a client-to-server ticket for *www.example.com,* it asks the KDC for a ticket to a service principal named *HTTP/www.example.com@example.com.* You must create the mapping between this SPN and the server user.

Note. PeopleSoft's Kerberos authentication implementation supports only Microsoft Active Directory as an LDAP version 3 directory server. Set up for Microsoft Active Directory is included in an appendix in this PeopleBook.

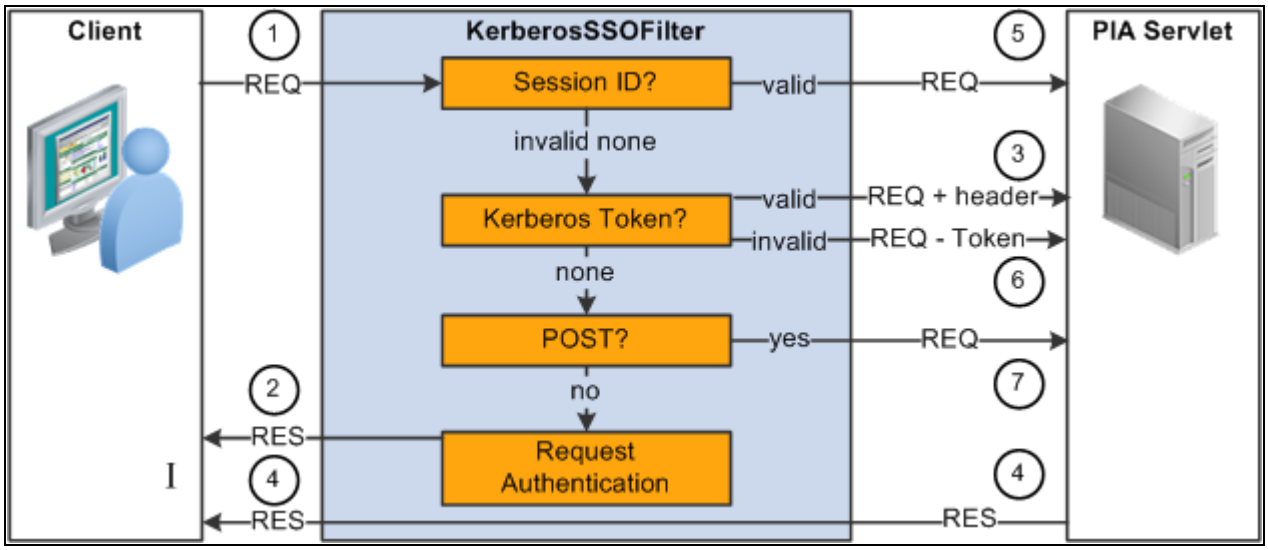
See Appendix A, "Enabling Kerberos Authentication in a Microsoft Active Directory Environment," page 355 .

The PeopleSoft Web Server KerberosSSOFilter Servlet

The PeopleSoft web server requests authentication from the client and then forwards the client credentials to the application server. This process is encapsulated in the KerberosSSOFilter servlet filter, which you attach to the PeopleSoft portal servlet. Servlet filters are intermediaries between the portal servlet and the client. Filters can:

- Pass messages through unchanged.
- Modify message contents.
- Stop messages entirely.

The following diagram illustrates the request and response flow between the client, the KerberosSSOFilter, and the portal servlet:



KerberosSSOFilter authentication process flow

This table describes the requests and responses between the client, the KerberosSSOFilter and the web server.

Request/Response	*Description*
1 REQ	*Request* from client (who may be authenticated, unauthenticated, or providing authentication)
2 RES	*Response* from KerberosSSOFilter requesting authentication
3 REQ + header	*Request* along with KRB_HEADER (Authentication) header which contains the authenticated user
4 RES	*Response* from the Portal Servlet (likely forwarded from the application server)
5 REQ	*Request* from client, unmodified because the client has a session 6

Request/Response	*Description*
6 REQ - token	*Request* from client with Kerberos token removed because authentication of at token failed
7 REQ	*Request* from client, unmodified because the request was a POST

The filter first checks to see if the client request (1) has a valid session id. If it does, then the client is already authenticated and no further action is required (5). If not, the filter checks the request's headers for a Kerberos token. This token is then validated using the server's credentials. If the token is valid, the authenticated username is added to the request in a special header, KRB_USER (3). The application server can then retrieve the username from this header, once it receives the request.

If the token is invalid, the token header is removed, making it appear to the servlet (and eventually the application server) as if no ticket exchange occurred (6). If no Kerberos token is present, then the filter must request authentication (2). Before doing so, it checks if the request is a POST. POST requests are always passed through to the Servlet (7), because the client browser will not resend POST parameters along with the authentication information. In addition, POST is used during regular web-based sign in, and performing single signon simultaneously with this is redundant and not a good practice. For the sake of correctness and simplicity, POST requests never trigger authentication requests or single signon.

After the filter makes the authentication request, the browser must determine how to proceed. If it can find appropriate credentials (a client-to-server ticket) then it will repeat the request (1) with the appropriate token in the header. Note that a browser may not respond to an authorization request if it cannot acquire credentials. The user will then not be able to view the site. Setting up the browser to avoid this is discussed in Browser Configuration. The filter is transparent to servlet responses. That is, responses from the servlet (4) pass through unchanged. Thus, responses from the application server are also unaffected by the filter. In other words, once Kerberos authentication completes and the user is issued a valid session ID, client/server messages will continue as if the filter was not present (the path going through 1, 5 and 4).

This table describes a typical single signon transaction in which Kerberos provides *valid* credentials during the authentication process:

Transaction Number	*Description*
1, 2	Unauthenticated client attempts to access a resource and KerberosSSOFilter requests authentication
1, 3, 4	Client responds with a Kerberos token, which KerberosSSOFilter validates and passes on to the servlet. The servlet passes it on to the application server, which signs the client on.
1, 5, 4, 1, 5, 4 ...	The rest of the communication passes through the KerberosSSOFilter unmodified

This table describes a typical single signon transaction in which Kerberos provides *invalid* credentials during the authentication process:

Transaction Number	*Description*
1, 2	Unauthenticated client attempts to access a resource and KerberosSSOFilter requests authentication
1, 6, 4	Client responds with a Kerberos token, which KerberosSSOFilter finds to be invalid. It removes the token from the request and forwards it on to the application server, which responds with a login page.
1, 7, 4	Client provides credentials using the login page (a POST operation). The servlet passes the credentials on to the application server which signs the client on.
1, 5, 4, 1, 5, 4 …	The rest of the communication passes through the KerberosSSOFilter unmodified

PeopleSoft Public Access and Signon PeopleCode for Kerberos Authentication

Although the web server performs Kerberos authentication, the application server is responsible for signon. The web server determines the network or directory username, but the application server must then map that name to a PeopleSoft user ID and sign the user in. To achieve single signon, the application server must perform these functions any time a user requests a PeopleSoft page, for example by clicking a link or by selecting a browser Favorite or Bookmark. You enable the application server to map the directory username to the PeopleSoft user ID by setting up public access, and then by adding a Kerberos authentication Signon PeopleCode function.

When a user who is not signed in requests a page, the application temporarily signs in as the public access user. You configure Single Signon settings that instruct the application server to run the `KRB_AUTHENTICATION` PeopleCode function in the FUNCLIB_LDAP record. This function checks for the request headers that the KerberosSSOFilter generates, and then performs the mapping and sign in operations.

The PeopleCode function that performs the user mapping depends on your specific implementation, but if you import your PeopleSoft users through LDAP from the same directory server that you use for Kerberos authentication, then the mapping is straightforward. For example, Kerberos authentication might return a username in this format: *johnsmith@EXAMPLE.COM,* while in your PeopleSoft application, the user ID is simply, *johnsmith.* In this case, the mapping operation merely strips *@EXAMPLE.COM* from the Kerberos username string.

Note. If the application server finds no authentication headers, then you should retrieve the PeopleSoft Sign In page so that a user is unaware of a failed attempt at authentication.

Browser Configuration

Internet Explorer default settings typically work with Kerberos single signon. However, if the browser settings are not the default, you might need to change the browser settings. In addition, Internet Explorer only uses Kerberos authentication for sites in the Local intranet zone. If your PeopleSoft applications are not within this zone, you must add them.

See Appendix B, "Enabling Kerberos Authentication in the Browser," page 359.

Considerations for Securing Kerberos Tokens Across the Enterprise

The previous sections discuss a web server-only implementation of Kerberos authentication. While this set up is a robust single signon solution, some elements remain less than optimally secure. In a web server-only implementation, the application server trusts *any* username that the web server places into the KRB_USER header. If the Kerberos token is accessed and manipulated between the web server and the application server, a hacker can very possibly sign in to PeopleSoft applications as any user. The web server-only configuration places an inordinate amount of trust on the PeopleSoft web server, yet ignores the connection between the web and application servers.

To make Kerberos authentication more secure, you can:

- Configure the application server to re-validate the Kerberos token.
- Configure the KerberosSSOFilter to request Kerberos tokens over a secure network connection only.
- Bypass KerberosSSOFilter authentication and forward the Kerberos token directly to the application server.

By using these methods, you make it almost impossible to trick the application server with a username and prevent network traffic sniffing. You also decrease the load on the web server.

Understanding Kerberos Authentication on the Application Server

In a typical configuration, the web server forwards the Kerberos token that it receives from the client to the application server, but the application server ignores it. To make the token more secure, set up the application server to run a Java program that accesses and validates the Kerberos token. The program should check the keytab for the username (the same way that the KerberosSSOFilter does), compare the username from the keytab to the value in the KRB_USER header, and then only if they match continue the sign in process.

Understanding Kerberos Authentication on an Encrypted Network

Kerberos single signon should be done through an encrypted channel by using SSL/TLS. Encrypting the channel substantially increases security. Using a secure channel requires *no* changes to the Kerberos single signon configuration, only that PeopleSoft pages are accessed through HTTPS rather than HTTP. To help enforce this policy, configure the KerberosSSOFilter to request Kerberos tokens over a secure connection only. When you implement this configuration, the filter passes any request that comes over an non-secure connection directly to the web server.

Note. Requiring this setting is not always appropriate. If the web server uses a proxy to connect securely to the client, then the connection between the proxy and the web server might not be secure, although the connection to the client is. In this case, the KerberosSSOFilter will not recognize that the connection is actually secure and will not request authentication.

Forwarding Kerberos Tokens Directly to the Application Server

Another possible configuration is to have the KerberosSSOFilter perform no authentication, but to forward tokens directly to the application server. The filter still must request the token, but does not validate it or add a KRB_USER header before passing the message on. The application server is responsible for both validation and sign in. Although this configuration only requires one validation step, rather than two, it also prevents the KerberosSSOFilter from appending a mutual authentication token.

Typically, a mutual authentication token is returned as part of the response to a successful authentication attempt. In theory, the browser can use the token to verify the identity of the website—making the authentication mutual. In practice, however, neither Mozilla Firefox nor Microsoft Internet Explorer does this. Removing authentication from the filter simply decreases the load on the web server.

Configuring the Directory Server to Act as the Key Distribution Center KDC

To configure your directory server to act as the KDC:

1. Create an appropriate server user account in the directory.
2. Generate the keytab file. The keytab file stores the name and encrypted password of the server user account.
3. Map the server user credentials to a Service Principal Name (SPN) matching the website that the server hosts.

   When a browser requests a client-to-server ticket for *www.example.com*, the browser asks the KDC for a ticket to a service principal named *HTTP/www.example.com@example.com.*

See Also

Appendix A, "Enabling Kerberos Authentication in a Microsoft Active Directory Environment," page 355

Setting up Kerberos Authentication on the Web Server

To set up Kerberos authentication on the web server:

- Configure the web server JVM for Kerberos authentication.
- Attach the KerberosSSOFilter to the PeopleSoft portal servlet.

Configure the Web Server JVM for Kerberos Authentication

To configure the web server JVM for Kerberos authentication:

1. Place the keytab file in a folder, such as C:\krb\, on the server.

   The actual location is not important as long as it is accessible by the server.

   Important! Use caution when you transfer the keytab; any party that gains access to it can pose as the server.

2. In the same folder, create two configuration files:

   - *krb5.conf*

     The *krb5.conf* file defines the domain to use for authentication and the domain controller to use as the KDC. For example, if you use the *example.com* domain with a domain controller at 192.168.1.1, then the krb5.conf file should contain:

     ```
     [libdefaults]
     default_realm = EXAMPLE.COM
     [realms]
     EXAMPLE.COM = {
     kdc = 192.168.1.1
     }
     ```

   - *krbLogin.conf.*

     The *krbLogin.conf* file defines how the server authenticates against the KDC. For example, if you use a server that hosts *www.example.com* with keytab *krb5.keytab* in *C:\krb*, then the krbLogin.conf file should contain:

     ```
     krbServer {
     com.sun.security.auth.module.Krb5LoginModule required
     storeKey=true
     useKeyTab=true
     keyTab="C:/krb/krb5.keytab"
     isInitiator=false
     principal="HTTP/www.example.com";
     };
     ```

     Note. Use forward slashes in the keytab path.

3. Configure the web server to use these files for Kerberos authentication, by adding JVM arguments.

   a. Edit the setEnv.cmd file in the <*PS_HOME*>\webserv\peoplesoft\bin\ directory.

   b. Find the line that begins with `SET JAVA_OPTIONS_WIN=` and append the line with this text:

      ```
      -Djava.security.auth.login.config="C:\krb\krbLogin.conf
      -Djava.security.krb5.conf="C:\krb\krb5.conf"
      ```

4. Save the file.

Attaching the KerberosSSOFilter to the PeopleSoft Portal Servlet

To attach the KerberosSSOFilter to the PeopleSoft portal servlet:

1. These Java class files should be present in the <*PS_HOME*> \webserv\peoplesoft\applications\peoplesoft\PORTAL.war\WEB INF\classes\com\peoplesoft\pt\desktopsso\kerberos directory.

   - KerberosSSOFilter.class
   - KerberosSSOFilter$1.class
   - KerberosSSOFilter$KerberosAuthWrapper.class
   - KerberosSSOFilter$KerberosHideWrapper.class

   Note. In addition to the compiled Java classes, all PeopleSoft applications include the Java source code for the Kerberos single signon. You can find the source code files in the <*PS_HOME*> \sdk\desktopsso\src\com\peoplesoft\pt\desktopsso\kerberos directory.

   ---

2. Open the web.xml file in the <*PS_HOME*> \webserv\peoplesoft\applications\peoplesoft\PORTAL.war\WEB INF\ directory.

3. Find the section that begins and ends with `<display-name> ... </display-name>` and below that section insert the following xml. This table describes the purpose of the xml.

Description	***XML***

Begin the filter definition.
Identify the filter as KerberosSSO.
Identify the location of the KerberosSSO Java class files (Java servlet) that the web server calls.
	`<filter>`
	`<filter-name>KerberosSSO</filter-name>`
	`<filter-class>com.peoplesoft.pt.`
	`desktopsso.kerberos.Kerberos`
	`SSOFilter</filter-class>`

Require a secure (HTTPS) connection for the authentication request.
If the parameter value is `true`, use HTTPS connections only.
If the parameter value is `false`, non-HTTPS connections are allowed.
	`<init-param>`
	`<param-name>checkSecureConnection`
	`</param-name>`
	`<param-value>true</param-value>`
	`</init-param>`

•
•
•
Check for a valid Kerberos token.
If parameter value is `true`, then the web server calls the KerberosSSOValidator Java class to validate the Kerberos token.
If the parameter value is `false`, then bypass the web server and forward the Kerberos token to the application server for validation.
Important! If set to false, you must set up the application server to call the KerberosSSOValidator Java class.
See Chapter 9, "Implementing Single Signon," Configuring the Application Server JVM to Validate the Kerberos Token, page 231.
	`<init-param>`
	`<param-name>validateToken</param-name>`
	`<param-value>true</param-value>`
	`</init-param>`

Description	*XML*

•
•
•
Turn on verbose tracing of the Java class so that you can debug it if there is an issue.
If the parameter value is `true` then the web server writes extensive comments to the Java console as the KerberosSSOValidator runs.
If the parameter value is `false` then the web server does not create a debug file in the Java console as the KerberosSSOValidator runs.
Note. The program doesn't create and store a physical log file on the web server.
End the filter definition.
	`<init-param>`
	`<param-name>verbose</param-name>`
	`<param-value>true</param-value>`
	`</init-param>`
	`</filter>`

Map specific URLs to specific filters.
Specify the Kerberos filter.
Specify the URLs by their pattern; `/*` indicates that all URLs should be directed to the filter.
	`<filter-mapping>`
	`<filter-name>KerberosSSO</filter-name>`
	`<url-pattern>/*</url-pattern>`
	`</filter-mapping>`

4. Save the file.

Setting up Kerberos Authentication on the Application Server

Setting up Kerberos authentication on the application server requires that you configure the application server JVM to validate the Kerberos token:

Configuring the Application Server JVM to Validate the Kerberos Token

To configure the application server JVM to validate the Kerberos token:

1. These Java class files should be in the <*PS_HOME*>\class\com\peoplesoft\pt\desktopsso\kerberos directory.
   - KerberosSSOValidator.class
   - KerberosSSOValidator$1.class

Note. In addition to the compiled Java classes, all PeopleSoft applications include the Java source code for the KerberosSSOValidator. You can find the source files in the <*PS_HOME*>\sdk\desktopsso\src\com\peoplesoft\pt\desktopsso\kerberos directory.

2. Open the application server configuration file: psappsrv.cfg
3. Find the line that begins with `JavaVM Options=`and append the line with this text:

```
-Djava.security.auth.login.config=C:\krb\krbLogin.conf
-Djava.security.krb5.conf=C:\krb\krb5.conf.
```

4. Save the file.

Writing Signon PeopleCode for Kerberos Authentication

The Signon PeopleCode function that you create for Kerberos authentication can vary based on where you want authentication to occur. This section discusses Signon PeopleCode that implements Kerberos authentication:

- At both the web and application servers.
- At the application server only

Implementing Kerberos Authentication at Both the Web and Application Servers

To implement Kerberos authentication at both the web and application servers only:

1. In PeopleSoft Application Designer, open the FUNCLIB_LDAP record definition. Right-click the LDAPAUTH field and select View PeopleCode.
2. Find the `Function getWWWAuthConfig()` PeopleCode function.
3. Change the `&defaultUserId` to *"PUBUSER"*.

4. In the same field and event, add a KRB_AUTHENTICATION function.

   This table describes the elements that make up a sample Kerberos authentication Signon PeopleCode program:

Code	***Description***

`Function KRB_AUTHENTICATION()`
 	
`  If %PSAuthResult = True And`
`  &authMethod <> "WWW" And`
`  &authMethod <> "OAMSSO" And`
`  &authMethod <> "OSSO" And`
`  &authMethod <> "SSO" And`
`  &authMethod <> "LDAP" Then`
	Declare the function.
	
	Check that the user is authorized and that the application is using no other authentication method.

`  getWWWAuthConfig();`
`  If %SignonUserId = &defaultUserId Then`
`  Local string &princName = %Request.Get⇒`
`Header("KRB_USER");`
`  Local string &krbToken = %Request.Get⇒`
`Header("Authorization");`
`  Local string &userName = &princName;`
`  Local number &foundDelim = Find("@",⇒`
` &userName);`
	Trust web server authentication.
	
	Verify that the public user is the current user and then,
	
	Retrieve the Authentication header from the KRB_USER and set it as the Kerberos token.
	
	Set the user name to be the principal name.
	
	Find the numeric location of the @ sign in the name.

`  If (&foundDelim > 0) Then`
`  &userName = Substring(&userName, 1,⇒`
` &foundDelim - 1);`
`  End-If;`
	If there is something in front of the @ symbol in the name, then
	
	Strip the text that precedes the @ symbol and set it as the user name..

`  If Len(&userName) > 0 Then`
`  &krbToken = Substring(&krbToken,⇒`
` 11, Len(&krbToken) + 1);`
`  &validator = GetJavaClass⇒`
`("com.peoplesoft.pt.desktopsso.kerberos.`
`  KerberosSSOValidator").getInstance();`
`  Local string &validUserName =⇒`
` &validator.validate(&krbToken);`
	If there is a user name, then
	
	Extract the user name from the Kerberos token.
	
	Call the Kerberos validation program and validate the Kerberos token.

`  If &validUserName <> "NULL" And`
`  &princName = &validUserName Then`
`  SetAuthenticationResult⇒`
`(True, Upper(&userName), "", False);`
`  &authMethod = "KRB";`
`  End-If;`
`  End-If;`
`  End-If;`
`  End-If;`
`End-Function;`
	If the user name contains a value, then
	
	Compare the principal name to the name in the Kerberos token, and if the values are the same,
	
	Authenticate the user and sign in.
	
	Specify to the system that Kerberos authentication was performed so that other Signon PeopleCode functions can perform similar checks to the one at the beginning of function.
	
	Close the if statements.
	
	Close the function.

5. Save the record definition.

Implementing Kerberos Authentication at the Application Server Only

To implement Kerberos authentication at the application server only:

Note. In the web.xml file on the web server, make sure that you have set the `validateToken` parameter to `false` so that the web server forwards the Kerberos token directly to the application server.

---

1. In PeopleSoft Application Designer, open the FUNCLIB_LDAP record definition. Right-click the LDAPAUTH field and select View PeopleCode.
2. Find the `Function getWWWAuthConfig()`PeopleCode function.
3. Change the `&defaultUserId` to *"PUBUSER"*.

4. In the same field and event, add a KRB_AUTHENTICATION function.

   This table describes the elements that make up a sample Kerberos authentication Signon PeopleCode program:

Code	*Description*

`Function KRB_AUTHENTICATION()`
 	
`  If %PSAuthResult = True And`
`     &authMethod <> "WWW" And`
`     &authMethod <> "OAMSSO" And`
`     &authMethod <> "OSSO" And`
`     &authMethod <> "SSO" And`
`     &authMethod <> "LDAP" Then`
	Declare the function.
	
	Check that the user is authorized and that the application is using no other authentication method.

`        getWWWAuthConfig();`
`        If %SignonUserId = &defaultUserId Then`
`           Local string &krbToken = %Request.Get⇒`
`Header("Authorization");`
	Trust web server authentication.
	
	Verify that the public user is the current user and then,
	
	Retrieve the Authentication header from the Kerberos token.

`           If Len(&krbToken) > 12 Then`
`               &krbToken = Substring(&krbToken,⇒`
` 11, Len(&krbToken) + 1);`
 	
`               &validator = GetJavaClass⇒`
`("com.peoplesoft.pt.desktopsso.kerberos.`
`        KerberosSSOValidator").getInstance();`
 	
`               Local string &userName =⇒`
` &validator.validate(&krbToken);`
	Extract the user name from the Kerberos token.
	
	Call the Kerberos validation program and validate the Kerberos token.

`               If &userName <> "NULL" Then`
`                   Local number &foundDelim = Find⇒`
`("@", &userName);`
`                   If (&foundDelim > 0) Then`
`                       &userName = Substring(&user⇒`
`Name, 1, &foundDelim - 1);`
`                   End-If;`
 	
`                   SetAuthenticationResult(True,⇒`
` Upper(&userName), "", False);`
`                   &authMethod = "KRB";`
`               End-If;`
`            End-If;`
`         End-If;`
`    End-If;`
`End-Function;`
	If the user name contains a value, then
	
	Find the @ symbol and strip the text that precedes the @ symbol and set it as the user name.
	
	Authenticate the user and sign in.
	
	Specify to the system that Kerberos authentication was performed so that other Signon PeopleCode functions can perform similar checks to the one at the beginning of function.
	
	Close the if statements.
	
	Close the function.

5. Save the record definition.

Configuring a PeopleSoft Application to Use Kerberos Authentication

To configure the PeopleSoft application to use Kerberos authentication:

- Enable public access.

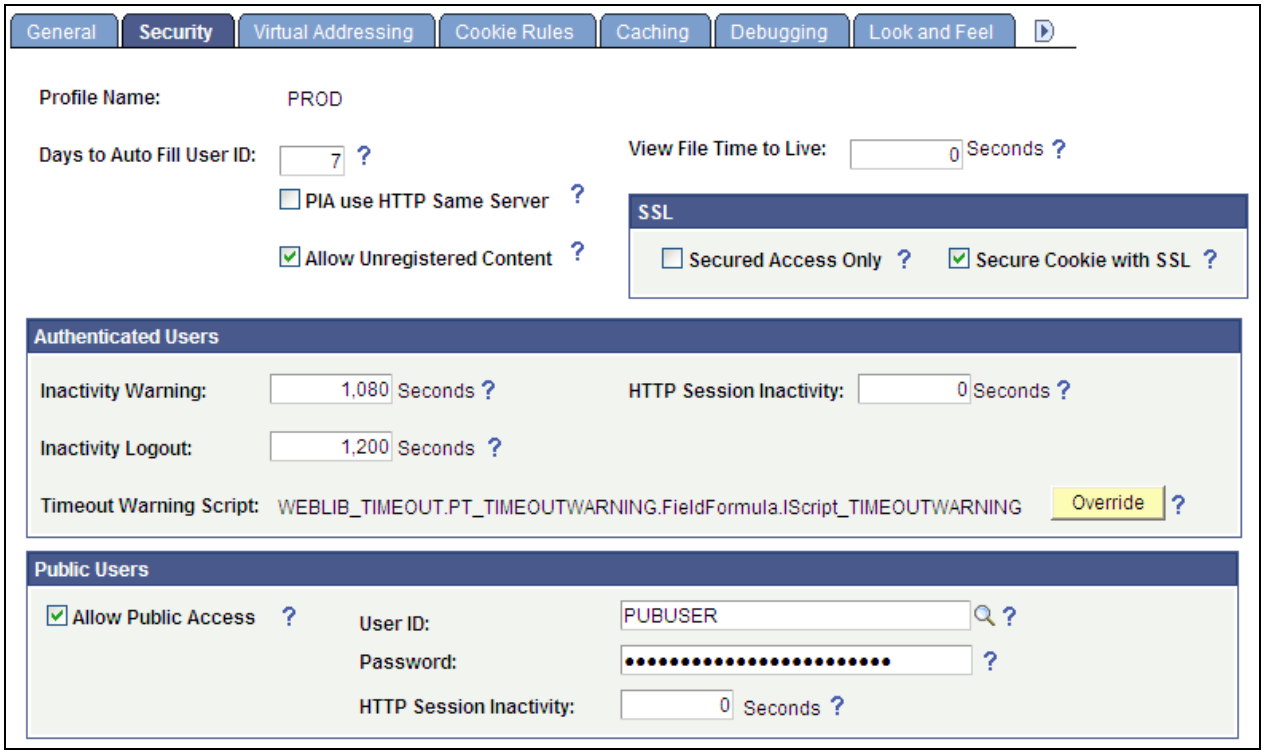
- Enable Kerberos authentication Signon PeopleCode.

Enabling Public Access

To enable public access:

1. Select PeopleTools, Web Profile, Web Profile Configuration and open the profile to which you are adding Kerberos authentication.
2. On the Security tab, select the Allow Public Access check box and then enter the user ID and password of the public access PeopleSoft application user. This user should have minimal permissions.

   In this example, you see that public access is enabled as the user ID, *PUBUSER*.



Web Profile: Security page showing PUBUSER as the public access user

---

Note. You are not limited to *PUBUSER* as the user ID, however, the user ID on this page must be the same user ID that you entered in the `getWWWAuthConfig()` function in the Signon PeopleCode.

---

3. Save the web profile.

Enabling Kerberos Authentication Signon PeopleCode

To enable Kerberos authentication Signon PeopleCode:

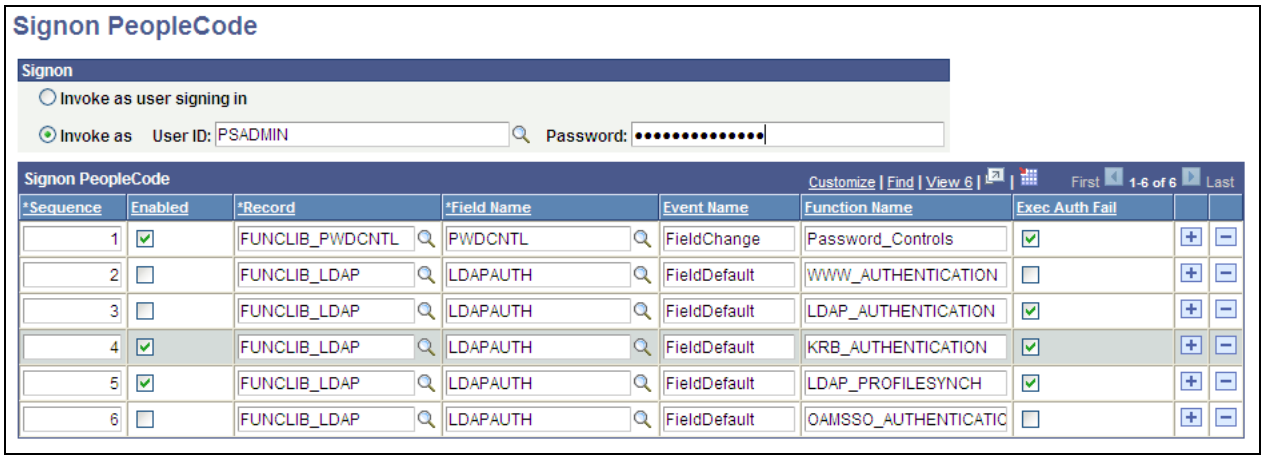
1. Select PeopleTools, Security, Security Objects, Signon PeopleCode.

2. Insert a new row and enter these values:

Field	*Value or State*
Enabled	Selected
Record	*FUNCLIB_LDAP*
Field Name	*LDAPAUTH*
Event Name	*FieldDefault*
Function Name	*KRB_AUTHENTICATION*
Exec Auth Fail	Selected
Sequence	Enter a value that does not conflict with the flow of other Signon PeopleCode functions.

3. Save the page.

   In this example, you see that the Kerberos authentication Signon PeopleCode, the *KRB_AUTHENTICATION* function, is enabled:



Signon PeopleCode page showing Kerberos authentication Signon PeopleCode

See Also

Chapter 5, "Administering User Profiles," Creating a New User Profile, page 99

Chapter 8, "Employing Signon PeopleCode and User Exits," Creating a Public Access User, page 187

PeopleTools 8.52: PeopleTools Portal Technologies, "Configuring the Portal Environment," Public Users

Chapter 10

Using Web Services for Object and Row-Level Data Authorization

This chapter discusses how to:

- Develop request messages for the security authorization service.
- Work with response messages for the security authorization service.
- Develop the security authorization service application class.
- Configure content references and components to use the security authorization service.
- Test and debug the security authorization service.

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 chapter.

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

The 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 Code Examples in this Chapter

This chapter 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 of this chapter for a list of all required and optional elements for authorization service request messages.

See Chapter 10, "Using Web Services for Object and Row-Level Data Authorization," Request Message Elements for the Security Authorization Service, page 245.

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 *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Configuring PeopleSoft Integration Broker for Handling Services."

- Service operation permissions.

  See *PeopleTools 8.52: PeopleSoft Integration Broker*, "Managing Service Operations," Setting Permissions to Service Operations.

- Authentication domain.

  See *PeopleTools 8.52: PeopleTools Portal Technologies*, "Configuring the Portal Environment," Configuring General Portal Properties.

- WS-Security.

  See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Setting Up Secure Integration Environments," Implementing Web Services Security.

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=RAJASIMHAN</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 executes 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*.
COMPONENT	Component name.	Required element for service type *UPGE*

Element	*Usage*	*Comments*
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 chapter. See Chapter 10, "Using Web Services for Object and Row-Level Data Authorization," Request Messages for Authorizing Access to Components, page 249.
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</KEYVAL>`
		`<KEYVAL>CUSTOMERID=CATHYPACIFIC</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:
		• *A*, Add.
		Constant value: *%Action_Add*
		• *U*. Update/Display.
		Constant value: *%Action_UpdateDisplay*
		• *L*. Update/Display All.
		Constant value: *%Action_UpdateDisplayAll*
		• *C*. Correction.

Element	*Usage*	*Comments*
		• *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.

See Also

Chapter 10, "Using Web Services for Object and Row-Level Data Authorization," Understanding Code Examples in this Chapter, page 242

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 ot 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..

See Also

Chapter 10, "Using Web Services for Object and Row-Level Data Authorization," Understanding Code Examples in this Chapter, page 242

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.

See Also

Chapter 10, "Using Web Services for Object and Row-Level Data Authorization," Understanding Code Examples in this Chapter, page 242

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>
```

See Also

Chapter 10, "Using Web Services for Object and Row-Level Data Authorization," Understanding Code Examples in this Chapter, page 242

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 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:AuthRequest);

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;
```

See Also

Chapter 10, "Using Web Services for Object and Row-Level Data Authorization," Understanding Code Examples in this Chapter, page 242

PeopleTools 8.52: PeopleCode API Reference, "Security Authorization Classes," Implementing a Security Authorization Handler

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 References and Components to Use the Security Authorization Service

This section discusses how to:

- Configure content references to use the security authorization service.
- Configure components to use the security authorization service.

Understanding Configuring Content References and Components to Use the Security Authorization Service

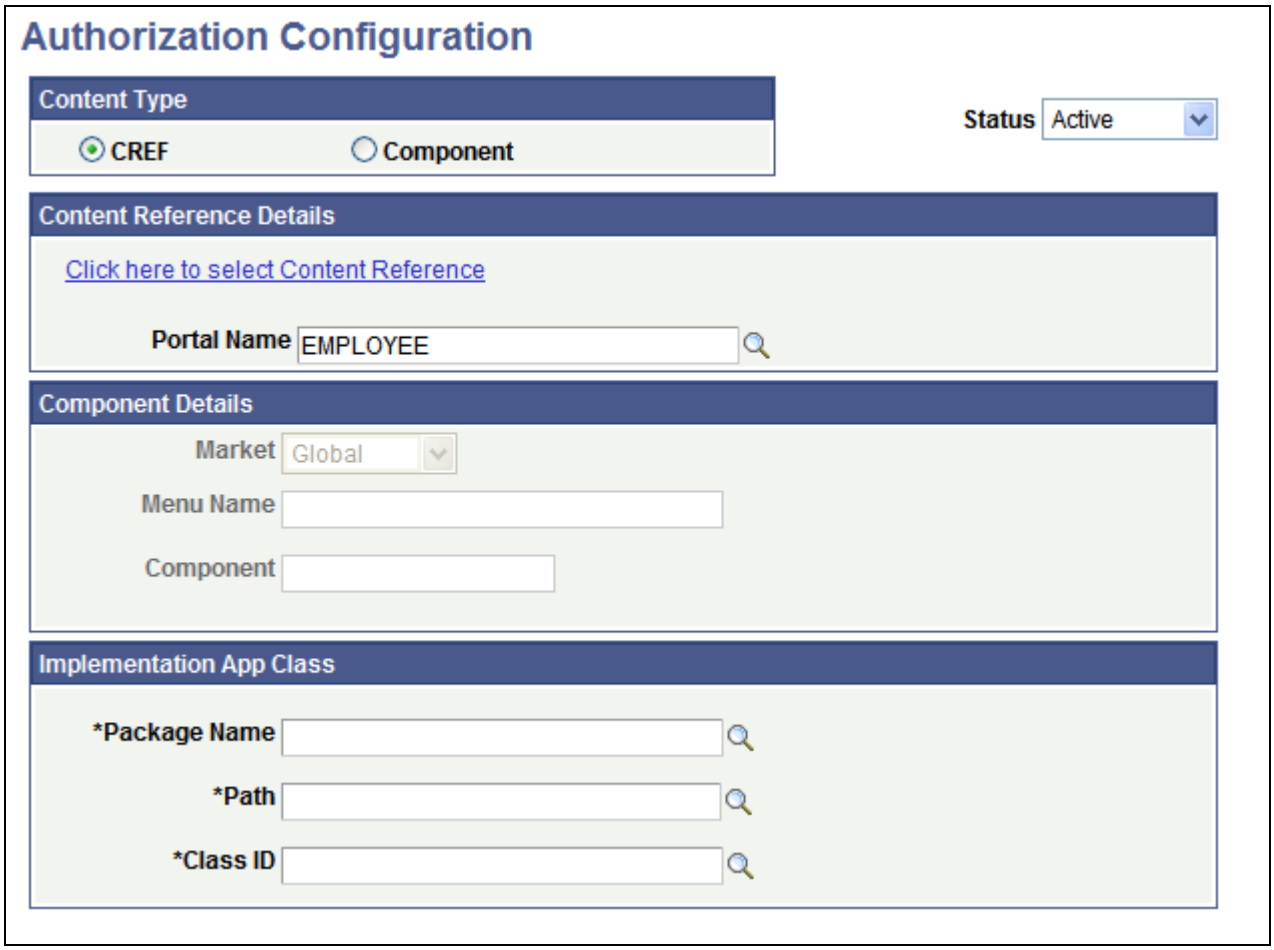
When using the authorization service to authorize row-level access to data in a component or content reference, you must configure the content reference or component to use the authorization service.

If you are using the authorization service to provide basic object authorization to a component or content reference you do not need to perform this configuration and you can invoke the authorization service as you would any other service.

Configuring Content References to Use the Security Authorization Service

Use the Authorization Configuration page (PTCAC_AUTH_CONFIG) to map content references to security authorization services.

To access the page, select PeopleTools, Security, Authorization Configuration. The following example shows the sections and fields that appear on the Authorization Configuration page when CREF is the content type. It is the default view of the page:

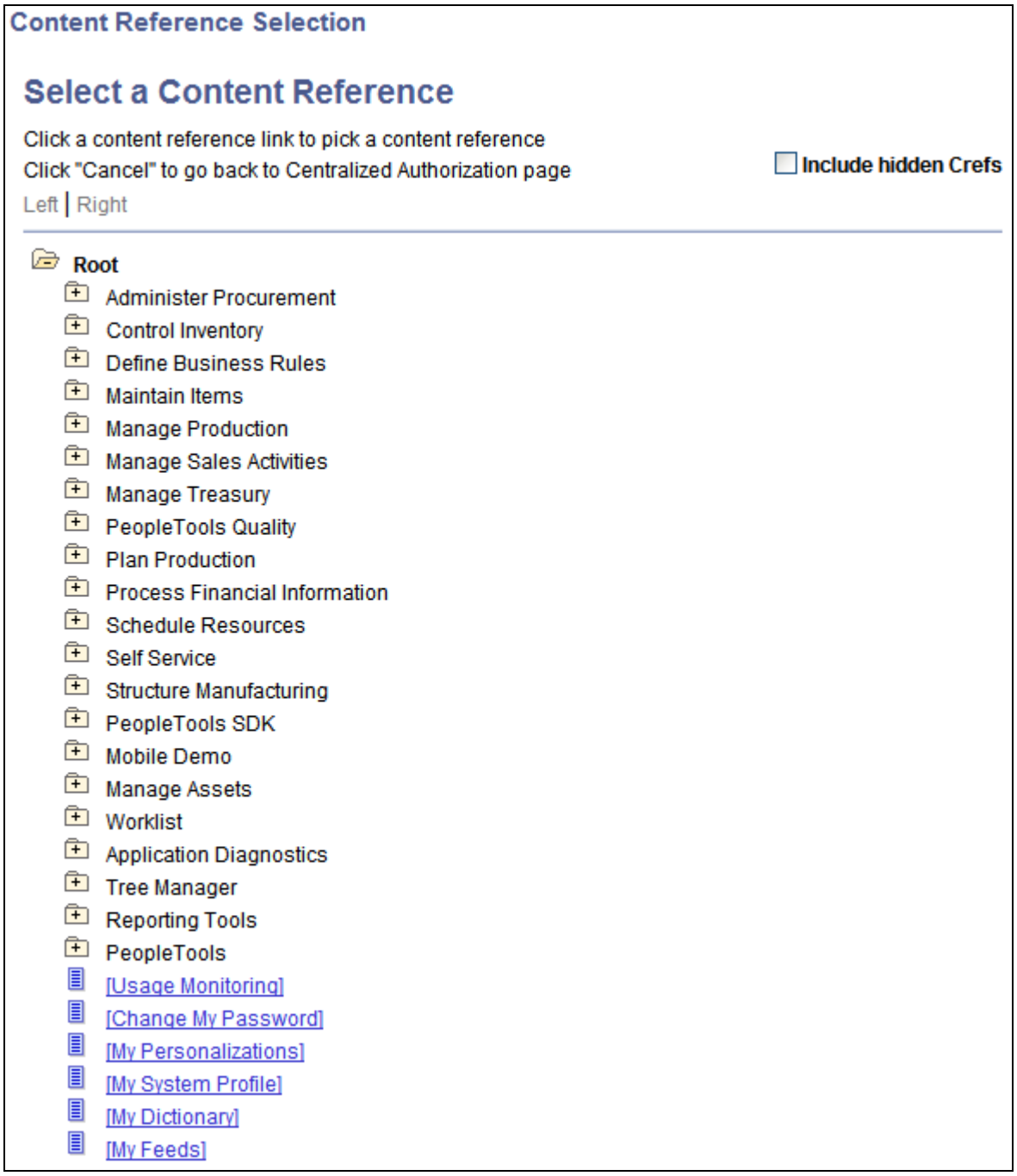


Configuring a content reference to use the authorization service.

Note. Note that the Status field is not currently used.

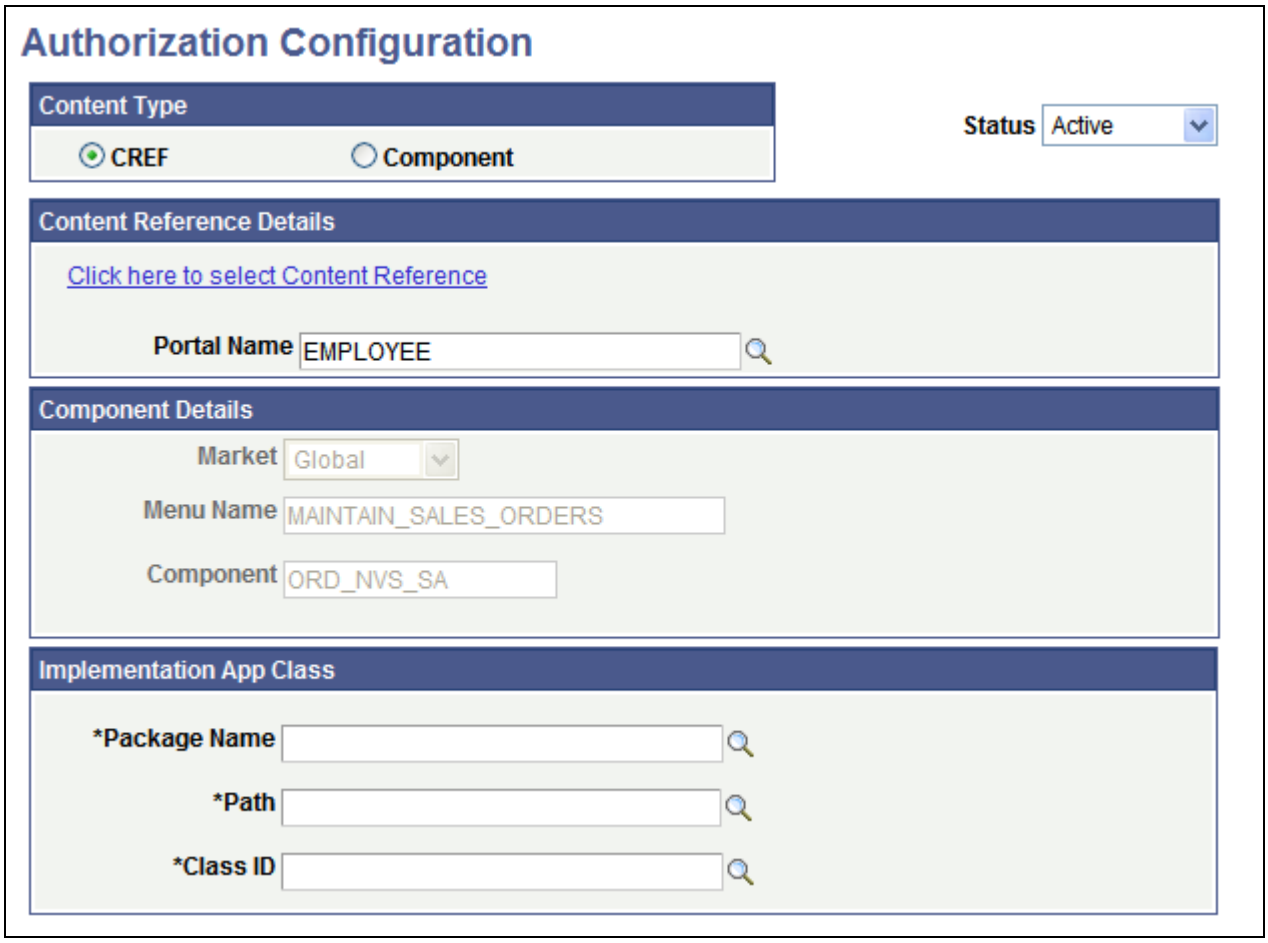
In the Content Reference Details section of the page select the content reference for which to configure for the service and the provider portal on which the content reference being authorized resides.

The Content Reference Details section features a Click here to select Content Reference link that you use to select the content reference. When you click the link the Select a Content Reference page (PTCAC_CRFURL_SELCT) appears as shown in the following example:



Selecting the content reference for the authorization service

Expand and collapse the folders on the page to select a content reference. The page also features an Include hidden Crefs box. Check the 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 values for the content reference in the Component Details section. The following example shows the Authorization Configuration page after selecting a content reference on the Select a Content Reference page:



Read-only component details of the content reference appear on the Authorization Configuration page.

The last configuration step is to use the Implementation App Class section of the page to define the details of the application class of the service.

To configure a content reference to use the authorization service:

1. Access the Authorization Configuration page (PeopleTools, Security, Authorization Configuration).
2. In the Content Type section, click the *CREF* radio button.
3. In the Content Reference Details section:
   a. Click the Click here to Select Content Reference link.

      The Select a Content Reference page appears.

   b. Navigate to and click the content reference to map to the service.

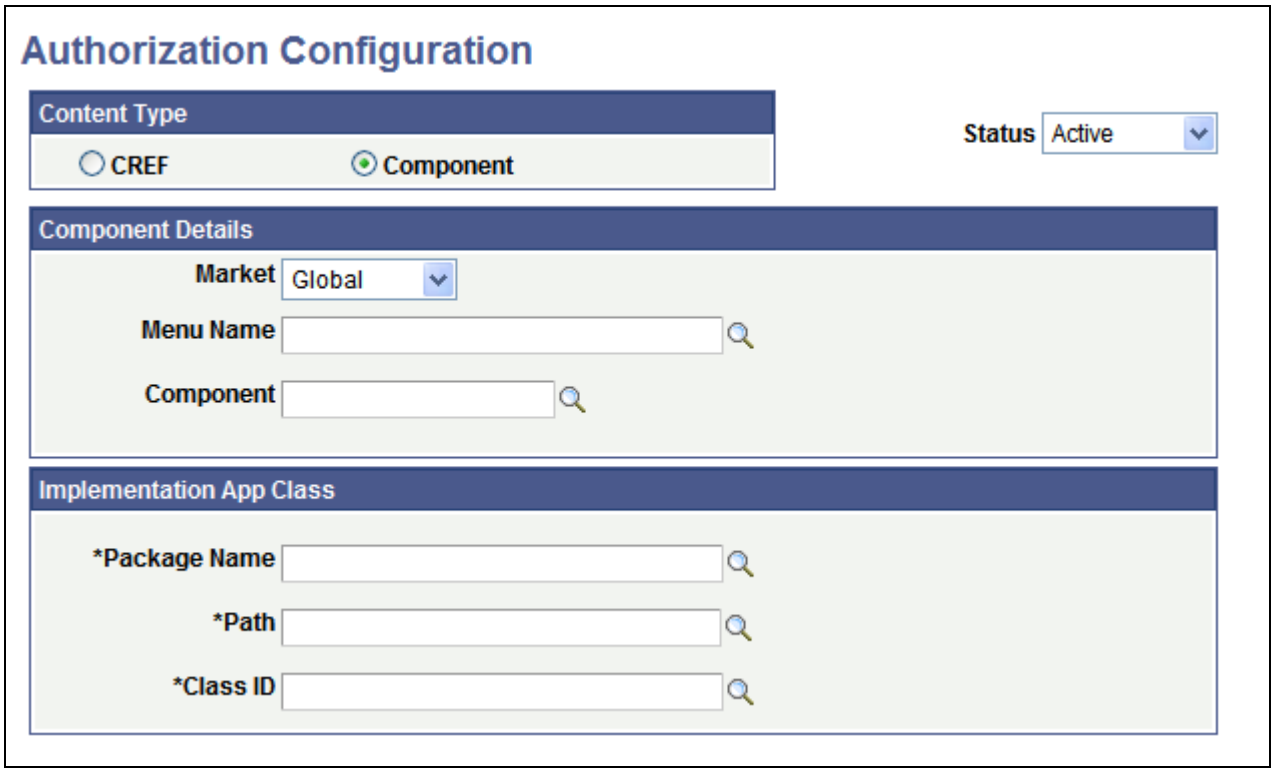
      The Authorization Configuration page appears.

   c. In the Portal Name field, select the provider portal where the authorized content reference resides.

4. In the Implementation App Class section, define the details of the application class for the service:
   a. In the Package Name field enter the package name.
   b. In the Path field enter the path.
   c. In the Class ID field enter the application class ID.
5. Click the Save button.

Configuring Components to Use the Security Authorization Service

If you select *Component* as the content type, the fields that appear on the page are those with which you work to configure a component to use an authorization service. The following example shows the Authorization Configuration page when the Content Type is set to *Component:*



Authorization Configuration page after selecting a Component to map to an authorization service.

Note. Note that the Status field is not currently used.

To configure a component to use the authorization service you must define the information about the component and the application class.

To configure a component to use the authorization service:

1. Access the Authorization Configuration page (PeopleTools, Security, Authorization Configuration).
2. In the Content Type section, click the *Component* radio button.

3. In the Component Details section:
    a. From the Market drop-down list select a value.
    b. In the Menu Name field enter the name of the menu where the component resides.
    c. In the Component field, enter the name of the component.
4. In the Implementation App Class section, define the details of the application class for the service:
    a. In the Package Name field enter the package name.
    b. In the Path field enter the path.
    c. In the Class ID field enter the application class ID.
5. Click the Save button

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.

  See *PeopleTools 8.52: Integration Broker Testing Utilities and Tools*, "Using the Handler Tester Utility."

- Generate SOAP Template Utility.

  Use this utility to test SOAP messages.

  See *PeopleTools 8.52: Integration Broker Testing Utilities and Tools*, "Using the Generate SOAP Template Utility."

Chapter 11

Working with SSL/TLS and Digital Certificates

This chapter provides an overview of Secure Sockets Layer/Transport Layer Security (SSL/TLS) and discusses how to configure 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 the message has not been altered.

The following table shows the PeopleSoft technologies that use HTTPS (HTTP over SSL/TLS) and how it is implemented in 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.
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 logon 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 logon 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, 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.

Type Select the type of certificate.

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).

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).

Chapter 12

Working with Web Service Security (WS-Security)

This chapter contains an overview of WS-Security and discusses how to:

- Implementing WS-Security for WSRP.
- Implementing WS-Security for PeopleSoft Integration Broker.

Understanding WS-Security

By implementing the 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, that 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 execute requested web services on behalf of the user without requiring a user to log in. Integration between web services consumer and producer feature is currently supported in PeopleSoft WSRP Portal, PeopleSoft Integration Broker, and BPEL product.

- 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.

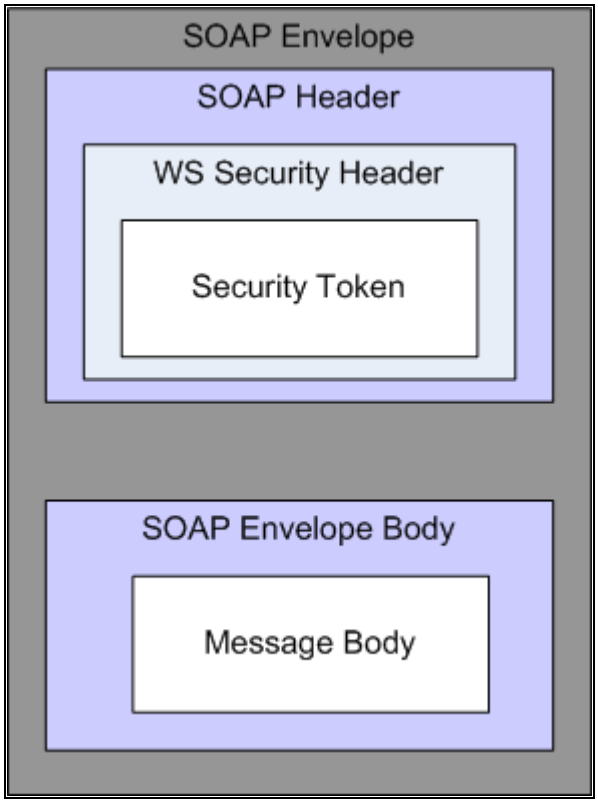
The Security Assertion Markup Language (SAML) token uses assertions to define a standard way to associate common information such as issuer ID, `NotBefore` and `NotOnOrAfter` conditions, assertion ID, subject, and so on.

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 chapter. 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:



WS-Security SOAP Message Structure

Implementing WS-Security for WSRP

If using the web services for remote portals technology, you implement WS-Security.

See *PeopleTools 8.52: PeopleTools Portal Technologies*, "Configuring WS-Security for WSRP Consumption and Production."

Implementing WS-Security for PeopleSoft Integration Broker

If using PeopleSoft Integration Broker, you configure WS-Security to ensure secure transmissions.

See *PeopleTools 8.52: PeopleSoft Integration Broker Administration*, "Setting Up Secure Integration Environments."

Chapter 13

Encrypting Text With PSCipher

This chapter contains an overview and discusses how to:

- Use the PSCipher utility.
- Generate a unique encryption key.
- Update the encryption key on Oracle WebLogic.
- Update the encryption key on IBM WebSphere.
- Secure the external key file.

Understanding the Triple Data Encryption Standard (DES) Encryption Implementation

The PSCipher utility provides Triple DES encryption (also 3DES) for increased data security. When you install Enterprise PeopleTools on your application servers and web servers, a default, Triple DES encryption key is provided. If your site decides to use the default key, no further configuration of your system is required to implement Triple DES encryption. However, if your site requires or desires a unique encryption key, you can generate a unique key using the PSCipher command line utility as described in this chapter.

The version of the default encryption key is version 1.1, or {V1.1}. If you generate a unique key, the current version used by the system would be {V1.2}. 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. You can now use the PSCipher command line utility to encrypt (with Triple DES) a variety of text values stored in various configuration files throughout your system. In addition, PSCipher also provides:

- Key generation: You can generate unique encryption keys if you do not want to use the default key.

- 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.

Note. PSCipher decrypts text encrypted in previous releases. For example, PSCipher in Enterprise PeopleTools 8.50 supports text encrypted with PSCipher in Enterprise PeopleTools 8.46.

To encrypt text, you submit text values in the form of arguments that PSCipher encrypts and then displays in its encrypted form. Suppose you needed to encrypt a user ID of "HRDMO" and a password of "DMOPSWD". You would submit these values to PSCipher as follows:

```
...\pscipher HRDMO
```

and

```
...\pscipher DMOPSWD
```

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. 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 Triple DES encryption keys. The `buildkey` command adds a new Triple DES 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 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 vender (as described in the following sections). 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 ({V1.1}), any server you add to the system and install PeopleTools 8.50 on will also have the default key version ({V1.1}). As such, no further steps are required. However, if you have generated a new key, giving the version number a value of {V1.2} 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 8.48 to PeopleTools 8.50, you will need to backup 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.

Updating the Encryption Key on Oracle WebLogic

On Oracle WebLogic, PSCipher.bat and psvault are stored in the following location:

<PIA_HOME>\webserv\<DOMAIN>.

Generating the Encryption Key on Oracle WebLogic

To update the encryption key:

1. Run <PIA_HOME>\webserv\<DOMAIN>\PSCipher –buildkey to create a new key in the key file.

   For example,

   ```
   c:\cd PIA_HOME\webserv\peoplesoft
   c:\PIA_HOME\webserv\peoplesoft>PSCipher.bat -buildkey
   Your environment has been set.
   A new key of version {V1.2} is generated successfully
   ```

2. Copy <PIA_HOME>\webserv\<DOMAIN>\psvault to the equivalent location on all other web server hosts and to <PS_HOME>\secvault\psvault on all application servers in your system.

3. 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_HOME\webserv*web_server*\applications\peoplesoft\PORTAL.war\WEB-INF\psftdocs*site_name*

The following encrypted text values in the configuration.properties file need to updated:

```
WebUserId={V1.1}et5LM5/C2fQPVWt5cztapg==
WebPassword={V1.1}et5LM5/C2fQPVWt5cztapg==
```

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_HOME\webserv*web_server*\applications\peoplesoft\PSIGW.war\WEB-INF

In the gatewayUserProfile.xml file, update the following text value:

```
<password>{V1.1}GD9klUFw8760HVaqeT4pkg==</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_HOME\webserv*web_server*\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:

- <PIA_HOME>\webserv\<DOMAIN>\applications\peoplesoft\PORTAL\WEB-INF\classes\
- <PIA_HOME>\webserv\<DOMAIN>\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.

Updating the Encryption Key on IBM WebSphere

On IBM WebSphere, PSCipher.bat and psvault key file are stored in the following location:

<PIA_HOME>\webserv\<Node_Server>\<APPLICATION>.ear

Generating the Encryption Key on IBM WebSphere

To update the encryption key:

1. Run <PIA_HOME>\webserv\<Node_Server>\<APPLICATION>.ear PSCipher –buildkey to create a new key in the key file.

   For example,

   ```
   c:\>cd ptinstall\webserv\DS9Node_DS9Node_server1\peoplesoft.ear
   c:\ptinstall\webserv\DS9Node_DS9Node_server1\peoplesoft.ear>PSCipher.bat -buildkey
   Your environment has been set.
   A new key of version {V1.2} is generated successfully
   ```

2. Copy <PIA_HOME>\webserv\<Node_Server>\<APPLICATION>.ear\psvault to the equivalent location on all other web server hosts and to <PS_HOME>\secvault\psvault on all application servers in your system.

3. 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:

<PIA_HOME>\webserv\<Node_Server>\<APPLICATION>.ear\PORTAL\WEB-INF\psftdocs\ps\

The following encrypted text values in the configuration.properties file need to updated:

```
WebUserId={V1.1}et5LM5/C2fQPVWt5cztapg==
WebPassword={V1.1}et5LM5/C2fQPVWt5cztapg==
```

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:

<PIA_HOME>\webserv\<Node_Server>\<APPLICATION>.ear\PSIGW\WEB-INF\

In the gatewayUserProfile.xml file, update the following text value:

```
<password>{V1.1}GD9klUFw8760HVaqeT4pkg==</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:

<PIA_HOME>\webserv\<Node_Server>\<APPLICATION>.ear\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:

- ...applications/peoplesoft/PSIGW.war\WEB-INF\classes (for Weblogic)
- <PIA_HOME>\webserv\<Node_Server>\<APPLICATION>.ear\PSIGW.war\WEB-INF\classes (for Websphere)

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.

Chapter 14

Securing Data with PeopleSoft Encryption Technology

This chapter provides overviews of data security, PeopleSoft Encryption Technology (PET), and the supported algorithms, and discusses how to:

- Load encryption libraries.
- Define algorithm chains.
- Define algorithm keysets.
- Define encryption profiles.
- Test encryption profiles.
- Invoke encryption profiles from PeopleCode.

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).

  ECB does not apply any special recombinations while encrypting. Plaintext blocks are simply encrypted with the key to produce blocks of ciphertext.

- Cipher Block Chaining (CBC).

  CBC takes a the previous block of ciphertext and XORs it with the current plaintext block before encrypting the plaintext.

- Cipher Feed Back (CFB).

  CFB produces ciphertext by XORing the plaintext with the result of a symmetric encryption operation on the previous ciphertext.

- Output Feed Back (OFB).

  OFB produces ciphertext by XORing plaintext blocks with a series of blocks resulting from repeated encryptions of the initialization vector.

There's a drawback with symmetric cryptography: The recipient of symmetrically encrypted ciphertext must posess 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.

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.

Encryption Algorithm	An encryption algorithm encrypts and decrypts data. As described in the previous sections of this chapter, PeopleSoft support symmetric and asymmetric encryption algorithms.
Encryption chain	An encryption chain is a sequence of encryption algorithms.
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, which you can license from PGP Corporation to enable its functionality.

   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 and *PS_HOME*\src\pspetpgp.

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 chapter.

Algorithm Library Glue Code

PeopleSoft delivers the glue code to interact with OpenSSL and PGP libraries. The location of the glue code is:

```
<PS_HOME>\src\pspet
```

The OpenSSL glue code has been tested on all supported PeopleSoft platforms with PKCS7 and 3DES. 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 *PeopleTools 8.52: System and Server Administration*, "Setting Application Server Domain Parameters."

- 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 *PeopleTools 8.52: PeopleSoft Process Scheduler*, "Using the PSADMIN Utility."

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 PET 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.

See Also

Chapter 14, "Securing Data with PeopleSoft Encryption Technology," Defining Encryption Profiles, page 304

PeopleTools 8.52: PeopleCode API Reference, "Crypt Class"

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 OS390. This algorithm functions the same as PSUnicodeToAscii on all platforms other than OS390.

Algorithm	Description
PSAsciiToUnicode_Generic_DEC	Convert ASCII text to Unicode **Note.** Use when performing cross-platform decryption where one platform is OS390. This algorithm functions the same as PSAsciiToUnicode on all platforms other than .OS390.

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 eight bytes (64 bits). It should be random but its secrecy isn't critical. For example: 0x0102030405060708. The longer the value the more secure the hash output.

Symmetric Encryption Algorithms

This table describes the supported OpenSSL symmetric encryption algorithms, which implement triple Data Encryption Standard (DES) encryption with various key sizes and modes.

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 feed back mode.
3des_ks112_cfb_decrypt	Decrypt data using a key size of 112 bits, in cipher feed back mode.
3des_ks112_ofb_encrypt	Encrypt data using a key size of 112 bits, in output feed back mode.
3des_ks112_ofb_decrypt	Decrypt data using a key size of 112 bits, in output feed back mode.
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.
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 feed back mode.
3des_ks168_cfb_decrypt	Decrypt data using a key size of 168 bits, in cipher feed back mode.
3des_ks168_ofb_encrypt	Encrypt data using a key size of 168 bits, in output feed back mode.
3des_ks168_ofb_decrypt	Decrypt data using a key size of 168 bits, in output feed back mode.

The following tables describe the supported OpenSSL symmetric encryption algorithms which implement Advanced Encryption Security (AES) encryption with various key sizes and modes. The information is divided by key size for ease of use.

The following table describes 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 feed back mode.
aes_ks128_cfb_encrypt	Encrypt data using a key size of 128 bits, in cipher feed back mode.
aes_ks128_ecb_decrypt	Decrypt data using a key size of 128 bits, in electronic code book mode.
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 feed back mode.
aes_ks128_ofb_encrypt	Encrypt data using a key size of 128 bits, in output feed back 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 electronic code book mode.
aes_ks192_cbc_encrypt	Encrypt data using a key size of 192 bits, in electronic code book mode.
aes_ks192_cfb_decrypt	Decrypt data using a key size of 192 bits, in cipher feed back mode.
aes_ks192_cfb_encrypt	Encrypt data using a key size of 192 bits, in cipher feed back 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 feed back mode.
aes_ks1928_ofb_encrypt	Encrypt data using a key size of 192 bits, in output feed back 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 electronic code book mode.
aes_ks256_cbc_encrypt	Encrypt data using a key size of 256 bits, in electronic code book mode.
aes_ks256_cfb_decrypt	Decrypt data using a key size of 256 bits, in cipher feed back mode.
aes_ks256_cfb_encrypt	Encrypt data using a key size of 256 bits, in cipher feed back mode.

Algorithm Name	Description
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 feed back mode.
aes_ks2568_ofb_encrypt	Encrypt data using a key size of 256 bits, in output feed back mode.

Most of these algorithms use the same two parameters:

- *IV* (Initialization Vector)

  This parameter isn't used by the listed ECB mode algorithms. Specify a hex encoded value to use 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 — eight bytes for triple DES. It should be random but its secrecy isn't critical. For example: *0x0102030405060708*

- *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.

Note. All algorithm chains that use 3 DES *encryption* algorithms must include either the base64_encode or PSHexEncode algorithm as a step in the encryption algorithm chain. All algorithm chains that use 3 DES *decryption* algorithms must include the corresponding base64_decode or PSHexDecode algorithm as a step in the decryption algorithm chain.

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 actual key value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*, depending on the algorithm.

- *SIGNERPKPASSPHRASE*

  Specify the pass phrase used to decrypt and unlock the signer's private key. This parameter's value is the actual pass phrase.

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. Its value should begin "-----BEGIN CERTIFICATE-----".

  Note. The API implementation of the rsa_sha1_verify algorithm requires that the Public Key be certified.

- *SIGNATURE*

  Specify, as a string, the hex encoded signature that's delivered with the input data or that's returned as the result of invoking a signing algorithm.

  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_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 actual certificate stored in the keyset database is an X.509 certificate. Its value should begin "-----BEGIN CERTIFICATE-----".

- *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 *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA*.

- *SIGNERPKPASSPHRASE*

  Specify the pass phrase used to decrypt and unlock the signer's private key. This parameter's value is the actual pass phrase.

Secure Messaging — pkcs7_signed_verify

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 value stored in the keyset database should begin with the line "-----BEGIN CERTIFICATE-----".

Secure Messaging — pkcs7_encrypted_encrypt

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. Its value should begin "-----BEGIN CERTIFICATE-----".

- *SYMMETRIC_ALGORITHM*

  Specify the name of the symmetric algorithm used for content encryption. This must be a symmetric encryption algorithm supported by an encryption plug-in.

  See Chapter 14, "Securing Data with PeopleSoft Encryption Technology," Symmetric Encryption Algorithms, page 288.

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 actual certificate in the keyset database should begin with the line "-----BEGIN CERTIFICATE-----"

- *RECIPIENTPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the recipient's private key in the PET keyset database. The actual key value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA.*

- *RECIPIENTPKPASSPHRASE*

  Specify the pass phrase used to decrypt and unlock the recipient's private key. This parameter's value is the actual pass phrase.

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 actual certificate stored in the keyset database is an X.509 certificate. Its value should begin "-----BEGIN CERTIFICATE-----".

- *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 *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA.*

- *SIGNERPKPASSPHRASE*

  Specify the pass phrase used to decrypt and unlock the signer's private key. This parameter's value is the actual pass phrase.

- *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. Its value should begin "-----BEGIN CERTIFICATE-----".

- *SYMMETRIC_ALGORITHM*

  Specify the name of the symmetric algorithm used for content encryption. This must be a symmetric encryption algorithm supported by an encryption plug-in.

  See Chapter 14, "Securing Data with PeopleSoft Encryption Technology," Symmetric Encryption Algorithms, page 288.

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 actual certificate stored in the keyset database is an X.509 certificate. Its value should begin "-----BEGIN CERTIFICATE-----".

- *RECIPIENTCERT*

  Specify, as a string, the keyset ID that represents the recipient's certificate in the PET keyset database. The actual certificate in the keyset database should begin with the line "-----BEGIN CERTIFICATE-----".

- *RECIPIENTPRIVATEKEY*

  Specify, as a string, the keyset ID that represents the recipient's private key in the PET keyset database. The actual key value in the keyset database should begin "-----BEGIN *xxx* PRIVATE KEY-----" where *xxx* is either *RSA* or *DSA.*

- *RECIPIENTPKPASSPHRASE*

  Specify the pass phrase used to decrypt and unlock the recipient's private key. This parameter's value is the actual pass phrase.

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.

This algorithm has one parameter: , which is

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.

See Also

Chapter 14, "Securing Data with PeopleSoft Encryption Technology," Loading Encryption Libraries, page 297

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.

Cross Platform Algorithm Chain Considerations

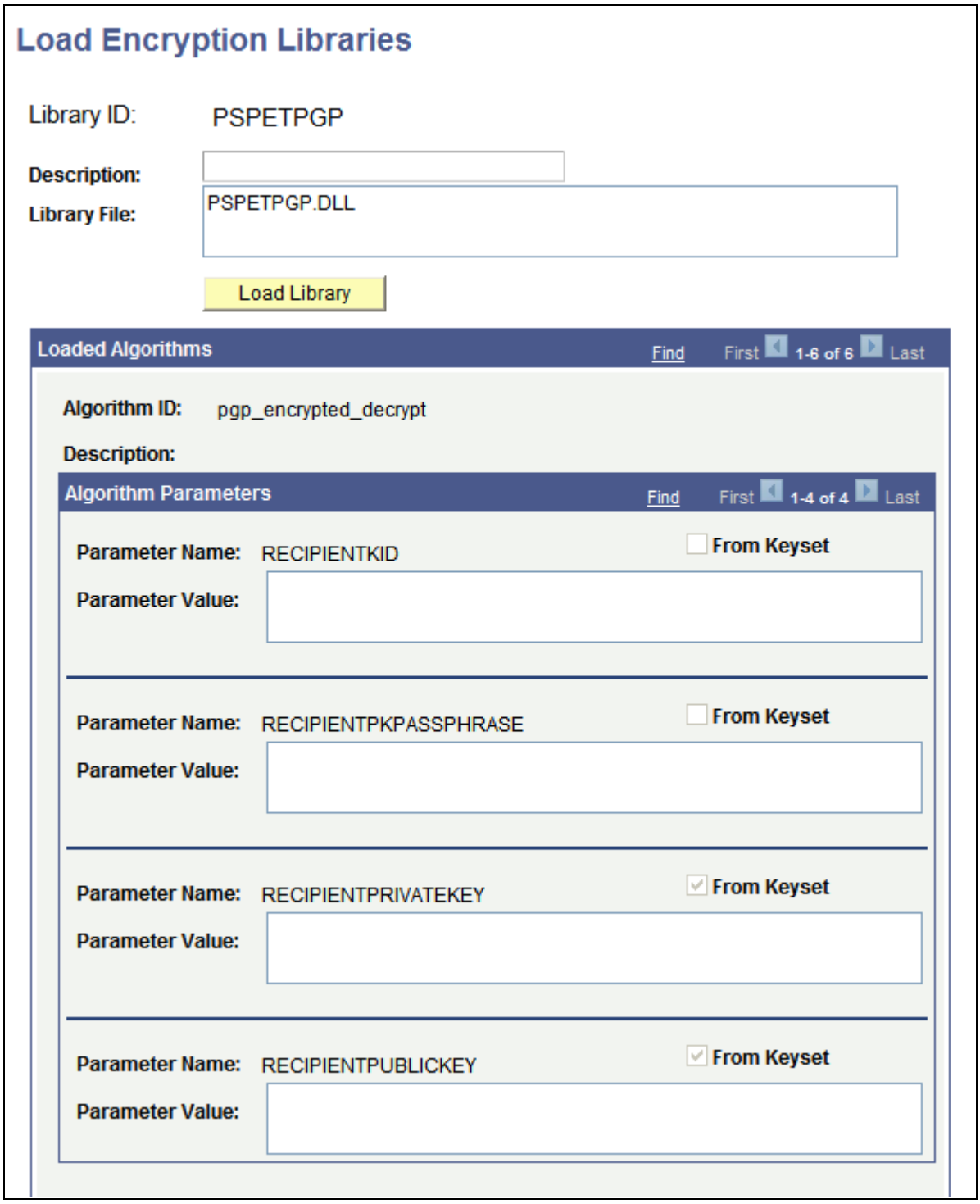
When encrypting and decrypting data across multiple platforms where OS390 is one of two or more platforms, the PSUnicodeToAscii_Generic_ENC algorithm must be the first algorithm in the encrypting algorithm chain. Conversely, PSAsciiToUnicode_Generic_DEC must be the last algorithm in the decrypting algorithm chain.

Note. If all participating encrypting and decrypting systems are on the OS390 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 OS390 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 OS390 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).



Load Encryption Libraries page

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.

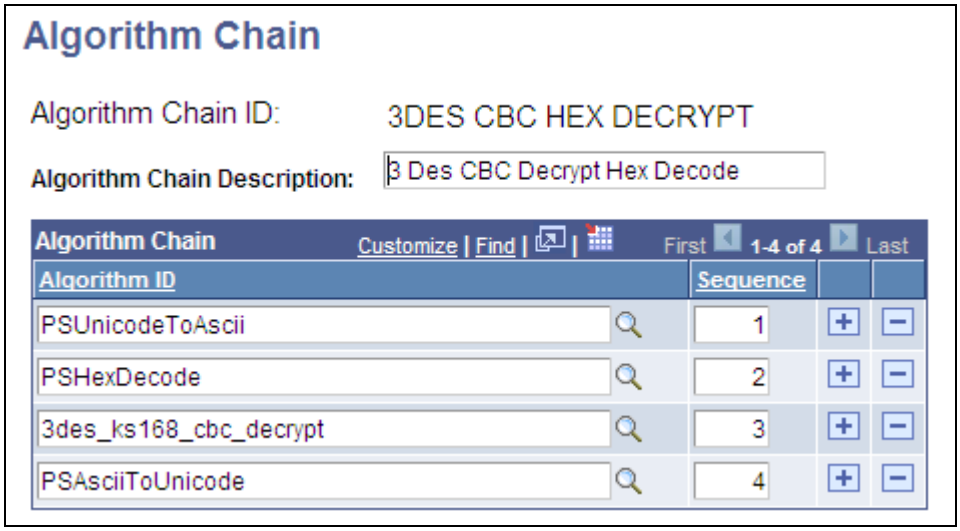
Important! If the library you specify fails to load, you must sign out of your application, then shut down and restart the application server before signing back in.

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, Algorithm Chain).



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, and the last algorithm must be PSAsciiToUnicode.

See Chapter 14, "Securing Data with PeopleSoft Encryption Technology," Cross Platform Algorithm Chain Considerations, page 297.

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
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

Algorithm Chain	*Algorithms*
PGP_ENCRYPTED	PSUnicodeToAscii pgp_encrypted_encrypt PSAsciiToUnicode
PGP_DECRYPTED	PSUnicodeToAscii pgp_encrypted_decrypt PSAsciiToUnicode
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

Defining Algorithm Keysets

Access the Algorithm Keyset page (PeopleTools, Security, Encryption, Algorithm Keyset).

Algorithm Keyset page

Specify 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.

Keyset ID Enter a name for the key value in the current row. Each row must have a unique keyset ID for this algorithm.

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.

The value that you enter has a length that depends on the keysize of the cipher. For triple DES with keysize 112, this is 16 bytes. For a keysize of 168, this is 24 bytes. This value should be represented in hex notation.

You must generate the key value that you enter here. 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.

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.

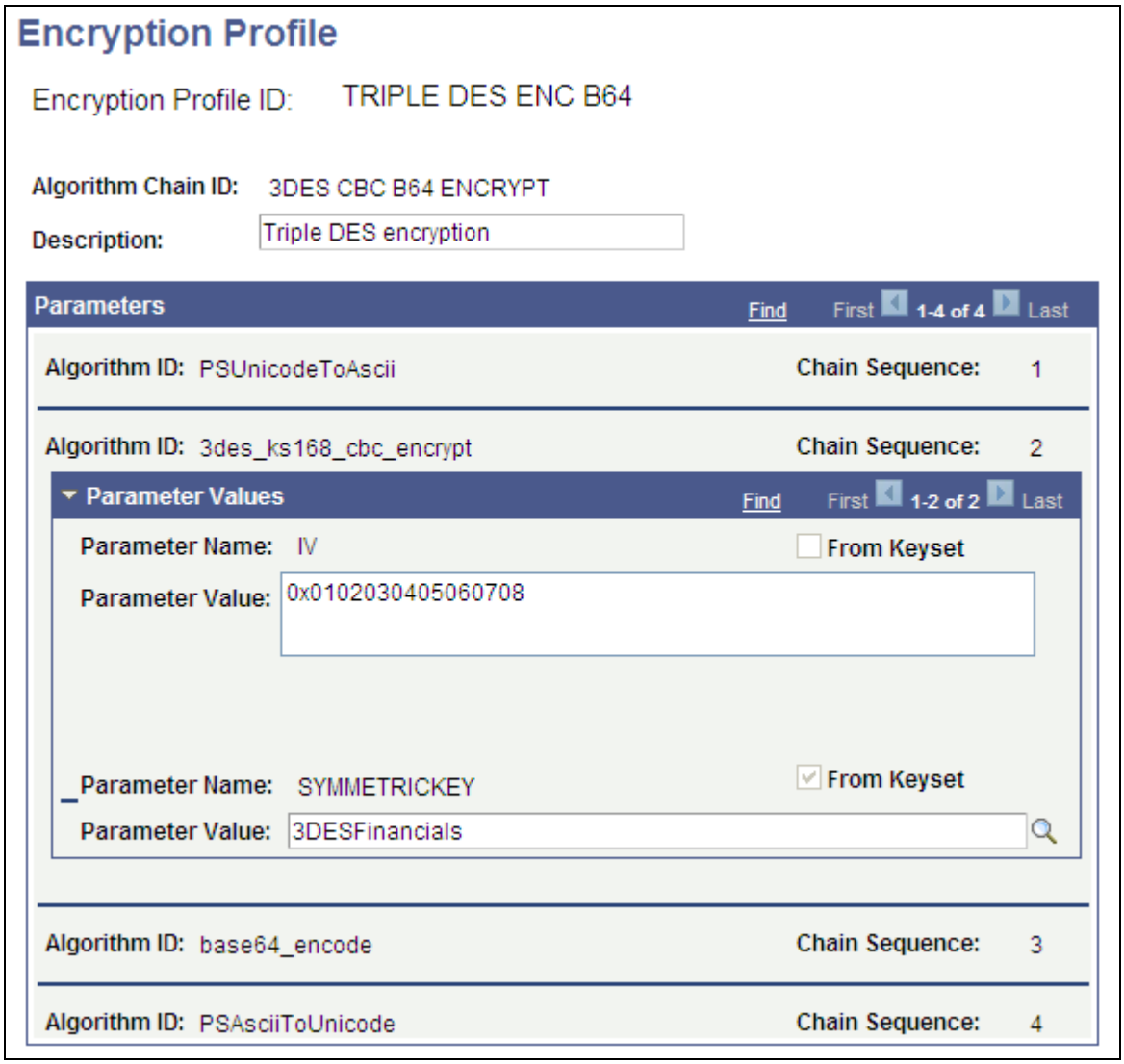
Note. The key value that you enter here is stored in the PET keyset table using a combination of the algorithm ID and the keyset ID as its identifier. Because this combination is unique for each algorithm, you can create identically defined keyset rows for multiple algorithms.

See Also

http://www.openssl.org/

Defining Encryption Profiles

Access the Encryption Profile page (PeopleTools, Security, Encryption, Encryption Profile).



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 Profile.).

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 Profile).

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;
```

See Also

PeopleTools 8.52: PeopleCode API Reference, "Crypt Class"

Using PeopleCode Encryption Methods

Two PeopleCode methods are provided by PET as part of PCI compliance which requires keys to be stored in encrypted format:

- `EncryptPETKey()`
- `DecryptPETKey()`

These methods are called and applied to keys wherever applicable when using PeopleSoft encryption technology. These functions are generally transparent to the application developer when using the PeopleTools PET pages. However, if you create applications which provide their own pages to display keys, you must use these functions to encrypt and decrypt keys to show them on application pages.

The two affected record.fields are:

- PSCRYPTKEYSET.CRYPT_KEY.
- PSCRYPTPRFLPRM.CRYPT_PARAM_VAL.

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 execute PeopleCode—Crystal Reports, 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.

See Also

PeopleTools 8.52: Application Engine, "Managing Application Engine Programs," Running Application Engine Programs

Chapter 15

Implementing Query Security

This chapter discusses how to:

- Define query profiles.
- Build query access group trees.
- Work with query trees.
- Define row-level security and query security records.

Note. You perform these setup tasks using the Query Access Manager, Application Designer, and permission lists. After you define Query Access Group trees, you provide user access using the Query tab in Permission Lists.

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: groupsandrecords.
- 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

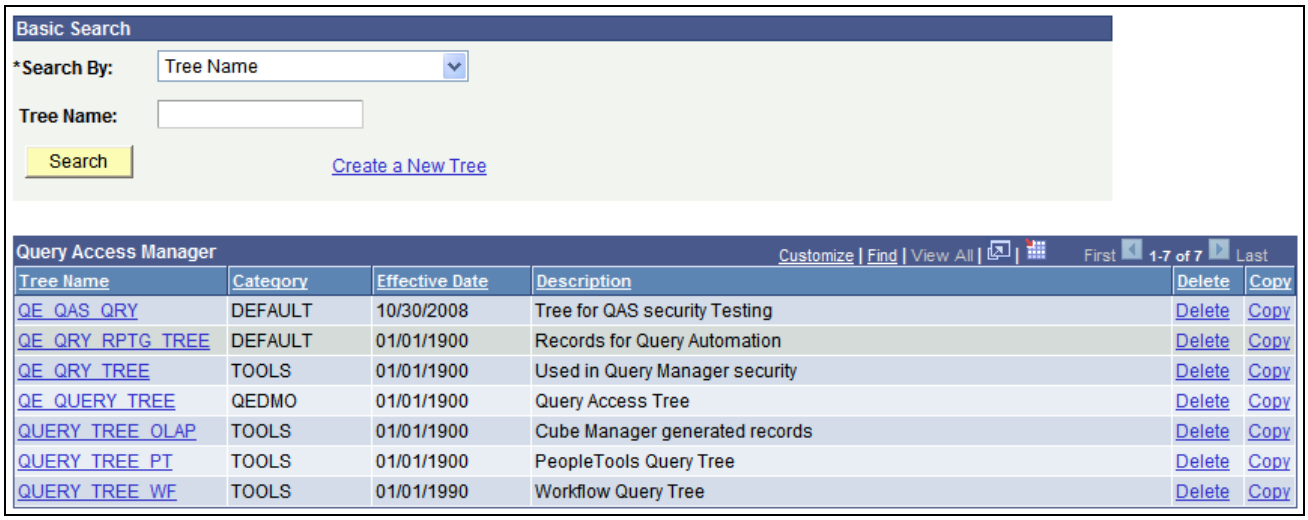
Regarding requirements, consider the following points:

- 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.).



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.

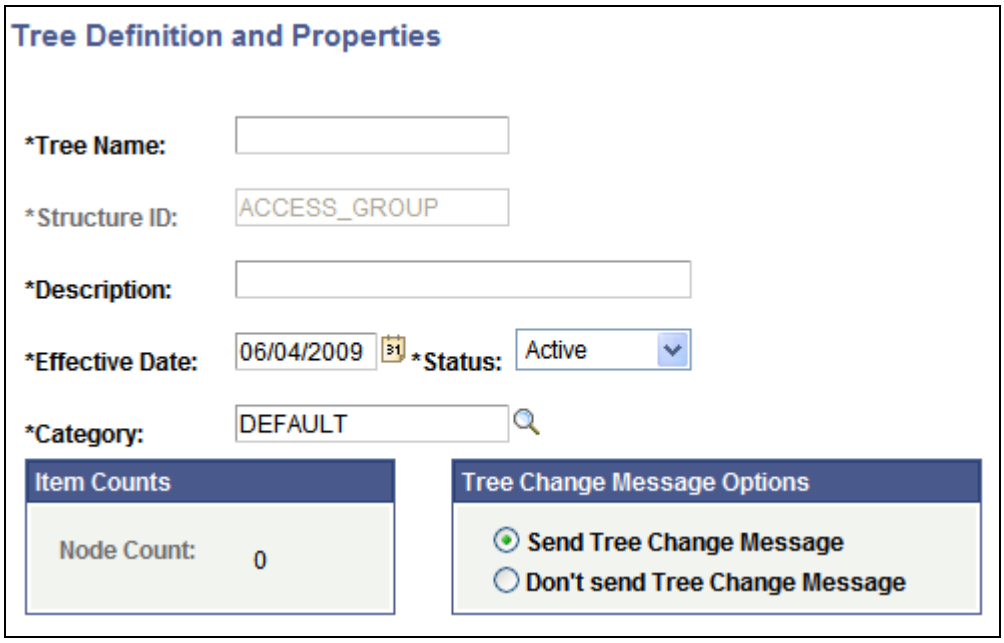
   The search page also enables you to delete or copy a tree. Click the Delete or Copy link to perform the desired task. If you click Delete, the system prompts you to confirm the action, and if you click Copy, the system displays the Copy Tree page where you can enter the name for the copied tree.

   Some of the trees in the grid may appear with no 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.



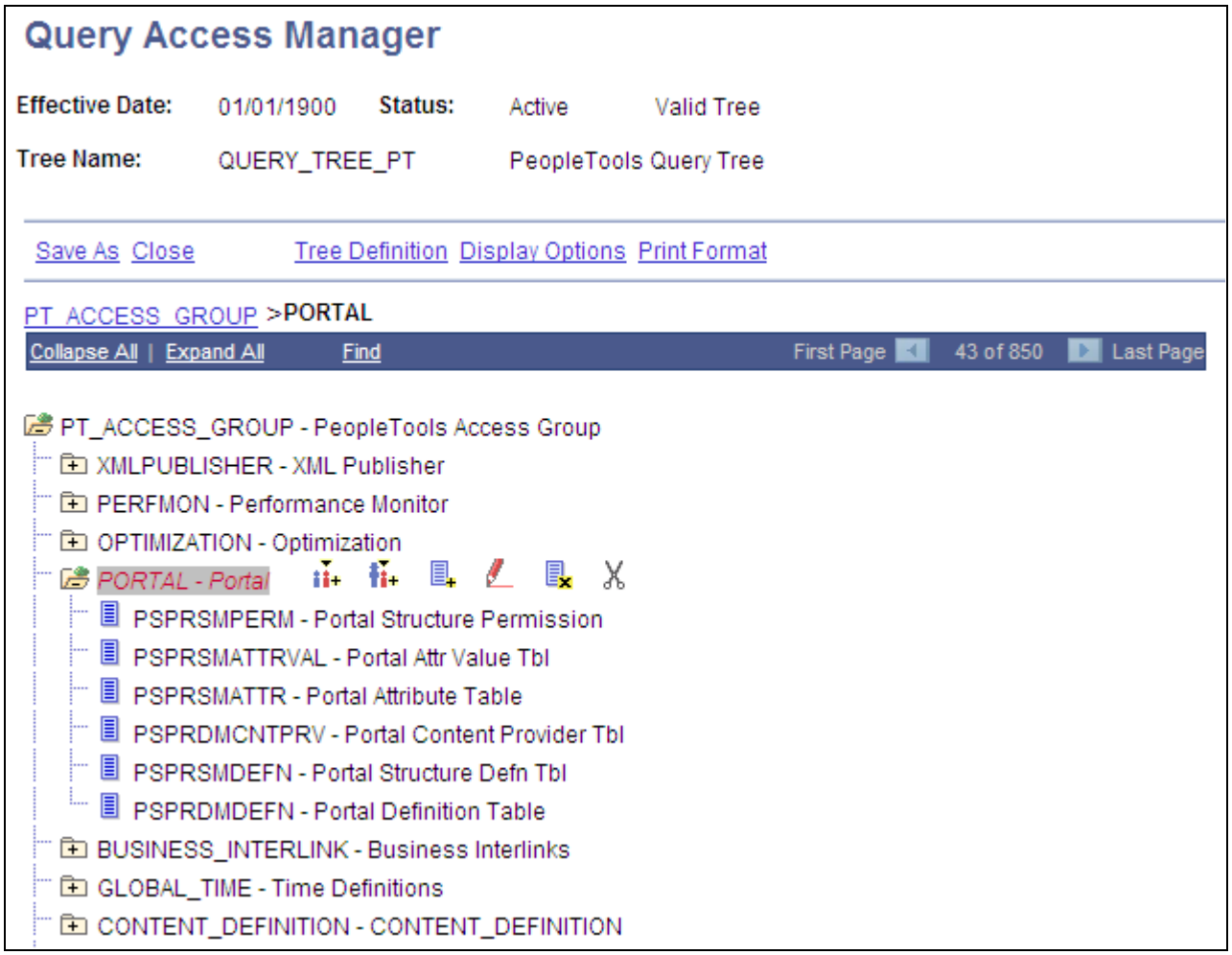
Tree Definition and Properties page

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_.
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.



Query Access Manager page

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.
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.

	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.
	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 execute 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. Select PeopleTools, Application Designer to open the Application Designer, and open the record on which you want to apply row-level security.
2. With the record definition open in the 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 16

Implementing Definition Security

This chapter provides an overview of definition security and discusses how to:

- Work with definition groups.
- View definition groups.
- Add and remove definitions.
- Assign definition groups to permission lists.
- Enable display only mode.
- View definition access by user and permission list.

Understanding Definition Security

This section discusses:

- Definition security.
- Definition groups and permission lists.
- Definition security rules.

Definition Security

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:

- Creating definition groups.
- Linking definition groups to predefined permission lists.

Definition security leverages the permission lists created in PeopleTools Security to restrict access to individual PeopleTools database definitions created using a PeopleTools designer utility, such as PeopleSoft Application Designer or PeopleSoft Tree Manager. Definition types include all of the definitions that appear in the following table. Most definition types are created in PeopleSoft Application Designer.

Definition Type	*Associated Designer Tool*
Activities	PeopleSoft Application Designer
Application Engine Programs	PeopleSoft Application Designer
Application Packages	PeopleSoft Application Designer
Approval Rule Sets	PeopleSoft Application Designer
Business Interlinks	PeopleSoft Application Designer
Business Processes	PeopleSoft Application Designer
Components	PeopleSoft Application Designer
Component Interfaces	PeopleSoft Application Designer
Fields	PeopleSoft Application Designer
File Layouts	PeopleSoft Application Designer
HTML	PeopleSoft Application Designer
Images	PeopleSoft Application Designer
Menus	PeopleSoft Application Designer
Messages	PeopleSoft Application Designer
Mobile Pages **Important!** PeopleSoft Mobile Agent is a deprecated product. These features exist for backward compatibility only.	PeopleSoft Application Designer
Pages	PeopleSoft Application Designer
Analytic Types	PeopleSoft Application Designer

Definition Type	*Associated Designer Tool*
Projects	PeopleSoft Application Designer
Queries	PeopleSoft Query
Records	PeopleSoft Application Designer
SQL	PeopleSoft Application Designer
Style Sheets	PeopleSoft Application Designer
Tree Structures	PeopleSoft Tree Manager
Trees	PeopleSoft Tree Manager
Translate Tables	PeopleSoft Application Designer

Note. You can restrict access to an entire definition type, such as records or pages, using the PeopleTools page in Security. 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.

Definition Security settings also works at the field level. To change a field on a record, you must be authorized to update all record definitions that contain the field. For example, to update or rename the EMPLID field on any record definition, you must have access to every record definition that contains the EMPLID field. If you are denied access to the ABSENCE_HIST record definition, which contains EMPLID, you won't be able to modify any field attributes of EMPLID on any other record that contains the field. This ensures the integrity of your system. In a fast-paced development environment, if PeopleTools definitions are not well secured, problems may result.

Before you start using Definition Security, it's a good idea to define the definition security needs of your users. Consider these types of questions:

- Should all developers have access to all PeopleTools definitions?
- Should payroll developers have access only to payroll definitions?
- Who will be allowed to access PeopleSoft Application Designer?

Definition Groups and Permission Lists

Use Definition Security to define definition groups and link them to permission lists that you 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.

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 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.

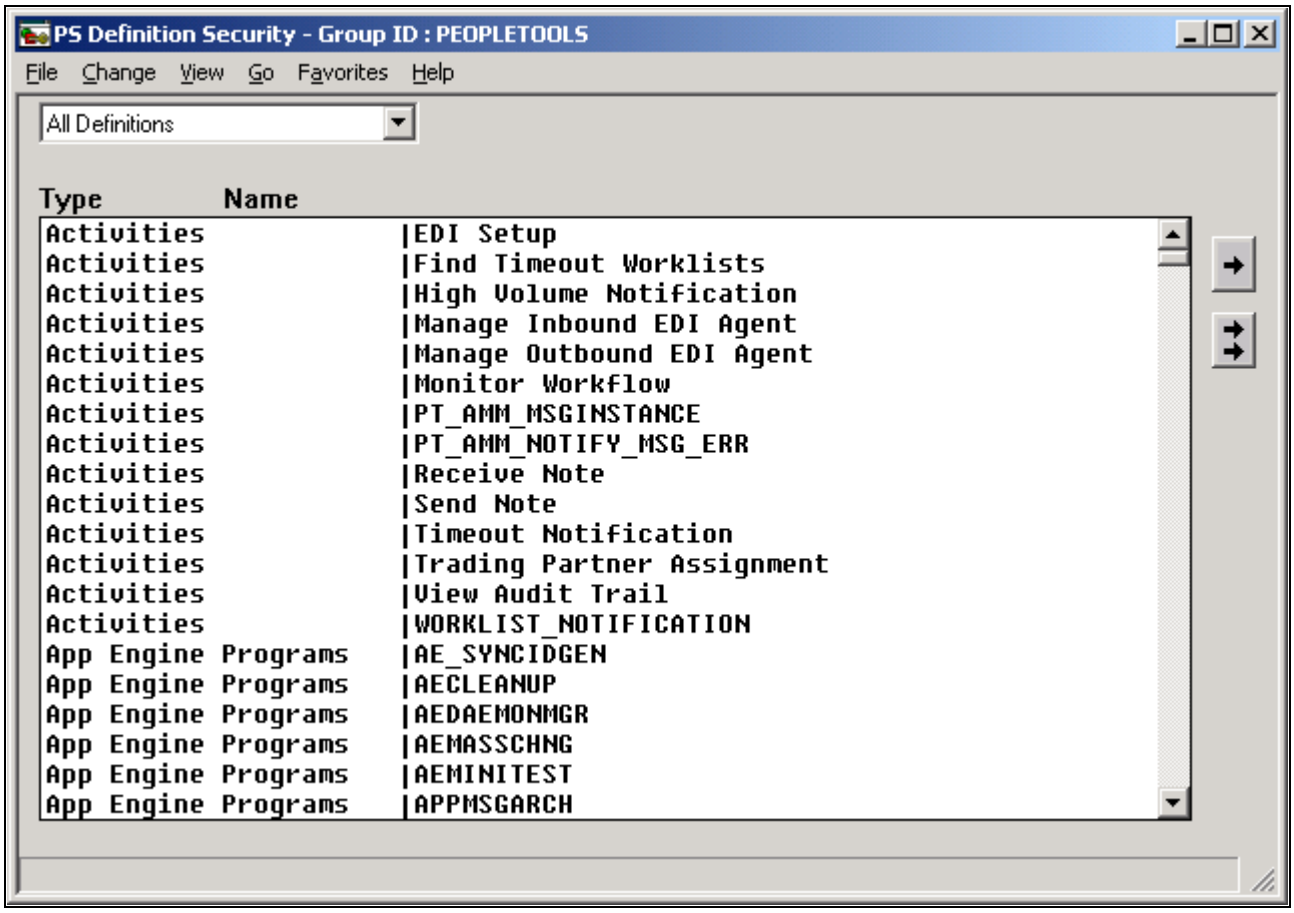
If the definition passes these system checks, the user is allowed to access and update it—unless it's a PeopleSoft Application Designer definition, in which case several other security checks are performed first. PeopleSoft Application Designer definitions are also controlled by the PeopleTools in permission lists.

Important! A user gets definition security permissions through the primary permission list, not through roles.

Working With Definition Groups

PeopleSoft Definition Security is a Microsoft Windows-based application that you can access from PeopleSoft Application Designer.

Access the PS Definition Security (Go, Definition Security).



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

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

This section 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

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 OK

   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

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

To view reports that detail *specific secured definitions* by user or by permission list, access the Common Queries - Definition Security Queries page (PeopleTools, Security, Common Queries-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 Chapter 3, "Setting Up Permission Lists," Running Permission List Queries, page 67.

See Chapter 5, "Administering User Profiles," Running User ID Queries, page 112.

Chapter 17

Managing PeopleSoft Personalizations

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

- Work with personalization options.
- Define personalization options.
- Work with category groups.
- Work with categories.
- Work with locale-based personalizations
- Add personalizations to permission lists.
- Create custom personalization options.
- Work with the My Personalizations interface.

Understanding Personalizations

PeopleSoft software offers a variety of options that enable end users, especially power users, to complete business transactions in a more efficient manner. 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 select, customize, and define personalizations using the Personalization pages.

To access the Personalization pages, select PeopleTools, Personalization.

Personalizations are grouped in three levels of categories to aid in development, organization, and deployment:

- The first level is the Option Category level.

  This level divides personalizations between functional area, such as PeopleTools personalizations and HCM personalizations. Also, there is a category for custom personalizations, which are those personalizations you develop and deploy in addition to the delivered personalizations.

- The second level is the Category Groups, which represent individual products within a Category Level.

  For example, within the PeopleTools Category Level some Category Groups are Application Designer, Process Scheduler, Security, and so on. Or, within the HCM Category Level one Category Group could be Payroll.

- The third level is the Personalization Categories themselves.

  This is the level that the end user sees. A category represents a product feature, such as navigation or system messages. A category contains a set of related personalizations.

After you have selected the personalizations for your site, you assign them to a user or role, using the Personalizations page of the permission lists component in PeopleTools Security. The Personalizations page enables the security administrator to assign role-based personalizations and enable user control for selected personalization options, if needed.

End users can view and modify their available personalization options from the My Personalization component (USER_SELF_PERSONAL).

The following sections provide more details on defining, customizing, and deploying PeopleSoft personalizations.

Working with Personalization Options

Before you begin defining and deploying personalization options, you need to be familiar with the personalization option categories delivered with PeopleSoft software, and the pages used to view and modify them. This section discusses:

- Navigation personalizations.
- Regional settings.
- General options.
- System messages.
- Internally controlled options.
- Pages used to define and modify personalizations.

Note. PeopleSoft Mobile applications use the standard personalizations.

PeopleSoft Mobile Agent is a deprecated product. Mobile personalization features exist for backward compatibility only.

Understanding Navigation Personalizations

The following table presents the delivered navigation personalization options.

Note. PTPT1000 is a delivered permission list that you can use as a starting point for a user permission list. The column shows whether PTPT1000 allows a user to set the option.

Option Code	*Description*	*Default Value*	*PTPT1000*
ACEGRDCOLS (Max columns for Show all Columns)	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	No
ACEGRDROWS (Max rows for View All)	Specify the maximum number of rows that are displayed in an analytic grid when the user selects View All.	100	No
ADBTN (Tab over Add/Del Buttons (+/-))	Enable tabbing over the Add (+) and Delete (-) buttons within grids and scrolls.	No	No
ANAVSORT (Navigation menu sort)	Enable top navigation sort.	Yes	No
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	Yes
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	No
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	No
BGLYPH (Tab over related content glyph)	Enable tabbing over the red glyphs, which indicate a field-level related content contextual menu.	Yes	No

Option Code	*Description*	*Default Value*	*PTPT1000*
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	No
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	No
BMOPOPUP (Enable mouse over pop-up windows)	Enable mouse over pop up pages, which appear over the main page when you hover over certain text fields.	Yes	No
BNEWWIN (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	Yes
CALBTN (Tab over Calendar Button)	Enable tabbing over the calendar controls, which appear as buttons on the page.	No	Yes
EXPERT	Enable expert entry.	Yes	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	No
GRDTAB (Tab over Grid Tabs)	Enable tabbing over the tabs or headings within grids.	No	Yes
HDRICN (Tab over Header Icons)	Enable tabbing over header icons, which appear at the top of each page and include Home,Add To Favorites, and Sign Out.	No	Yes

Option Code	*Description*	*Default Value*	*PTPT1000*
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	Yes
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	Yes
NONPS (Tab over Browser Elements)	Restrict tabbing to include only the PeopleSoft elements of the page, and tab over non-PeopleSoft elements.	No	Yes
PGLNK (Tab over Page Links)	Enable tabbing over links to other pages in the same component.	No	Yes
POPUP (Tab over Related Page Links)	Enable tabbing over the pop-up menu icon that opens a page of associated menu items.	No	Yes
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	Yes
TYPEAHD	Enable autocomplete on prompt edit boxes. The system performs a prompt lookup as you type, suggesting appropriate values.	Yes	Yes

Understanding Regional Settings

The following table presents the delivered regional settings.

Note. PTPT1000 is a delivered permission list that you can use as a starting point for a user permission list. The column shows whether PTPT1000 allows a user to modify the option.

Option Code	Description	Default Value	PTPT1000
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	Yes
AUTOGREGCAL	Specify whether the system automatically recognizes and converts date values to Gregorian calendar dates.	Yes	Yes
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	Yes
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.	.	No
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	Yes

Option Code	Description	Default Value	PTPT1000
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.	/	No
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	Yes
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	Yes
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	Yes
TMSP (Time Separator)	Specify the time separator character to separate hours, minutes, and seconds, such as (:) or (.). You can enter any single character.	:	No
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.	,	No

Option Code	Description	Default Value	PTPT1000
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	Yes
WEEKFIRSTDAY	(Locale-based) Specify which day begins the week.	Sunday	Yes

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 Personalizations component, 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.

See Also

Chapter 17, "Managing PeopleSoft Personalizations," Working with Locale-Based Personalizations, page 348

Understanding General Options

The following table presents the delivered general options.

Note. PTPT1000 is a delivered permission list that you can use as a starting point for a user permission list. The column shows whether PTPT1000 allows a user to modify the option.

Option Code	*Description*	*Default Value*	*PTPT1000*
ACCESS (Accessibility Features)	Specify accessibility features. This option provides better support for assistive technologies. Select from the following values: • *Use accessible layout mode* **—** For use with screen readers. Page elements (fields, links, buttons, and so on) are presented in linear fashion to assistive software. • *Use standard layout mode* **—** Supports assistive technologies without altering the page design. • *Accessibility features off* **—** This disables accessibility features.	Accessibility features off	Yes
EXCEL97 (Excel 97 grid download)	Indicate that you want to use the character set defined in the user language instead of the default UTF-8 character set when you download a page grid to Microsoft Excel 97. Enter *Y* to enable, or *N* to disable this option. **Note.** This option is recommended only for non-English speaking users who use Microsoft Excel 97. It isn't recommended for Excel in Microsoft Office 2000 and later.	N	Yes

Option Code	Description	Default Value	PTPT1000
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 *PeopleTools 8.52: PeopleSoft Application Designer Developer's Guide*, "Creating Component Definitions," Setting Component Properties.	Yes	No
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	Yes
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	Yes
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	Yes

See Also

PeopleTools 8.52: PeopleTools Portal Technologies, "Using Portal Caching Features"

Understanding System Messages

System 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.

Note. PTPT1000 is a delivered permission list that you can use as a starting point for a user permission list. The column shows whether PTPT1000 allows a user to modify the option.

Option Code	Description	Default Setting	PTPT1000
SCNFRM (Save Confirmation)	Display a brief message confirming each save action.	Yes	No
SWARN (Save Warning)	Display a warning when the user makes a change and attempts to leave the transaction without saving.	Yes	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 Personalizations, even if you assign them to a permission list.

Instead of accessing these options in My Personalizations, 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:

AUTOJOIN This option appears as the Enable Auto Join check box on the Query Preferences page. It's selected by default.

NAMESTYLE This option appears as the Name Style setting on the Query Preferences page. Its default value is *Name and Description.*

DICTIONARY This option is not used in the current release.

SORTBY This option is not used in the current release.

See *PeopleTools 8.52: PeopleSoft Query*, "Creating and Running Simple Queries," Specifying Query Preferences.

Portal Preference

The following personalization option is used by PeopleTools portal technologies:

PAGEHDRCACHE

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:

TMLINES 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 *PeopleTools 8.52: PeopleSoft Tree Manager*, "Using PeopleSoft Tree Manager," Setting Display Options.

Pages Used to Define and Modify Personalizations

Page Name	*Definition Name*	*Navigation*	*Usage*
Define Personalizations	PSUSEROPTNDEFN	PeopleTools, Personalization, Personalization Options	View, modify, or add personalization option definitions and their formats. View or modify the explanations that end users see in the My Personalization interface.
Category Groups	USEROPTN_CAT_GRP	PeopleTools, Personalization, Category Groups	View or modify the grouping of options for administrative and ownership purposes.

Page Name	*Definition Name*	*Navigation*	*Usage*
Category	USEROPTN_CAT	PeopleTools, Personalization, Categories	View or modify the categories in which personalization options are grouped for end users.
Locale Definition	PSLOCALEDEFN	PeopleTools, Personalization, Locales	Control the locales for which you can specify defaults.
Locale Defaults	PSLOCALEOPTNDFLTS	PeopleTools, Personalizations, Locale Defaults	Specify defaults for locales appearing on the Locale Definition page.
My Personalizations	PSUSERPRSNLCAT	My Personalizations	End users access this page to view and modify personalizations

Defining Personalization Options

This section provides an overview of the Search page and discusses how to:

- Use the Definition tab.
- Use the Format tab.
- Use the Explanation tab.

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 the Search Page

To access the personalization definition pages, select PeopleTools, Personalization, Personalization Options. On the search page, you have the option 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).

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.

You add and modify the delivered personalization options using the Define Personalizations component.

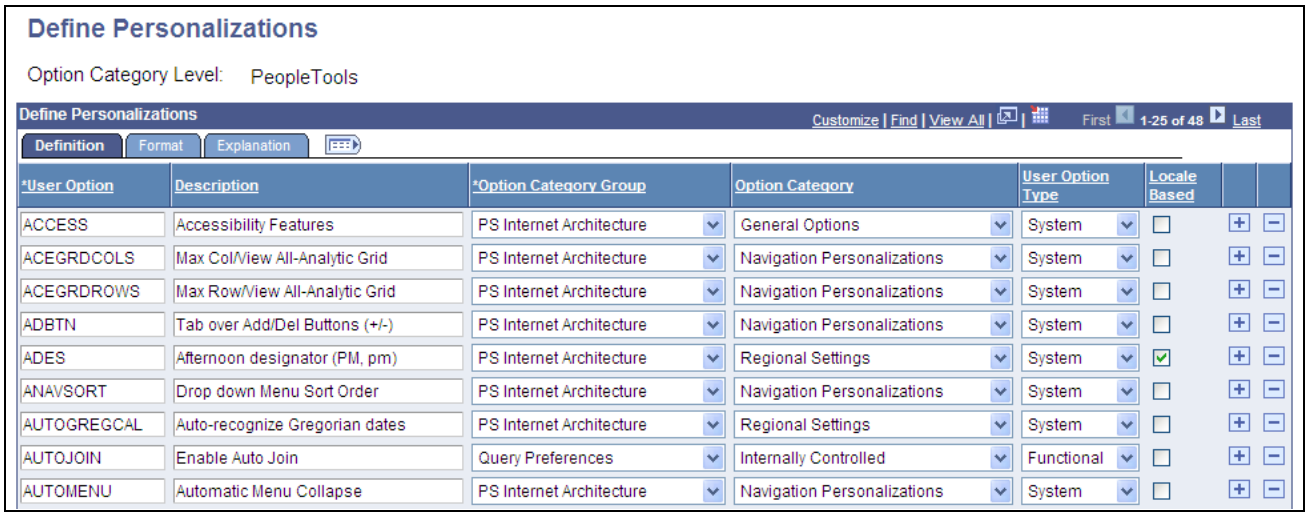
Access the Define Personalization page (PeopleTools, Personalization, Personalization Options). This grid contains the following tabs:

- Definition
- Format
- Explanation

You use this grid to view and to modify the personalizations within the Category Level you selected on the search page.

Using the Definition Tab

Click the Definition tab.



Define Personalizations - Definition tab

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 Personalizations 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 option you select determines the button the end user clicks 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 options:

- *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 they authority.

Locale Based Indicates that the option derives the default values based on the Locale of the browser.

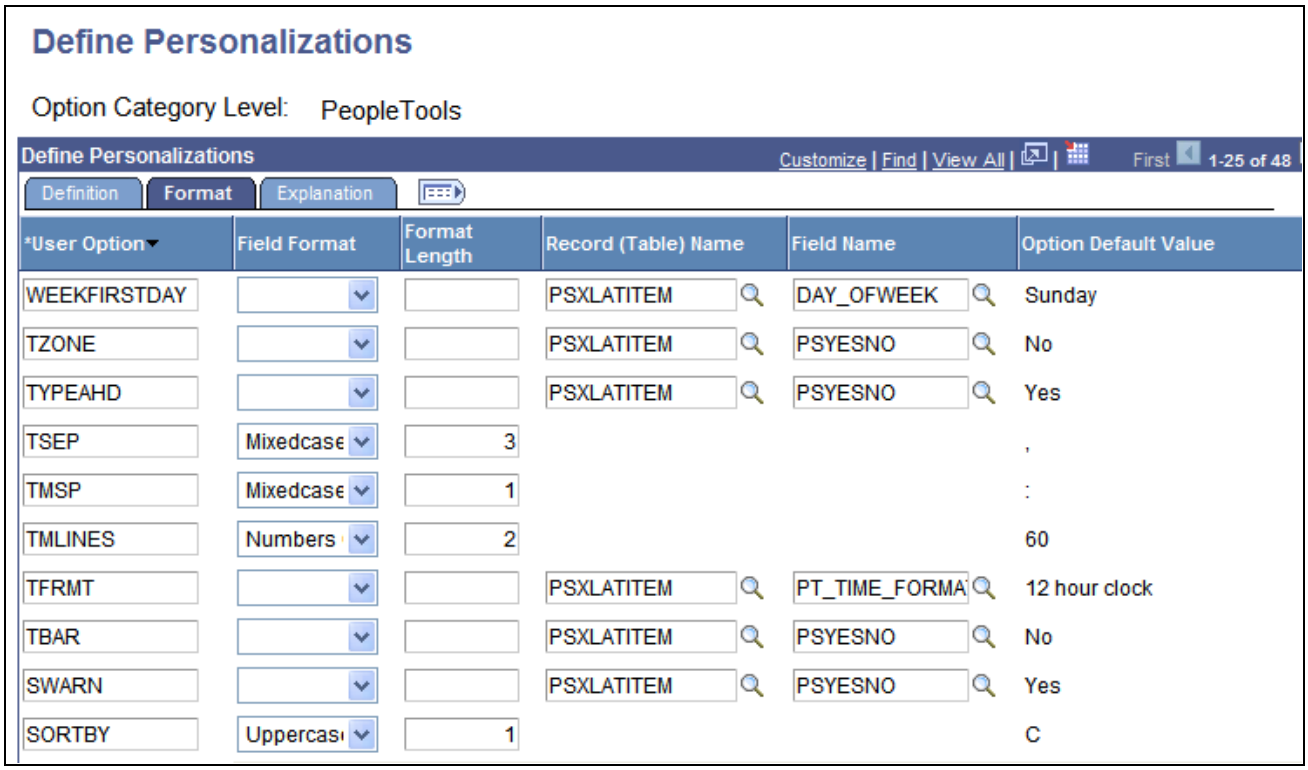
To add an option, use the insert row (+) button. To delete an option, use the delete row (-)button.

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 Personalizations interface, end users see only options that posses 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 Format Tab

Click the Format tab.



Define Personalizations - Format tab

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. This value is set through the Set Option Default Value.

Set Option Default Value This is a link to the secondary page used to set Option Default Values.

Set Option Default Value

The following items appear on the Set Option Default Value page:

Option Category Level Shows the high-level category to which the option belongs, such as PeopleTools or HCM.

User Option Shows the code associated with the option.

Description Shows the description of the option.

Current Default Value Displays the current default value

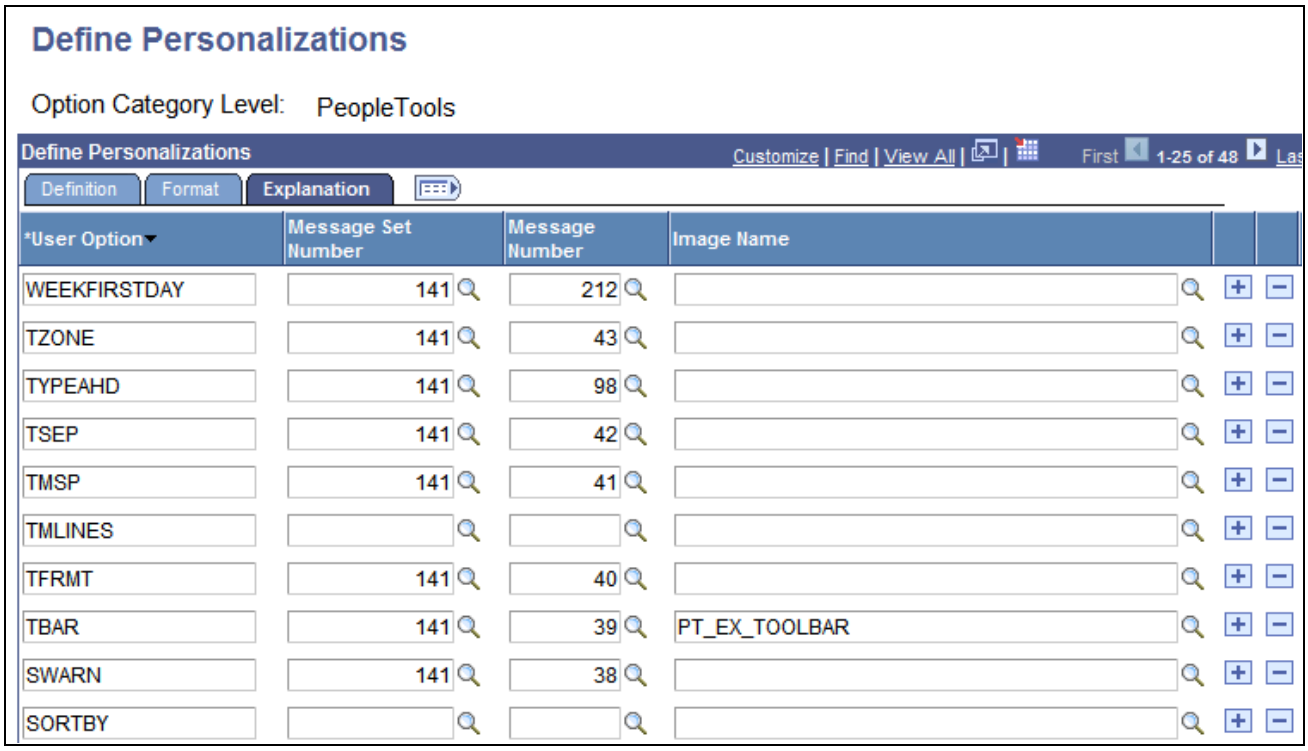
Option Default Value Select the appropriate value from the drop down list, or add the appropriate option manually. For options that derive default values from a prompt table, the system displays a drop down list. Otherwise, the system displays an edit box.

Using the Explanation Tab

The Explanation tab enables you to reference the message text and the image (if needed) that the end user sees after clicking the Explain button in the My Personalizations interface.

If you are adding a custom personalization, you'll need to create the message in the message catalog and create the image (if needed).

Click the Explanation tab.



Define Personalizations - Explanation tab

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.

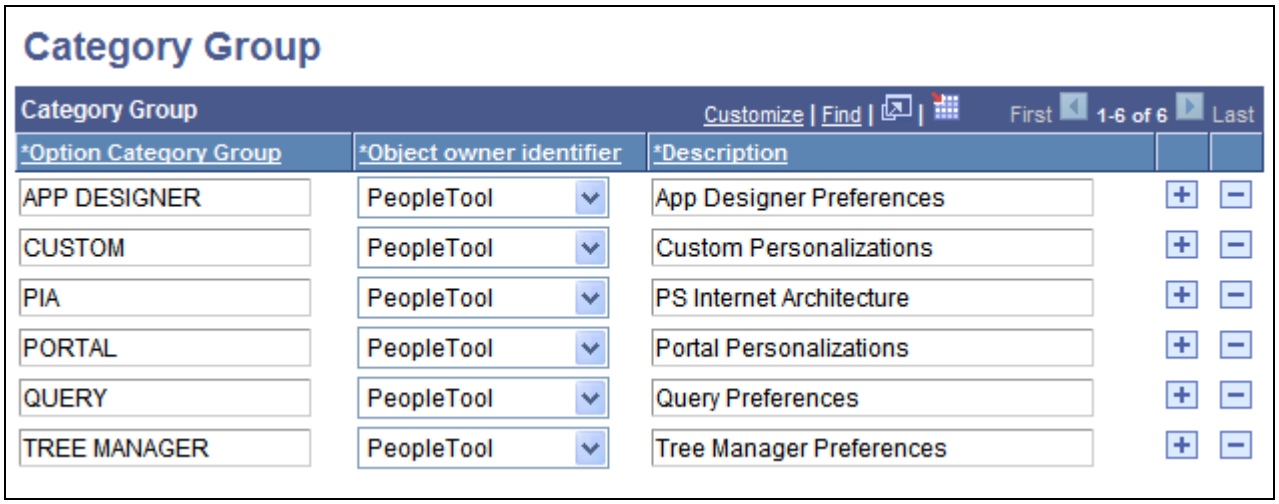
Image Name Points to the image that the system presents to the end user to provide clarification and context for the personalization. For example, for the "Tab over add button" option, the image of the add button is included so the user can recognize the object.

Working with 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.

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, Category Groups).



Category Group page

Option Category Group Displays the name of the category group.

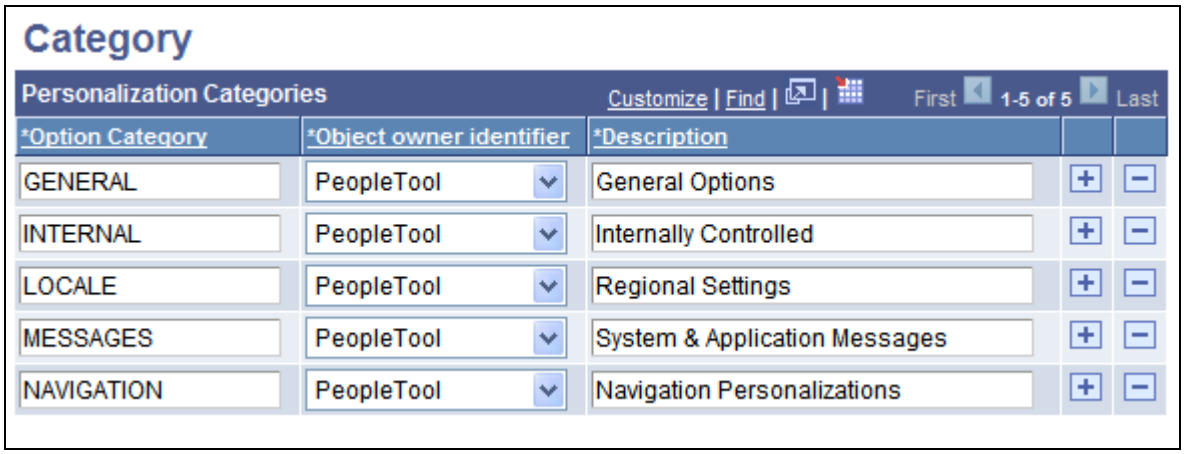
Object owner identifier Displays the name of the group responsible for the maintenance of the category group.

Description Provides a description of the category group for identification purposes. This field has a 30-character limit.

Working with Categories

Categories are the way that you group and present personalization options to your end users. For example, for the Navigation option category, the end user sees the description (Navigation Personalizations) on the My Personalizations page. When the end user clicks the adjacent Personalize Options button, they access the options you have grouped in the Navigation category.

Access the Category page (PeopleTools, Personalization, Categories).



Category page

Option Category Shows the name of the category in which options are displayed on the My Personalizations 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 Personalization page. If you add custom categories make sure the text is meaningful for end users.

Working with Locale-Based 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.
- Locale Defaults.

The system derives the locale information based on the locale specified in the browser. PeopleSoft software provides these pages populated with the codes that represent the current browser locales.

This topic is discussed in more detail in the *PeopleTools 8.52: Global Technologies PeopleBook.*

See Also

PeopleTools 8.52: Global Technology, "Controlling International Preferences," Setting Up Locale-Based Formatting for the PeopleSoft Pure Internet Architecture

Adding 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.

See Also

Chapter 3, "Setting Up Permission Lists," Setting Personalization Permissions, page 59

Creating Custom Personalization Options

Creating custom personalization options involve the following steps:

1. Define the option using the Define Personalization interface.

   See Chapter 17, "Managing PeopleSoft Personalizations," Defining Personalization Options, page 341.

2. Implement the behavior using PeopleCode personalization options (discussed in the following section).

   See Chapter 17, "Managing PeopleSoft Personalizations," Working with the My Personalizations Interface, page 350.

3. To enable users to control the personalization, you need to make the option accessible on the appropriate permission list through PeopleTools Security.

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.

See Also

PeopleTools 8.52: PeopleCode Language Reference, "PeopleCode Built-in Functions"

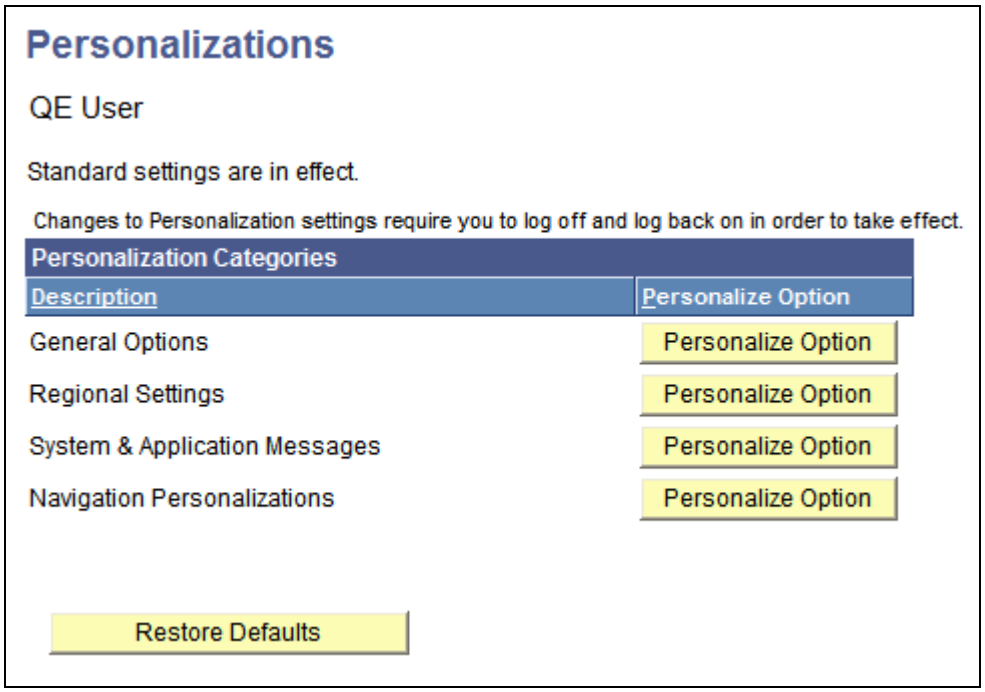
Working with the My Personalizations Interface

This section discusses how to:

- Use the Personalizations page.
- Set personalize options.
- Use the Personalization Explanation page.
- Modify a personalization option.

Using the Personalizations Page

Select My Personalizations to access the Personalizations page.



Personalizations page

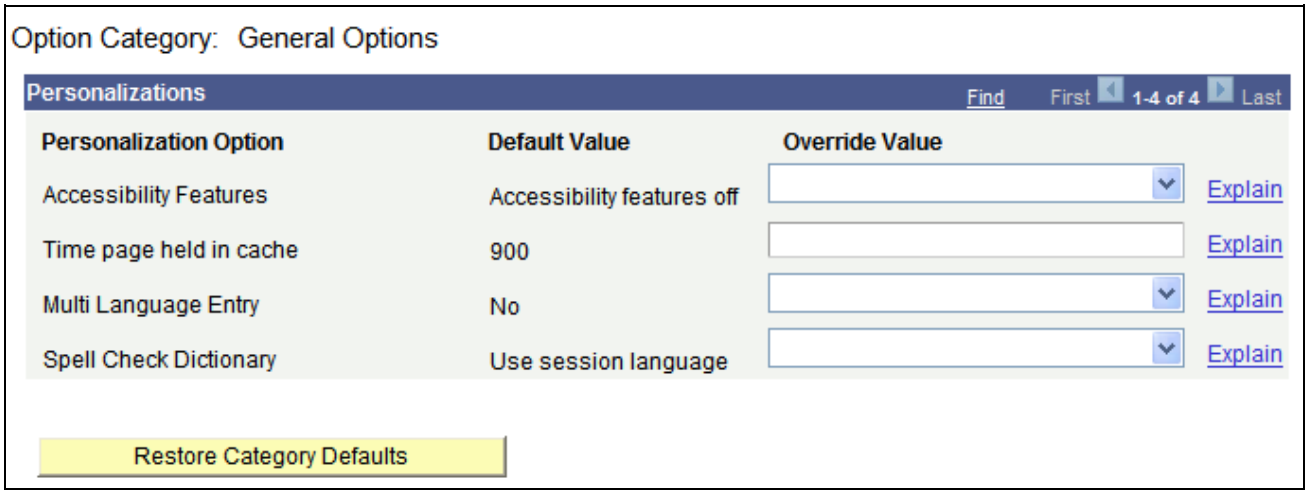
Description The description column contains a brief description for identifying a particular category of personalization options.

Personalize Options Click this button to view and modify the options within a category.

Restore Defaults Click this button to restore the default values for all options in each personalization category. Defaults refer to the initial values that your system administrator has set for each available option—before you modified the option. So, you only use this feature if you have modified one or more personalization option and you want to revert to the initial settings.

Setting Personalize Options

Access the My Personalizations - Personalize Options page (From the homepage, click My Personalizations, then click the Personalize Options button in the General Options row).



My Personalizations - Personalize Options page

Option Category Shows the description of the category of personalizations. This helps you to make sure that you have the correct category open.

Personalization Option This column lists all of the personalization options available for you to modify. The text that appears in the list is a brief description of the option. For more information on the option, click the Explain link.

Default Value Refers to the initial settings that your administrator has specified for the option. If you do not modify the default value, the option assumes the value provided by the system administrator.

Override Value Enter any custom value you want to assign to the personalization option. To override a default setting means to use the new value in place of the default setting.

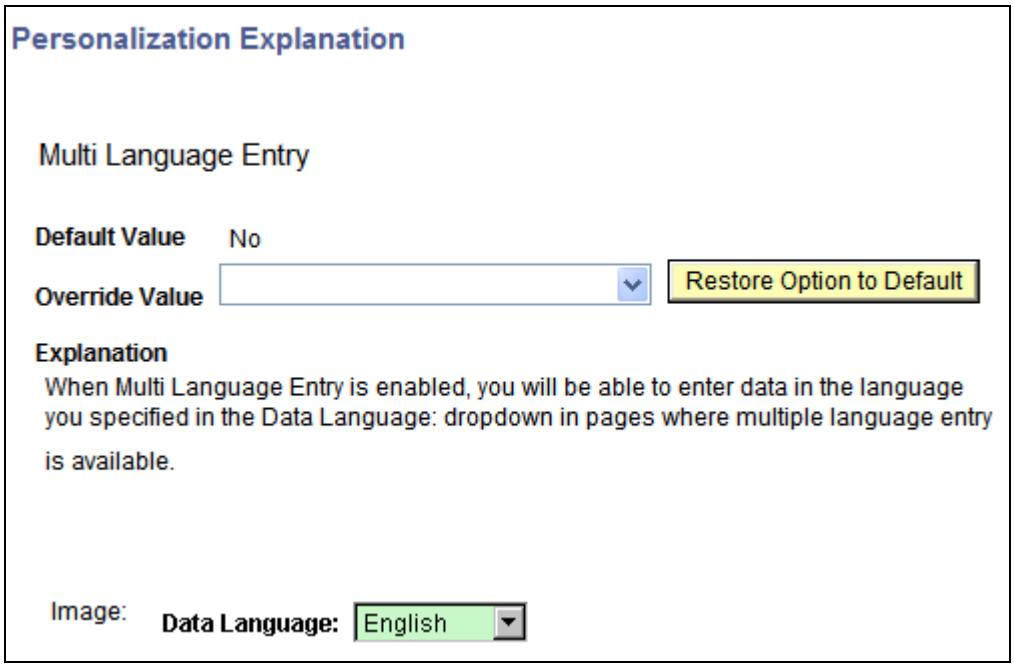
Explain Click this link to view more information on what the personalization option provides. See the following section for more information on the Explanation page.

Restore Category Defaults Returns all modified options to the default values. This button applies only to the current category, as in the category you have open.

OK/Cancel After you have made any modifications, click OK so that the system records your changes. If you do not want your changes recorded click Cancel. If you have not made any changes and just viewed the options, you can use either button to return to the Personalizations page.

Using the Personalization Explanation Page

Access the Personalization Explanation page (click the Explain link on the Option Category page).



Personalization Explanation page

Personalization Name The name of the individual personalization appears at the top of this page so that you can make sure you are viewing or modifying the appropriate option.

Default Value Shows the value that your system administrator has set as the default value for an option. The personalization assumes the default value unless you override it.

Override Value Overrides the default value. For example, if the default value for an option is No, you can override the default value to be Yes.

Restore Option to Default Enables you to change any option value that you've modified to assume the original default value specified by your system administrator.

Explanation Contains the description of what the personalization option provides when activated. For longer descriptions, use the scroll bar to view. This box is read-only.

Image	In many cases, especially with the Navigation options, an image appears to provide further clarification as to a specific control or item that the option affects. For example, on the explanation page for the Tab Over Toolbar option, an image of the toolbar appears in the image section to show exactly the area on the page that the personalization affects.
OK/Cancel	Returns you to the current Option Category page. If you've made changes to the personalization option that you want to keep, click OK. If you do not want to keep the changes you have made, click Cancel. If you have made no changes, use either button.

Modifying a Personalization Option

The following procedure describes the steps you need to complete to modify a personalization option.

To modify a personalization option:

1. Select My Personalizations from the portal menu.
2. On the Personalizations page, click the Personalize Options button adjacent to the category of personalization options you want to modify.
3. In the Personalization Option list, locate the option you want to modify.
4. In the corresponding Override Value edit box specify the appropriate override value.

   Depending on the option, you will see one of the following controls.

   - A drop-down list box.

     Select the appropriate option from the drop-down list.

   - An edit box.

     Manually enter an override value.
5. Click OK.

   This saves the change to the system.
6. Sign out and then sign in again to view your changes.

Appendix A

Enabling Kerberos Authentication in a Microsoft Active Directory Environment

This appendix describes how to configure Microsoft Active Directory for Kerberos authentication. To enable Kerberos authentication in Microsoft Active Directory, you must:

- Add a server user for Kerberos single signon.
- Create a keytab file and map user credentials to a service principal.

Adding a Server User for Kerberos Single Signon

To add a server user:

1. On a Windows 2003 domain controller, select Start, Control Panels, Administrative Tools, Active Directory Users and Computers.
2. From the menu bar, select Action, New, User.
3. Enter values in the Full name and User logon name fields. You should use your own internal naming conventions. For example,

   Full name: *Kerberos Server*

   User logon name: *krbsrv*

   Note. The First name,Last name, and Initials fields are not important, but you must specify the Full name and User logon name. Kerberos authentication uses the User logon name only.
4. Click Next.

5. Use this table to set the password and check box values:

Field	***Value***
Password	*<your_complex_password>*
User must change password at next logon	Cleared
User cannot change password	Selected
Password never expires	Selected
Account is disabled	Cleared

6. Click Next and then click Finish.

Generating the Keytab File and Mapping the Service Principal Name

To generate the keytab file and map the service principal name:

Note. These steps assume that the server user is *krbsrv* and the domain is *example.com.*

1. Open a command window by selecting Start, Run and then entering *cmd* in the Open field.

2. In the command window, enter

```
C:\>ktpass -princ HTTP/www.example.com@EXAMPLE.COM -mapuser krbsrv@example.com -⇒
crypto RC4-HMAC-NT -ptype KRB5_NT_PRINCIPAL -pass krbPass! -out c:\temp⇒
\krb5.keytab
```

   This calls the ktpass utility with these parameters:

Parameter	*Value*	*Description*
`-princ`	`HTTP/www.example.com@EXAMPLE.C⇒ OM`	Specifies the service principal name in the form user@realm.
`-mapuser`	`krbsrv@example`	Maps the name of the Kerberos principal specified by the princ parameter to the specified local user name.
`-crypto`	`RC4-HMAC-NT`	Sets the encryption type to use.
`-ptype`	`5_NT_PRINCIPAL`	Sets the principal type to Kerberos 5 for Microsoft Windows
`-pass`	`*`	Causes the utility to prompt you for a password
`-out`	`c:\temp\krb5.keytab`	Specifies the name and location of the Kerberos version 5 .keytab file to generate.

3. When prompted for the password, enter some value. This resets the password and does not have to match the one used when the user was created.

   Note. Make sure that the password meets domain security requirements or the utility fails.

4. Verify that the command window output is similar to the following text. If so, the mapping is complete and the keytab file *krb5.keytab* is in the C:\temp directory.

```
Key created.
Output keytab to c:\temp\krb5.keytab:
Keytab version: 0x502
keysize 83 HTTP/www.example.com@EXAMPLE.COM

ptype 1 (KRB5_NT_PRINCIPAL) vno 15 etype 0x17 (RC4-HMAC)
keylength 16 (0xdd74540caa4a230af2ed75558a37995d)
```

Service Principal Name Considerations

The SPN can include any possible URL. Valid SPNs for the example.com domain include:

- HTTP/mail.example.com@EXAMPLE.COM
- HTTP/other.domain.net@EXAMPLE.COM
- HTTP/localhost@EXAMPLE.COM

- HTTP/192.168.1.100@EXAMPLE.COM

Browsers request the client-to-server tickets based on the URL that the user enters. If a page, for example *http://www.example.com/page.html?param=value,* requests authentication, then the browser requests a client-to-server ticket for an SPN that is based on the website domain name: *HTTP/www.example.com@EXAMPLE.COM.*

Although *http://192.168.1.100* and *http://www.example.com* might refer to the same physical machine, an authentication request from http://192.168.1.100 requests a client-to-server ticket for HTTP/192.168.1.100@EXAMPLE.COM only, not HTTP/mail.example.com@EXAMPLE.COM. In other words, an SPN mapping which uses the server DNS name is not applicable when the client visits the server site using its IP address.

In addition, Microsoft Active Directory will not proceed with the client-to-server ticket exchange unless the server machine is either in the same domain as the directory server or in a trusted domain. For example if *http://mail.example.com* references a machine on the directory server domain (or a domain it trusts), then the Kerberos ticket exchange proceeds. If *http://mail.example.com* is not in the same domain or a trusted domain of the directory server, then the exchange does *not* proceed, regardless of the site's URL.

See Also

Consult your Microsoft Active Directory documentation for more information.

http://technet.microsoft.com/en-us/library/cc753771(WS.10).aspx

Appendix B

Enabling Kerberos Authentication in the Browser

This appendix describes how to enable Kerberos authentication in:

- Microsoft Internet Explorer.
- Mozilla Firefox.

Enabling Kerberos Authentication in Internet Explorer

To enable Kerberos authentication in Internet Explorer:

1. Open Internet Explorer and select Tools, Internet Options. Then, select the Security tab.
2. In the zones display, select Local intranet and then, click the Sites button.
3. Select the check boxes that apply to the PeopleSoft site.
4. If these settings do not meet your needs, then click the Advanced button and add the site specifically. After you add the site, click the Closebutton.
5. On the Local Intranet dialog box, click the OK button.
6. On the Internet Options dialog box, select the Advanced tab. Then, scroll down to the Security settings. Select the Enable Integrated Windows Authentication check box.
7. Click the OK button and then, restart the browser so that the settings take effect.

Enabling Kerberos Authentication in Firefox

Firefox does not automatically perform Kerberos authentication against any sites. You must manually add sites to a trusted sites list.

To enable Kerberos authentication in Firefox:

1. Open Firefox and enter *about:config* in the address bar. Dismiss any warnings that appear.
2. In the Filter field, enter *negotiate.*

3. Double-click the network.negotiate-auth.trusted-uris preference.

   This preference lists the trusted sites for Kerberos authentication.

4. In the dialog box, enter the PeopleSoft domain, such as *example.com.*

5. Click the OK button.

   The domain that you just entered in the network.negotiate-auth.trusted-uris should now appear in Value column. The setting takes effect immediately; you do not have to restart Firefox.

Index