Apache and Apache-ssl Proxy setup to Paradox Web Server OCX for Internet Enabled Databases

by Dennis Santoro

Last updated on Dec. 11, 2000

As shown in the paper Paradox Web Server by Tony McGuire (also available on the Paradox resources page on http://www.RDAWorldWide.com) Paradox can be a very simple and powerful solution for setting up a database oriented web site. The power and simplicity of this solution is a real strength of Paradox. One of the apparent weaknesses of the Paradox Web Server OCX is the inability of it to directly provide secure links to a browser through Secure Sockets Layer (SSL) encryption. Fortunately this issue has been solved thanks to some diligent work from a number of people in the Paradox community. The information below shows how the Paradox Web Server OCX can be used to serve data to a secure site running SSL. It also shows how the OCX can be used in conjunction with a web server not running SSL. In either case the web server can provide the primary site while passing data handling tasks off to Paradox. You can even combine all 3 to provide a non secure site and a secure site both with Paradox data handling.

As described below this is done with Apache and Apache-ssl. Both are available free for the download. This description also focuses on setup in Corel Linux running the web server and a windows machine running the OCX. You could extrapolate this to the Windows version of Apache and Apache-ssl to run the whole system on a single machine. You could also use a different distribution of Linux or any other OS which has an Apache version available. The basics will be the same although the details may vary.

This paper was put together based on ideas and news group postings provided by Dan Alder, Tony McGuire, Liz Woodhouse, Thad Van Ry and me. This paper will be revised as additional information becomes available. Anyone who wishes to submit the details of their experiences setting up Apache and the OCX in any environment please contact me through our web site or post them to the coreldevelopers.paradox-web news group on the cnews.corel.com server.

Getting Started:

To begin with you must understand how to set up and use the Paradox OCX to provide data and handle data requests. This is covered in Tony McGuire’s paper mentioned above. Also if you want to handle password protected tables and security at the paradox level you will either have to write that handling yourself or you can use Resource Development Associates’ Security Guard for the Internet.

Once you have your site functioning in the OCX the trick is to integrating it with a web server that will host static pages, other Internet services, and SSL if you want encrypted traffic. This is done more simply than you might expect. All that is required is to set up the Apache web server or Apache-ssl and then establish a proxy to the OCX location. This requires the Apache proxy module be installed and loaded. The details of this are below.

Setting Up Apache:
In Corel Linux, you can select Apache web server for installation when you set up the OS. If you have done that all you will need to do is configure the proxy. If you have not done so, go to Corel Update after logging in as root. Once there, select Apache from the available software and run the update.

Once Apache is installed you configure the proxy as follows:

1. Open your favorite editor and select the file /etc/apache/httpd.conf.

2. Find the line that looks like this:

```
# LoadModule proxy_module /usr/lib/apache/1.3/libproxy.so
```

and uncomment it by removing the #.

3. Add the lines

```
ProxyPass         /pdx/ http://127.0.0.1:8000/
ProxyPassReverse  /pdx/ http://127.0.0.1:8000/
```

where /pdx/ is the proxy you want to use to point to your OCX location, http://127.0.0.1 is the tcp/ip address of the machine running the OCX and :8000 is the port which the OCX is monitoring. If you have the OCX running as the only web server on the machine in question and litening to port 80, which is standard, then the port argument can be omitted. If you are running Apache and the OCX on the same machine then you will need to reset the OCX port and use the parameter. If you are using naming on your network or the OCX based machine is elsewhere on the Internet as a names host you can substitute the name for the tcp/ip address such as http://yourmachinename. (Note that if you need more than one proxy you will need to repeat the LoadModule line and then add additional ProxyPass and ProxyPassReverse lines.)

4. Save the file

5. You can then simply reboot. If you would rather not reboot you can do the following:

6. Find the line in the file you just edited that starts with "PidFile". Make a note of that location. Generally it is /var/run/apache.pid)

7. In the console or other xterm window issue the following line

```
kill -HUP `cat <location of pidfile>`
```

to restart apache. Note the quotes around the `cat <file>` using the back quote (under the tilde).

That’s it. Now Apache is listening on the Linux machine and becomes your web site’s primary location set up to your domain name if you are directly on the Internet or as the machine name or tcp/ip address if you are on an Intranet. When you need to use the OCX you simply point to the proxy. You can do this in the location line of your browser (where you normally see http://whatever) by simply typing in http://sitemachinename/pdx/ (where you substitute the name of your site machine and the name of the proxy you set up) or you can have your web pages pass requests to the proxy for you through a standard hyperlink.

Setting Up Apache-ssl:

Setting up Apache-ssl is only slightly more complex than the description above for regular Apache. To begin with, the needed components do not come with Corel Linux. But they can be added. You can download the components and install or install from the Internet. Two places to start for locating the components are: http://www.modssl.org and/or http://http://www.apache-ssl.org. These are two alternative SSL implementations for Apache. Once at these sites you will have to follow the links to the version for your distribution of Linux. Since Corel Linux is based on the Debian version you will need to use that version.

To enable SSL on Apache using Corel Linux OS doing the install from the Internet.

In Corel Update:

1. In Options/Set Package Sources
2. Add a new Custom Path with the following settings:

URL: ftp://ftp.de.debian.org/debian
Version: stable
Category: remove everything except "main"
(there will be four items in this list box by default)

After creating the Custom Path above:

1. On the Current Software Profile tab, expand the "base" category by clicking the + next to it.
2. Double-click on "ldso" until the arrows appear below Status
3. Click the upgrade icon in the toolbar.
4. Do the same for "libc6"

If you prefer you can download the relevant files from the site instead. File download locations are:

libdb2_2.4.14-2.7.7.1.c.deb
  http://packages.debian.org/stable/libs/libdb2.html
apache-common_1.3.9-13.1.deb
  http://packages.debian.org/stable/web/apache-common.html
libssl09_0.9.4-5.deb
  http://packages.debian.org/stable/non-us/libssl09.html
openssl_0.9.4-5.deb
  http://packages.debian.org/stable/non-us/openssl.html
apache-ssl_1.3.9.13-2.deb

  http://packages.debian.org/stable/non-us/apache-ssl.html
apache-doc_1.3.9-13.1.deb (optional)
  http://packages.debian.org/stable/doc/apache-doc.html

Once you have done either the download or the path setup you set up Apache-ssl as follows:

Using File/Install DEB File, install the following in this order:
libdb2_2.4.14-2.7.7.1.c.deb
apache-common_1.3.9-13.1.deb
libssl09_0.9.4-5.deb
openssl_0.9.4-5.deb
apache-ssl_1.3.9.13-2.deb
apache-doc_1.3.9-13.1.deb (optional)

Now to set up the proxy to the secure server you follow the proxy instructions for regular Apache shown above except that the configuration file will be: /etc/apache-ssl/httpd.conf instead of the one listed above.

To access the site you now type https://yourMachineName/yourProxy/ (note the s on http which indicates an ssl site). During the setup you will have to generate the basics for your site certificate. This is a fairly straight forward process. If you wish you can also get outside digital certificates which are recognized as from 3rd party digital certificate suppliers but this is not required.

Amazingly, that is all there is to the basic setup. However, setting up the secure site will make the unsecure site, http://yourmachine/ quit working. See the FAQ at http://www.apache-ssl.org/ for more information on how to have both run at the same time. Pay particular attention to the question: "I want to run secure and non-secure servers on the same machine. Is that possible?"

An additional disclaimer: for those files with "non-us" in the URL, please read the following (from http://packages.debian.org/stable/libs/):
"Software restricted in the US. Most of these packages can be used in the US, but they can't be exported (not even re-exported!). Some of these packages may not be used in the US due to software patents. You may need to check the regulations in your country before using this software."

Conditions of Use and Reproduction:

This content in this paper is provided as is, with no warranties, guarantees, or claims regarding its accuracy, completeness, or usefulness. While all efforts have been made to ensure its quality and accuracy, you are solely responsible for your own use of this information.

Any statements of fact contained in this article should be interpreted as the opinions of the author, which may or may not reflect the opinions of any other entity involved in the transactions that led you to this article.