10

Managing Fine-Grained Access in PL/SQL Packages and Types

Oracle Database provides PL/SQL packages and types for fine-grained access to control access to external network services and wallets.

- About Managing Fine-Grained Access in PL/SQL Packages and Types
  You can configure user access to external network services and wallets through a set of PL/SQL packages and one type.
- About Fine-Grained Access Control to External Network Services
  Oracle Application Security access control lists (ACL) can implement fine-grained access control to external network services.
- About Access Control to Oracle Wallets
  Encrypting communication between a remote web service and the Oracle database, acting as a client to this service, is an established industry best practice.
- Upgraded Applications That Depend on Packages That Use External Network Services
  Upgraded applications may have `ORA-24247` network access errors.
- Configuring Access Control for External Network Services
  The `DBMS_NETWORK_ACL` packages configures access control for external network services.
- Configuring Access Control to an Oracle Wallet
  Fine-grained access control for Oracle wallets provide user access to network services that require passwords or certificates.
- Examples of Configuring Access Control for External Network Services
  You can configure access control for a variety of situations, such as for a single role and network connection.
- Specifying a Group of Network Host Computers
  You can use wildcards to specify a group of network host computers.
- Precedence Order for a Host Computer in Multiple Access Control List Assignments
  The access control list assigned to a domain has a lower precedence than those assigned to the subdomains.
- Precedence Order for a Host in Access Control List Assignments with Port Ranges
  The precedence order for a host in an access control list is determined by the use of port ranges.
- Checking Privilege Assignments That Affect User Access to Network Hosts
  Both administrators and users can check network connection and domain privileges.
- Configuring Network Access for Java Debug Wire Protocol Operations
  Before you can debug Java PL/SQL procedures, you must be granted the `jdwp` ACL privilege.
- Data Dictionary Views for Access Control Lists Configured for User Access
  Oracle Database provides data dictionary views that you can use to find information about existing access control lists.

10.1 About Managing Fine-Grained Access in PL/SQL Packages and Types

You can configure user access to external network services and wallets through a set of PL/SQL packages and one type.

These packages are the UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, and UTL_INADDR ,and the DBMS_LDAP PL/SQL packages, and the HttpUriType type.

The following scenarios are possible:

- **Configuring fine-grained access control for users and roles that need to access external network services from the database.** This way, specific groups of users can connect to one or more host computers, based on privileges that you grant them. Typically, you use this feature to control access to applications that run on specific host addresses.
- **Configuring fine-grained access control to Oracle wallets to make HTTP requests that require password or client-certificate authentication.** This feature enables you to grant privileges to users who are using passwords and client certificates stored in Oracle wallets to access external protected HTTP resources through the UTL_HTTP package. For example, you can configure applications to use the credentials stored in the wallets instead of hard-coding the credentials in the applications.

10.2 About Fine-Grained Access Control to External Network Services

Oracle Application Security access control lists (ACL) can implement fine-grained access control to external network services.

This guide explains how to configure the access control for database users and roles by using the DBMS_NETWORK_ACL_ADMIN PL/SQL package.

This feature enhances security for network connections because it restricts the external network hosts that a database user can connect to using the PL/SQL network utility packages UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, and UTL_INADDR; the DBMS_LDAP and DBMS_DEBUG_JDWP PL/SQL packages; and the HttpUriType type. Otherwise, an intruder who gained access to the database could maliciously attack the network, because, by default, the PL/SQL utility packages are created with the EXECUTE privilege granted to PUBLIC users. These PL/SQL network utility packages, and the DBMS_NETWORK_ACL_ADMIN and DBMS_NETWORK_ACL_UTILITY packages, support both IP Version 4 (IPv4) and IP Version 6 (IPv6) addresses. This guide explains how to manage access control to both versions.

Be aware that outbound Transport Layer Security (TLS) connections with UTL_HTTP cannot use the default trust store. You must create an Oracle wallet to hold the trust certificates.

Related Topics

- Tutorial: Adding an Email Alert to a Fine-Grained Audit Policy
  This tutorial demonstrates how to create a fine-grained audit policy that generates an email alert when users violate the policy.
- About *Oracle Database Net Services Administrator's Guide*

- Managing Oracle Database Wallets and Certificates
  You can use the orapki command line utility and sqlnet.ora parameters to manage public key infrastructure (PKI) elements.

10.3 About Access Control to Oracle Wallets

Encrypting communication between a remote web service and the Oracle database, acting as a client to this service, is an established industry best practice.

Oracle Database supports network encryption using Transport Layer Security (TLS) when invoking remote services. It also supports authentication methods that may be required. The Oracle database must be aware of the remote site's server certificate before it can securely establish the connection.

There are two ways to handle this configuration:

- **Using the system certificate store.** This method can be used for common TLS-protected web services (that is, HTTPS calls). To configure the system certificate store, you can use the UTL_HTTP PL/SQL package.
- **Storing the certificate in an Oracle wallet.** The use of Oracle wallets is beneficial because it provides secure storage of passwords and client certificates necessary to access protected Web pages. The Oracle wallet provides secure storage of user passwords and client certificates. To configure access control to a wallet, you must have the following components:
  - An Oracle wallet, which you can create by using the Oracle Database orapki or mkstore utility. The HTTP request will use the external password store or the client certificate in the wallet to authenticate the user.
  - An access control list, which you use to grant privileges to the user to use the wallet. To configure the access control list, you use the DBMS_NETWORK_ACL_ADMIN PL/SQL package.

Related Topics

- Configuring Access Control to an Oracle Wallet
  Fine-grained access control for Oracle wallets provide user access to network services that require passwords or certificates.

10.4 Upgraded Applications That Depend on Packages That Use External Network Services

Upgraded applications may have ORA-24247 network access errors.

If you have upgraded from a release before Oracle Database 11*g* Release 1 (11.1), and your applications depend on PL/SQL network utility packages (UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, UTL_INADDR, and DBMS_LDAP) or the HttpUriType type, then the ORA-24247 error may occur when you try to run the application.

The error message is as follows:

```
ORA-24247: network access denied by access control list (ACL)
```

Use the procedures in this chapter to reconfigure the network access for the application.

> **See Also:**
>
> *Oracle Database Upgrade Guide* for compatibility issues for applications that depend on the PL/SQL network utility packages

10.5 Configuring Access Control for External Network Services

The `DBMS_NETWORK_ACL` packages configures access control for external network services.

- Syntax for Configuring Access Control for External Network Services
  You can use the `DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE` procedure to grant the access control privileges to a user.
- Enabling the Listener to Recognize Access Control for External Network Services
  A `TNS-01166: Listener rejected registration or update of service ACL` error can result if the listener is not configured to recognize access control for external network services.
- Example: Configuring Access Control for External Network Services
  The `DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE` procedure can configure access control for external network services.
- Revoking Access Control Privileges for External Network Services
  You can remove access control privileges for external network services.
- Example: Revoking External Network Services Privileges
  The `DBMS_NETWORK_ACL_ADMIN.REMOVE_HOST_ACE` procedure can be used to revoke external network privileges.

10.5.1 Syntax for Configuring Access Control for External Network Services

You can use the `DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE` procedure to grant the access control privileges to a user.

This procedure appends an access control entry (ACE) with the specified privilege to the ACL for the given host, and creates the ACL if it does not exist yet. The resultant configuration resides in the `SYS` schema, not the schema of the user who created it.

The syntax is as follows:

```
BEGIN
 DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE (
  host          => 'host_name',
  lower_port    => null|port_number,
  upper_port    => null|port_number,
  ace           => ace_definition);
END;
```

In this specification:

- `host`: Enter the name of the host. It can be the host name or an IP address of the host. You can use a wildcard to specify a domain or an IP subnet. Be aware of the precedence order for a host computer in multiple access control list assignments when you use wildcards in domain names.) The host or domain name is case insensitive. Examples are as follows:

```
host     => 'www.example.com',
```

```
host     => '*example.com',
```

- lower_port: (Optional) For TCP connections, enter the lower boundary of the port range. Use this setting for the connect privilege only. Omit it for the resolve privilege. The default is null, which means that there is no port restriction (that is, the ACL applies to all ports). The range of port numbers is between 1 and 65535.

  For example:

  ```
  lower_port => 80,
  ```

- upper_port: (Optional) For TCP connections, enter the upper boundary of the port range. Use this setting for connect privileges only. Omit it for the resolve privilege. The default is null, which means that there is no port restriction (that is, the ACL applies to all ports). The range of port numbers is between 1 and 65535

  For example:

  ```
  upper_port => 3999);
  ```

  If you enter a value for the lower_port and leave the upper_port at null (or just omit it), then Oracle Database assumes the upper_port setting is the same as the lower_port. For example, if you set lower_port to 80 and omit upper_port, the upper_port setting is assumed to be 80.

  The resolve privilege in the access control list has no effect when a port range is specified in the access control list assignment.

- ace: Define the ACE by using the XS$ACE_TYPE constant, in the following format:

  ```
  ace    => xs$ace_type(privilege_list => xs$name_list('privilege'),
                        principal_name => 'user_or_role',
                        principal_type => xs$ace_type_user));
  ```

  In this specification:

  - privilege_list: Enter one or more of the following privileges, which are case insensitive. Enclose each privilege with single quotation marks and separate each with a comma (for example, 'http', 'http_proxy').

    For tighter access control, grant only the http, http_proxy, or smtp privilege instead of the connect privilege if the user uses the UTL_HTTP, HttpUriType, UTL_SMTP, or UTL_MAIL only.

    - http: Makes an HTTP request to a host through the UTL_HTTP package and the HttpUriType type

    - http_proxy: Makes an HTTP request through a proxy through the UTL_HTTP package and the HttpUriType type. You must include http_proxy in conjunction to the http privilege if the user makes the HTTP request through a proxy.

    - smtp: Sends SMTP to a host through the UTL_SMTP and UTL_MAIL packages

    - resolve: Resolves a network host name or IP address through the UTL_INADDR package

    - connect: Grants the user permission to connect to a network service at a host through the UTL_TCP, UTL_SMTP, UTL_MAIL, UTL_HTTP, and DBMS_LDAP packages, or the HttpUriType type

    - jdwp: Used for Java Debug Wire Protocol debugging operations for Java or PL/SQL stored procedures.

- `principal_name`: Enter a database user name or role. This value is case insensitive, unless you enter it in double quotation marks (for example, `'"ACCT_MGR"'`).
- `principal_type`: Enter `XS_ACL.PTYPE_DB` for a database user or role. You must specify `PTYPE_DB` because the `principal_type` value defaults to `PTYPE_XS`, which is used to specify an Oracle Database Real Application Security application user.

Related Topics

- Precedence Order for a Host Computer in Multiple Access Control List Assignments
  The access control list assigned to a domain has a lower precedence than those assigned to the subdomains.
- Configuring Network Access for Java Debug Wire Protocol Operations
  Before you can debug Java PL/SQL procedures, you must be granted the `jdwp` ACL privilege.

> **See Also:**
>
> *Oracle Database Real Application Security Administrator's and Developer's Guide* for information about additional `XS$ACE_TYPE` parameters that you can include for the `ace` parameter setting: `granted`, `inverted`, `start_date`, and `end_date`

10.5.2 Enabling the Listener to Recognize Access Control for External Network Services

A `TNS-01166: Listener rejected registration or update of service ACL` error can result if the listener is not configured to recognize access control for external network services.

1. Add the following line to the `listener.ora` file:

   ```
   LOCAL_REGISTRATION_ADDRESS_LISTENER = ON
   ```

2. Restart the listener.

   ```
   ./lsnrctl stop
   ./lsnrctl start
   ```

10.5.3 Example: Configuring Access Control for External Network Services

The `DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE` procedure can configure access control for external network services.

Example 10-1 shows how to grant the `http` and `smtp` privileges to the `acct_mgr` database role for an ACL created for the host `www.example.com`.

Example 10-1 Granting Privileges to a Database Role External Network Services

```
BEGIN
 DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
  host       =>  'www.example.com',
  ace        =>   xs$ace_type(privilege_list => xs$name_list('http', 'smtp'),
                             principal_name => 'acct_mgr',
                             principal_type => xs_acl.ptype_db));
```

```
END;
/
```

10.5.4 Revoking Access Control Privileges for External Network Services

You can remove access control privileges for external network services.

- To revoke access control privileges for external network services, run the `DBMS_NETWORK_ACL_ADMIN.REMOVE_HOST_ACE` procedure.

Related Topics

- *Oracle Database PL/SQL Packages and Types Reference*

10.5.5 Example: Revoking External Network Services Privileges

The `DBMS_NETWORK_ACL_ADMIN.REMOVE_HOST_ACE` procedure can be used to revoke external network privileges.

Example 10-2 shows how to revoke external network privileges.

Example 10-2 Revoking External Network Services Privileges

```
BEGIN
 DBMS_NETWORK_ACL_ADMIN.REMOVE_HOST_ACE (
  host       => 'www.example.com',
  lower_port => 80,
  upper_port => upper_port => 3999,
  ace        => xs$ace_type(privilege_list => xs$name_list('http', 'smtp'),
                            principal_name => 'acct_mgr',
                            principal_type => xs_acl.ptype_db),
  remove_empty_acl => TRUE);
END;
/
```

In this specification, the `TRUE` setting for `remove_empty_acl` removes the ACL when it becomes empty when the ACE is removed.

10.6 Configuring Access Control to an Oracle Wallet

Fine-grained access control for Oracle wallets provide user access to network services that require passwords or certificates.

- About Configuring Access Control to an Oracle Wallet
  You can configure access control to grant access to passwords and client certificates.
- Step 1: Configure the Operating System Certificate Store as the Default Wallet Path
  You can use the `UTL_HTTP`, `UTL_TCP`, or `UTL_SMTP` PL/SQL packages to configure a the system's certificate store to act in place of an Oracle wallet.
- Step 2: Configure Access Control Privileges for the Oracle Wallet
  After you have created the wallet, you are ready to configure access control privileges for the wallet.
- Step 3: Make the HTTP Request with the Passwords and Client Certificates
  The `UTL_HTTP` package can create an HTTP request object to hold wallet information, which can authenticate using a client certificate or a password.
- Revoking Access Control Privileges for Oracle Wallets
  You can revoke access control privileges for an Oracle wallet.

- Troubleshooting ORA-29024 Errors
  The `ORA-29024: Certificate validation failure` error occurs when the facility, component, or product or a failing operation is expecting an Oracle wallet.

10.6.1 About Configuring Access Control to an Oracle Wallet

You can configure access control to grant access to passwords and client certificates.

These passwords and client certificates are stored in an Oracle wallet. The access control that you configure enables users to authenticate themselves to an external network service when using the PL/SQL network utility packages.

This enables the user to gain access to the network service that requires password or certificate identification.

10.6.2 Step 1: Configure the Operating System Certificate Store as the Default Wallet Path

You can use the `UTL_HTTP`, `UTL_TCP`, or `UTL_SMTP` PL/SQL packages to configure a the system's certificate store to act in place of an Oracle wallet.

In previous releases, you used `orapki` to create a wallet. If you choose to create a wallet, then make a note of the directory in which you created the wallet. You will need this directory path when you complete the procedures in this section. However, using the operating system certificate in place of a wallet greatly improves Oracle Database performance.

In a new connected session, `UTL_HTTP` uses the default system certificate store. If `UTL_HTTP.SET_WALLET` had been set, then setting `UTL_HTTP.SET_WALLET` to `system:` overrides the previous `UTL_HTTP.SET_WALLET` setting.

- To use the system certificate, specify `system:` (including the colon), in the following comands:
  - Run the `UTL_HTTP.SET_WALLET('system:')` procedure to explicitly request to use the system's certificate store. (In the absence of any configuration, the `UTL_HTTP` package uses the system's certificate store as the default wallet.)
  - Pass `wallet_path => 'system:'` to the `UTL_HTTP.REQUEST()` procedure and related functions in the package.
  - For the `UTL_TCP` and `UTL_SMTP` packages, set any procedures that use the `wallet_path` parameter to the `'system:'` setting.

Related Topics

- Example: Configuring ACL Access Using Passwords in a Non-Shared Wallet
  The `DBMS_NETWORK_ACL_ADMIN` and `UTL_HTTP` PL/SQL packages can configure ACL access using passwords in a non-shared wallet.
- Example: Configuring ACL Access for a Wallet in a Shared Database Session
  The `DBMS_NETWORK_ACL_ADMIN` and `UTL_HTTP` PL/SQL packages can configure ACL access for a wallet in a shared database session.

10.6.3 Step 2: Configure Access Control Privileges for the Oracle Wallet

After you have created the wallet, you are ready to configure access control privileges for the wallet.

- Use the `DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE` procedure to configure the wallet access control privileges.

  The syntax for the `DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE` procedure is as follows:

  ```
  BEGIN
   DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE (
    wallet_path => 'directory_path_to_wallet',
    ace         => xs$ace_type(privilege_list => xs$name_list('privilege'),
                               principal_name => 'user_or_role',
                               principal_type => xs$ace_type_user));
  END;
  ```

  In this specification:

  – `wallet_path`: Enter the path to the directory that contains the wallet that you just created. When you specify the wallet path, you must use an absolute path and include `file:` before this directory path. Do not use environment variables, such as `$ORACLE_HOME`, nor insert a space after `file:` and before the path name. For example:

    ```
    wallet_path   => 'file:/oracle/wallets/hr_wallet',
    ```

  – `ace`: Define the ACL by using the `XS$ACE_TYPE` constant. For example:

    ```
      ace         =>  xs$ace_type(privilege_list => xs$name_list(privilege),
                                  principal_name => 'hr_clerk',
                                  principal_type => xs_acl.ptype_db);
    ```

    In this specification, *`privilege`* must be one of the following when you enter wallet privileges using `xs$ace_type` (note the use of underscores in these privilege names):

    * `use_client_certificates`

    * `use_passwords`

    Be aware that for wallets, you must specify either the `use_client_certificates` or `use_passwords` privileges.

> **See Also:**
>
> *Oracle Database Real Application Security Administrator's and Developer's Guide* for information about additional `XS$ACE_TYPE` parameters that you can include for the `ace` parameter setting: `granted`, `inverted`, `start_date`, and `end_date`

Related Topics

- Step 1: Configure the Operating System Certificate Store as the Default Wallet Path
  You can use the `UTL_HTTP`, `UTL_TCP`, or `UTL_SMTP` PL/SQL packages to configure a the system's certificate store to act in place of an Oracle wallet.
- Syntax for Configuring Access Control for External Network Services
  You can use the `DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE` procedure to grant the access control privileges to a user.

10.6.4 Step 3: Make the HTTP Request with the Passwords and Client Certificates

The UTL_HTTP package can create an HTTP request object to hold wallet information, which can authenticate using a client certificate or a password.

- Making the HTTPS Request with the Passwords and Client Certificates
  The UTL_HTTP package makes Hypertext Transfer Protocol (HTTP) callouts from SQL and PL/SQL.
- Using a Request Context to Hold the Wallet When Sharing the Session with Other Applications
  You should use a request context to hold the wallet when other applications share the database session.
- Use of Only a Client Certificate to Authenticate
  Only a client certificate can authenticate users, as long as the user has been granted the appropriate privilege in the ACL wallet.
- Use of a Password to Authenticate
  If the protected URL being requested requires username and password authentication, then set the username and password from the wallet to authenticate.

10.6.4.1 Making the HTTPS Request with the Passwords and Client Certificates

The UTL_HTTP package makes Hypertext Transfer Protocol (HTTP) callouts from SQL and PL/SQL.

- Use the UTL_HTTP PL/SQL package to create a request context object that is used privately with the HTTP request and its response.

  For example:

  ```
  DECLARE
   req_context UTL_HTTP.REQUEST_CONTEXT_KEY;
   req         UTL_HTTP.REQ;
  BEGIN
   req_context := UTL_HTTP.CREATE_REQUEST_CONTEXT (
                wallet_path          => 'file:path_to_directory_containing_wallet',
                wallet_password      => 'wallet_password'|NULL);
   req := UTL_HTTP.BEGIN_REQUEST(
                url                  => 'URL_to_application',
                request_context      => 'request_context'|NULL);
   ...
  END;
  ```

  In this specification:

  - req_context: Use the UTL_HTTP.CREATE_REQUEST_CONTEXT_KEY data type to create the request context object. This object stores a randomly-generated numeric key that Oracle Database uses to identify the request context. The UTL_HTTP.CREATE_REQUEST_CONTEXT function creates the request context itself.
  - req: Use the UTL_HTTP.REQ data type to create the object that will be used to begin the HTTP request. You will refer to this object later on, when you set the user name and password from the wallet to access a password-protected Web page.
  - wallet_path: Enter the path to the directory that contains the wallet. Ensure that this path is the same path you specified when you created access control list earlier when

configuring access control privileges for the Oracle wallet. You must include `file:` before the directory path. Do not use environment variables, such as `$ORACLE_HOME`.

For example:

```
wallet_path          => 'file:/oracle/wallets/hr_wallet',
```

– `wallet_password`: Enter the password used to open the wallet. The default is `NULL`, which is used for auto-login wallets. For example:

```
wallet_password      => 'wallet_password');
```

– `url`: Enter the URL to the application that uses the wallet.

For example:

```
url                  => 'www.hr_access.example.com',
```

– `request_context`: Enter the name of the request context object that you created earlier in this section. This object prevents the wallet from being shared with other applications in the same database session.

For example:

```
request_context      => req_context);
```

Related Topics

- Step 2: Configure Access Control Privileges for the Oracle Wallet
  After you have created the wallet, you are ready to configure access control privileges for the wallet.
- *Oracle Database PL/SQL Packages and Types Reference*

10.6.4.2 Using a Request Context to Hold the Wallet When Sharing the Session with Other Applications

You should use a request context to hold the wallet when other applications share the database session.

If your application has exclusive use of the database session, you can hold the wallet in the database session by using the `UTL_HTTP.SET_WALLET` procedure.

- Use the `UTL_HTTP.SET_WALLET` procedure to configure the request to hold the wallet.

For example:

```
DECLARE
 req         UTL_HTTP.REQ;
BEGIN
 UTL_HTTP.SET_WALLET(
             path          => 'file:path_to_directory_containing_wallet',
             password      => 'wallet_password'|NULL);
 req := UTL_HTTP.BEGIN_REQUEST(
             url           => 'URL_to_application');
 ...
END;
```

If the protected URL being requested requires the user name and password to authenticate, then you can use the `SET_AUTHENTICATION_FROM_WALLET` procedure to set the user name and password from the wallet to authenticate.

10.6.4.3 Use of Only a Client Certificate to Authenticate

Only a client certificate can authenticate users, as long as the user has been granted the appropriate privilege in the ACL wallet.

If the protected URL being requested requires only the client certificate to authenticate, then the `BEGIN_REQUEST` function sends the necessary client certificate from the wallet. assuming the user has been granted the `use_client_certificates` privilege in the ACL assigned to the wallet.

The authentication should succeed at the remote Web server and the user can proceed to retrieve the HTTP response by using the `GET_RESPONSE` function.

10.6.4.4 Use of a Password to Authenticate

If the protected URL being requested requires username and password authentication, then set the username and password from the wallet to authenticate.

For example:

```
DECLARE
 req_context UTL_HTTP.REQUEST_CONTEXT_KEY;
 req         UTL_HTTP.REQ;
BEGIN
...
 UTL_HTTP.SET_AUTHENTICATION_FROM_WALLET(
  r                => HTTP_REQUEST,
  alias            => 'alias_to_retrieve_credentials_stored_in_wallet',
  scheme           => 'AWS|Basic',
  for_proxy        => TRUE|FALSE);
END;
```

In this specification:

- `r`: Enter the HTTP request defined in the `UTL_HTTP.BEGIN_REQUEST` procedure that you created above, in the previous section. For example:

  ```
  r                => req,
  ```

- `alias`: Enter the alias used to identify and retrieve the user name and password credential stored in the Oracle wallet. For example, assuming the alias used to identify this user name and password credential is `hr_access`.

  ```
  alias            => 'hr_access',
  ```

- `scheme`: Enter one of the following:
  - `AWS`: Specifies the Amazon Simple Storage Service (S3) scheme. Use this scheme only if you are configuring access to the Amazon.com Web site. (Contact Amazon for more information about this setting.)
  - `Basic`: Specifies HTTP basic authentication. The default is `Basic`.

  For example:

  ```
  scheme           => 'Basic',
  ```

- `for_proxy`: Specify whether the HTTP authentication information is for access to the HTTP proxy server instead of the Web server. The default is `FALSE`.

  For example:

```
    for_proxy       => TRUE);
```

The use of the user name and password in the wallet requires the `use_passwords` privilege to be granted to the user in the ACL assigned to the wallet.

10.6.5 Revoking Access Control Privileges for Oracle Wallets

You can revoke access control privileges for an Oracle wallet.

- To revoke privileges from access control entries (ACE) in the access control list (ACL) of a wallet, run the `DBMS_NETWORK_ACL_ADMIN.REMOVE_WALLET_ACE` procedure.

For example:

```
BEGIN
 DBMS_NETWORK_ACL_ADMIN.REMOVE_WALLET_ACE (
  wallet_path   => 'file:/oracle/wallets/hr_wallet',
  ace           =>  xs$ace_type(privilege_list => xs$name_list(privilege),
                             principal_name => 'hr_clerk',
                             principal_type => xs_acl.ptype_db),
  remove_empty_acl  => TRUE);
END;
/
```

In this example, the `TRUE` setting for `remove_empty_acl` removes the ACL when it becomes empty when the wallet ACE is removed.

10.6.6 Troubleshooting ORA-29024 Errors

The `ORA-29024: Certificate validation failure` error occurs when the facility, component, or product or a failing operation is expecting an Oracle wallet.

You can troubleshoot this error by using the following methods, in this order:

1. Check is the relevant Oracle documentation for the steps related to the failing configuration.
   For example, if this error is occurs while using `UTL_HTTP`, then it means that a secure web site is being accessed without a wallet and this operation needs a wallet created. See *Oracle Database PL/SQL Packages and Types Reference* for information about using the `UTL_HTTP` PL/SQL package.

   In another example, the error can occur while making a remote connection to the database server over a TLS connection, which indicates that this connection is expecting an Oracle wallet. Troubleshooting this problem requires a proper understanding of Oracle Wallets and certificates. See Configuring PKI Certificate Authentication.

2. After the wallet is configured according to the documentation, if the error still occurs, then try the following solutions:

   - Open the wallet using the `orapki` utility as follows:

     ```
     orapki wallet display -wallet wallet_file_directory
     ```

     If this command fails, then it means that the wallet is corrupt. Create a new wallet and recheck the scenario.

   - If the current configuration needs a wallet with a user and trusted certificates, then check whether both the user and trusted certificates are valid and not expired or revoked.

- If this error occurs while using the wallet with a UTL_HTTP configuration, then check whether all the certificates of the secure web site are there in the wallet and the certificate chain is complete.
- If there is a proxy server involved, then ensure that the target website is in the proxy `allowlist`.

See the following My Oracle Support notes for information about getting a complete certificate chain of a secure site for a `UTL_HTTPS` call.

- Note 169768.1 Configuring Wallet Manager to enable HTTPS connections via UTL_HTTP.REQUEST
- Note 230917.1 Troubleshooting the UTL_HTTP Package

10.7 Examples of Configuring Access Control for External Network Services

You can configure access control for a variety of situations, such as for a single role and network connection.

- Example: Configuring Access Control for a Single Role and Network Connection
  The `DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE` procedure can configure access control for a single role and network connection.
- Example: Configuring Access Control for a User and Role
  The `DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE` can configure access control to deny or grant privileges for a user and a role.
- Example: Using the DBA_HOST_ACES View to Show Granted Privileges
  The `DBA_HOST_ACE` data dictionary view shows privileges that have been granted to users.
- Example: Configuring ACL Access Using Passwords in a Non-Shared Wallet
  The `DBMS_NETWORK_ACL_ADMIN` and `UTL_HTTP` PL/SQL packages can configure ACL access using passwords in a non-shared wallet.
- Example: Configuring ACL Access for a Wallet in a Shared Database Session
  The `DBMS_NETWORK_ACL_ADMIN` and `UTL_HTTP` PL/SQL packages can configure ACL access for a wallet in a shared database session.

10.7.1 Example: Configuring Access Control for a Single Role and Network Connection

The `DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE` procedure can configure access control for a single role and network connection.

Example 10-3 shows how you would configure access control for a single role (`acct_mgr`) and grant this role the `http` privilege for access to the `www.us.example.com` host. The privilege expires January 1, 2013.

Example 10-3 Configuring Access Control for a Single Role and Network Connection

```
BEGIN
 DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
  host       => 'www.us.example.com',
  lower_port => 80,
  ace        =>  xs$ace_type(privilege_list => xs$name_list('http'),
                             principal_name => 'acct_mgr',
                           principal_type => xs_acl.ptype_db,
                           end_date => TIMESTAMP '2013-01-01 00:00:00.00 -08:00');
END;
/
```

10.7.2 Example: Configuring Access Control for a User and Role

The DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE can configure access control to deny or grant privileges for a user and a role.

Afterwards, you can query the DBA_HOST_ACES data dictionary view to find information about the privilege grants.

Example 10-4 grants to a database role (acct_mgr) but denies a particular user (psmith) even if that user has the role. The order is important because ACEs are evaluated in the given order. In this case, the deny ACE (granted => false) must be appended first or else the user cannot be denied.

Example 10-4   Configuring Access Control Using a Grant and a Deny for User and Role

```
BEGIN
 DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
  host       => 'www.us.example.com',
  lower_port => 80,
  upper_port => 80,
  ace        =>  xs$ace_type(privilege_list => xs$name_list('http'),
                             principal_name => 'psmith',
                             principal_type => xs_acl.ptype_db,
                             granted        => false));

 DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
  host       => 'www.us.example.com',
  lower_port => 80,
  upper_port => 80,
  ace        =>  xs$ace_type(privilege_list => xs$name_list('http'),
                             principal_name => 'acct_mgr',
                             principal_type => xs_acl.ptype_db,
                             granted        => true));
END;
```

10.7.3 Example: Using the DBA_HOST_ACES View to Show Granted Privileges

The DBA_HOST_ACE data dictionary view shows privileges that have been granted to users.

Example 10-5 shows how the DBA_HOST_ACES data dictionary view displays the privilege granted in the previous access control list.

Example 10-5   Using the DBA_HOST_ACES View to Show Granted Privileges

```
SELECT PRINCIPAL, PRIVILEGE, GRANT_TYPE FROM DBA_HOST_ACE WHERE PRIVILEGE = 'HTTP';

PRINCIPAL    PRIVILEGE  GRANT_TYPE
------------ ---------- --------------------
PSMITH       HTTP       FALSE
ACCT_MGR     HTTP       TRUE
```

10.7.4 Example: Configuring ACL Access Using Passwords in a Non-Shared Wallet

The `DBMS_NETWORK_ACL_ADMIN` and `UTL_HTTP` PL/SQL packages can configure ACL access using passwords in a non-shared wallet.

Example 10-6 configures wallet access for two Human Resources department roles, `hr_clerk` and `hr_manager`. These roles use the `use_passwords` privilege to access passwords stored in the wallet. In this example, the wallet will not be shared with other applications within the same database session.

Example 10-6   Configuring ACL Access Using Passwords in a Non-Shared Wallet

```
/* 1. At a command prompt, create the wallet. The following example uses the
      user name hr_access as the alias to identify the user name and password
      stored in the wallet. You must use this alias name when you call the
      SET_AUTHENTICATION_FROM_WALLET procedure later on. */
$ mkstore -wrl $ORACLE_HOME/wallets/hr_wallet -create
Enter password: password
Enter password again: password
$ mkstore -wrl $ORACLE_HOME/wallets/hr_wallet -createCredential hr_access hr_usr
Your secret/Password is missing in the command line
Enter your secret/Password: password
Re-enter your secret/Password: password
Enter wallet password: password

/* 2. In SQL*Plus, create an access control list to grant privileges for the
      wallet. The following example grants the use_passwords privilege to the
      hr_clerk role.*/
BEGIN
 DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE (
  wallet_path => 'file:/oracle/wallets/hr_wallet',
  ace         => xs$ace_type(privilege_list => xs$name_list('use_passwords'),
                             principal_name => 'hr_clerk',
                             principal_type => xs_acl.ptype_db));
END;
/

/* 3. Create a request context and request object, and then set the authentication
      for the wallet. */
DECLARE
  req_context  UTL_HTTP.REQUEST_CONTEXT_KEY;
  req          UTL_HTTP.REQ;

BEGIN
 req_context := UTL_HTTP.CREATE_REQUEST_CONTEXT(
     wallet_path          => 'file:/oracle/wallets/hr_wallet',
     wallet_password      => NULL,
     enable_cookies       => TRUE,
     max_cookies          => 300,
     max_cookies_per_site => 20);
  req := UTL_HTTP.BEGIN_REQUEST(
     url                  => 'www.hr_access.example.com',
     request_context      => req_context);
  UTL_HTTP.SET_AUTHENTICATION_FROM_WALLET(
     r                    => req,
     alias                => 'hr_access'),
     scheme               => 'Basic',
     for_proxy            => FALSE);
```

```
END;
/
```

10.7.5 Example: Configuring ACL Access for a Wallet in a Shared Database Session

The `DBMS_NETWORK_ACL_ADMIN` and `UTL_HTTP` PL/SQL packages can configure ACL access for a wallet in a shared database session.

Example 10-7 configures the wallet to be used for a shared database session; that is, all applications within the current database session will have access to this wallet.

Example 10-7 Configuring ACL Access for a Wallet in a Shared Database Session

```
/* Follow these steps:
   1. Use the orapki utility to create the wallet and add the client
      certificate. For example:

         orapki wallet create -wallet wallet_location
         orapki wallet add -wallet wallet_location -trusted_cert -cert
certificate_location

   2. In SQL*Plus, configure access control to grant privileges for the wallet.
      The following example grants the use_client_certificates privilege
      to the hr_clerk and hr_mgr roles. */
BEGIN
 DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE (
  wallet_path => 'file:/oracle/wallets/hr_wallet',
  ace         => xs$ace_type(privilege_list => xs$name_list('use-client_certificates'),
                             principal_name => 'hr_clerk',
                             principal_type => xs_acl.ptype_db));

 DBMS_NETWORK_ACL_ADMIN.APPEND_WALLET_ACE (
  wallet_path => 'file:/oracle/wallets/hr_wallet',
  ace         => xs$ace_type(privilege_list => xs$name_list('use_client_certificates'),
                             principal_name => 'hr_mgr',
                             principal_type => xs_acl.ptype_db));
END;
/
COMMIT;

/* 3. Create a request object to handle the HTTP authentication for the wallet.*/
DECLARE
  req  UTL_HTTP.req;
BEGIN
  UTL_HTTP.SET_WALLET(
   path            => 'file:/oracle/wallets/hr_wallet',
   password        => NULL);
 req := UTL_HTTP.BEGIN_REQUEST(
   url             => 'www.hr_access.example.com',
   method          => 'POST',
   http_version    => NULL,
   request_context => NULL);
END;
/
```

10.8 Specifying a Group of Network Host Computers

You can use wildcards to specify a group of network host computers.

- To assign an access control list to a group of network host computers, use the asterisk (*) wildcard character.

For example, enter `*.example.com` for host computers that belong to a domain or `192.0.2.*` for IPv4 addresses that belong to an IP subnet. The asterisk wildcard must be at the beginning, before a period (.) in a domain, or at the end, after a period (.), in an IP subnet. For example, `*.example.com` is valid, but `*example.com` and `*.example.*` are not. Be aware that the use of wildcard characters affects the order of precedence for multiple access control lists that are assigned to the same host computer. You cannot use wildcard characters for IPv6 addresses.

The Classless Inter-Domain Routing (CIDR) notation defines how IPv4 and IPv6 addresses are categorized for routing IP packets on the internet. The `DBMS_NETWORK_ACL_ADMIN` package supports CIDR notation for both IPv4 and IPv6 addresses. This package considers an IPv4-mapped IPv6 address or subnet equivalent to the IPv4-native address or subnet it represents. For example, `::ffff:192.0.2.1` is equivalent to `192.0.2.1`, and `::ffff:192.0.2.1/120` is equivalent to `192.0.2.*`.

10.9 Precedence Order for a Host Computer in Multiple Access Control List Assignments

The access control list assigned to a domain has a lower precedence than those assigned to the subdomains.

For multiple access control lists that are assigned to the host computer and its domains, the access control list that is assigned to the host computer takes precedence over those assigned to the domains.

The access control list assigned to a domain has a lower precedence than those assigned to the subdomains.For example, Oracle Database first selects the access control list assigned to the host `server.us.example.com`, ahead of other access control lists assigned to its domains. If additional access control lists were assigned to the sub domains, their order of precedence is as follows:

1. `server.us.example.com`
2. `*.us.example.com`
3. `*.example.com`
4. `*.com`
5. `*`

Similarly, for multiple access control lists that are assigned to the IP address (both IPv4 and IPv6) and the subnets it belongs to, the access control list that is assigned to the IP address takes precedence over those assigned to the subnets. The access control list assigned to a subnet has a lower precedence than those assigned to the smaller subnets it contains.

For example, Oracle Database first selects the access control list assigned to the IP address `192.0.2.3`, ahead of other access control lists assigned to the subnets it belongs to. If additional access control lists were assigned to the subnets, their order of precedence is as follows:

1. `192.0.2.3` (or `::ffff:192.0.2.3`)
2. `192.0.2.3/31` (or `::ffff:192.0.2.3/127`)
3. `192.0.2.3/30` (or `::ffff:192.0.2.3/126`)
4. `192.0.2.3/29` (or `::ffff:192.0.2.3/125`)
5. ...
6. `192.0.2.3/24` (or `::ffff:192.0.2.3/120` or `192.0.2.*`)
7. `...`
8. `192.0.2.3/16` (or `::ffff:192.0.2.3/112` or `192.0.*`)
9. ...
10. `192.0.2.3/8` (or `::ffff:192.0.2.3/104` or `192.*`)
11. ...
12. `::ffff:192.0.2.3/95`
13. `::ffff:192.0.2.3/94`
14. ...
15. `*`

10.10 Precedence Order for a Host in Access Control List Assignments with Port Ranges

The precedence order for a host in an access control list is determined by the use of port ranges.

When an access control list is assigned to a host computer, a domain, or an IP subnet with a port range, it takes precedence over the access control list assigned to the same host, domain, or IP subnet without a port range.

For example, suppose you have TCP connections to any port between port 80 and 99 at `server.us.example.com`. Oracle Database first selects the access control list assigned to port 80 through 99 at `server.us.example.com`, ahead of the other access control list assigned to `server.us.example.com` that is without a port range.

10.11 Checking Privilege Assignments That Affect User Access to Network Hosts

Both administrators and users can check network connection and domain privileges.

- About Privilege Assignments that Affect User Access to Network Hosts
  Oracle provides DBA-specific data dictionary views to find information about privilege assignments.
- How to Check User Network Connection and Domain Privileges
  A database administrator can query the `DBA_HOST_ACES` data dictionary view to find the privileges that have been granted for specific users or roles.

- Example: Administrator Checking User Network Access Control Permissions
  The DBA_HOST_ACES data dictionary view can check the network access control permissions for users.
- How Users Can Check Their Network Connection and Domain Privileges
  Users can query the USER_HOST_ACES data dictionary view to check their network and domain permissions.
- Example: User Checking Network Access Control Permissions
  The USER_HOST_ACES data dictionary view shows network access control permissions for a host computer.

10.11.1 About Privilege Assignments that Affect User Access to Network Hosts

Oracle provides DBA-specific data dictionary views to find information about privilege assignments.

Database administrators can use the DBA_HOST_ACES data dictionary view to query network privileges that have been granted to or denied from database users and roles in the access control lists, and whether those privileges take effect during certain times only

Using the information provided by the view, you may need to combine the data to determine if a user is granted the privilege at the current time, the roles the user has, the order of the access control entries, and so on.

Users without database administrator privileges do not have the privilege to access the access control lists or to invoke those DBMS_NETWORK_ACL_ADMIN functions. However, they can query the USER_HOST_ACES data dictionary view to check their privileges instead.

Database administrators and users can use the following DBMS_NETWORK_ACL_UTILITY functions to determine if two hosts, domains, or subnets are equivalent, or if a host, domain, or subnet is equal to or contained in another host, domain, or subnet:

- EQUALS_HOST: Returns a value to indicate if two hosts, domains, or subnets are equivalent
- CONTAINS_HOST: Returns a value to indicate if a host, domain, or subnet is equal to or contained in another host, domain, or subnet, and the relative order of precedence of the containing domain or subnet for its ACL assignments

If you do not use IPv6 addresses, database administrators and users can use the following DBMS_NETWORK_ACL_UTILITY functions to generate the list of domains or IPv4 subnet a host belongs to and to sort the access control lists by their order of precedence according to their host assignments:

- DOMAINS: Returns a list of the domains or IP subnets whose access control lists may affect permissions to a specified network host, subdomain, or IP subnet
- DOMAIN_LEVEL: Returns the domain level of a given host

10.11.2 How to Check User Network Connection and Domain Privileges

A database administrator can query the DBA_HOST_ACES data dictionary view to find the privileges that have been granted for specific users or roles.

The DBA_HOST_ACES view shows the access control lists that determine the access to the network connection or domain, and then determines if each access control list grants

(GRANTED), denies (DENIED), or does not apply (NULL) to the access privilege of the user. Only the database administrator can query this view.

10.11.3 Example: Administrator Checking User Network Access Control Permissions

The DBA_HOST_ACES data dictionary view can check the network access control permissions for users.

Example 10-8 shows how a database administrator can check the privileges for user preston to connect to www.us.example.com.

In this example, user preston was granted privileges for all the network host connections found for www.us.example.com. However, suppose preston had been granted access to a host connection on port 80, but then denied access to the host connections on ports 3000–3999. In this case, you must configure access control for the host connection on port 80, and a separate access control configuration for the host connection on ports 3000–3999.

Example 10-8   Administrator Checking User Network Access Control Permissions

```
SELECT HOST, LOWER_PORT, UPPER_PORT,
       ACE_ORDER, PRINCIPAL, PRINCIPAL_TYPE,
       GRANT_TYPE, INVERTED_PRINCIPAL, PRIVILEGE,
       START_DATE, END_DATE
  FROM (SELECT ACES.*,
DBMS_NETWORK_ACL_UTILITY.CONTAINS_HOST('www.us.example.com', HOST) PRECEDENCE
          FROM DBA_HOST_ACES ACES)
 WHERE PRECEDENCE IS NOT NULL
 ORDER BY PRECEDENCE DESC,
          LOWER_PORT NULLS LAST,
          UPPER_PORT NULLS LAST,
          ACE_ORDER;
HOST               LOWER_PORT UPPER_PORT ACE_ORDER PRINCIPAL PRINCIPAL_TYPE   GRANT_TYPE
INVERTED_PRINCIPAL PRIVILEGE START_DATE END_DATE
------------------ ---------- ---------- --------- --------- ---------------- ----------
------------------ --------- ---------- --------
www.us.example.com         80        80          1 PRESTON DATABASE USER      GRANT
NO                 HTTP
www.us.example.com         80        80          2 SEBASTIAN DATABASE USER    GRANT
NO                 HTTP
*.us.example.com                                 1 ACCT_MGR DATABASE USER     GRANT
NO                 CONNECT
*                                                1 HR_DBA DATABASE USER       GRANT
NO                 CONNECT
*                                                1 HR_DBA DATABASE USER       GRANT
NO                 RESOLVE
```

10.11.4 How Users Can Check Their Network Connection and Domain Privileges

Users can query the USER_HOST_ACES data dictionary view to check their network and domain permissions.

The USER_HOST_ACES view is PUBLIC, so all users can query it.

This view hides the access control lists from the user. It evaluates the permission status for the user (GRANTED or DENIED) and filters out the NULL case because the user does not need to know when the access control lists do not apply to them. In other words, Oracle Database only shows the user on the network hosts that explicitly grant or deny access to them. Therefore, the output does not display the *.example.com and * that appear in the output from the database administrator-specific DBA_HOST_ACES view.

10.11.5 Example: User Checking Network Access Control Permissions

The USER_HOST_ACES data dictionary view shows network access control permissions for a host computer.

Example 10-9 shows how user preston can check their privileges to connect to www.us.example.com.

Example 10-9   User Checking Network Access Control Permissions

```
SELECT HOST, LOWER_PORT, UPPER_PORT, PRIVILEGE, STATUS
  FROM (SELECT ACES.*,
DBMS_NETWORK_ACL_UTILITY.CONTAINS_HOST('www.us.example.com', HOST) PRECEDENCE
          FROM USER_HOST_ACES ACES)
 WHERE PRECEDENCE IS NOT NULL
 ORDER BY PRECEDENCE DESC,
          LOWER_PORT NULLS LAST,
          UPPER_PORT NULLS LAST;

HOST               LOWER_PORT UPPER_PORT PRIVILEGE STATUS
------------------ ---------- ---------- --------- -------
www.us.example.com         80         80 HTTP      GRANTED
```

10.12 Configuring Network Access for Java Debug Wire Protocol Operations

Before you can debug Java PL/SQL procedures, you must be granted the jdwp ACL privilege.

If you want to debug Java PL/SQL procedures in the database through a Java Debug Wire Protocol (JDWP)-based debugger, such as SQL Developer, JDeveloper, or Oracle Developer Tools For Visual Studio (ODT), then you must be granted the jdwp ACL privilege to connect your database session to the debugger at a particular host.

The jdwp privilege is needed in conjunction with the DEBUG CONNECT SESSION system privilege.

If you have not been granted the jdwp ACL privilege, then when you try to debug your Java and PL/SQL stored procedures from a remote host, the following errors may appear:

```
ORA-24247: network access denied by access control list (ACL)
ORA-06512: at "SYS.DBMS_DEBUG_JDWP", line line_number
```

- To configure network access for JDWP operations, use the DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE procedure.

The following example illustrates how to configure network access for JDWP operations.

```
BEGIN
  DBMS_NETWORK_ACL_ADMIN.APPEND_HOST_ACE(
    host         => 'host',
    lower_port   => null|port_number,
    upper_port   => null|port_number,
    ace => xs$ace_type(privilege_list => xs$name_list('jdwp'),
                       principal_name => 'username',
                       principal_type => xs_acl.ptype_db));
```

```
END;
/
```

In this specification:

- *host* can be a host name, domain name, IP address, or subnet.
- *port_number* enables you to specify a range of ports. If you want to use any port, then omit the lower_port and upper_port values.
- *username* is case-insensitive unless it is quoted (for example, principal_name => '"PSMITH"').

See Also:

- *Oracle Database Java Developer’s Guide* for more information about debugging server applications with JDWP
- *Oracle SQL Developer User's Guide* for information about remote debugging in SQL Developer

10.13 Data Dictionary Views for Access Control Lists Configured for User Access

Oracle Database provides data dictionary views that you can use to find information about existing access control lists.

Table 10-1 lists these views.

Table 10-1 Data Dictionary Views That Display Information about Access Control Lists

View	Description
DBA_HOST_ACES	Shows the network privileges defined for the network hosts. The SELECT privilege on this view is granted to the SELECT_CATALOG_ROLE role only.
DBA_WALLET_ACES	Lists the wallet path, ACE order, start and end times, grant type, privilege, and information about principals
DBA_WALLET_ACLS	Shows the access control list assignments to the wallets. The SELECT privilege on this view is granted to the SELECT_CATALOG_ROLE role only.
DBA_HOST_ACLS	Shows the access control list assignments to the network hosts. The SELECT privilege on this view is granted to the SELECT_CATALOG_ROLE role only.
USER_HOST_ACES	Shows the status of the network privileges for the current user to access network hosts. The SELECT privilege on the view is granted to PUBLIC.
USER_WALLET_ACES	Shows the status of the wallet privileges for the current user to access contents in the wallets. The SELECT privilege on the view is granted to PUBLIC.

Related Topics

- *Oracle Database Reference*