Security Guide

Oracle Utilities Smart Grid Gateway
Version 2.1.0.3 (OUAF 4.2.0.3)

E63092-01

May 2015

ORACLE

Security Guide, Oracle Utilities Smart Grid Gateway, Version 2.1.0.3 (OUAF 4.2.0.3)

E63092-01

Table of Contents

Preface

Welcome to Oracle Utilities Smart Grid Gateway Security Guide. This guide describes how you can configure security for Oracle Utilities Smart Grid Gateway by using the default features.

This preface contains these topics:

Audience

Oracle Utilities Smart Grid Gateway Security Guide is intended for product administrators, security administrators, application developers, and others tasked with performing the following operations securely and efficiently:

- Designing and implementing security policies to protect the data of an organization, users, and applications from accidental, inappropriate, or unauthorized actions
- Creating and enforcing policies and practices of auditing and accountability for inappropriate or unauthorized actions
- Creating, maintaining, and terminating user accounts, passwords, roles, and privileges
- Developing interfaces that provide desired services securely in a variety of computational models, leveraging product and directory services to maximize both efficiency and ease of use

To use this document, you need a basic understanding of how the product works, and basic familiarity with the security aspects of the Oracle WebLogic (or IBM WebSphere) and Database security.

Documentation Accessibility

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

Access to Oracle Support

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

Related Documents

For more security-related information, see these Oracle resources:

- *Oracle Utilities Smart Grid Gateway Server Administration Guide*

- *Oracle Utilities Smart Grid Gateway Batch Server Administration Guide*
- *Oracle Utilities Smart Grid Gateway DBA Guide*
- *Oracle Database Security Guide*
- *Oracle Utilities Application Framework Advanced Security (Doc Id: 1375615.1)*
- *Technical Best Practices for Oracle Utilities Application Framework Based Products (Doc Id: 560367.1)*
- *Batch Best Practices for Oracle Utilities Application Framework based products (Doc Id: 836362.1)*
- *Production Environment Configuration Guidelines (Doc Id: 1068958.1)*
- *Database Vault Integration (Doc Id: 1290700.1)*
- *Oracle Identity Management Suite Integration with Oracle Utilities Application Framework based products (Doc Id: 1375600.1)*

These documents are available from My Oracle Support and/or Oracle Delivery Cloud.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

Note: Screen images in this document are for illustrative purposes only.

Note: Menu options in this document assume the use of Alphabetic sorting. If alternatives are used, then adjust the advice accordingly.

What's New in Security?

The security features and enhancements described in this section comprise the overall effort to provide superior access control, privacy, and accountability with this release of Oracle Utilities Smart Grid Gateway.

The following sections describe new security features of Oracle Utilities Smart Grid Gateway (Version 2.1.0.3) and provide pointers to additional information.

Authentication User Support

With the advent of Single Sign On (SSO) solutions and greater flexibility of security credentials in the marketplace, the existing eight (8) character User ID limit is a restriction.

To address this, an alias has been added, known as Login ID, on the User object. The Login ID supports user identifiers up to 256 characters in length. The Login ID is now used for authentication, where the preexisting User ID (still restricted to 8 characters for backward compatibility) is now used for authorization.

The Login Id allows the following to be implemented:

- The Login ID can default to the same value as the User ID for backward compatibility.
- The Login ID can be changed at any time, by an authorized user, without affecting audit information.
- The Login ID can be provisioned from a third party user provisioning engine like Oracle Identity Manager or similar.
- Customers can choose to auto generate User IDs from Login IDs using class extensions or use the third party provisioning engine to generate User IDs (as in Oracle Identity Manager).

Security Cache Refresh

The web application server keeps a cache of user/application service security information in order to enhance performance. In previous releases, the lifetime of an entry in the cache is until midnight, system time or 30 minutes, whichever is shorter. The default cache time is now much shorter, changing it to 1 minute, so that changes to a user's security information propagate more rapidly.

Business Service Security enhanced

In previous releases Business Services inherited their security attributes from the Application Service they were based upon. In this release, it is possible to explicitly define a specific application service as the basis for the security attributes on the Business Service definition.

Session Identification on Database

For performance reasons, database connections are pooled and shared across all active users.

In this situation identification of the active user on a connection is typically hard to track making database level tracking hard to perform. In this release the user attributes and other session parameters are available to the user on the database connection. A DBA Administrator can now recognize the active user and other session parameters via the `v$session` view. This information is also available for use by other Oracle Database technologies such as Real Application Clusters and Audit Vault.

Enhanced Application Service Page

In past release, determining the access rights for user groups and users to a particular service involved using a number of queries and navigation. In this release a new portal has been developed to streamline the management of authorization permissions between Application Services, User Groups and Users.

Audit On Inquiry

In past releases the internal Audit facility can be used to track changes to records through the online web application. In this release it is now possible to configure individual zones to track whenever particular records (or groups of records) are read or inquired upon to provide additional levels of security.

Encrypted Passwords in Configuration

Administration passwords are stored in various configuration files for use by utilities and the runtime of the product. These passwords are now encrypted using AES-256 level encryption.

Removal of Session Cookie

In past releases a session cookie was issued by the J2EE container (Oracle WebLogic or IBM WebSphere) whenever a user logged on to pass credentials to servers. This has been replaced with native J2EE calls to provide a more secure connection.

Addressed Security Vulnerabilities

With every release of the product, the security facilities within the product are enhanced to address new and specific identified potential security vulnerabilities from customer security assessments and internal assessments.

These enhancements include addressing cross scripting vulnerabilities, SQL injection and internal information display. The valid statements that are available in scripting is now controlled by a product whitelist. Additionally the comments in the generated screens can now be stripped to prevent display of internal information that is typically used by developers but not necessary for runtime use.

User Enablement

One of the limitations of the user object was that you cannot delete a user that has been used.

This is enforced to maintain audit information that is captured by the product. A User Enable flag has been added to the user object, thus allowing a user to be logically deleted but retain the information for that user to satisfy audit purposes. The user record is still retained in the product for audit purposes.

Implementations can use this flag to mark user records as inactive for user retrenchments or temporary contractor workers. From an authentication purposes only active users will be allowed to authenticate successfully or execute any object within the product.

JAAS Support

Java Authentication and Authorization Service (JAAS) has be implemented to control security across application tiers and for securing JMX interfaces to the product for online, web service and batch channels.

Enhanced Web Services Security

Over the last few releases of the product, the number of standards in respect to Web Service Security supported has increased. In this release it possible to use a wider range of security standards as well as externalized security offered by Oracle Web Services Manager to offer additional levels of security. Security has been expanded to support WS-Policy, X.509, SAML and other approaches to extend the HTTP Basic Digest Security and WS-Security approaches already supported.

Enhanced Patch Permissions

Some implementations prefer running Database Upgrade or Patch utilities as non-schema owner and non-system account for site security policy reasons. A new feature configuration has been added enable sites to run these utilities without using system account.

To use this facility, the utility scripts will use a preconfigured Feature Configuration that stores the schema username/password and then connect to the database using these credentials. As with other passwords in the product the database password is saved in encrypted format in the framework database tables and it is be displayed on the screen as masked.

Identity Management Suite Support

Oracle Identity Management Suite is a comprehensive set of utilities to control and manage security across an enterprise. The product has a set of adapters and configuration settings that allow the following components of Identity Management Suite to be used as part of a solution:

- **Oracle Identity Manager** – The provisioning of user records and managing password rules in a centralized fashion.
- **Oracle Access Manager** – Providing Single Sign On and session management capability.
- **Oracle Internet Manager** – Provide an LDAP based security store.
- **Oracle Adaptive Access Manager** – Providing a mechanism for detecting fraudulent

activity using patterns.

- **Oracle Virtual Directory** – Providing a standardized virtualized security interface for all security needs.

Refer to *Oracle Identity Management Suite Integration with Oracle Utilities Application Framework based products* (Doc Id: 1375600.1) available from My Oracle Support for more details of this integration.

Database Vault Support

The database installation of the product now includes a set of optional additional installation files to enable a default Database Vault solution for the product. This restricts system and DBA accounts to appropriate access to product data. By default, the system users in Oracle (i.e. the SYS and SYSTEM user), DBA users (SPLADM/CISADM) and any user with the SYSDBA role has full access to the application data. Whilst, this is generally acceptable for most sites, some sites have considered this a potential security issue. These particular sites wish to use an option for the Oracle database called Database Vault that allows additional security to be defined to restrict system and DBA users to their allocated tasks. A new default set of configuration files for Database Vault allows restriction of Data Manipulation Language (DML) access to the product data for system and DBA users whilst allowing appropriate access to Data Definition Language (DDL) and Data Control Language (DCL). Customers wishing to use this facility must license and enable the Database Vault option on the Oracle Database prior to enabling the Database Vault product solution provided.

Refer to *Database Vault Integration* (Doc Id: 1290700.1) available from My Oracle Support for more details of this integration.

Audit Vault Support

With the introduction of enhanced database connection tagging, it is now possible to use Audit Vault as an alternative for inbuilt auditing. This will allow customers to centralize audit information external to the product using Oracle's Audit Vault product and allow auditors to use the specialized compliance reporting and tracking facilities in Audit Vault to deliver higher levels of compliance and monitoring. This solution will be a combination of configuration on both the product and Audit Vault side.

Debug Mode Security Control

The product allows an end user to enter debug mode to display additional support information to logs to be used by developers and support personnel. As long as the end user has access to the system they can use debug mode to assist support personnel.

The use of debug mode is now restricted to authorized users only using the **F1DEBUG** security group.

Cache Management Security Control

Typically the online server cache is automatically managed by the product but a number of additional utilities were provided to manually manage the contents of the cache. This was

primarily used by developers and testers in their activities. As with Debug mode this facility was open to any valid user of the product. While this was feature was not destructive, it was decided to add similar security controls as provided with the debug mode.

The use of cache management utilities is now restricted to authorized users only using the **F1ADMIN** security group.

Note: This facility is provided for backward compatibility and for developers as it is expected that customers will use the JMX facilities to control the cache.

Logoff Support

It is now possible to logoff the browser session explicitly in additional to closing the browser session to terminate the session. The logoff button will return to the logon screen. For example:

User Identification on User Interface

The name of the user is clearly identified at the top of the logon screen in line with Fusion and Oracle standards.

Channel Based Passwords

In line with practices used other Oracle products, each mode of access has the ability to use the same or different database users. This technique allows implementers to assign different resource profiles at the database level for different modes of access. It is now possible to setup individual database users for the following modes of access at installation/configuration time:

- Online
- Batch
- XAI

Secure Password input

Passwords that are specified during the installation and configuration processes are now input in a secure silent mode.

Enhanced File Permissions

In previous versions of the product, file permissions were set at the global level for the environment. These are now fine grained permissions in accordance with Oracle Security policy.

LDAP Import can be scheduled

In past releases, the LDAP Import function was initiated online to interactively import user records from an LDAP source into the security model on a regular basis. Whilst this was sufficient for some sites, most sites wish to regularly schedule synchronization in the background, especially for large numbers of users. It is now possible to schedule the LDAP import via a new job that takes the location of the LDAP, credentials; mapping file and import filter as parameters to execute the job with. The existing online version of the LDAP import will be retained in the short term for backward compatibility and as a test bed for the mapping and import filter prior to executing this new job.

Note: Deletions of users will not result of deletion of user records. Deletions will disable users using the new "User Enable" facility.

IP Address Tracking in Audit

The audit facility within the framework allows registration of user credentials as part of the audit information captured. It is now possible to capture IP Address of the end user for use on the audit record (if desired). This feature also supports recording real and proxy IP addressed and proxy connections (if the proxy is configured to place the pre-proxy address in the request header context variable **`$REQUESTING-IP-ADDRESS`**).

Note: To use this facility the base audit facility must be extended. Please refer to your product SDK documentation to see if this has been enabled in the product.

Generic User Portal

To support extensions to the authorization model, a generic user tab has been added to the user object to allow products and implementations to add specialist zones to display, modify or add additional information on the user object. This portal can be used by products to support security for specific market requirements as well as provide implementations for a means to provide customers with additional site specific security facilities.

Privacy Notice Redirect

In some markets the site must provide a privacy policy that can be accessed from the product and/or the logon screen. A standard redirect has been implemented to allow product groups and/or implementations to add a privacy screen to comply with site standards. The privacy policy can be in HTML format and must be located as cm/privacy.html which is then able to be accessed using the http://<host>:<port>/<server>/privacy URL

Centralized JMX Security

Over the past few releases Java Management Extensions (JMX) has been introduced to provide a management mechanism for operators from JSR160 compliant consoles and Oracle Enterprise Manager. Security for this interface has been centralized and managed using JAAS. By default the security scheme is file based but can be configured to use other security mechanisms.

SYSUSER can be disabled

By default, **SYSUSER** is created as an initial user, used to populate other users definitions after installation. It is not meant to be used as active user after populating. This enhancement will allow for the logical deletion of the **SYSUSER** account by allowing the user record to be disabled, if desired, post installation.

Enhanced Menu Item Security

In past releases, if a user had any access to a function then the function appeared on the menu. Once a user entered the function security access would be either applied on entry or when attempting a function. For example, function buttons would not appear if the user was authorized or an error message would be displayed if a user attempted to access the function.

In this enhancement the product menu items and service calls are assessed on loading time or upon calling (for example from a BPA).

Advanced Database Security support

In past releases of the Oracle Utilities Application Framework we have supported the use of secure protocols in the architecture. This enhancement extends this by allowing customers to encrypt database connections using Oracle Advanced Security.

This enhancement, if enabled, will require customers to purchase the Advanced Security option on the database and configure the connections to be secure. At this point no unsecure transactions will be allowed to be performed on the database.

After this enhancement is delivered every channel within the product architecture can be secured.

Note: Securing the network traffic of any tier may result in performance degradation due to the extra processing of encryption and decryption.

Keystore Support

In past releases, keys used for encrypting data such as passwords and product data were managed internally by the Oracle Utilities Application Framework. In this release these keys are now externalized in a JCEKS based keystore.

Introducing Security

One of the key aspects of the product is security which not only confirms the identity of an individual user but, once identity is confirmed, what data and what functions that user has access to within the product.

Security Features

Security is one of the key features of the product architecture protecting access to the product, its functionality and the underlying data stored and managed via the product.

From an architecture point of view the following summarizes the approach to security:

- **Web Based Authentication** – The product provides a default method, using a traditional challenge and response mechanism, to authenticate users.
- **Support for J2EE Web Application Server security** – The supported J2EE Web Application Servers can integrate into a number of internal and external security stores to provide authentication services. The product can use those configurations, to liaise via the J2EE Web Application Server, to authenticate users for online and Web Services based security.
- **Operating System Security** – For non-online and non-web service based channels, the product utilizes the operating system security (including any additional products used to enhance the base operating system security).
- **Non-Cookie based security** – After authentication the user's credentials form part of each transaction call to correctly identify the user to the internal authorization model to ensure the user is only performing permitted actions. This support is not browser cookie based.
- **Secure Transport Support –** Transmission of data across the network can utilize the secure encryption methods supported for the infrastructure.
- **Inter-component security -** Calls within the product and across the tiers are subject to security controls to ensure only valid authenticated and authorized users using Java Authentication and Authorization Services (JAAS).
- **Inbuilt Authorization Model –** Once a user is authenticated then the internal authorization model is used to determine the functions and data the user has access to within the product.
- **Native Web Services Security –** Web Services available from the product are natively available from the J2EE Web Application Server. A wide range of security policies are available.
- **Keystore Support -** Keys for encryption can be externalized in JCEKS based keystore.
- **Integration with other security products –** Implementation of security varies from customer to customer so the product allows integration of other security products to offer enhanced security implementations, either directly or indirectly.

Additional Security Resources

In addition to the security resources described in this guide, Oracle Utilities Smart Grid

Gateway provides the following additional security resources:

- **Oracle Database Vault** – Oracle Database Vault provides fine-grained access control to your sensitive data, including protecting data from privileged users. *Oracle Database Vault Administrator's Guide* and *Database Vault Integration* (Doc Id: 1290700.1) describes how to use Oracle Database Vault.
- **Oracle Audit Vault** – Oracle Audit Vault collects database audit data from sources such as Oracle Database audit trail tables, database operating system audit files, and database redo logs. Using Oracle Audit Vault, you can create alerts on suspicious activities, and create reports on the history of privileged user changes, schema modifications, and even data-level access. *Oracle Audit Vault Administrator's Guide* explains how to administer Oracle Audit Vault.
- **Oracle Advanced Security** - See *Oracle Database Advanced Security Administrator's Guide* for information about advanced features such as transparent data encryption, wallet management, network encryption, and the RADIUS, Kerberos, Secure Sockets Layer authentication.
- **Oracle Identity Management Suite** – Oracle offers a range of specialist security products to manage user identities, password management, single sign on, access management, identity governance, fraud detection and directory services. The *Oracle Identity Management Suite Administrator Guides* and *Oracle Identity Management Suite Integration with Oracle Utilities Application Framework based products* (Doc Id: 1375600.1) provides additional information about these products and integration capabilities.

Authentication

About Authentication

From a security point of view authentication is about identification of the user. It is the first line of *defense* in any security solution. In simple terms it can be as simple as the *challenge-response* mechanism we know as userid and password. It can be also as complex as using digital certificates as the identification mechanism and numerous other schemes for user identification.

The authentication aspect of security for the product is delegated to the infrastructure used to run the product. This is due to a number of reasons:

- **Authentication scheme support** – The J2EE Web Application Server supports a number of industry standard security repositories and authentication methods. These can be native to the J2EE Web Application Server or additional products that can be are integrated.
- **Enterprise Level Identity Management** – Identity Management is typically performed at an enterprise level rather than managed at an individual product level. The product typically is not the only application used at any site and managing security across the enterprise is more efficient.

Online Authentication

The product delegates the responsibility of authentication of the online users to the J2EE Web Application Server. This means that any integration that the J2EE Web Application Server has with specific security protocols or security products can be used with the product for authentication purposes. The configuration of authentication is therefore performed within the J2EE Web Application Server itself.

Typically the J2EE Web Application Server support one or more of the following:

- **Inbuilt Security** – The J2EE Web Application Server typically supplies a default basic security store and associated security management capability that can be used if no other security repository exists.
- **LDAP Based Security** – The Lightweight Directory Access Protocol (LDAP) is a protocol for accessing and maintaining distributed directory information services. LDAP is used to standardize the interface to common security repositories (such as Oracle Internet Directory, Microsoft Active Directory etc). LDAP support may be direct or indirect via Identity Management software like Oracle Virtual Directory or Oracle Identity Federation.
- **SAML Based Security** – Security Assertion Markup Language (SAML) is an XML based data format for exchanging authentication and authorization information between parties.
- **DBMS Based Security** – The J2EE Web Application Server can store, manage and retrieve security information directly from a database.

- **Operating System Based Security** - The J2EE Web Application Server can store, manage and retrieve security information directly from the underlying operating system.

These security configurations can be natively support or can be augmented with additional products.

Refer to the Security Guides supplied with your J2EE Web Application Server for details of the security configuration process.

Batch Authentication

The Batch component of the architecture utilizes the operating system based security (including any extensions to that security) to authenticate users to execute batch processes. From an authentication point of view:

- Batch users must be defined in the operating system and associated with the operating system security group assigned at product installation time. This ensures users have appropriate access to product resources and the ability to write logs.
- Threadpools can be started by any valid operating system user but ideally threadpools and submitters should be executed by the same operating system user.
- Before any threadpool or submitter is executed the user must execute the `splenviron` utility to set the environment variables for the product correctly. This can be done at the command line for each threadpool and submitter or globally using the logon profile for the operating system user.

Web Service Authentication

The Web Service component of the product is housed in the J2EE Web Application Server and utilizes the native Web Services security mechanism supported by that server.

From an authentication point of view:

- The Web Service is deployed using an administration account using the utilities provided from the product online (for developers) or using command line utilities.
- The Web Service is managed using the administration account using the administration console provided with the J2EE Web Application Server.
- The J2EE Web Application Server allows security policies and/or security access rules to be configured at an individual Web Service point of view. Any of the valid policies and security rules supported by the J2EE Web Application Server can be used.
- Web Service management products such as Oracle Web Services Manager can be used to augment security for Inbound Web Services.

Authorization

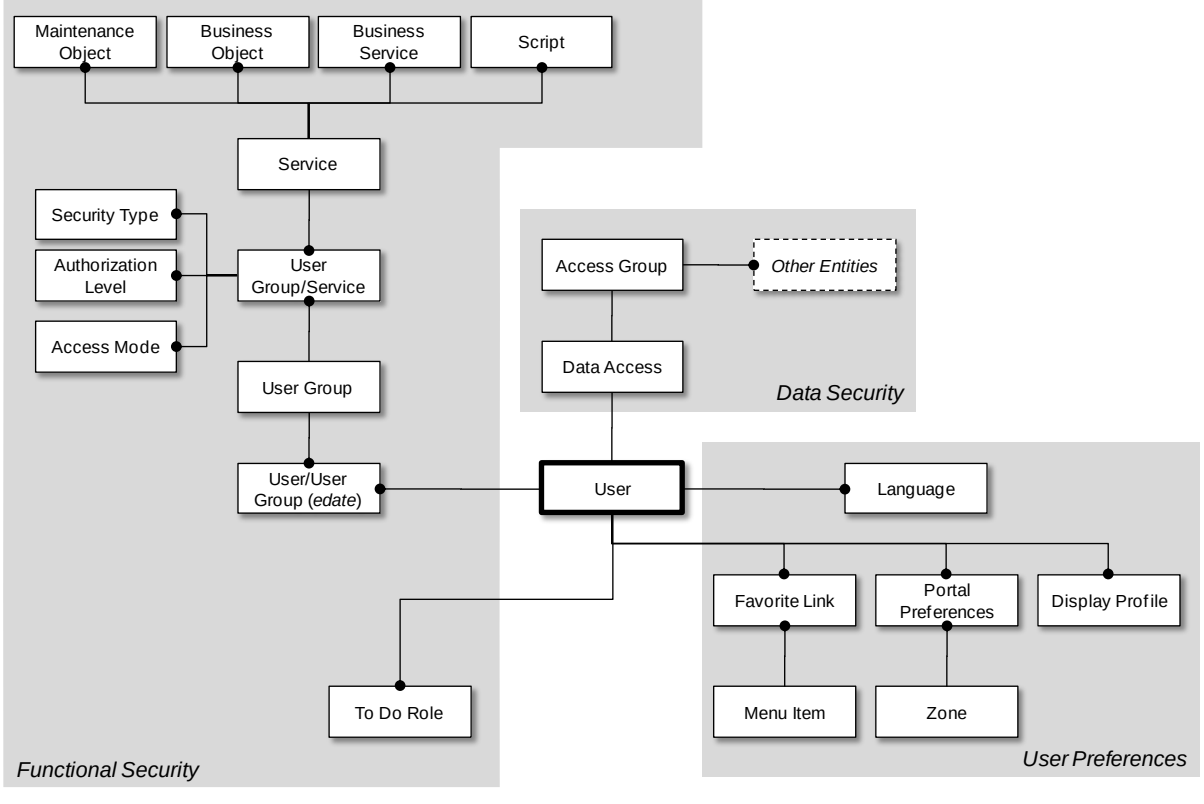
Once a user is identified they must be authorized to specific functions and data within the product.

About Authorization

The Oracle Utilities Application Framework uses an inbuilt security model for authorization. This model contains all the data necessary for the definition of authorizations to function and data. The following data model describes the security authorization model.

Authorization Model

The Oracle Utilities Application Framework uses an inbuilt security model for authorization. This model contains all the data necessary for the definition of authorizations to function and data. The following data model describes the security authorization model.



A record of each user is stored in the User entity, which defines the attributes of the user including identifier, name, Portal preferences, Favorites, Display Profile (such as format of dates etc), and Language used for screens and messages and other attributes. Users are attached to To Do roles which allow the user to process any error records for background processes. For example, if the XXX background process produces an error it is possible to define which users will process and address those errors.

Users are also attached to User Groups. This relationship is effective dated which means that the date period it is active across is also defined. This can be useful for temporary employees such as contractors or for people who change roles regularly.

User Groups are a mechanism for grouping users usually around job roles. Each User Group is then attached to the Application Services that the group is authorized to access. The Application Services are the functions within product. Loosely they correspond to each of the screens accessible in product. In this attachment the Access Mode is also defined with standards being Add, Modify, Read and Delete. With this combination it is possible to define what functions and what access is allowed to those functions for user groups (and hence users).

Additionally it is possible to define the authorization level that is allowed for the User Group to that function. For example, you may find that a certain group of users can only approve payments of a certain level unless additional authorization is obtained. The Authorization Level is associated with a Security Type which defines the rules for that Application Service.

Note: To use security types, the implementation must develop server side or client side user exits to implement code necessary to implement the security level.

Services can be attached to individual Maintenance Objects, Business Objects, Business Services and Scripts to denote the service to be used to link user groups to access these objects. In this case Business Object security overrides any Maintenance Object security. The same applies to Business Services security overriding the Application Service it is based upon.

The Oracle Utilities Application Framework allows you to limit a user's access to specific data entities to prevent users without appropriate rights from accessing specific data. By granting a user access rights to an account, you are actually granting the user access rights to the account's bills, payment, adjustments, orders, etc.

An Access Group defines a group of Accounts that have the same type of security restrictions. A Data Access Role defines a group of Users that have the same access rights (in respect of access to entities that include access roles). When you grant a data access role rights to an access group, you are giving all users in the data access role rights to all entities in the access group.

The following points summarize the data relationships involved with data security:

- Entities reference a single access group. An access group may be linked to an unlimited number of relevant entities.
- A data access role has one or more users associated with it. A user may belong to many data access roles.
- A data access role may be linked to one or more access group. An access group may be linked to one or more data access roles.

Information in the security model can be manually entered using online transactions and also can be imported and synchronized using a LDAP import function provided with the Web Services Adapter. The latter is typically used with customers who have lots of online users to manage.

The authorization model is used by all modes of access to the product. Native interfaces (java classes) are used by all objects and a PL/SQL procedure is provided for reporting interfaces.

Managing Security

About Managing Security

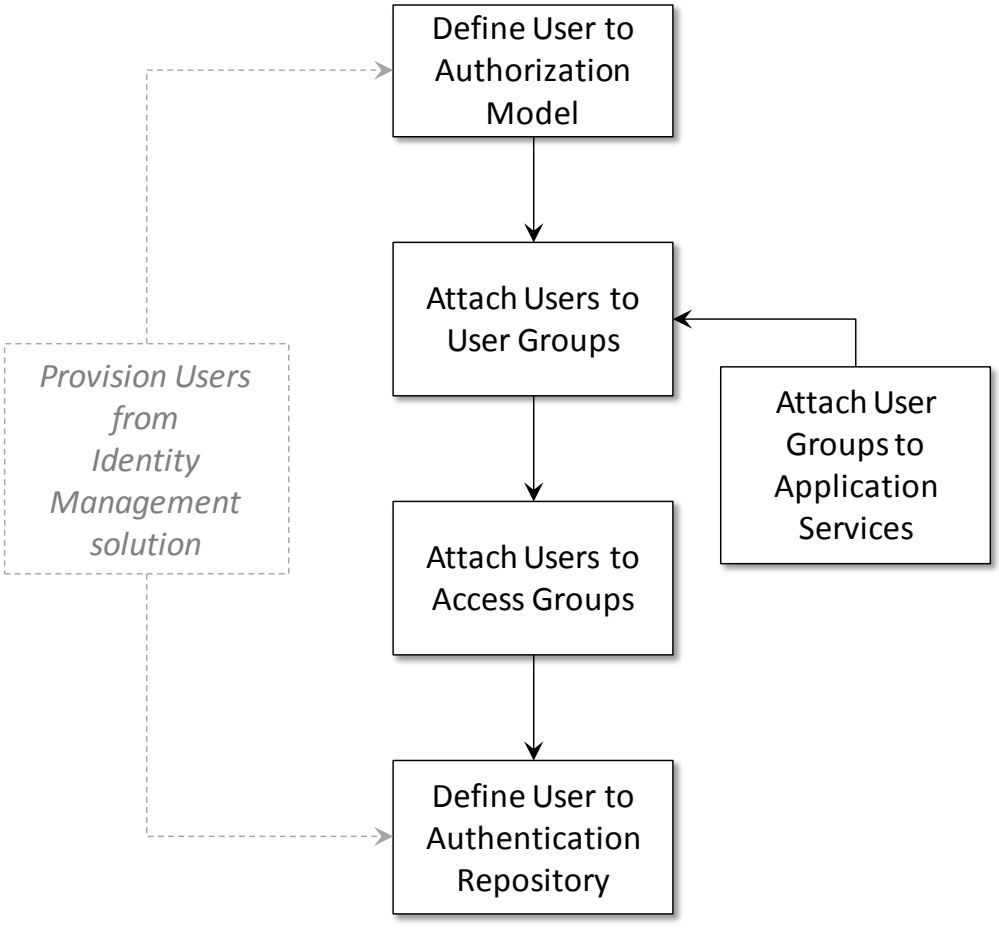
Once the security definitions are established they must be managed from the product itself, security infrastructure and security repositories used.

Managing Online Users

To manage online users a number of facilities must be configured:

- The security repository and rules must be configured in the J2EE Web Application Server to enable authentication. Refer to the J2EE Web Application Server *Administration Guides* for more information.
- The product group used to connect users to J2EE resources should be created in the security repository and configured in the product configuration. The default value for this setting is `cisusers`. Refer to the *Server Administration Guide* for more information on this setting.
- Users need to be connected to the product group within the security repository to indicate that they can access the J2EE resources.

The process for managing online users is outlined in the following process:



- Users should be defined to the authorization model to define their profile and permissions within the product. Refer to Adding Users for more details of this process.
- Attach user groups to application services to define the subset of service and actions valid for that group of users. Refer to Defining User Groups to Application Services for more details of this process.
- Attach Data Access Groups to the users. This defines the subset of data that the user has access to. Refer to Define Users to Data Access Groups for more details of this process.
- Attach users to the appropriate user groups to define the subset services and valid actions the user can perform within the product. Refer to Defining Users to User Groups for more details of this process.

Managing Users

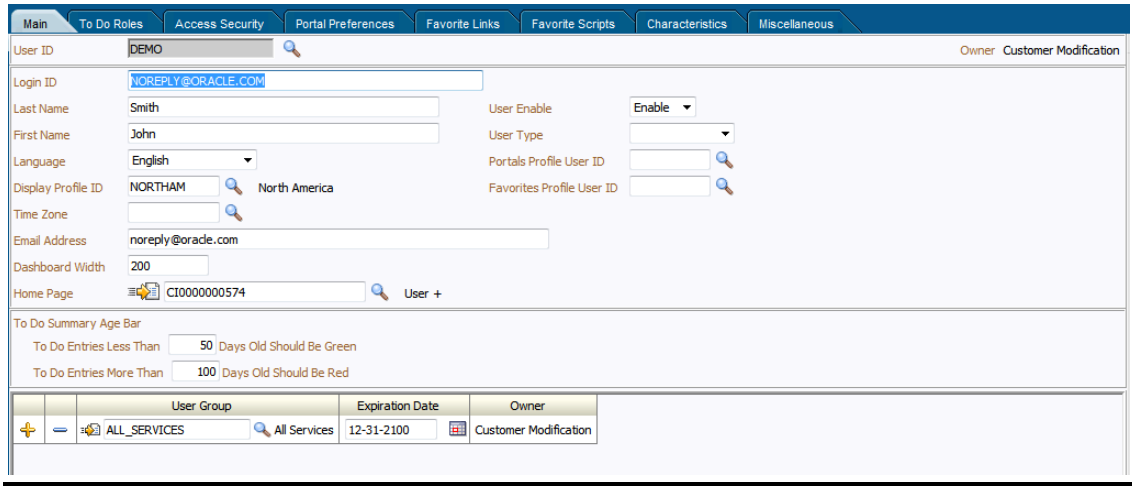
The user object in the product is used to record the security information used for identification of the user and their permissions.

The product provides a maintenance function to maintain these definitions within the product. To maintain the users the following is performed:

- Navigate to the *Administration Menu* → *U* → *User menu* option. Using the + option on the menu allows navigation to the add function.
- The User maintenance object is displayed which maintains the security information

for a user.

- A screen similar to the one shown below is displayed:



Field	Comments
Userid	This is the unique user identifier used within the product used for authorization activities. Limited to eight (8) characters in length.
Login Id	This is the unique user identifier used within the product used for authentication purposes. This must match the value used in the security repository to successfully use the product. Limited to 256 characters in length. This value can be the same or different to the Userid.
Last Name	Last Name of user. Limited to 50 characters in length.
First Name	First Name of user. Limited to 50 characters in length.
User Enable	Whether the user is active in the security system or not. Valid Values: **Yes**(*default*) – User is active and can use the system, **No** – User is disabled and cannot use the system. Refer to User Enable and Disable for more details.
User Type	The type of user. Valid Values: **Blank** = Normal user, **Template** = Template User.
Language	Default Language used for user. For non-English languages, Language pack must be installed to use specific languages.
Display Profile Id	The display profile associated with the user. This controls the display of currency, dates etc…
Time Zone	Time Zone allocated to user account[1].

[1] This feature is only applicable to specific products. Check your product online documentation for more details about applicability.

Field	Comments
Email Address	Optional Email address associated with user. This is used by utilities and can be used for interfaces requiring email addresses.
Dashboard Width	Default width for Dashboard Portal. Setting this value to zero (0) will disable the dashboard altogether.
Home Page	The default home page associated with the user.
Portals Profile User Id	The userid used to inherit portal definitions from. Refer to Template Users for more information.
Favorites Profile User Id	The userid used to inherit favorite definitions from. Refer to Template Users for more information.
To Do Summary Age Bar	The settings for the color coding of the To Do Summary portal in the dashboard. This can be used to indicate relative age of to do entries.
User Groups	This is a list of user groups and their associated expiry dates. Refer to Define Users to User Groups for more information.

- Save the additions/changes for the user using the *Save* function on the top of the screen.

Template Users

By default portal preferences and favorites are set at an individual user level. It is possible to inherit the portal preferences and/or favorites from other users to reduce the maintenance effort for security information. Changes to the profile user are automatically inherited to any users where the profile user is attached to.

To use this functionality the following must be performed:

- Setup each user to be used as template and indicate the user type is set to **`Template`** to indicate such.
- For any user that will inherit the portal preferences and/or favorites specify the appropriate template user in the following fields:
  - **Portal Preferences** – Use the *Portals Profile User Id* to indicate which Template user can be used to inherit the portal preferences from.
  - **Favorites** – Use the *Favorites Profile User Id* to indicate which Template user can be used to inherit the favorites and favorite scripts from.
- Once any changes are made to the Template users portal preferences and/or favorites they will automatically apply to any attached users for these features.

Assigning To Do Types

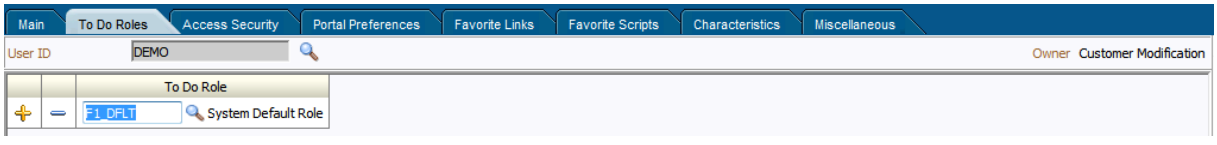
Note: To Do records can be assigned to explicit users or groups of users. This section covers the latter condition.

Note: Refer to the online Administration Help for a discussion about the To Do functionality. To Do

roles must be setup prior to using this functionality.

The product generates To Do records for any function or error condition that requires human intervention. The To Do record contains a type and role to be used assist in assigning the appropriate resources to work on the condition indicated by the To Do.

For security purposes, users need to be attached to the relevant roles for the To Do facility to limit which To Do Types an individual user can work upon. To define the To Do roles for a user, navigate to the *To Do Roles* tab of user maintenance function. This will display a screen similar to the one below:



To manage the To Do Roles to be assigned to a user the following must be performed:

- Use the + to add a new To Do Role
- Use the − to remove an existing To Do Role from the list.

The Search icon ( ) can be used to find the existing To Do Role or it can be typed in.

Once the users have been attached to the To Do Roles then they can access the associated TO Do types assigned to that role or any To Do directly assigned to them.

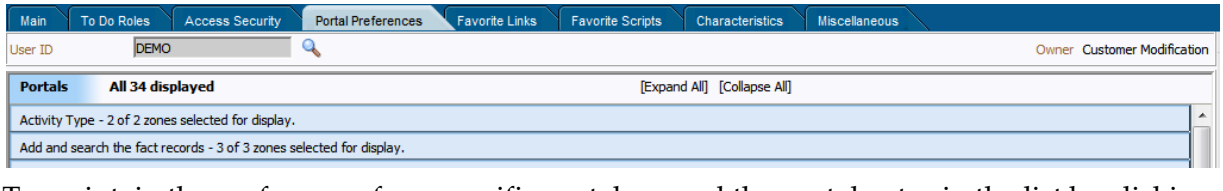
Assigning User Portal Preferences

Note: Refer to the online Administration Help for a discussion about the Portal/Zone functionality. Portals and Zones must be setup prior to using this functionality.

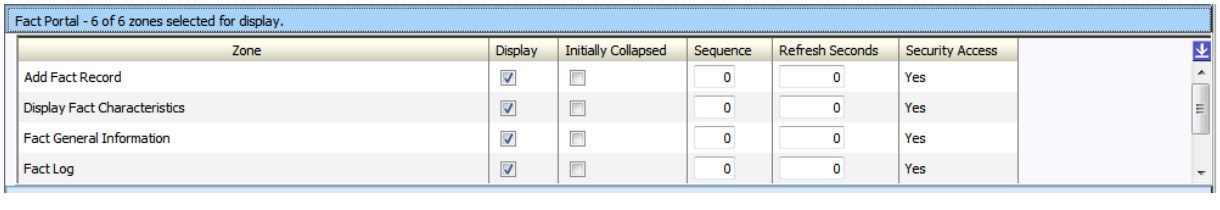
Note: Portal Preferences can be inherited from other users if Template users are used. In this case the ability to set for portal preferences for users attached to a template user are disabled.

The product user interface is made up of Portals containing individual Zones. Each of the portals and zones can be associated with an application service for security purposes. Users that are attached to User Groups that are also attached to those application services can view and use the portals and zones.

The order of display and other factors are defined at an individual user basis. To define the portal preferences for a user, navigate to the *Portal Preferences* tab of user maintenance function. This will display a screen similar to the one below with a list of the portals the user has access to, via the user groups they are attached to:



To maintain the preferences for a specific portal expand the portal entry in the list by clicking the name of the portal or using the *Expand All* functionality. For example:



The following zone preferences can be set for the user:

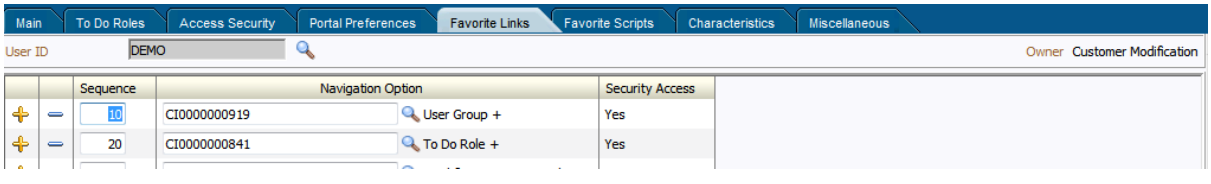
- **Display** – Whether the zone is included or not in the portal. This allows specific zones to be displayed at startup time while other zones can be hidden and only displayed upon conditions in other zones. See *Zone Visibility* in the online Administration guide for more information.
- **Initially Collapsed** – Whether the zone is displayed collapsed on initial load. Zones are only executed when they are expanded. Marking zones as *Initially Collapsed* can prevent them from being executed and can speed up portal rendering times.
- **Sequence** – Defines the relative order of the zones within the portal. A value of zero (0) takes the default sequence from the portal definition.
- **Refresh Seconds** – Defines the zone auto refresh rate (this is only applicable to particular zone types). A value of zero (0) disables auto-refresh.
- **Security Access** – This is an information field that indicates whether the user has access to the zone or not[2]. Refer to the online documentation for more information.

Assign Favorite Links

Note: Favorites can be inherited from other users if Template users are used.

Each individual user can set a number of favorite functions or menu items that they can access using keyboard shortcuts or via the Favorites zone on the Dashboard.

The definition of the users Favorite Links can be configured by navigating to the *Favorite Links* tab of user maintenance function. This will display a screen similar to the one below:



The following fields can be set for the favorites:

- **Sequence** – The relative sequence number of the favorite used for sorting purposes.
- **Navigation Option** – The Navigation option to use to display the favorite. This can reference the zone or maintenance function to display when this favorite is chosen.
- **Security Access** – This is an information field that indicates whether the user has access to the Navigation Option or not.

To manage the Favorites to be assigned to a user the following must be performed:

- Use the ✚ to add a new Favorite with the appropriate Navigation Option with the appropriate Sequence to indicate where in the favorites list the option should be placed.
- Use the ➖ to remove an existing Navigation Option from the list.

The Search icon (🔍) can be used to find the existing Navigation Option or it can be typed in.

Favorites are then available to be displayed in the Favorites portal on the Dashboard.

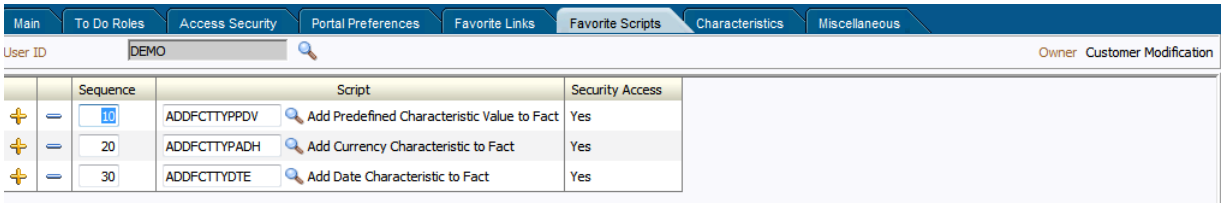
[2] While unlikely, it is possible to have a portal contain particular zones not permitted for access to an individual user.

Assign Favorite Scripts

Note: Favorites can be inherited from other users if Template users are used.

Each individual user can set a number of favorite BPA Scripts that they can access using the Favorite Scripts zone on the Dashboard.

The definition of the users Favorite Scripts can be configured by navigating to the *Favorite Scripts* tab of user maintenance function. This will display a screen similar to the one below:



The following fields can be set for the favorites:

- **Sequence** – The relative sequence number of the favorite used for sorting purposes.
- **Script** – The BPA Script to use to display the favorite.
- **Security Access** – This is an information field that indicates whether the user has access to the Script or not.

To manage the Favorites to be assigned to a user the following must be performed:

- Use the ✚ to add a new Favorite indicating the Script with the appropriate Sequence to indicate where in the favorites list the option should be placed.
- Use the ➖ to remove an existing Script from the list.

The Search icon (🔍) can be used to find the existing Script or it can be typed in.

Favorites are then available to be displayed in the Favorite Scripts portal on the Dashboard.

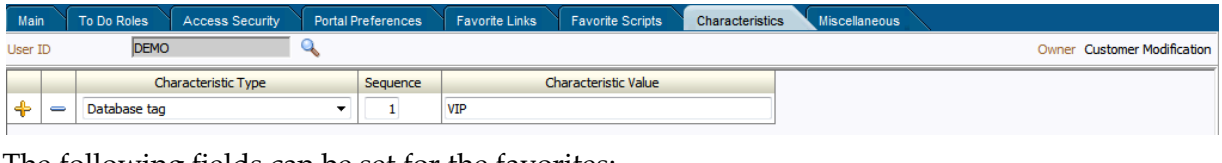
Assign User Characteristics

Note: To use this facility the appropriate characteristic types must be created and attached to the user object. Refer to the online Administration documentation for more information.

Note: The product ships with a predefined set of characteristic types.

One of the features of the product is the ability to extend the object within the product using user defined fields called *Characteristics*. Characteristics act as additional data attributes that can be used to simply provide additional information or used in custom algorithms for processing.

The user object in the product can also be customized using characteristics. This can be achieved by navigating to the *Characteristics* tab of user maintenance function. This will display a screen similar to the one below:



The following fields can be set for the favorites:

- **Characteristic Type** – The characteristic type associated with the user object. This is a drop down list of the valid characteristic types associated with the object.

- **Sequence** – The relative sequence number of the characteristic used for processing purposes.
- **Characteristic Value** – This is the value of the characteristic. Depending on the configuration of the characteristic type this value may be free format, an attachment, a specific format or a specific set of values.

To manage the Characteristics to be assigned to a user the following must be performed:

- Use the + to add a new Characteristic indicating the Characteristic Type, the appropriate Sequence to indicate where in the favorites list the option should be placed and the value associated with the Characteristic Type.
- Use the − to remove an existing Characteristic from the list.

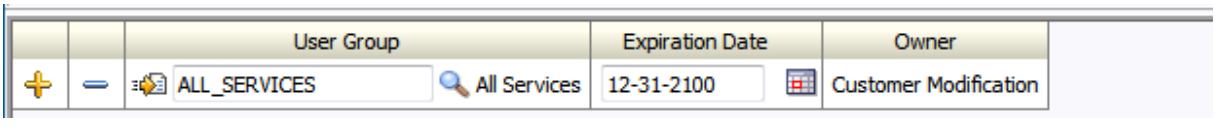
Defining Users to User Groups

To access the services within the product users must be connected to user groups which are in turn connected to application services. This defines the linkage for functionality that the user has access to.

The link between users and user groups has the following attributes:

- The linkage between users and users groups is subject to an expiry date to allow representation of transient security configurations.
- Each link between a user and user group is owned and subject to Data Ownership Rules. By default, all site created links are owned as *Customer Modifications*.
- User Groups are setup according to your site preferences. They can be job related, organization level related or a combination of factors.
- A user can be a member of any number of users groups but should be at least a member of one group to access the system.
- Users can be members of groups with overlapping permissions to application services. In the case of overlapping permissions, then the highest valid permission is used.

This can be achieved by navigating to the *Main* of user maintenance function. This will display a screen with zones at the bottom of the screen similar to the one below:



The user groups are listed that the user has access to and can be manipulated using the following:

- Use the + to add a new User Group indicating the User Group Name with the appropriate expiry date to indicate relevance of the connection. The Search icon (🔍) can be used to find the existing User Group or it can be typed in.
- Use the − to remove an existing User Group from the list.

The users security is then used for menu and function access regardless of access channel used (i.e. online, web service or batch).

Defining User Groups to Application Services

Note: A starter set of User Groups are loaded with the product that can be used as the basis for further security user groups.

Note: The product ships with all the application services predefined for base functions. These can be used or replaced with custom definitions as desired.

One of the fundamental security configurations for the product is to define the user groups to the application service. An application service can represent an individual service within the product, an individual menu or an individual object. When linking a user group to a service the access modes can be configured which defines the valid actions the user group can perform against the service.

Additionally each service can specify Security Types which allow for custom security rules to be applied at runtime. Refer to Security Types for more details of this facility.

To maintain the linkages between user groups and application services there are two different methods:

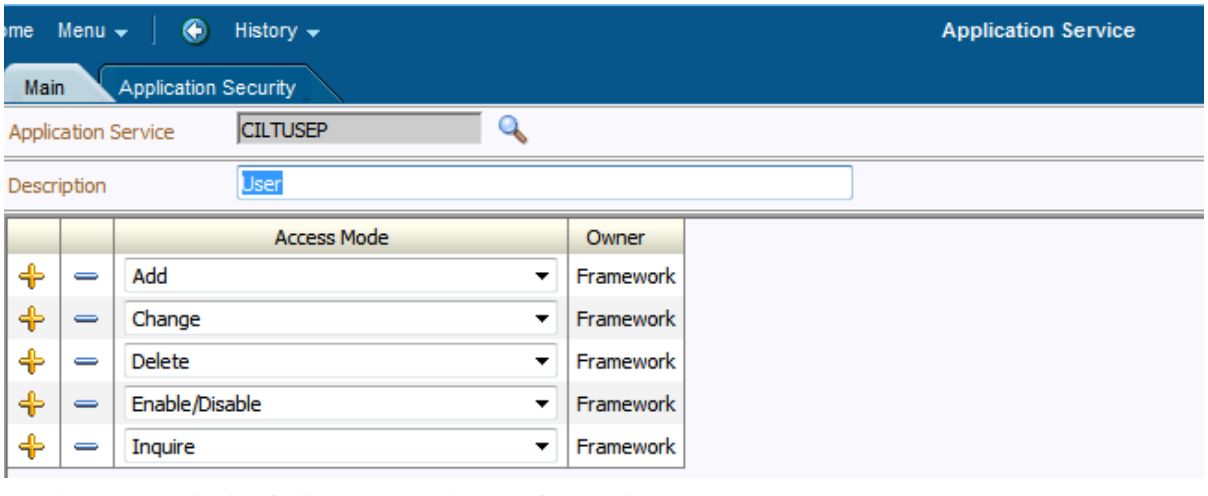
- Application Services Portal – When maintaining each Application Service it is possible to connect and disconnect the user groups and determine which groups have access to what functions.
- User Group Maintenance – When maintaining User Group definitions it is possible to connect Application Services to the group and manage users in that group from a single maintenance function.

Both methods are valid for most sites and can be used to manage the same information from different prospective.

Using the Application Services Portal

The Application Service portal allows administrators to define an application service, the valid access modes available for the Application Service and the user groups the application service is connected to.

To access the Application Services Portal, navigate to the *Administration Menu* → *A* → *Application Service* option. This will display a screen similar to the following:



On the main tab the following can be configured:

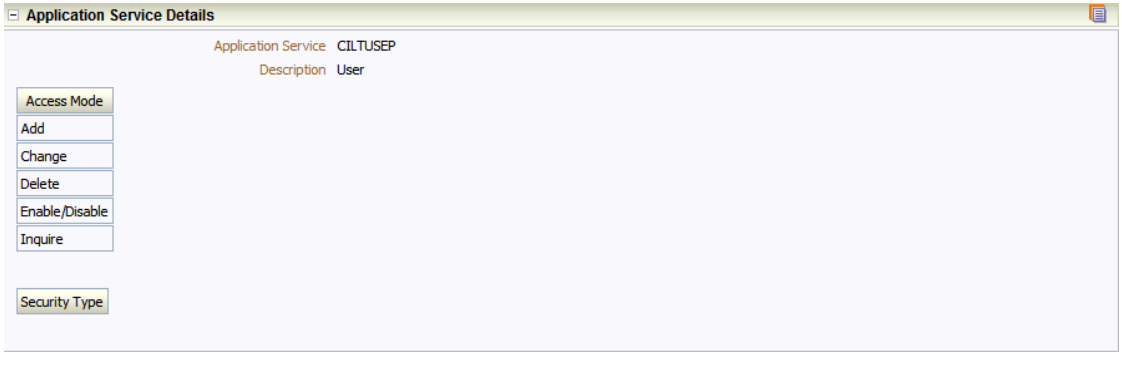
- **Application Service** – The Application Service identifier. This is a service

identification token used in configuration of security on the objects, menu or service etc. For custom definitions it is recommended to prefix this value with *CM* to avoid conflict with product provided application services.

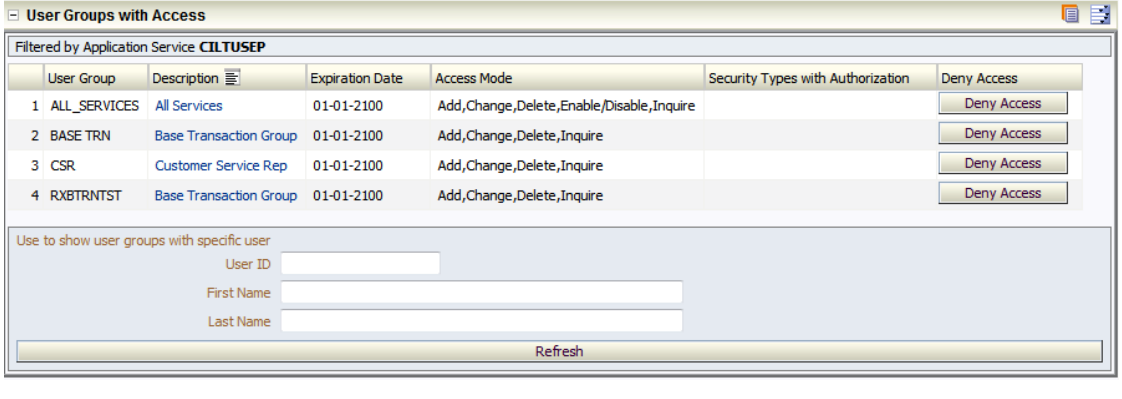
- **Description** – This is a short description used for documentation purposes. This value is displayed on any security screen when the Application Service is specified.
- **Access Modes** – A list of valid access modes is defined and displayed for the Application Service. These modes must match the internal actions supported by the objects which this Application Service is used. When using this list
  - Use the ✚ to add a new Access mode from the drop down list of valid actions. An individual Access Mode can only be defined once for an individual Application Service.
  - Use the ➖ to remove an existing Access Mode from the list.
  - The Access Mode link to the Application Service is ownership controlled. By default, all created links are owned as Customer Modifications. Refer to Data Ownership Rules for more details on ownership of data.

The *Application Security* tab is a portal that provides the ability to display the user group membership and manage that relationship. The portal is made up with a number of zones to provide information and maintenance capabilities:

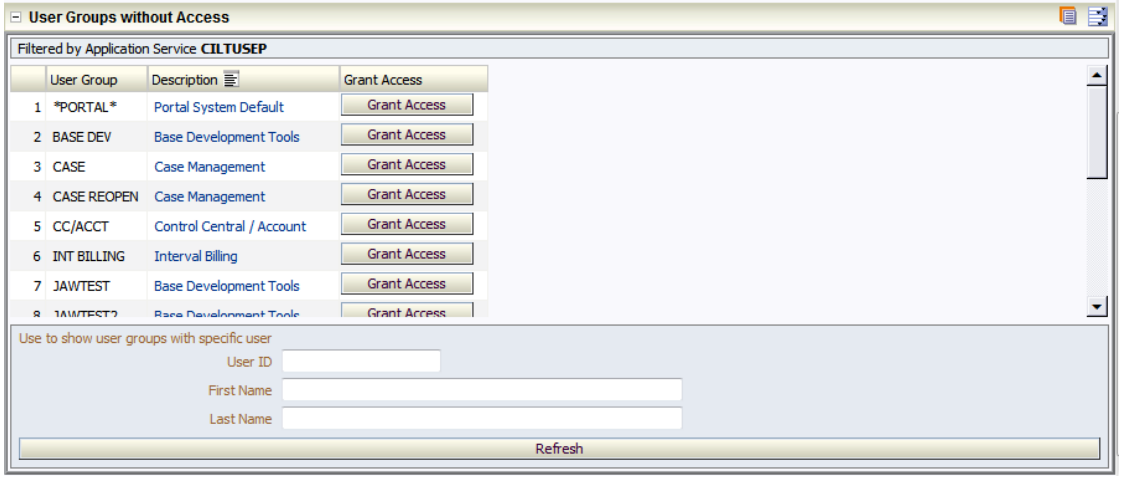
- **Application Service Details** – This zone summarizes the access modes and security types defined for an application service. For example:



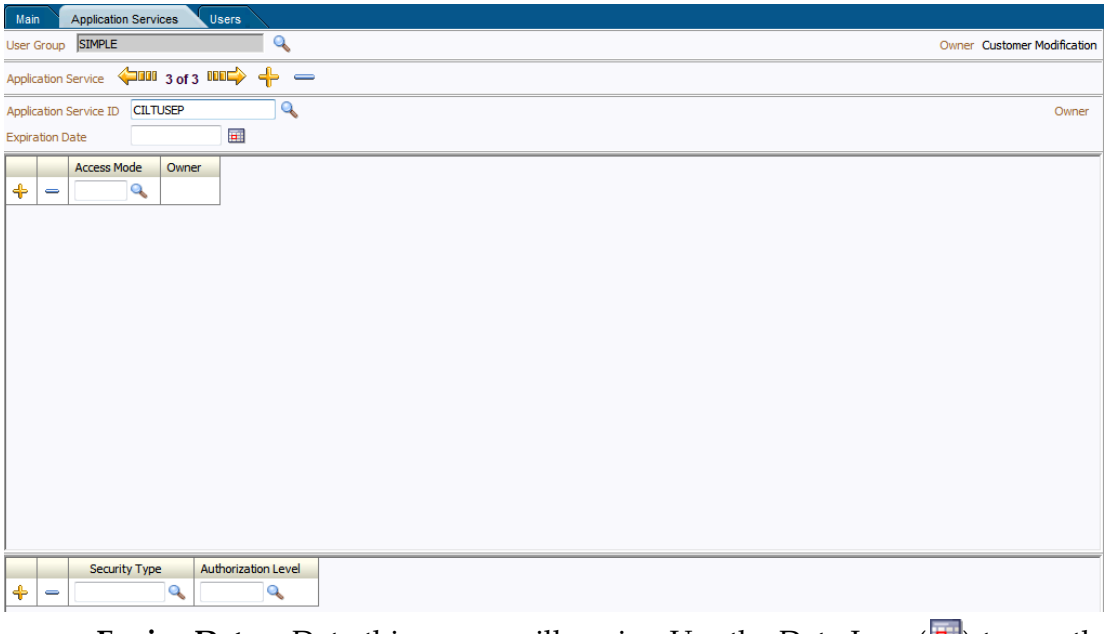
- **User Groups With Access** – This zone lists the user groups that have access to the Application Service along with the associated expiry date, access modes and security types (and associated authorization level). It is possible to deny access by a particular group to the application service using the *Deny Access* functionality. The list can be filtered to user groups for a particular user to assist in isolating particular user groups. For example:



- **User Groups Without Access** – This zone list the user groups that do not have access to the Application Service to grant access, if desired, using the *Grant Access* functionality. The list can be filtered to user groups for a particular user to assist in isolating particular user groups. For example:



Once a group is granted access then the specification of the valid access modes and security groups can be provided for the particular user group. For example:



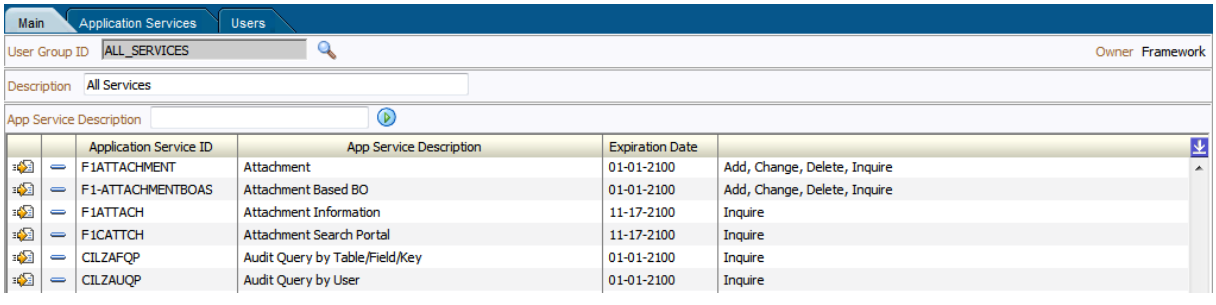
  - **Expiry Date** – Date this access will expire. Use the Date Icon ( ) to use the Date selection widget.
  - **Access Mode** – Valid Access mode as defined on Application Service definition.
    - Use the to add a new Access Mode. The Search icon ( ) can be used to find the existing Access Mode or it can be typed in.
    - Use the to remove an existing Access Mode from the list.
  - **Owner –** Ownership of link (refer to Data Ownership Rules)
  - **Security Type** – Security Type code associated with Application Service.
    - Use the to add a new Security Type. The Search icon ( ) can be used to find the existing Security Type or it can be typed in.

- Use the ⊖ to remove an existing Security Type from the list.

- **Authorization Level** – The Authorization Level assigned to this User Group when executing this Application Service for the Security Type. The Search icon (🔍) can be used to find the existing Authorization Level or it can be typed in.

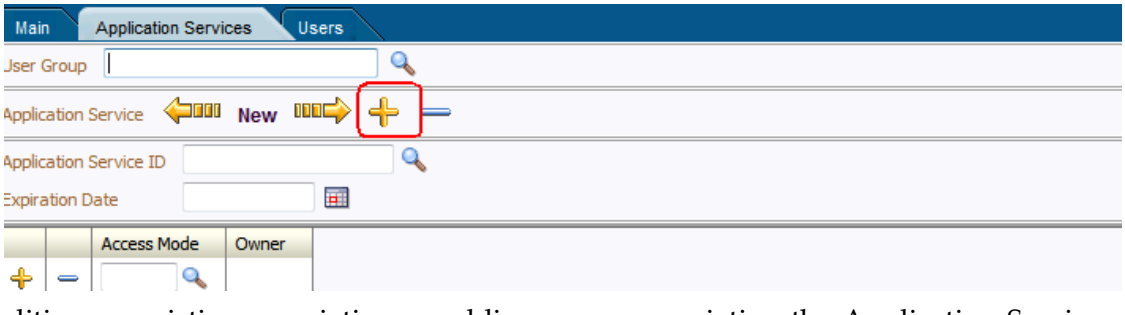
Using User Group Maintenance

When editing an individual user group it is possible to define the accessible application services and connect users to the user group from the user group maintenance function. To do this, navigate to the *Administration Menu* → *U* → *User Group* menu option. This will display a screen similar to the following:

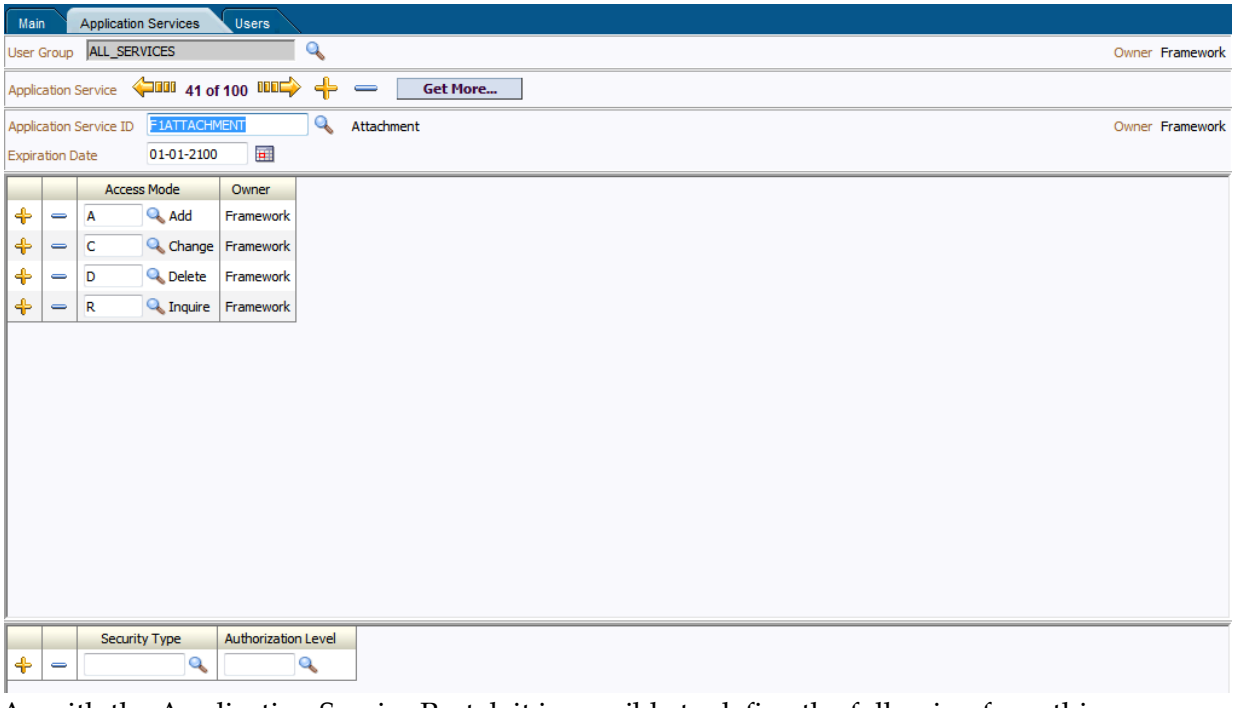


The services that this user group has access to are shown with the associated expiry date and access modes for the user group. The following actions maintain the information:

- Use the icon to edit an existing permission.
- Use the ⊖ icon to remove an associate between a user group and an application service.
- Use the ⊕ on the *Application Services* tab to add a new association between an application service and an individual user group. For example:



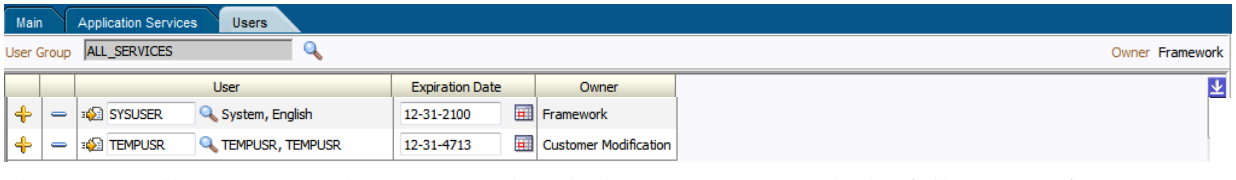
When editing an existing association or adding a new association the Application Services tab is displayed to maintain the association with associated Access Modes and Security Types. For example:



As with the Application Service Portal, it is possible to define the following from this screen:

- **Expiry Date** – Date this access will expire. Use the Date Icon ( ) to use the Date selection widget.
- **Access Mode** – Valid Access mode as defined on Application Service definition.
  - Use the to add a new Access Mode. The Search icon ( ) can be used to find the existing Access Mode or it can be typed in.
  - Use the to remove an existing Access Mode from the list.
- **Owner –** Ownership of link (refer to Data Ownership Rules)
- **Security Type** – Security Type code associated with Application Service.
  - Use the to add a new Security Type. The Search icon ( ) can be used to find the existing Security Type or it can be typed in.
  - Use the to remove an existing Security Type from the list.
- **Authorization Level** – The Authorization Level assigned to this User Group when executing this Application Service for the Security Type. The Search icon ( ) can be used to find the existing Authorization Level or it can be typed in.

Additional from the User Group maintenance screen it is possible to manage the users associated with this user group. The *Users* tab is used to define this information. For example:



The screen allows users to be associated with the user group with the following information:

- **User** – This is the authorization user identifier to be connected to the user group. The Search icon ( ) can be used to find the existing User or they can be typed in
- **Expiration Date** – Date the association between the user and user group will expire. Use the Date Icon ( ) to use the Date selection widget.

- **Owner** - Ownership of link (refer to Data Ownership Rules).

Use the + to add a new User or use the − to remove an individual user from the list.

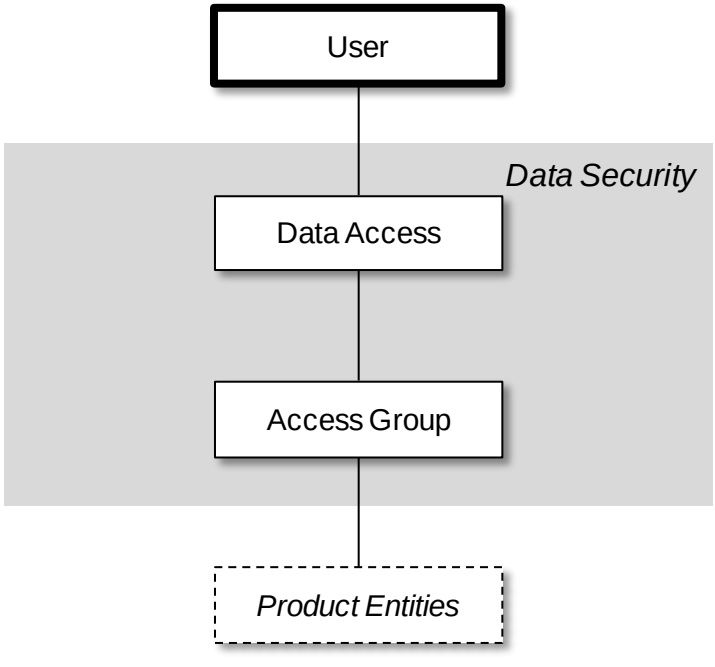
Define Users to Data Access Groups

Note: Not all products support Data Access Roles and Data Access Groups. Refer to the online Administration Guide for more details.

Data Access Groups are used to define the subset of data objects the user is permitted to access. There are two levels to the definition of data access:

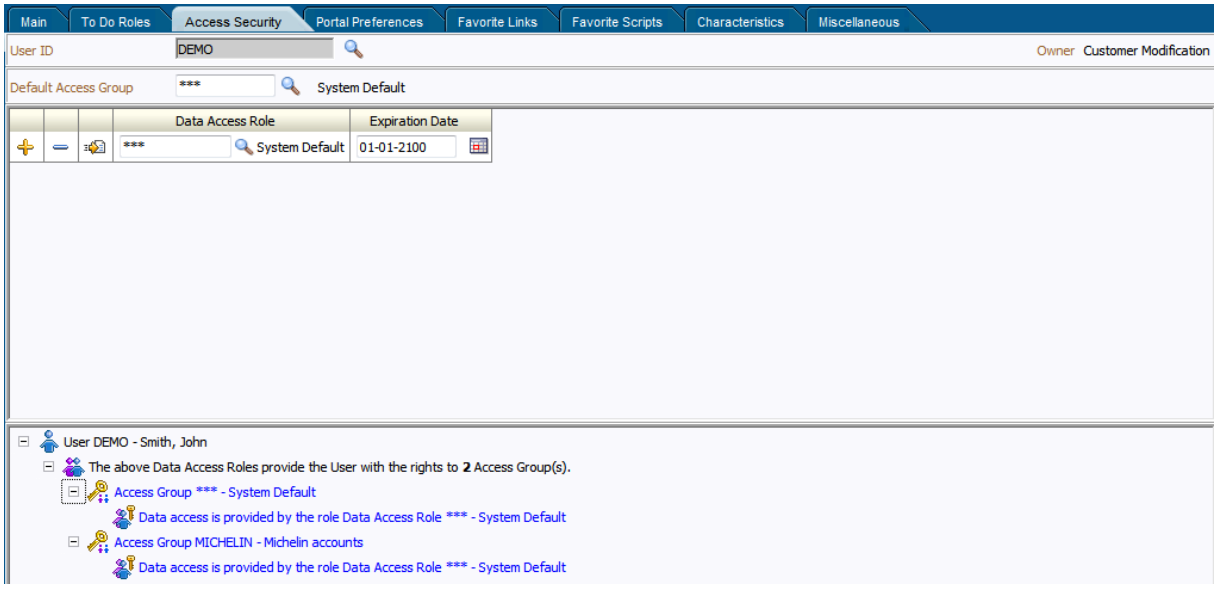
- **Data Access Roles** – User are connected to Data Access Roles which defines the groups of data permissions the user has access to. Data Access Roles are connected to Data Access Groups (a.k.a. Access Groups).
- **Data Access Groups** – Data Access Groups are tags that are attached to entities in the product to implement data security[3]. Data Access Groups are maintained using Access Group maintenance. Refer to the online Administration Guide for more details of this facility.

The relationships between these objects are illustrated in the figure below:



To maintain the Data Access Roles and Access Groups a user has access to, navigate to the *Access Security* tab of the user maintenance function. This will display a screen similar to the following:

[3] Attaching a Data Access Group to a product entity it does not automatically implement data security. Queries for that object must be altered to take into account the Data Access Group. Refer to the Oracle Utilities SDK for more details.



The screen will allow the definition and display of the following information:

- **Default Access Group** – When this user creates a new object that is subject to Access Security then this default is used for the value of the Access Group of the new object. This can be overridden by logic within the object if necessary.
- **Data Access Role** – List of Data Access Roles this user is attached to. The Search icon ( ) can be used to find the existing Data Access Role or they can be typed in
- **Expiration Date** - Date the association between the user and data access role will expire. Use the Date Icon ( ) to use the Date selection widget.

Use the to add a new Data Access Role or use the to remove a Data Access Role from the list.

User Enable and Disable

One feature of security is that the user record is attached to some objects for audit purposes (some objects are automatic, such as financials, and some are configurable). When a user does any work in the product and the user has been attached to some audit object across the whole product, the user cannot be deleted. This is due to auditing requirements.

There is a feature on the user object to enable or disable a user by setting the appropriate value for **User Enable** on the User object. This has the following implications:

User Enable	Implications
Enable	• User can access system. • User can process records according to the authentication model. • User must be active in Security repository to fully access the product.
Disable	• User cannot access the system regardless of other security setup • User record is retained for audit purposes only. • User does not have to exist in the Security Repository.

This facility has a number of key use cases:

- **Support for personnel (permanent or temporary) leaving** – It is possible to manually disable users once they leave the organization yet keep the
- **Logical deletion** – If the user needs to be deleted for any reason then setting Disable
- **Temporary disablement** – If business rules need to isolate users then setting the **User Enable** for appropriate users can effectively disable them from the product.

Note: When a user is disabled, it will apply when the user next attempts to login or when the security cache is refreshed.

Managing Batch Users

Each time a batch process is executed the security components of the product must authenticate the user against a security repository and authorize the user to access the components the batch process needs to complete its operations.

The batch component of the architecture uses a number of security mechanisms:

- **Authorization** – Any batch users must be defined to the operating system configured security repository and be a member of the operating system group assigned to the product.

  Note: The userid used does not have to match the authorization user used within the product.

- **Authorization** – The authorization user is defined within the product as per the online users and is specified as a job parameter at execution time or in configuration files supplied for the batch process. Refer to the Batch Administration Guide for details of the parameters used for batch processing.

To manage batch user therefore the following is recommended:

- Add the authentication user used to initiate the threadpool and submitter processes for a batch process to the configured operating system repository.
- Specify a valid user authorization identifier as a parameter for the batch process. This identifier must be authorized to the valid actions against the main objects used in the batch process. Refer to the product functional documentation on the objects used in each of the product batch processes.

Managing Web Services Users

Note: Native Security support is only supported for XAI Inbound Services using the ***BusinessAdapter*** *Adapter.*

From a product perspective a Web Service is a channel into the objects within the product. Any of the objects, services and scripts available in the product can be exposed as JAX-WS 2.0 based Web Service. From a security perspective Web Services uses the following security mechanisms:

- **Authentication** – The Web Services component of the product uses the Web Services support native to the J2EE Web Application Server. This allows security tokens supporting many standards to be used to authentication individual web service calls.
- **Authorization** – The Web Services component uses the same authorization model as the online user and batch components use.

Note: The user for authentication is used to map to the authorization user within the user object in the same way that online users are mapped.

To manage Web Services security users the following is recommended:

- Users for authentication are added to the security repository configured with the J2EE Web Application Server. This should match the Login Id used for the authorization model.
- Security Policies need to be attached to Web Services using the J2EE Web Application Server. For Oracle WebLogic the security policies available using Oracle Web Services Manager is available for use with individual Web Services. Multiple policies are supported. Refer to the *Oracle Fusion Middleware Security and Administrator's Guide for Web Services* for more information and the policies available.
- Users must be defined to the authorization model with appropriate access to underlying services used by the Web Service. For Web Services based upon Business Objects, Business Services and Scripts, users need appropriate access to the Application Service defined on these objects.
- Transaction Types in the Web Services translate to Access Modes within the Application Service calls.

For more information about Web Services, refer to the *XAI Best Practices* (Doc Id: 942074.1).

Authentication User

In the user object there are two different user identifiers namely Userid and Login Id. The different identifiers have differing roles:

- The Userid, which is up to 8 characters in length, is used internally for authorization and is passed to the database connection as the **`CLIENT_IDENTIFIER`** on the database connection. This user cannot be changed after the user has created any records in the system as it is used for record ownership in some objects and in auditing.
- The Login Id, which is up to 256 characters in length, is used for authentication to the security repository configured on J2EE Web Application container. The Login Id can match the Userid but can differ to reflect site standards. Unlike the Userid the Login Id can changed at anytime to reflect changes in the organization such name changes or acquisition.

Note: The Login Id must match, in the same case, as the entry in the configured security repository for the J2EE Web Application Server.

When maintaining a user, it is important that the Login Id is only changed using the maintenance function, LDAP Import or any XAI/Inbound Web Service based upon the USER object and not directly using other means (such as direct SQL) as a Security Hash is generated at maintenance time and is checked at login time. At application login time, if the security hash does not match the user is deemed not authorized and will be refused access to the product. To ensure security hashes are correct use the Synchronize Data Encryption function to reset the user security hash.

Advanced Security

About Advanced Security

While the default security settings are adequate for most sites, there are a number of additional advanced settings that can be configured to support a wider range of security requirements. This section outlines the various security settings available and the configurations supported.

J2EE Authentication Group

The default installation of the product includes a default authentication group (**role-name**) defined within the J2EE Web Application web descriptor (web.xml). This role name is used by the J2EE Web Application to link the authorized users within the product to the associated J2EE physical resources (i.e. pages, configuration files) within the J2EE Web Application Server. The specification of the group in the web descriptor is in the security section[4]. For example:

```
<security-role>
   <description>OUAF Users</description>
   <role-name>cisusers</role-name>
</security-role>
```

By default, this group is set to **cisusers**, which is configurable for each web component. When the product is deployed to the J2EE Web Application Server, this group is instantiated ready to be allocated to individual users. Users of the product must be attached to this group to use the product.

From a configuration point of view there are a number of options for this setting:

- The default group may be changed at installation and configuration time using the configuration settings as shown below as outlined in the *Server Administration Guide*. The group name should have no embedded blanks.

Component	Principal Name	Role Name
Online/Help	WEB_PRINCIPAL_NAME	WEB_ROLE_NAME
AppViewer	WEB_APPVIEWER_PRINCIPAL_NAME	WEB_APPVIEWER_ROLE_NAME

- If the J2EE Web Application Server is configured to use an external security repository the configured administration group must exist in the security repository and the users must be connected to this group.

Note: If the J2EE administration group is changed after installation time, users will need to be migrated to the new J2EE administrations group either manually, using tools provided with the security repository or J2EE Web Application Server.

[4] The security role is used in a number of sections of the web application descriptor.

Logon Configuration

The default configuration for online authentication is using a logon screen for the online product, online help and online AppViewer applications. The product supplies a prebuilt logon screen for all three components preconfigured.

At logon it detects that a user has not logged on before (the presence of a JSESSIONID cryptographically-secure session cookie issued by the Web Application Server is used). Depending on the configuration (in the web.xml) of the applications, housed in the J2EE Web Application Server, the following is performed:

- **FORM** – This is the default setting to support a logon screen with an associated error screen in case of unsuccessful logon. The product provides a prebuilt logon screen but can be replaced with custom logon screens[5] by setting the following configuration settings appropriately for each web component as outlined in the Server Administration Guide:

Component	Login Screen	Login Error Screen
Online	WEB_FORM_LOGIN_PAGE	WEB_FORM_LOGIN_ERROR_PAGE
Help	WEB_HELP_FORM_LOGIN_PAGE	WEB_HELP_FORM_LOGIN_ERROR_PAGE
AppViewer	WEB_APPVIEWER_FORM_LOGIN_PAGE	WEB_APPVIEWER_FORM_LOGIN_ERROR_PAGE

- **BASIC** – The browser will issue a call to the operating system to display the default logon dialog supplied with the operating system. No logon dialog is supplied.
- **CLIENT-CERT**[6] - This is an advanced configuration to allow for certificated (one way or two way) to be used. Refer to the documentation supplied with the J2EE Web Application Server for more details of the additional configuration required.

Data Ownership Rules

On each of the objects (and on selected child objects) an owner flag is included to determine the origin of the data. The owner is used by the product to determine the maintenance owner of key data as well as protect important data shipped with the product from accidental deletion.

The value of the flag is displayed on maintenance screens to visually indicate the data owner. The location of the information varies from the top left of maintenance pages, within lists of information (to apply to individual rows) and within sections of maintenance pages.

The flag has a number of valid values:

- **Base** – This is important information shipped with the product and cannot be deleted or modified using the delete or medication functions, respectively, regardless of user permissions. This is reserved for use by the product to ship key important information and to protect that information. Deletion of this information directly

[5] Custom logon screens should be placed in the **cm** directory of the web application server as outlined in the Oracle Utilities SDK.

[6] **CLIENT-CERT** is supported but requires manual changes to configuration files. Refer to the Server Administration Guide on implementing custom templates.

from a product database will result in unexpected results.

- **Product Name** – The product name that owns the data is displayed. This is similar to the **Base** data ownership value but indicates which component the data is applicable to. All the rules that apply with the **Base** data ownership value apply to this value.
- **Customer Modification** – This indicates that the data was added by the implementation using the various methods and is owned by the implementation. Deletion of data is permitted using the valid deletion functions for authorized users.

In general sites, can only maintain Customer Modification owned records. Other ownership values are reserved to protect product installation supplied data.

Configuring JMX Security

The operations interface to the product is based upon Java Management Extensions (JMX) allowing components of the product to be managed and monitored from JSR160 compliant consoles including jconsole or Oracle Enterprise Manager.

Refer to the *Server Administration* and *Batch Server Administration Guides* for more details of the JMX operations interface.

By default the JMX implementation and configuration uses the default simple file based security as outlined in the JMX specification.

Default Simple File Based security

The default configuration is based upon a properties file containing name/value pairs corresponding to role/password pairs and authorization can be also based on a properties file containing name/value pairs corresponding to role/access pairs where access can be any of **readonly** access which grants read access to any remote operation and **readwrite** access which grants access to read and update operations in the interface.

*Note: By default the user (**BSN_JMX_SYSUSER**) and password (**BSN_JMX_SYSPASS**) for the administrator are automatically added to the configuration files.*

To use this facility the following file should be maintained using an appropriate editor (located in **$SPLBASE/scripts** directory or **%SPLEBASE%\scripts** in Windows):

- **ouaf.jmx.access.file** – This file contains the userid and access permissions in the format separated by a *blank* space:

Field	Comments
Userid	Authentication user to access JMX.
Permission	Permission assigned to user. Valid values are: **readonly** – No update access and **readwrite** – Update access and can access update operations

- **ouaf.jmx.password.file** - This file contains the userid and password in the format separated by a *blank* space:

Field	Comments
Userid	Authentication user to access JMX

Field	Comments
Password	Password in plain text or encrypted format.

Note: These files are also tailored using custom templates. The ***ouaf.jmx.access.file.template*** *and* ***ouaf.jmx.password.file.template*** *are used for the configuration.*

SSL based Security

Note: For a full description of SSL setup refer to the ***To Setup SSL*** *section of Monitoring and Management Using JMX Technology.*

To secure communications for JMX using the Java SSL support the following process needs to be performed:

- Security has to be setup using the Simple File Based Security or Using Other Security Sources.
- A key pair and certificate need to be setup on your server. Refer to the Monitoring and Management Using JMX Technology or J2EE Web Application Server Administration documentation for details and utilities available for this process.
- Set additional java parameters using the **WEB_ADDITIONAL_OPT** for the online/Web Services and **BATCH_MEMORY_ADDITIONAL_OPT** for Batch. Refer to the Server Administration Guide and Batch Server Administration Guide for details of these parameters. The following additional system properties must be set:

System Property	Comments
javax.net.ssl.keyStore	Keystore location
javax.net.ssl.keyStoreType	Default keystore type
javax.net.ssl.keyStorePassword	Default keystore password
javax.net.ssl.trustStore	Truststore location
javax.net.ssl.trustStoreType	Default truststore type
javax.net.ssl.trustStorePassword	Default truststore password
com.sun.management.jmxremote.ssl	Set to **true**
com.sun.management.jmxremote.registry.ssl	Set to **true**
com.sun.management.jmxremote.ssl.need.client.auth	Set to **true**

Note: Additional options are also supported as documented in Monitoring and Management Using JMX Technology.

Note: Specification of system properties for java are as per the java command line.

Note: For sites using Oracle WebLogic in native mode, configuration of SSL requires configuring Oracle WebLogic to use SSL and altering the startup scripts for Oracle WebLogic to include the above options.

Using Other Security Sources

Whilst, by default, the file based repository is supported it is possible to configure the authentication of JMX to use an alternative data source such as an LDAP Server. This involves changing the JAAS configuration stored in the **java.login.config** file **$SPLEBASE/splapp/config** directory (or **%SPLEBASE%\splapp\config** directory on Windows).

In the JAAS configuration file there is a default **jmxrealm** that contains the default JMX LoginModule. This can be changed, using custom templates, to support an alternative source for authentication. Refer to the **LdapLoginModule** for information and examples of login configurations.

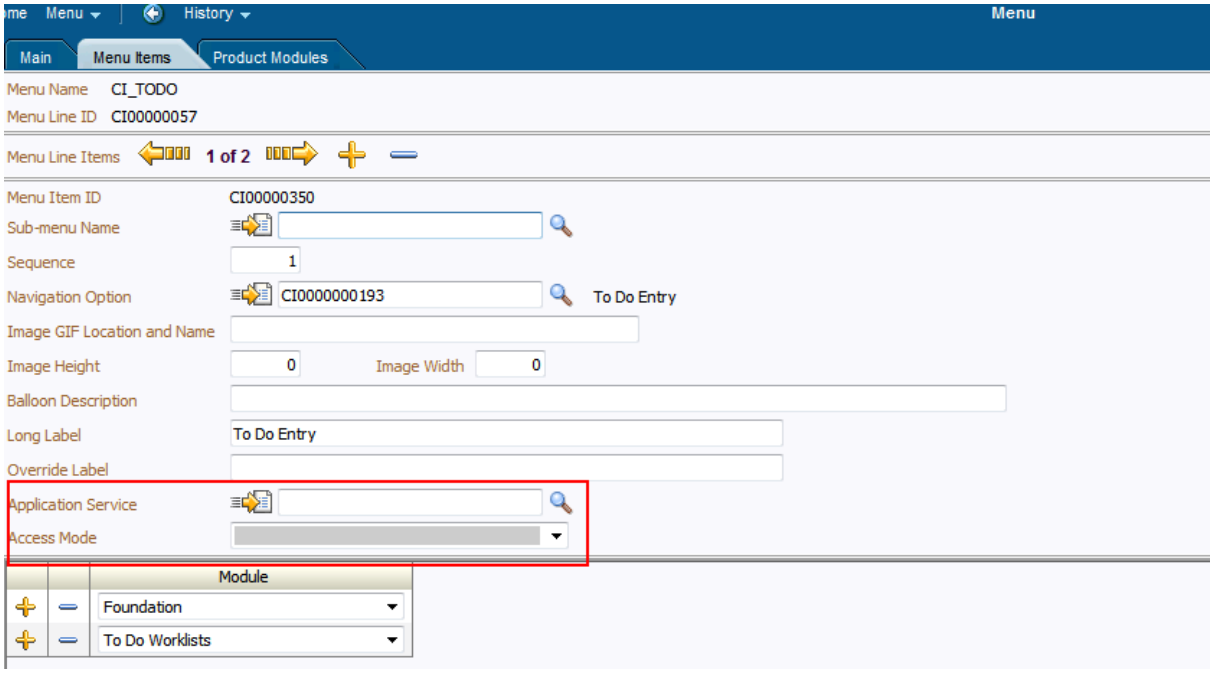
*Note: To implement the custom security source custom templates for **java.login.config** must be implemented according to the process outlined in the Server Administration Guide. This configuration affects all modes of access (i.e. online, Web Services and Batch).*

Menu Security Guidelines

By default, a menu option is displayed whenever a user has access to the underlying application service definition attached to objects that are indirectly linked to a menu entry. Whilst this behavior is sufficient for most needs, it is possible to place an override on an individual menu item to override the lower level security levels. This is particularly useful where implementations wish to replace base supplied menu items with custom menu items.

By linking a menu item to a new service that can reference the underlying objects and specifying an Application Service (optionally also including an Access Mode) would override the permissions on the underlying objects.

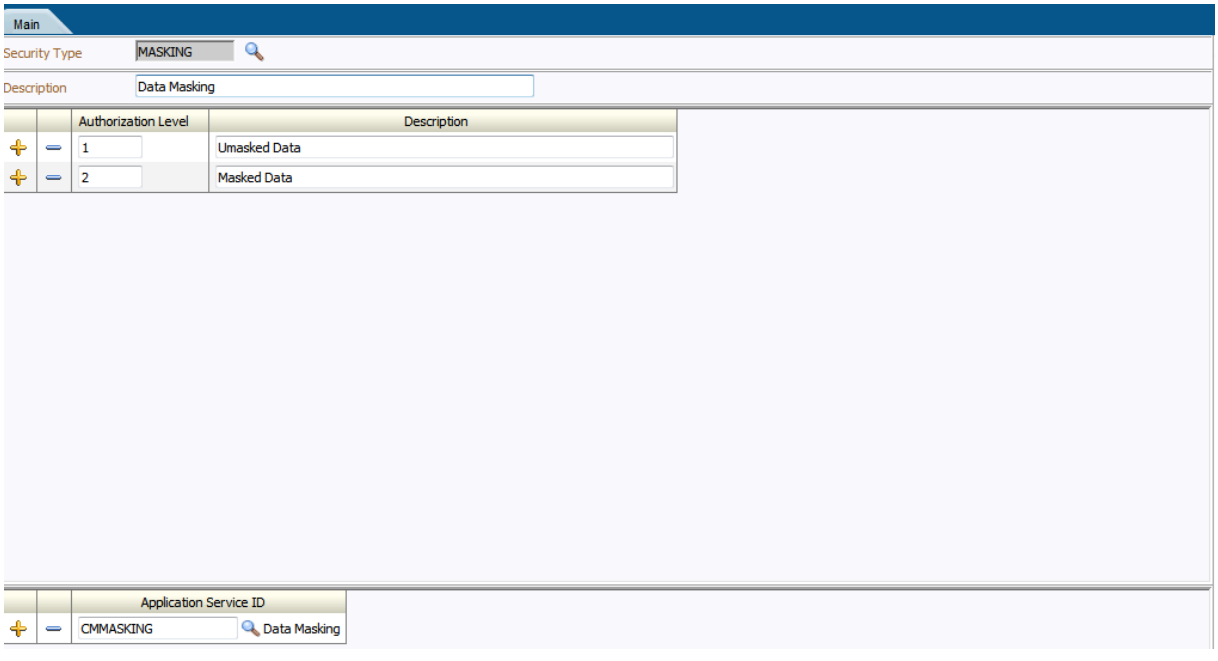
It is possible to specify the Application Service on a menu item on the *Menu Items* tab of the *Administration* → *M* → *Menu* option. For example:



Security Types

By default users have full access to the objects via the access methods specified in their user groups. If the implementation wishes to implement additional levels or rules then the application service must use Service Types. The definition of a Service Type allows additional tags to be attached to service definitions and then code written to detect and take advantage of the presence of the tag to limit security access to specific object data. For example, whether data is masked or not or some limit is placed on values of data.

To define Security Types, *Administration* → *S* → *Security Types* option to display the Security Types maintenance function. For example:



On this function define the following in relation to the Security Type:

- **Security Type** – Identifier for Security Code
- **Description** – A short description of the use of the Security Code.
- **Authorization Levels** – A list of codes (Authorization Level) and associated descriptions. Use the + to add a new Authorization Level or use the − to remove an existing Authorization Level from the list. The Authorization Level values are free format but should be representative of the desired function. The Description is used to explain the value.
- **Application Service Id** – A list of associated Application Services to use this Security Code. Use the + to add a new Application Service or use the − to remove an existing Application Service from the list. The Search icon (🔍) can be used to find the existing Application Server or it can be typed in.

Note: To fully implement the rules associated with the security types, code must be included in objects to implement security logic.

Default Generic Application Services

By default all a set of Application Services are defined against base functions. In line with data ownership rules, some of these records can be altered and new functions added. A set

of generic application services are also shipped with the product to provide a mechanism for defining new zones, new objects or new menu items for rapid deployment.

There are two generic Application Services that can be used to secure objects, zones and menu items:

- **F1-DFLTAPS** – This is a generic execution Application Service which is designed to secure zones and menu options. It only supports the *Execute* Access Method.
- **F1-DFLTS** – This is a generic maintenance Application Service which is designed to secure business objects. It supports the Add, Modify, Delete and Inquire Access Methods.

Use of these generic Application Services is optional.

Administration Delegation

By default, the product provides a single administration account, as configured in the **SPLADMIN** configuration setting, in the **ENVIRON.INI** configuration file, to manage the operational aspects of the product. This operating system user is the owner of the product when it is installed and is typically used for all operational aspects of the product.

Note: It is not possible to change the product administration account after installation. If this is desired it is recommended to remove the product and reinstall using the alternative administration account.

Whilst the single administration account is sufficient for most needs it is possible to provide additional administration accounts to delegate administration tasks. To delegate administration the following must be configured:

- Any administration user must be a member of the operating system group allocated to the product as outlined in the **SPLADMINGROUP** configuration setting in the **ENVIRON.INI** configuration file.
- If you are using Oracle WebLogic in embedded mode or using IBM WebSphere and using the **spl** utility to manage the startup and shutdown of the product then the utility permissions must be altered to set the *sticky* bit[7], using the **chmod +t** or **chmod +s** command, so that the utility must run as the product administration account.

  Note: Permissions on the directories are set to restrict the administration functions. Do not alter the permissions on individual directories and file unless otherwise directed.

- If you are using Oracle WebLogic in native mode, then the console will execute the native facilities to start and stop the product. It is recommended that the user allocated to Oracle WebLogic at installation time be a member of the operating system group outlined in **SPLADMINGROUP** configuration setting in the **ENVIRON.INI** configuration file.

Note: Customers using Oracle Enterprise Manager, with or without Application Management packs, should use the administration delegation and credential management capabilities of that product to manage administration delegations.

[7] Support for sticky bit varies from operating system to operating system

Secure Communications (SSL)

By default, the product uses HTTP for communication to the browser and across the tiers. The transport protocol can be encrypted using SSL/TLS to secure transmission of data across networks.

Note: Oracle strongly recommends that customers use SSL to secure transmission for production environments.

To implement SSL the following process must be completed:

- Configure the J2EE Web Application Server to use the SSL protocol. For Oracle WebLogic customers refer to *Configuring SSL* in *Oracle Fusion Middleware Securing Oracle WebLogic Server*.

  Note: Customers using IBM WebSphere or IBM WebSphere ND should refer to the WebSphere documentation on enabling the SSL protocol.

- Set the SSL Port Number using the **WEB_WLSSLPORT** configuration parameter as outlined in the product Server Administration Guide.
- Once the setup has been tested and verified refer to the J2EE Web Application Server documentation on disabling insecure protocols.

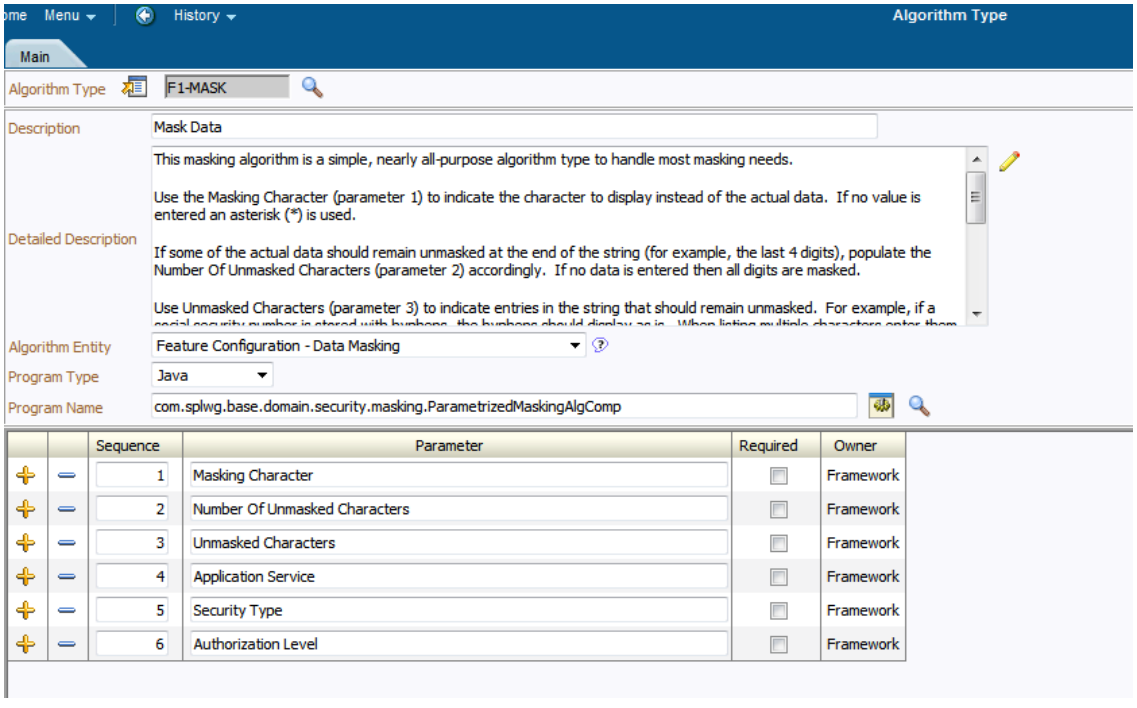
Data Masking Support

By default, the data in the product is unmasked for authorized users. If particular data within the object is considered a candidate for data masking then the masking capabilities with the product can be used to mask the data in an appropriate fashion.

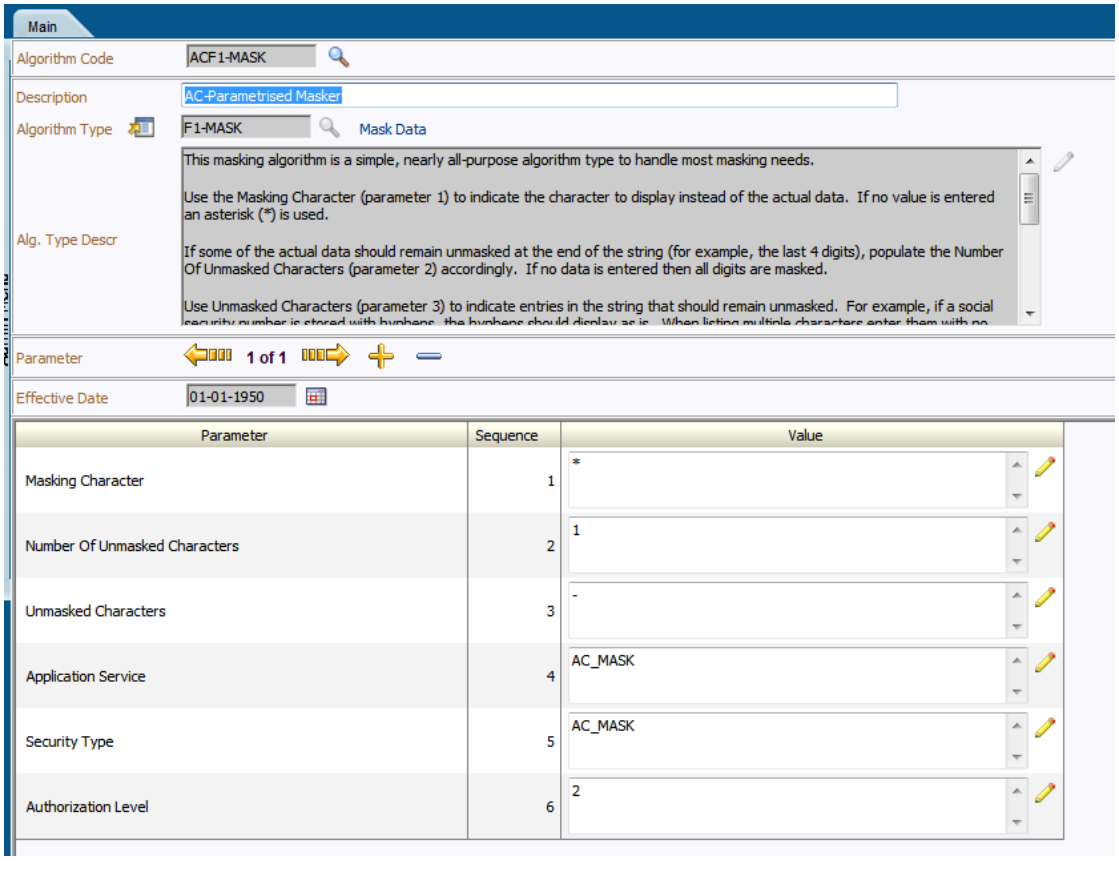
Note: The data is not stored in masked fashion; it is configured to be displayed in masked format for particular users using the Security Types.

To mask data using the internal data masking capability:

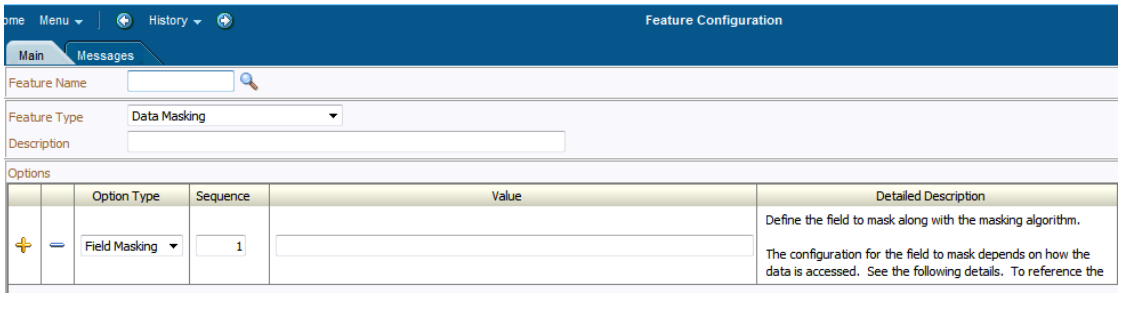
- An internal algorithm type of **F1-MASK** is supplied with the product to perform basic data masking. For example:



- The following parameters are applicable to the algorithm:
  - **Masking Character** – The character to be used as a mask. By default the ***** character is used.
  - **Number of Unmasked Characters** – Number of suffix characters to unmask. Commonly the last x characters are displayed unmasked to allow some identification. A value of zero masks all characters.
  - **Unmasked Characters** – A list of characters, with no spaces, to leave unmasked. Commonly, this is used to denote delimiter characters to enhance recognition.
  - **Application Service** – Application Service used for security authorization checking. This allows global (or local) services to be configured to indicate security access to data masks.
  - **Security Type** – The Security Type used to flag which users will view the data in masked on unmasked format. User groups need to be connected to the Application Service, Security Type and given the Authorization Level to determine the level of data masking.
  - **Authorization Level** – The authorization level used to determine if a user has access to the data unmasked. All other authorization levels in the security type indicate masked data.
- Configure an Algorithm entry of Algorithm Type **F1-MASK** for the desired masking configuration. Algorithm entries can be shared across fields to be masked using *the Administration → A → Algorithm* option. For example:



- Attach user groups to the Application Service with the appropriate Authorization Level for the Security Type.
- Create or update a Feature Configuration of Feature Type **`Data Masking`** using the *Administration → F → Feature Configuration* option. For example:



- For each field to mask add an entry to the Options section of the Feature Configuration with the following values:
  - **Option Type** – Select **`Field Masking`** for Data Masking.
  - **Sequence** – Specify a sequence number for order of evaluation.
  - **Value** – Specify the tag string, delimited by "**`,`**", to indicate the definition of the data masking with the following tags depending on how the data is accessed:
    - Only fields defined as strings are supported by the supplied algorithm.
    - To reference the masking algorithm, enter **`alg="`*`algorithm name`*`"`**. The algorithm's algorithm type must reference the Data Masking algorithm entity.

- For data that is accessed via a schema-based object call, the field to be masked must reference a meta-data field name in its schema definition. For example, if you want to mask a credit card number, let's assume that field is defined in the schema is `<creditCard mdField="CCNBR" mapField="EXT_ACCT_ID"/>`. In this case, the option value should be `field="CCNBR", alg="algorithm name"`.
- For data that is accessed via a page maintenance service call, indicate the table name and the field name where the data resides, for example `table="table_name", field="fld_name", alg="algorithm name"`.
- A `WHERE` clause may also be specified. This is useful for data that resides in a child table where only data of a certain type needs to be masked. For example `table="CI_PER_ID", field="PER_ID_NBR", alg="algorithm name", where="ID_TYPE_CD='SSN'"`
- For data that is stored as a characteristic, simply indicate the characteristic type `CHAR_TYPE_CD='char type', alg="algorithm name"`. This needs to be defined only once regardless of which characteristic entity the char type may reside on

Note: Only ad-hoc characteristics are supported at the present time.

- For data that is displayed via a search service call, indicate the search name and the appropriate field to mask along with the masking algorithm. For example: `search="SearchServiceName", field="PER_ID_NBR", where="ID_TYPE_CD='SSN'", alg="algorithm name"`. To find the name of the search service, launch the search in question, right click in the filter area and choose *View Source*. Search for *ServiceName*. The service name is listed there. To find the field name to mask, go back to the search window and right click on the results area and choose View Source. Look for the Widget Info section and find the field name in the SEARCH RESULTS (do not include the $).

Note: The `WHERE` statement can only apply to fields that are also part of the search results.

- Use the + to add a new Data Masking definition or use the − to remove an existing Data Masking definition from the list.

Securing Files

Note: The utilities mentioned in this section apply to Linux and Unix environments only.

The product file structure is protected by permissions set at the operating system level. By default, the settings provided with the product upon installation comply with Oracle standards in respect to permissions. For more details of the individual user permissions on product directories and subdirectories, refer to the product *Server Administration Guide*.

If at any time, the permission are manually altered and need to be reset to the defaults then

the following process can be used:

- Execute the **splenviron.sh** utility to set the environment variables for the product environment to reset. Refer to the product *Server Administration Guide* for details of this process.
- Execute the **setpermissions.sh** utility to reset the environment permissions back to the defaults.

The environment permissions will be reset to the defaults supplied with the product.

Password Management

On a regular basis passwords are changed to maintain security rules. The product uses a number of passwords that may require changing on a regular basis. The following table lists all the passwords used in the product and guidelines for changing the password values used by the product.

Password Owner	Location	Comments
Online User	J2EE Authentication Source	No configuration changes. User changes password in security repository directly or indirectly using security products. Security repository is configured in J2EE Web Application Server[8].
Web Service User	J2EE Authentication Source	No configuration changes. User changes password in security repository directly or indirectly using security products. Security repository is configured in J2EE Web Application Server.
Batch User	Operating System	No configuration changes. User changes password in security repository directly or indirectly using security products. Security repository is configured in operating system.
Database Users	**BATCH_DBPASS** **DBPASS** **XAI_DBPASS**	The database users are stored in **ENVIRON.INI**. Refer to the *Server Administration Guide* on process to change values. New Passwords need to be re-encrypted.
JMX Users	**BSN_JMX_SYSPASS**	The default JMX user is stored in **ENVIRON.INI**. Refer to the *Server Administration Guide* on process to

[8] **WEB_SPLPASS** specifies the default password for the initial user. If this user is used past the installation the password may need to be changed. Refer to the *Server Administration Guide* for more details.

Password Owner	Location	Comments
		change values. New Passwords need to be re-encrypted.
J2EE Administration Account	**WLS_WEB_WLSYSPASS** **WEB_WLSYSPASS**	The default administration users are stored in **ENVIRON.INI**. Refer to the *Server Administration Guide* on process to change values. New Passwords need to be re-encrypted.

Securing Online Debug Mode

The product features an online debug mode which is used for problem solving and development personnel to trace their code or diagnose problems. As with other functions within the product the debug function is security controlled.

To use this facility any of the user groups an individual user must include *Inquire* access to the **F1DEBUG** application service. This will enable the debug facility from the URL.

For more information about the Debug facility refer to the *Server Administration Guide*.

Securing Online Cache Management

The product features an online cache management facility which is used to reset the online cache to force new values to be loaded. As with other functions within the product the cache management function is security controlled.

To use this facility any of the user groups an individual user must include *Change* access to the **F1ADMIN** application service. This will enable the cache management facility from the URL.

For more information about the cache management facility refer to the *Server Administration Guide*.

Audit Facilities

The product has an inbuilt auditing capability to register accesses to data from online and Web Services users. Batch processing is not audited by default but can be enabled using the Oracle Utilities SDK using programmatic methods.

About Audit

Auditing allows for the configurable tracking of changes to key data by online and Web Services users. The product has an inbuilt, configurable audit facility that tracks changes and allows authorized users to track changes on an individual user and change basis.

The use of the inbuilt audit facility is optional and can be enabled or disabled at any time.

Audit Configuration

Note: This section covers the soft table implementation of Auditing. There is also specialist Audit algorithm support on Business Objects and Maintenance Objects to add information to log entries attached to these objects. Refer to the Oracle Utilities SDK and online Administration documentation for a description of programmatic implementation of Auditing.

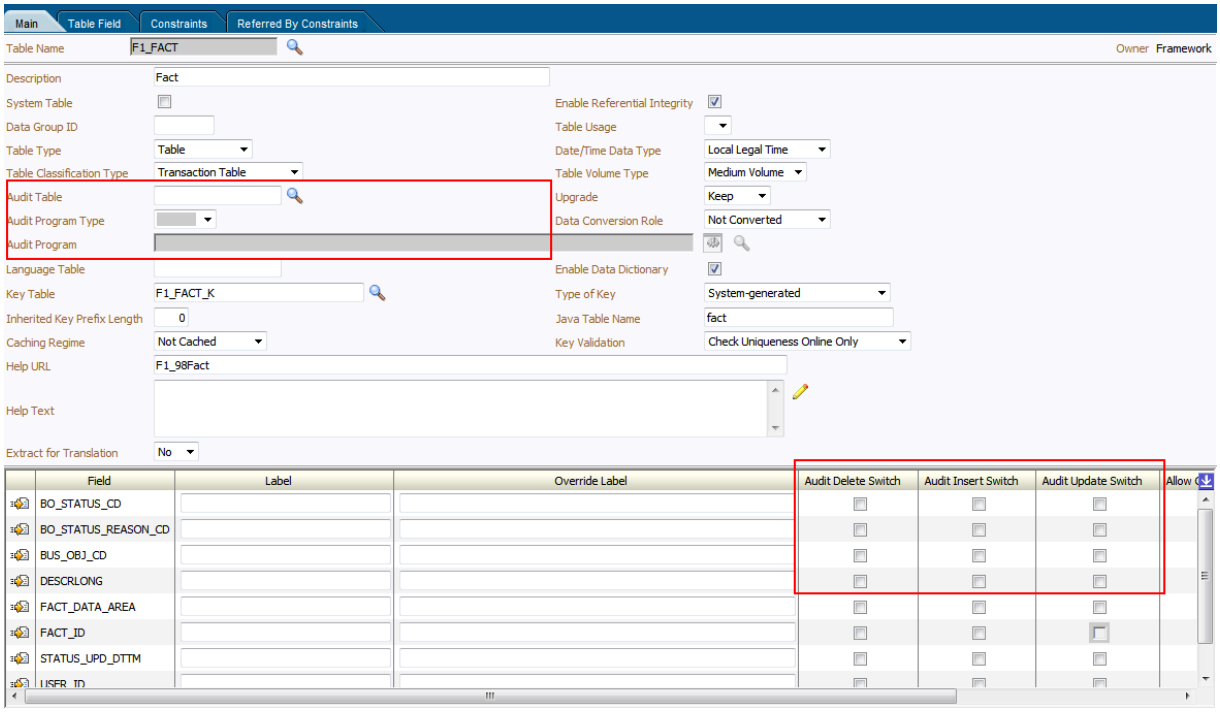
The inbuilt Audit facility is configured at a table level. For each table you wish to enable audit upon the following needs to be configured:

- **Audit Table** – To store the audit information a database table must be configured to hold the audit information. By default, the `CI_AUDIT` table can be used for this purpose. If a custom table is used to store the information it should have the same structure as `CI_AUDIT` for compatibility purposes.
- **Audit Program** – To process the audit information a class or program must be configured to record the audit information. By default a number of prebuilt Audit programs are available for use:
  - `com.splwg.base.domain.common.audit.DefaultTableAuditor` – This is the default java based audit class provided by the product. It audits any changes to any fields configured to track auditing information.
  - `com.splwg.base.domain.common.audit.ModifiedTableAuditor` – This is an alternative to the `DefaultTableAuditor` but it will not audit inserts or deletes of empty string field data. For example, changes from null values to empty spaces and vice versa would not be logged.
  - `CIPZADTA` – For backward compatibility purposes, products which use COBOL based extensions can use a COBOL version of the `DefaultTableAuditor`. It is recommended for customers to use the java version in preference to the COBOL version.

  Note: It is possible to implement custom Audit handlers using the base classes as parent classes. Refer to the Oracle Utilities SDK documentation on how to extend the product.
- **Audit conditions** – A set of switches are configurable on each field you wish to include in auditing to determine the conditions of auditing. At least one of these switches must be enabled for auditing to be registered:

- **Audit Delete Switch** – Enable this switch to audit delete operations against this field.
- **Audit Insert Switch** – Enable this switch to audit insert operations against this field.
- **Audit Update Switch** - Enable this switch to audit update operations against this field.

To maintain the audit information, navigate to the *Administration → T → Table* option and specify the table to enable auditing against. For example:

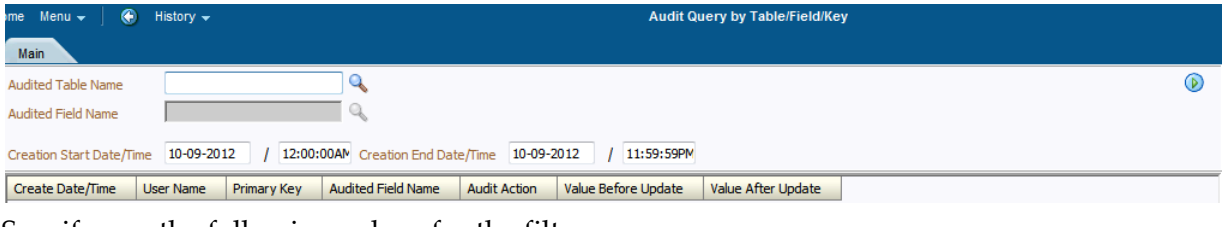


Specify the Audit Table, Audit Program (and associated type) and configure the Audit Switches on the fields you wish to track.

Note: To enable Auditing on a running version of the product, the online data cache must be flushed or the product restarted. Refer to the Server Administration Guide for more details.

Audit Query by Table/Field/Key

Once Auditing is enabled changes are logged in the configured Audit Table using the Audit Program specified in the configuration. It is possible to query this Audit information by Table, Field and Key value to isolate changes. To access this query navigate to the *Administration → A → Audit Query By Table/Field/Key* option. For example:



Specify any the following values for the filters:

- **Audit Table Name** – The name of the table that has been audited to query. When this table is chosen the screen will list additional fields to filter upon.

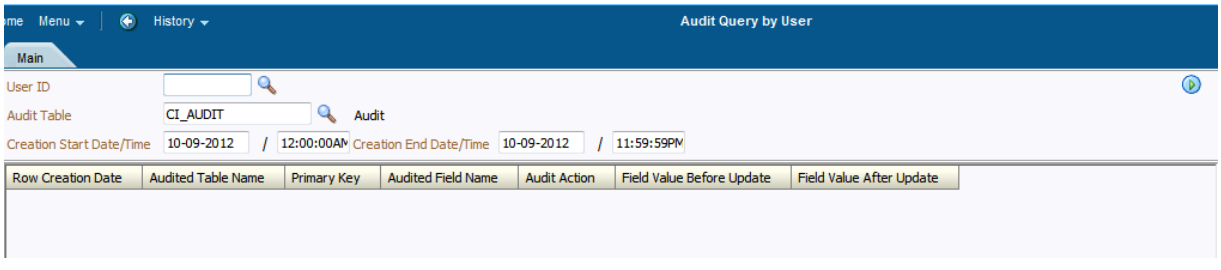
- **Audit Field Name** – The name of the field to track to filter the results.
- **Creation Start Date/Time and Creation End Date/Time** – Date and time range to limit the records returned.

The query will return the following results:

- **Create Date/Time** – Date and time the changes were made.
- **User Name** – Name of user who made the changes.
- **Primary Key** – Record key of the change.
- **Audited Field Name** – Name of field that was changed.
- **Audit Action** – Action that was recorded with change (i.e. Insert, Update or Delete)
- **Value Before Audit** – The field value before the change was made.
- **Value After Audit** - The field value after the change was made.

Audit Query By User

Once Auditing is enabled changes are logged in the configured Audit Table using the Audit Program specified in the configuration. It is possible to query this Audit information by individual users to isolate changes made by that user. To access this query navigate to the *Administration → A → Audit Query By User* option. For example:



Specify any the following values for the filters:

- **User ID** – Authorization User to track.
- **Audit Table** – The name of the table containing the audit information.
- **Creation Start Date/Time and Creation End Date/Time** – Date and time range to limit the records returned.

The query will return the following results:

- **Row Creation Date** – Date and time the changes were made.
- **Audited Table Name** – Name of table that was audited.
- **Primary Key** – Record key of the change.
- **Audited Field Name** – Name of field that was changed.
- **Audit Action** – Action that was recorded with change (i.e. Insert, Update or Delete)
- **Field Value Before Audit** – The field value before the change was made.
- **Field Value After Audit** - The field value after the change was made.

Read Auditing

Whilst the inbuilt Audit facility is mainly used to register changes in data, it can also be used to register whenever data is accessed for auditing purposes. The concept of read auditing is

different from the standard auditing as it is related to zones[9]. On the zone configuration there is an ability to configure an Audit Service Script which is called whenever the zone is displayed to determine which criteria and result records are displayed.

The information audited can be programmatically determined and which information is logged according to your requirements. Refer to the online zone help for descriptions and samples to configure Read Auditing.

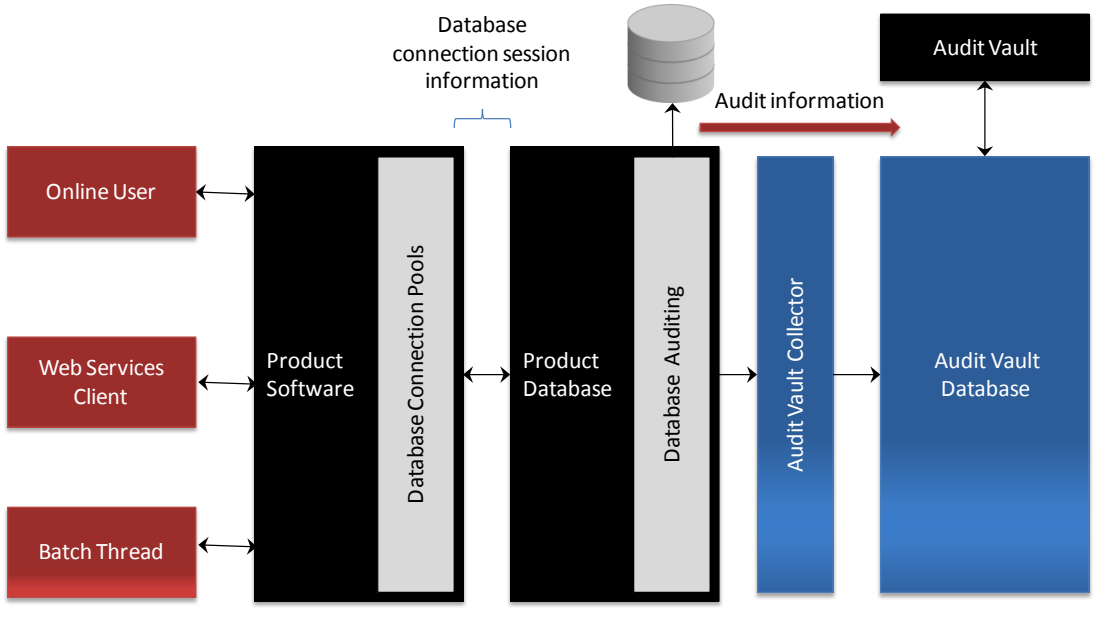
Note: Products ship with sample generic inquiry Audit code specific to the product. These can be reused or altered to suit your needs. Refer to the product documentation for details of these samples.

Integrating to Audit Vault

The Oracle Utilities Application Framework contains an internal audit facility that provides a basic audit facility for recording changes and optionally, inquiring, data by online users. Whilst this facility is sufficient for most needs it may be replaced with using Oracle Audit Vault to provide an enterprise wide audit facility.

Oracle Utilities Application Framework supports the use of Audit vault in association or as a complete replacement for the inbuilt Auditing feature.

Audit Vault collects audit information at the database level, using the Database Auditing features of the Oracle Database, and loads them into a separate Audit Vault database. The information in that database can be queried, reported and managed using the Audit Vault front end. For example:



To use Audit Vault the following must be configured:

- **Setup Database Auditing** - The Database auditing feature must be enabled to store the relevant audit information. The level of auditing information and the location of the audit information is configurable. Refer to the Oracle Database Security Guide for a discussion of Database Auditing and Best Practices of Auditing for a discussion of the various methods available.

[9] At the present time this parameter is available for **F1-DE**, **F1-DE-QUERY**, **F1-DE-SINGLE**, **F1-MAPDERV** and **F1-MAPEXPL** zone types only.

- **Design Database Auditing** - The tables, users and SQL statements to audit need to be specified on the product database. This typically done by the database administrator using the **AUDIT** statement.
- **Install and Configure Audit Collector** - On the host holding the product database an Audit Vault Data Collector needs to be configured to pass audit information to Audit Vault and implement data retention policies for audit information.
- **Configure Audit Vault** - Audit Vault can be configured to implement policies, alerts and reports on the Audit data. Audit Vault can be configured to set an Audit Data Retention Policy for its internal audit information.

Database Security

About Database Security

The Oracle Database supports a wide range of security configurations natively or via additional options available. For a full discussion of the available security options for the database refer to the *Oracle Database Security Guide*.

Database Users

The product installation ships with a predefined set of users to be used by the product at configuration and runtime. These users are specified in the installation of the product to build the database and load its initial dataset.

The following users are available:

- **SPLADM** – This is the default DBA administration account which owns the product schema. This user is used to create and maintain the structures of the database. It is used by DBA personnel to maintain the product schema and indexes.
- **SPLUSER** – This is the default main product user used by the product to access the **SPLADM** schema. The product uses this physical userid as a pooled user with pooled connections to the database. Variations on this account can be created for each channel of access using the following configuration settings

Configuration Parameter	Comments
BATCH_DBUSER	Database User for Batch
DBUSER	Database User for online (Default: **SPLUSER**)
XAI_DBUSER	Database User for Web Services

- **SPLREAD** – This is the default read only user available for reporting tools or external direct interfaces to use on the product database. This user is not used by the product[10].
- **CISOPR, OPRPLUS** – These are optional operator users that can be used to delegate backup and restore operations on the product.

Note: The values of these users can be altered to customer specific values at installation time. Refer to the product Installation Guide and product DBA Guide for more information.

Database Roles

The product ships with a set of database roles to allow administrators to allocate new database users to the relevant components of the product. The following roles are shipped by default for the product:

- **SPL_USER** – This role is available for database users who require update, insert,

[10] For customers on older versions of particular products this user was also used for the ConfigLab component.

delete and select access to the product schema. This role is used for product users.

- **SPL_READ** – This role is available for database users who require read only access to the product schema.

To use the roles the DBA grants the role to the database user to connect them to the schema in the desired fashion.

Database Permissions

Database permissions for the product are allocated at the role level with the role setting permissions to the schema objects. By default the roles have full access to all the objects in the product schema, as dictated by the role.

Unless otherwise stated, it is not recommended to alter the database users used by the product to specific additional permissions on the product schema as this may cause permission issues.

Customers wishing to restrict external parties, such as external tools or reporting engines, to specific objects may use all of the desired security facilities available in the database to implement those restrictions.

Using Transparent Data Encryption

Transparent Data Encryption (TDE) allows data to be encrypted at the storage level to protect the data files at the lowest level. From a product perspective, the implementation of Transparent Data Encryption requires no product configuration changes on the application server.

Note: To implement Transparent Data Encryption, DBAs will have to execute appropriate alter statements on product tables to indicate the level of encryption.

Note: For product tables with large amounts of data it is recommended to use the NOMAC feature to save disk space.

For more information about implementing Transparent Data Encryption refer to the *Oracle Advanced Security Guide*.

Using Database Vault

By default, the database administration account as SQL Data Manipulation Language (DML) access to the product schema, as dictated by the default permissions of the Oracle Database. It is possible to restrict the permissions of the DBA to SQL Data Definition Language (DDL) statements only using Database Vault. Refer to the *Database Vault Administrators Guide* for details of this facility.

The product includes a prebuilt database vault solution, refer to the *Database Vault Integration (Doc Id: 1290700.1)* available from My Oracle Support.

Security Integration

About Security Integration

Whilst the product provides a set of security facilities natively or via the J2EE Web Application Server, it is possible to augment the security with additional security features or security products.

LDAP Integration

By default, Oracle WebLogic includes an internal security repository that uses the Lightweight Directory Access Protocol (LDAP) to provide authentication facilities[11]. It is possible to replace the internal security repository with another LDAP compliant security source.

To use an alternative source as a security repository the following process must be used:

- The J2EE Web Application Server must be configured to use the external LDAP security source for authentication. Refer to the documentation provided with the J2EE Web Application Server for more details. For Oracle WebLogic customers, refer to the *Configuring LDAP Authentication Providers* section of the *Oracle Fusion Middleware Securing Oracle WebLogic Server* Guide.
- The product LDAP import feature can be used to initially populate the authorization model from the LDAP source as outlined in the *LDAP Integration for Oracle Utilities Application Framework based product* (Doc Id: 774783.1) available from My Oracle Support.

Note: Whilst LDAP sources are the most common security repository, it is possible to use alternative security authentication sources as supported by the J2EE Web Application Server. Refer to the documentation provided with the J2EE Web Application Server for more details.

Single Sign On Integration

One of the common security integrations is the ability to implement Single Sign On with the product. This enables end users to access the product minimizing the need to re-authenticate each time.

The J2EE Web Application Server can be configured to support Single Sign On. For more details refer to *Single Sign On Integration for Oracle Utilities Application Framework based products* (Doc Id: 799912.1) and *Oracle Identity Management Suite Integration with Oracle Utilities Application Framework based products* (Doc Id: 1375600.1) available from My Oracle Support.

[11] It also provides authorization services but these are not typically utilized by the product.

Kerberos Support

Single Sign-On (SSO) with Microsoft clients allows cross-platform authentication between Web applications running in the J2EE Web Application Server and .NET Web service clients or browser clients (for example, Microsoft Internet Explorer) in a Microsoft domain. The Microsoft clients must use Windows authentication based on the Simple and Protected Negotiate (SPNEGO) mechanism.

Refer to Configuring Single Sign-On with Microsoft Clients for details of configuring Oracle WebLogic to use Kerberos.

Oracle Identity Management Suite Integration

Oracle offers a comprehensive set of security products as part of the Oracle Identity Management Suite that can be used to augment the security setup at your site. The product can be integrated with the following components of Oracle Identity Management Suite:

- **Oracle Identity Manager** – Oracle Identity Manager can be used to centralize user provisioning to the product, password rule management and identity administration.
- **Oracle Access Manager** – Oracle Access Manager can be used to provide authentication, single sign on, access controls and user tracking.
- **Oracle Adaptive Access Manager** – Oracle Adaptive Access Manager can be used to provide fraud tracking and multi-faceted authentication.
- **Oracle Virtual Directory** – Oracle Virtual Directory can be used to provide virtualized LDAP security access to LDAP and non-LDAP security sources.
- **Oracle Internet Directory** – Oracle Internet Directory can be used as a LDAP security store.

Refer to the *Oracle Identity Management Suite Integration with Oracle Utilities Application Framework based products* (Doc Id: 1375600.1) whitepaper for more information, available from My Oracle Support.

Keystore Support

The Oracle Utilities Application Framework supports the ability to store cryptographic keys and/or certificates in the keystore file. The keys from the keystore are used for encrypting and decrypting data such as passwords and product data.

Creating the Keystore

Note: For backward compatibility, customers on older versions of the Oracle Utilities Application Framework will use a default keystore created upon upgrade with appropriate values. If you wish to reuse this legacy cryptography refer to Reverting to Legacy Cryptography for details.

Note: If the keystore is not present, Oracle Utilities Application Framework will revert to the internal cryptography used in previous releases.

Note: Passwords encrypted using this keystore will be prefixed with ENCKS and legacy password encryption uses prefix ENC.

Typically a keystore is created using the java **keytool** utility manually but the Oracle Utilities Application Framework utilities have been extended to allow customers to create and manage the keystore from the command line.

Before creating the keystore the following settings must be set in the installation, as per the Server Administration Guide:

- **KS_ALIAS** - The alias of the system key that used for encrypting/decrypting passwords. By default this is set to **ouaf.system**.
- **KS_ALIAS_KEYALG** - The encryption algorithm for the system key. By default it is set to **AES** (Advanced Encryption Standard).
- **KS_ALIAS_KEYSIZE** - The size of the generated system key.
- **KS_HMAC_ALIAS** - The alias of the system HMAC (Hash-based Message Authentication Code) key that used for the product user authentication.
- **KS_HMAC_ALIAS_KEYALG** - The algorithm to be used by the **KS_HMAC_ALIAS** entry in keystore to encrypt the data. By default this is set to **HmacSHA256**.
- **KS_HMAC_ALIAS_KEYSIZE** - The size of the generated system HMAC key.
- **KS_KEYSTORE_FILE** - Location of the keystore file.
- **KS_MODE** - The system key algorithm encryption mode. By default it is set to **CBC** (cipher-block chaining).
- **KS_PADDING** - the system key algorithm encryption padding. By default it is set to **PKCS5Padding**.
- **KS_STOREPASS_FILE** - Keystore Password file.
- **KS_STORETYPE** - Keystore type. By default this is set to **JCEKS**.

Once these settings are specified the keystore is created using the following command:

Linux/UNIX

```
initialSetup.sh -k
```

Windows

```
initialSetup.cmd -k
```

This generates the keystore using the credentials outlined in the Keystore Password file.

Altering the KeyStore Setting

Note: This process should be used for any keystore change including copying keystores across environments.

Once the keystore is created if any of its settings, that has been described in the Creating the KeyStore section, need to be changed the system needs to be updated with the new settings.. The following process must be performed:

- Logon to the machine where you wish to make the changes to the settings.
- Execute the `splenviron[.sh] -e <environment>` command where `<environment>` is the environment on the machine to change.
- Shutdown the environment.
- Alter the keystore parameters to suit the new desired configuration using the `configureEnv[.sh] -a` utility.
- Execute the `initialSetup[.sh] -k` utility to recreate the keystore with the new settings.
- Execute the `configureEnv[.sh]` once more and press enter on each password prompt to re-encrypt the passwords with the new settings.
- Execute the `initialSetup[.sh]` command to apply the changes to the configuration files.

  Note: For customers using native installation, update the Deployments using the Oracle WebLogic console or Oracle Enterprise Manager to load the new versions of the product EAR files.

- The encrypted data that is stored in the database must be re-encrypted using the new settings. Follow the process that is outlined in the Synchronize Data Encryption.

Synchronize Data Encryption

Note: Failure to synchronize the data when the keystore settings have changed will cause outages and unexpected behavior in the product.

Note: The product should be shutdown while running this process.

If at any time the keystore settings change the data that is encrypted using the old settings must be updated. A new utility `com.splwg.shared.common.ChangeConfigurationKey` is provided for re-encrypting the data. The following data is updated using this utility:

- Database Passwords used in Feature configurations such as Database Update features.
- XML Application Integration legacy passwords for JDBC.
- XAI Sender and Receiver Passwords (depending on Sender and Receiver type)
- Reporting tool integration passwords
- Multi-Purpose Listener passwords (for selected products(
- Email Adapter configuration.

- Web Services Passwords (legacy only)
- Security Hashes on user records

The following process is to be performed:

- Logon to the machine you have made the changes upon as the product administrator.
- If you have not already done so, use the **splenviron** utility to set the environment variables for the product environment.
- Set the **CLASSPATH** to point to the required jar files for this utility.

  Windows:

  ```
  set CLASSPATH=%CLASSPATH%;%SPLEBASE%\splapp\standalone\lib\spl-shared-4.2.0.2.0.jar;%SPLEBASE%\splapp\standalone\lib\commons-cli-1.1.jar;%SPLEBASE%\splapp\standalone\lib\log4j-1.2.17.jar;%SPLEBASE%\splapp\standalone\lib\commons-codec-1.6.jar
  ```

  Unix/Linux:

  ```
  export CLASSPATH=$CLASSPATH:$SPLEBASE/splapp/standalone/lib/spl-shared-4.2.0.2.0.jar:$SPLEBASE/splapp/standalone/lib/commons-cli-1.1.jar:$SPLEBASE/splapp/standalone/lib/log4j-1.2.17.jar:$SPLEBASE/splapp/standalone/lib/commons-codec-1.6.jar
  ```

- Execute the **com.splwg.shared.common.ChangeConfigurationKey** java class using the following command:

  ```
  java com.splwg.shared.common.ChangeConfigurationKey [-t/-l/-h/-p] [settings]
  ```

  where options are:

`-t`	Test Mode (no commit of changes)
`-l`	Convert Legacy/OUAF System key
`-h`	Convert User hashes only
`-p`	Convert encrypted passwords only
`[settings]`	List of original settings as per below (other above options should not be used with these settings)

  ```
  java com.splwg.shared.common.ChangeCryptographyKey -l -h -Dcom.oracle.ouaf.system.old.keystore.file=<oldfile> -Dcom.oracle.ouaf.system.old.keystore.passwordFileName=<oldpassfile> -Dcom.oracle.ouaf.system.old.keystore.type=<oldtype> -Dcom.oracle.ouaf.system.old.keystore.alias=<oldalias> -Dcom.oracle.ouaf.system.old.keystore.padding=<oldpadding> -Dcom.oracle.ouaf.system.old.keystore.mode=<oldmode>
  ```

Where:

<oldfile>	Original Key Store file
<oldpassfile>	Original Password Store file
<oldtype>	Original Key store type
<oldalias>	Original alias
<oldpadding>	Original Padding
<oldmode>	Original Mode

Note: Only specify the values that have been changed.

Note: This command has to be run once for each alias changed.

Upgrading from Legacy to Keystore

When upgrading from past releases of Oracle Utilities Application Framework and adopting the new keystore it is recommended to use the following process:

- Ensure all passwords have been updated by executing the **configureEnv** and pressing enter at each password prompt.
- Execute the process outlined in the Synchronize Data Encryption section by running the **com.splwg.shared.common.ChangeConfigurationKey** utility with the **-l** option to re-encrypt data that has been encrypted using the legacy encryption key to the new encryption key. For example:
  `java ChangeConfigurationKey -l`
- Optionally, it is possible to update the individual passwords using the **LegacyCryptographerUpdater** utility as the following command::
  `java LegacyCryptographyUpgrader [-f <file>| -p <password>]`
  where options are:

`-f <file>`	File name for reading the old encrypted password from and output the newly encrypted password to
`-p <password>`	Encrypted password that is prefixed with ENC for re-encrypting with the new key. The output will be sent to the console (stdout). The **password** should be already in ENC format.

Reverting To Legacy Cryptography

For backward compatibility the product includes a legacy cryptography library. Customers on earlier versions of the product can retain the original cryptography libraries if they do not want to take advantage of the newer cryptography standards.

Note: The legacy cryptography libraries are limited in scope. If cryptography is important to your implementation it is recommended to use the newer encryption libraries.

To migrate back to the legacy cryptography libraries the following process should be executed:

- Backup the product software and product database prior to reverting. This will act as a fall back situation if commands are not executed correctly.
- Login to the machine housing the desired environment using the product administrator account.
- Execute the **splenviron[.sh] -e *<environment>*** command where **<environment>** is the name of the environment to revert. This should be executed from the bin directory of the environment to ensure the correct environment variables are set.
- Execute the **configureEnv[.sh]** utility and respecify all the user/passwords used in the product especially the database users/passwords. Save the changes.
- Navigate to the **$SPLEBASE/splapp/standalone/lib** directory (**%SPLEBASE%\splapp\standalone\lib** on Windows).

- Set the classpath using the following command:

  Linux/Unix:

  ```
  export MYCP=$CLASSPATH:$SPLEBASE/splapp/standalone/lib/spl-shared-4.2.0.2.0.jar:$SPLEBASE/splapp/standalone/lib/commons-cli-1.1.jar:$SPLEBASE/splapp/standalone/lib/log4j-1.2.17.jar:$SPLEBASE/splapp/standalone/lib/commons-codec-1.6.jar
  ```

  Windows:

  ```
  set MYCP=%CLASSPATH%:%SPLEBASE%\splapp\standalone\lib\spl-shared-4.2.0.2.0.jar:%SPLEBASE%\splapp\standalone\lib\commons-cli-1.1.jar:%SPLEBASE%\splapp\standalone\lib\log4j-1.2.17.jar:%SPLEBASE%\splapp\standalone\lib\commons-codec-1.6.jar
  ```

- Execute the class that regenerates the encryption files:

  Linux/Unix:

  ```
  java -cp $MYCP com.splwg.shared.common.ChangeConfigurationKey -l
  ```

  Windows:

  ```
  java -cp %MYCP% com.splwg.shared.common.ChangeConfigurationKey -l
  ```

- The base cryptography is now in operation.

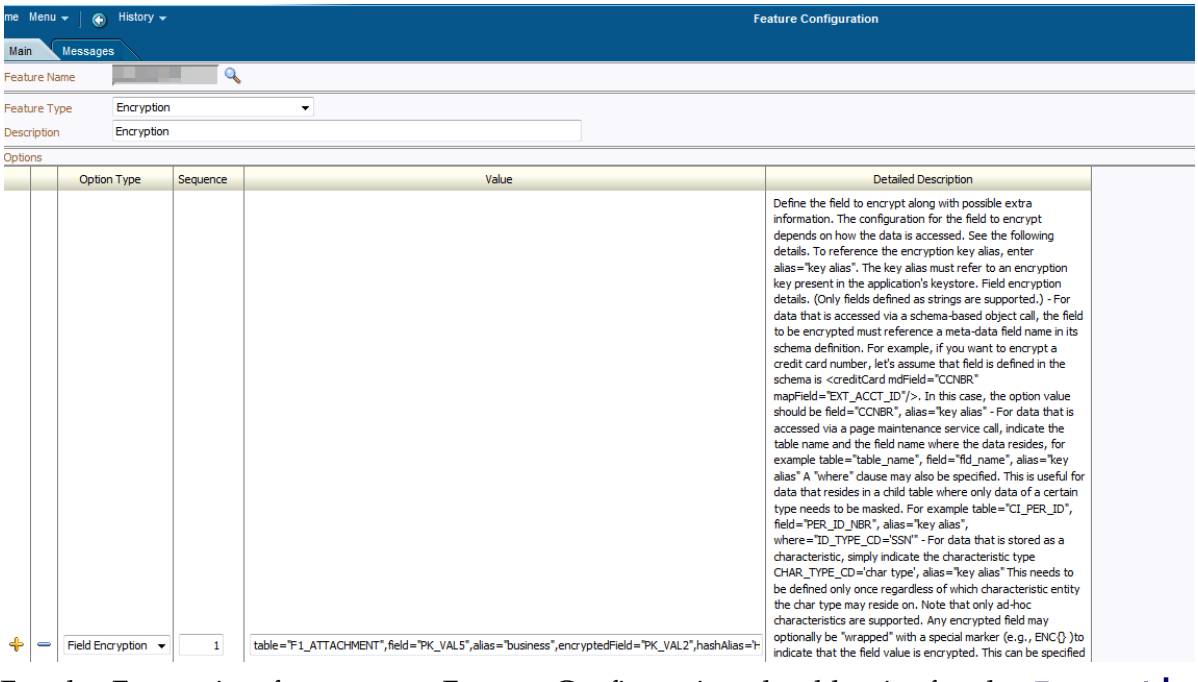
Encryption Feature Type

One of the major features of the Oracle Utilities Application Framework is the ability to mask and encrypt data within the product to protect sensitive information. This encryption is implemented in a Feature Configuration using the Encrypted Feature Type.

Overview

The Oracle Utilities Application Framework supports Feature Configuration which store specific configuration settings for features in the product to be implemented. Feature Configurations allow simple configurations to be implemented for specific features.

Feature Configurations can be maintained using the *Admin Menu* → *F* → *Feature Configuration* menu item. For example:



For the Encryption feature, one Feature Configuration should exist for the **`Encryption`** Feature Type with an option per field to encrypt.

Note: If the product does not ship a Feature Configuration for Encryption, then it can be created as a Customer Modification. Prefix the name of the Feature Name with CM.

Configuration of Encrypted Fields

To define a field to encrypt an option must be added with the following attributes:

- Option Type should be set to **`Field Encryption`**.
- Sequence should be an appropriate sequence number. Typically this is a number that is not used already. Higher number values override lower level sequences.
- In the value you need to specify the specification of the encryption in the format of a command string.

table	Table Name. Table must exist in meta data.	table="SC_USER"
field	Field to encrypt. Field must exist in metadata.	field="FIRST_NAME"
alias	The encryption key alias to use to encrypt the data	alias="ouaf.system"
where	Filter for data. Useful for child tables to determine specific values to encrypt	where="ID_TYPE_CD='SSN'"
wrap	Whether the value should wrapper with the ENC() marker. [true|**false**]	wrap=false
maskAlg	If the field is also to be masked then the algorithm to mask the data.	maskAlg="CMCCR"
maskField	If the field is also to be masked then the field to use as the mask	maskField="CNBR_MASK"
hashAlias	If the field should be hashed then the alias in the keystore to use	hashAlias="ouaf.hmac.system"
hashField	If the field should be hashed then the field to use as the hash value	hashField="CNBR_HASH"
encryptedField	If the output from the encryption is to be stored on another field in the table, specify the field name.	encryptedField="PK_VAL2"

For example:

```
table="F1_ATTACHMENT",field="PK_VAL5",alias="ouaf.system",encryptedField="PK_VAL2",hashAlias='HmacSHA256-1024',hashField="PK_VAL3",
where="PK_VAL1='Encrypted'"
```

There are a few guidelines when using this facility:

- The key aliases specified in **alias** and/or **hashAlias** must exist in the keystore used for the product..
- Fields to be encrypted must be in string format only. Other field formats are not supported.
- If using a higher level of encryption may increase the storage requirements for a field. If this is the case, adding an **encryptedField** to hold the larger encrypted value.
- The **wrap** field should be set to false unless additional processing in your code is included to handle the special marker. Product fields should use **wrap=false**. Wrapping an encrypted value can be useful in knowing whether a specific data is

encrypted in cases where only some data on the table is encrypted.

- Ad-hoc characteristics cannot be specified in the **WHERE** tag.
- Hashing the value is handy for additional verification and indexing values.