B.1 Appendix: Troubleshooting UTL_HTTP

This appendix guides you through the steps to troubleshoot issues that may arise while using the `UTL_HTTP` package.

The `UTL_HTTP` package is used to access a given URL from either an internal, external, or a secure website. Troubleshooting this package may require expertise from different competence areas. This guide takes you through a checklist of items. Depending on where an error occurs in the context of the checklist, this guide determines the competence area best suited for further assistance with the actual issue at hand.

`UTL_HTTP` Package Overview

You can use a `UTL_HTTP` package (PL/SQL package) method to obtain HTML text from a given web server page. The obtained text can be used within your application (usually by parsing the text for data). Your application would execute a PL/SQL procedure within the database to call the `UTL_HTTP` package after passing in the desired parameters. The derived result can be processed to perform a variety of tasks.

The `UTL_HTTP` package is used within the database. The package makes an HTTP or HTTPS connection on the internet or intranet, and brings back text, which an application can process. No HTTP server, except that which it is accessing, is required for this purpose. It can be any web server (not necessarily an Oracle server) that serves HTTP or HTTPS requests. When making HTTP callouts from PL/SQL or SQL, it turns the database into a text-based browser.

The `UTL_HTTP` package is not the same as the PL/SQL web toolkit, which is used with the Oracle HTTP Server to access procedures in the database and generate HTML pages to return back to the browser. While the `UTL_HTTP` package can be used within an application that is accessed over the web, its processing does not send anything back to the browser (unless at a further point in your application, the code calls the PL/SQL Toolkit `OWA` and `HTP` packages).

Troubleshooting Steps

The troubleshooting involves the following checks that are enumerated in the following steps. It is recommended that you perform these tasks in the given order, and also ensure that each step is completed before moving on to the next step.

- Database Check (Steps 1 to 3) wherein you verify that the `UTL_HTTP` package is valid and the required privileges are set correctly
- Secure Website Access (HTTPS) Check (Steps 4 and 5) wherein you verify if a non-secure website is being accessed, which includes the following: if Oracle Wallet is being used, verify the wallet location, check if Oracle has the permission to open the wallet, and verify if the wallet password is correct
- Configuration Check (Step 6) wherein you verify if the `UTL-HTTP` package is being used in conjunction with another Oracle product
- Language Check (Step 7) wherein you ascertain if the language-handling group should be involved to troubleshoot programmatic issues

Step 1: Verify that the `UTL_HTTP` Package is Valid

To verify if the UTL_HTTP package is valid, use the following command.

Note:

The dependent objects may differ based on the database version.

```
SQL> column object_name format a15

SQL>SELECT object_name,
           object_type,
           status
    FROM dba_objects
      WHERE object_name IN
         (SELECT referenced_name
            FROM dba_dependencies
            WHERE name='UTL_HTTP')
      ORDER BY object_name, object_type;
```

An example of the output is as follows:

```
OBJECT_NAME     OBJECT_TYPE        STATUS
--------------- ------------------ -------
PLITBLM         PACKAGE            VALID
PLITBLM         SYNONYM            VALID
STANDARD        PACKAGE            VALID
STANDARD        PACKAGE BODY       VALID
UTL_HTTP        PACKAGE            VALID
UTL_HTTP        PACKAGE BODY       VALID
UTL_HTTP        SYNONYM            VALID
UTL_HTT_LIB     LIBRARY            VALID
UTL_RAW         PACKAGE            VALID
UTL_RAW         PACKAGE BODY       VALID
UTL_RAW         SYNONYM            VALID
```

Another example of the output is as follows:

```
OBJECT_NAME     OBJECT_TYPE        STATUS
--------------- ------------------ -------
STANDARD        PACKAGE            VALID
STANDARD        PACKAGE BODY       VALID
UTL_HTTP        PACKAGE            VALID
UTL_HTTP        PACKAGE BODY       VALID
UTL_HTTP        SYNONYM            VALID
```

If any of the objects are not valid, run the `htlrp.sql` script to validate them.

```
cd $ORACLE_HOME/rdbms/admin sqlplus

SQL> SELECT object_name FROM DBA_OBJECTS WHERE status = 'INVALID';
SQL> connect / as sysdba
SQL> @utlrp.sql
SQL> quit
```

Step 2: Verify if the Required Privileges are Set Correctly

If the packages are all being returned with a status of `VALID`, then check the privileges.

To check the privileges, use the following commands:

```
SQL> column grantee format a10
SQL> column owner format a6
SQL> column table_name format a15
SQL> column grantor format a10
SQL> column privilege format a10
SQL> SELECT * FROM dba_tab_privs WHERE table_name='UTL_HTTP';
```

An example of the output is as follows:

```
GRANTEE    OWNER  TABLE_NAME      GRANTOR    PRIVILEGE  GRA HIE
---------- ------ --------------- ---------- ---------- --- ---
PUBLIC     SYS    UTL_HTTP        SYS        EXECUTE    NO  NO
```

Step 3: Check the Alert Logs (`alert.log`)

If no errors are returned from the previous step, check the `alert.log` file for additional information that is relevant to the time the `UTL_HTTP` package was executed. If you are a DBA (Database Administrator), you can query the `background_dump_dest` initialization parameter to find the location of the `alert.log` file.

```
SQL> column value format a40
SQL> column name format a30
SQL> SELECT name, value FROM v$parameter WHERE name='background_dump_dest';
```

DBA Database Checks

If any of the following is true, then any further troubleshooting should be within the purview of the database DBA group.

- Package(s) is invalid.
- Privileges are not granted.
- There are errors in the `alert.log` file.
- Running any `UTL_HTTP` function or procedure results in an `ORA-00600` or `ORA-03113` error.

Step 4: Check if it is a Secure Website Access (HTTPS Access)

If the target URL of the website contains `https` instead of `http`, then the website is a secure website.

Note:

> If a browser is available on the server, you should verify that the secure URL can be accessed. If you cannot access the URL on the browser, then the `utl_http` package cannot access it either. Also, ensure that a dialog box requesting client authentication does not appear (as this is not yet supported).

Just as with the browser, the `UTL_HTTP` package also supports HTTP over the Secured Socket Layer (SSL) protocol (also known as HTTPS), directly or through an HTTP proxy. For releases prior to Oracle Database Release 23ai, an Oracle Wallet is required to make an HTTPS request using the `UTL_HTTP` package (Non-HTTPS fetches do not require an Oracle Wallet). Starting Oracle Database 23ai, you can use the operating system's certificate store instead of an Oracle Wallet (provided the web service you are connecting to is "trusted" by the operating system).

About Oracle Wallet

Oracle Wallet contains the list of certificate authorities that the user of the `UTL_HTTP` package trusts. When a wallet is created, it is populated with a set of well-known certificate authorities as trust points. If the certificate authority that signs the certificate of the remote HTTPS web server is not among the trust points, then you should obtain the root certificate of that certificate authority, and install it as a trust point in the wallet using Oracle Wallet Manager.

Step 5: Verify the Oracle Wallet Location

When the `UTL_HTTP` package is executed in the Oracle Database server, the wallet must be accessible. To confirm the existence of the wallet and also the file permissions, navigate to the wallet directory using the system shell. The directory should have a file named `ewallet.p12` and the file permissions should be set with at least read permissions for the Oracle user.

For example, on Unix, you should see something similar to the following:

```
[sunsys]/etc/ORACLE/WALLETS/oracle> ls -al

drwxr-xr-x   2 oracle   dba          512 Jan 28 11:33 ./
drwxr-xr-x  10 oracle   dba          512 Jan 30 08:39 ../
-r--------   1 oracle   dba         8581 Jan 17 11:31 ewallet.p12
```

With the wallet configured, you can test the access to the secure website using the following SQL:

```
SELECT utl_http.request('', '', 'file:', '') FROM DUAL;
```

For example:

```
SELECT utl_http.request('https://www.xyz.com','proxy.<Domin Name>:<Port
Number>','file:/etc/ORACLE/WALLETS/oracle','welcome1') FROM DUAL;
```

Step 6: Verify if the `UTL_HTTP` package is Being Used in Conjunction with Another Oracle Product

> **Note:**
>
> Using SQL*Plus or PL/SQL does not constitute the use of another product.

Verify if the `UTL_HTTP` package is being used in conjunction with another Oracle product or component (such as, Reports, Portal, Discoverer). If so, any Support Service Request should be transferred to the associated competence area. However, it is recommended to test a plain or simple `.html` page. If a simple page works with `UTL_HTTP`, but another component's pages do not work, the issue is within the component being used; perhaps the way the page is rendered back to the `UTL_HTTP` package.

The reason for transferring to the associated group is because, at this point, the `UTL_HTTP` package has been confirmed to be successfully installed, and a connection to an internal and external website can be made (along with a secure site, if applicable). For example, when used with Reports, the URL for the Reports server is generally used and since connection to other sites works fine, the issue may be related to the actual Reports Server and the URL.

Step 7: Check the Language

With the addition of new functions within the `UTL_HTTP` package, the chance of integrating within PL/SQL-specific functionality has increased, and the language group should be involved if the customer has programmatic issues. This falls under the guidelines of using the results from the `UTL_HTTP` package within expressions, control structures (`IF-THEN-ELSE`), or loops (`Simple, While, For`).

If you can successfully accomplish all the tasks discussed so far, but the issue remains unresolved, it could likely be because of custom application usage, or a bug with the package or interaction with the database. Ensure that you always apply the latest database patch set, because the `UTL_HTTP` package would then include any new patch updates.

Step 8: Contact Oracle Support

If you are still unable to troubleshoot the issue, you can contact Oracle Support for further assistance. Be prepared with a test case that can be used to try and reproduce the issue.