

Installing the Apache Web Server and PL/SQL Gateway

22 July 2008

By Joe Lennon (<http://www.joelennon.com>)

In order to develop PL/SQL applications on Oracle XE, we are going to need to install a web server (Apache) and a PL/SQL gateway for Apache called mod_owa. On a regular Oracle Application Server you would probably be using Oracle HTTP Server (a modified Apache) and mod_plsql, and you should also be able to follow my future tutorials if you have this type of setup.

Once again I will be assuming throughout this guide that you have followed my Oracle XE installation tutorial. If you didn't, you may need to change usernames, passwords and service names to fit your setup. I will be walking through the setup process on a Windows XP machine, but it should be similar on another Windows version. Let's get started!

The first thing you'll need to do is download the relevant software. The easiest way to install Apache is to download and run the installer binary from the Apache website. At the time of writing, the latest stable version of Apache available is 2.2.9, but again the process should be similar for whatever version you are installing. You can download Apache from <http://httpd.apache.org/download.cgi>. You will also need to grab the Apache PL/SQL Gateway Module (mod_owa) from <http://www.sharemation.com/~dmcmahon/modowa.htm>. Click on the link for "Zip file for Windows", which includes the source code and binaries for the Windows version of Apache.

The first stage in setting up our PL/SQL web development environment is the installation of the Apache web server. To start the installation, run the installer package you downloaded earlier (the filename should be something like apache_2.2.9-win32-x86-no_ssl-r2.msi). The screen should look similar to the one in Figure 3a below:



Figure 3a: Apache HTTP Server 2.2 Installation Wizard

Click "Next >" to continue. On the screen that follows, read the license agreement and if you are happy with the terms, select "I accept the terms in the license agreement" and click "Next >" to move on to the readme screen, from which you can simply press "Next >" again. You should now be presented with the following screen:

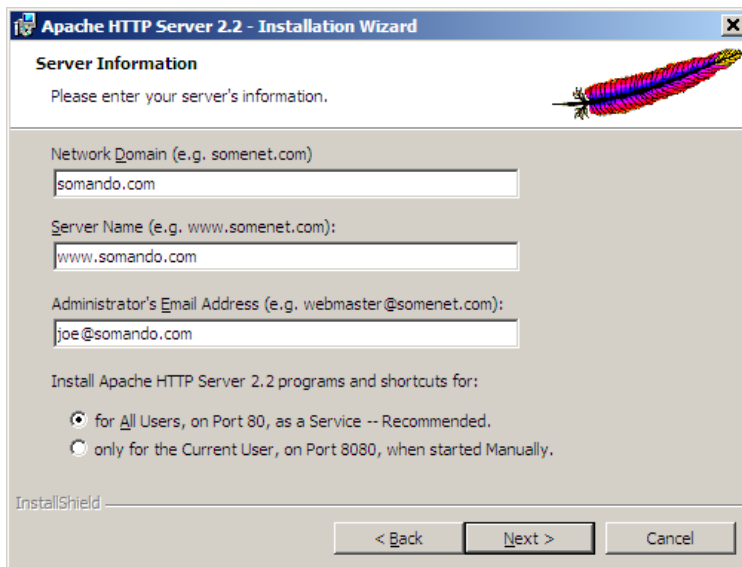


Figure 3b: Apache HTTP Server 2.2 Installer – Server Information

Unless you are installing Apache for production use, it doesn't matter too much what you enter into Network Domain, Server Name or Administrator's email, but be sure to select the option to Install Apache "for All Users, on Port 80, as a Service". Installing on Port 8080 would likely cause

a conflict between Apache and the Application Express software that comes with Oracle XE, which runs on port 8080. As soon as you are ready, click "Next >" to continue.

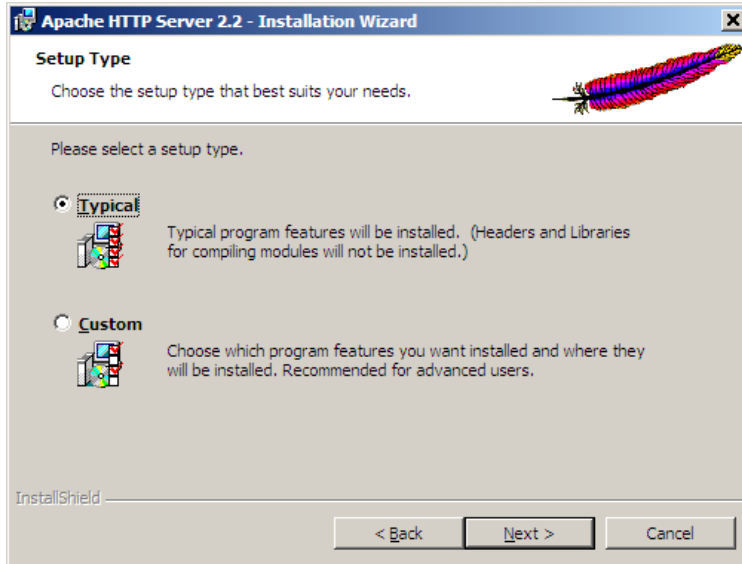


Figure 3c: Apache HTTP Server 2.2 Setup Type

On the "Setup Type" screen, you can safely leave the default option of "Typical" selected and click "Next >" to continue with the installation. Advanced users may wish to tweak their installation using the Custom option, but for the sake of this tutorial, there is no need to do so.

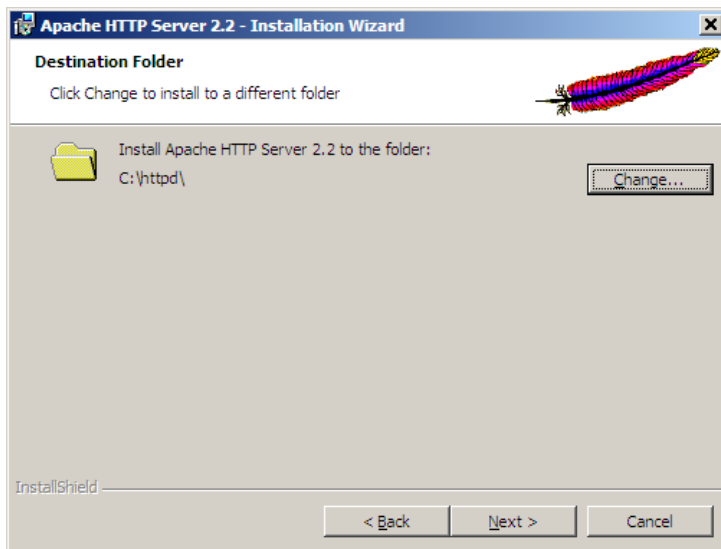


Figure 3d: Apache HTTP Server 2.2 Destination Folder

The Apache installer will now ask you where to install Apache on your hard disk. By default Apache is installed into a subfolder of the Program Files folder, but I find it easier to install Apache into a folder just beneath the root of the C drive, which I usually name "httpd". You can safely accept the default destination, but from here on I will refer to all Apache configuration files as if you chose to install to C:\httpd\. To install to this directory, click on the "Change" button as in Figure 3d, which should popup a screen like the one in Figure 3e. In the textbox for Folder Name, enter C:\httpd\ and click "OK". This will close the popup window and return you to the screen shown in Figure 3d. You can now click on "Next >".

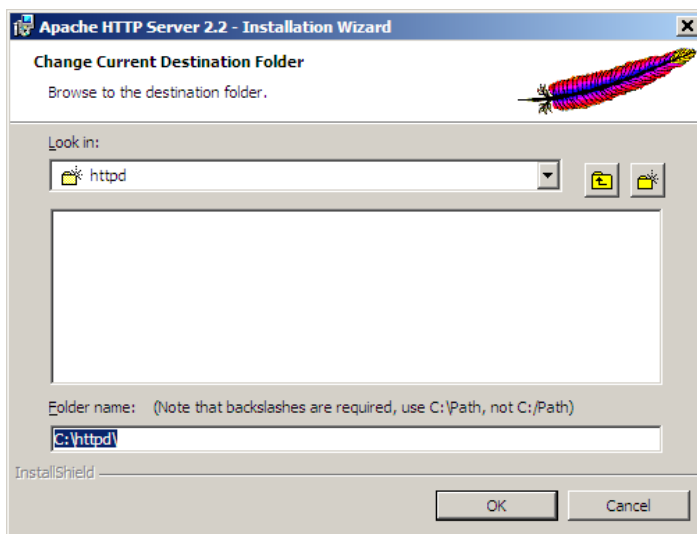


Figure 3e: Apache HTTP Server 2.2 – Change Current Destination Folder

You can now safely click on the "Install" button, which will start the installation process. This should not take long, especially if you are using a modern PC. During the installation you should see a screen similar to that in Figure 3f below.

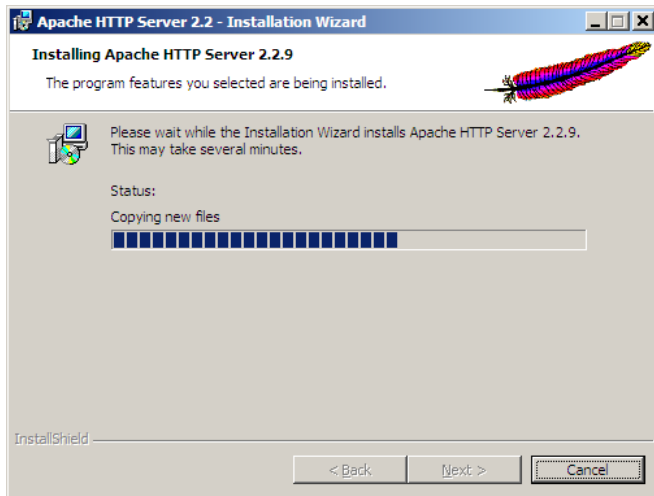


Figure 3f: Installing Apache HTTP Server 2.2.9

When the installation has completed, simply click the "Finish" button to exit the installer. You should now have a new icon in your system tray, like the one highlighted in Figure 3g. It should have a tiny green play symbol in it. If it has a red square stop symbol instead, left-click on the icon, and from the Apache2.2 menu, click "Start" to start the Apache service.



Figure 3g: Apache icon in System Tray

Before we move on to installing mod_owa, we will first check that Apache is up and running and functioning correctly. To do so, open your favourite web browser (I use Firefox), enter <http://localhost/> in the address bar and press enter. If Apache is working, you will see a message like in Figure 3h:

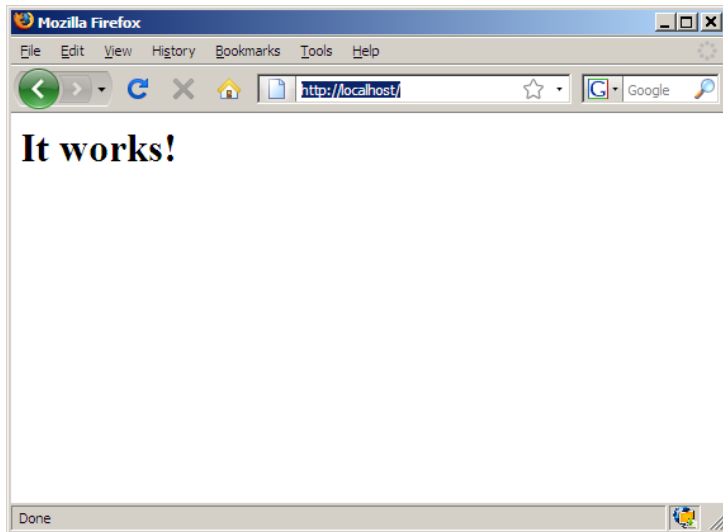


Figure 3h: Apache Test Page – It Works!

Now that we have Apache up and running, it is time to install our PL/SQL gateway, mod_owa. The first thing we need to do is to unzip the archive we downloaded earlier, usually named **windows_all.zip**. You don't need to worry about where you extract the files to, as we are only really interested in one of them. When the archive has extracted, open the folder you extracted them, and go into the modowa folder. From here, double click on apache22 and right-click on mod_owa.dll and click **Copy**.

We will now paste this into the **modules** folder under the Apache installation directory. Hold down the Windows key and press R to open the Run dialog and enter **C:\httpd\modules** as shown in Figure 3i:

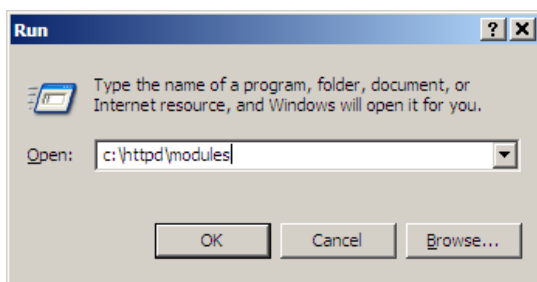


Figure 3i: Open Apache modules folder

This will open the modules folder. Go to the Edit menu and click on Paste to put a copy of the mod_owa.dll file here. Now that we have stored the module DLL file, we now need to tell Apache to load this module into the web server and set up a new Apache Location, which will act like an Oracle Document Access Descriptor (DAD) from where we can run our PL/SQL web applications.

If none of this makes any sense to you at this point, don't worry about it, just follow the instructions closely and you'll be fine.

To open the Apache configuration file (httpd.conf) open the Run dialog once again (Windows+R or Start->Run). This time, enter **notepad c:\httpd\conf\httpd.conf** and click the OK button. Notepad should launch, with the Apache config file opened and ready for editing. Scroll down to the bottom of this file (or hold down the Ctrl button and press the End button) and below all other code, insert the text from listing 3a:

```
LoadModule owa_module modules/mod_owa.dll

<Location /somando>
    AllowOverride None
    Options          None
    SetHandler       owa_handler
    OwaUserid        somando/somando1
    OwaNLS           WE8ISO8859P1
    OwaDiag          COMMAND ARGS CGIENV POOL SQL MEMORY
    OwaLog           "/usr/local/apache/logs/mod_owa.log"
    OwaPool          20
    OwaStart         "doc_pkg.homepage"
    OwaDocProc       "doc_pkg.readfile"
    OwaDocPath       docs
    OwaUploadMax     10M
    OwaCharset       "iso-8859-1"
    order            deny,allow
    allow            from all
</Location>
```

Listing 3a: mod_owa Options for Apache's httpd.conf

When you have added the text above to your httpd.conf file, save the file, and on the Apache icon in your system tray, left-click and from the Apache2.2 menu choose the "Restart" option to restart the Apache service. If all has gone well, Apache will restart just fine and you will have a play icon in your system tray icon once again. If you did not follow my guide to install Oracle XE, you will need to change the OwaUserid parameter in the text above to your own database's connect identifier.

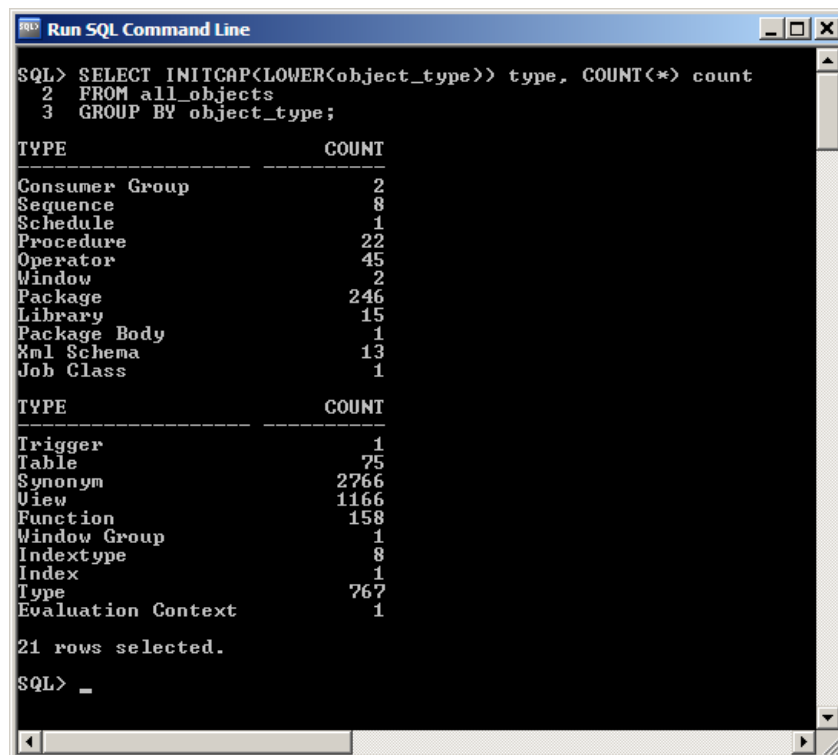
Once Apache has been restarted, mod_owa should now be running from <http://localhost/somando>. To test this, enter that address into your favourite web browser, and if mod_owa was installed you should see an error message like the one displayed in Figure 3j. If you see a different error message, you may not have installed mod_owa correctly.

The final thing we are going to do to make sure our Apache+PL/SQL setup is working correctly is create a sample application which performs a database **SELECT** and displays the output in a HTML table. Before we create the PL/SQL procedure to do this, let's run the query from the SQL*Plus command line to see what data we should expect to appear in our sample app. To open SQL*Plus, go to Start -> Programs -> Oracle Database 10g Express Edition -> Run SQL Command Line. When the command line appears log on by typing **CONNECT** and pressing Enter. When prompted, enter your database username and password (Username: somando Password: somando1 if you followed our Oracle XE installation guide) and as soon as you are connected, enter the SQL statement from Listing 3b to query the database.

```
SELECT INITCAP(LOWER(object_type)) type, COUNT(*) count
FROM all_objects
GROUP BY object_type;
```

Listing 3b: Query all_objects table by object_type

You should see a result similar to that illustrated in Figure 3j:



```
Run SQL Command Line
SQL> SELECT INITCAP(LOWER(object_type)) type, COUNT(*) count
 2 FROM all_objects
 3 GROUP BY object_type;

TYPE          COUNT
-----
Consumer Group      2
Sequence            8
Schedule            1
Procedure           22
Operator            45
Window              2
Package            246
Library             15
Package Body        1
Xml Schema          13
Job Class           1

TYPE          COUNT
-----
Trigger          1
Table             75
Synonym          2766
View             1166
Function          158
Window Group      1
Indextype         8
Index              1
Type              767
Evaluation Context 1

21 rows selected.

SQL> _
```

Figure 3j: Executed SQL Statement

Now we are going to create a PL/SQL procedure that will output this data to a HTML table that we can access using our web browser. From the SQL Command Line, enter the command **ed test_page**. Notepad will open and say that it cannot find the test_page.sql file, asking if you would like to create a new file. Click on the "Yes" button, and paste the text from Listing 3c into the Notepad window:

```
CREATE OR REPLACE PROCEDURE test_page IS
  CURSOR get_data IS
    SELECT INITCAP(LOWER(object_type)) type, COUNT(*) count
    FROM all_objects
    GROUP BY object_type;
BEGIN
  http.p('<table border="1">');
  http.p('<tr><th>Type:</th><th>Count:</th></tr>');
  FOR i IN get_data LOOP
    http.p('<tr><td>' || i.type || '</td><td>' || i.count || '</td></tr>');
  END LOOP;
  http.p('</table>');
END test_page;
/
```

Listing 3c: Create test_page PL/SQL Procedure

When you have pasted the above code into Notepad, click on File -> Save and then quit Notepad. Back in the SQL Command Line, you should have an SQL prompt. Here, enter the command **@test_page** to run the script we just created. If all goes according to plan, you should see a message "Procedure created". We can now test this procedure from our web browser by navigating to http://localhost/somando/test_page which should display something like what you see in Figure 3k below:

Type:	Count:
Consumer Group	2
Sequence	8
Schedule	1
Procedure	23
Operator	45
Window	2
Package	246
Library	15
Package Body	1
Xml Schema	13
Job Class	1
Trigger	1
Table	75
Synonym	2766
View	1166
Function	158
Window Group	1
Indextype	8
Index	1
Type	767
Evaluation Context	1

Figure 3k: Our sample PL/SQL web application

That's it! You are now ready to start developing PL/SQL web applications. In our next tutorial I will look at installing Oracle SQL Developer and how to set up a development environment for PL/SQL web application programming. Very shortly I will be writing tutorials on creating some neat web applications in PL/SQL.