

# Web Enabling your Hp e3000

## *Introduction*

Web enabling your HP e3000 can be done in many ways. The preferred is a three tier approach using an NT or Linux as the application/web server and the HP e3000 as the back-end database server.

The software used for the application server/web server is: Orion application server running on NT or Linux. The Orion application server is a very robust server and very inexpensive. Orion is written in 100% Java, so Orion will run on any platform that implements Java, including the HP e3000. We use Java Server Pages and Java servlets for the HTML presentation and data retrieval from an HP e3000. We use Adager Database Connectivity (ADBC ) as the application program interface (API) to your HP e3000. For the following examples, NT will be used as the middleware.

## *Installing your application.web server*

Orion is very easy to use. First, download the server from Orion ([www.orionserver.com](http://www.orionserver.com)), then unzip the contents of the zip file. All the directory structures will be created for you. Again, Orion can run on any platform that implements Java.

After the directories are created , start the Orion server from the Orion root directory.

```
D:\orion\java -jar orion.jar
```

Please note: Orion will only work with Java version 1.2 and above.

Orion resolves your customer classes from the d:\orion\lib directory. Any class you use in your JSP, EJB application will need to reside in this directory. You can store the files as class files or jar files.

You need to change the configuration files located in D:\orion\config so that Orion will know where your JSP files will be located. In the examples, the files are placed in D:\orion\application\agfa-apps\

## ***Installing ADBC for Back-end Access to your HP e3000***

ADBC has a listener program and a data manager program that run on your HP e3000. The listener program listens for Java clients and the data manager will process request and send back results to the Java client. You will need to run the listener program and data manager program in any account or simply create a new account to run the programs. The accounts will need MR and PH capabilities. Create a job stream to run the listener program in the background. Both programs must reside in the same group and account.

```
!JOB MGR.ADBC
!RUN LISTENER
!EOJ
```

The ADBC API java classes are located in a jar file called advnetsys.jar. You will need to copy the jar file into the D:\orion\lib directory.

After streaming the above listener program on your HP e3000, you are ready to begin programming.

## ***Using JSP and ADBC to Access Data from your HP e3000 for Web Access.***

The example is a very simple serial read of an IMAGE data set. Once the JSP page is accessed, it will serial read and report the results in an HTML table format. Any web browser will work ok with this solution.

The ADBC API comes with JSP tags. Whenever you make changes to the JSP file, the JSP server will recompile the JSP tags into pure Java servlet code.

First using any editor or IDE tool you need to creat a file with a JSP extension. In this case the file name is example.jsp.

The first code that is added is to indicate we will be running this page in session mode. Also, we need to tell the page where the ADBC tags are located.

```
<%@ page session="true" %>
<%@ taglib uri="looptags.jar" prefix="adbc" %>
```

Using HTML code, a title is created for my web page:

```
<html>
  <head>
    <title>ADBC Example</title>
  </head>
</body>
```

```
<adbc:mpe3000 id="mpe3000" url="161.153.226.2:30803" scope="session" />
```

mpe3000 is a variable that is used to identify an HP e3000 connection. The url will be the hosts ip address or hostname.

We now need to use JSP to identify a database, dataset and a result set to place the query results into so we can extract them for our web page. ADBC Does not use the term result set, we use the term TurboBuffer. The TurboBuffer will contain all the results for all database queries.

```
<adbc:database id="findb" name="findb.fdatas" mpe3000Id="mpe3000" mode="5"
password="ASK" scope="session" />

    <adbc:dataset id="promas" name="promas" dbId="findb" scope="session" />

    <adbc:turboBuffer id="product" dbId="findb" datasetId="promas"
dataset="promas" columns="pronum" fetchSize
```

In the above example we are using ManMan databases as examples. The database we are using is “findb.fdatas”. The dataset is “promas” and the turboBuffer is “product”.

All that is left is to query the dataset and report the results on the web page:

```
<adbc:read id="product" />

<TABLE BORDER="1">
    <%      while ( product.next() ) { %>
        <TR><TD>
            <A
href="/adbc/productdetail.jsp?product=<%=product.getString("pronum")%>">
            <%=product.getString("pronum")%>
            </A>
        </TD></TR>
    <%      } %>
</TABLE>
```

The read tag will do the dbfind and the next method will serial read the dataset. The field results “pronum” is extracted from the turbo buffer and is reported as a hyperlink to activate the jsp file product.

The complete code:

```
<%@ page session="true" %>
<%@ taglib uri="looptags.jar" prefix="adbc" %>

<html>
  <head>
    <title>ADBC Example</title>
  </head>
  <body>

    <adbc:mpe3000 id="mpe3000" url="161.153.226.2:30803" scope="session" />

    <adbc:database id="findb" name="findb.fdatas" mpe3000Id="mpe3000"
mode="5" password="ASK" scope="session" />

    <adbc:dataset id="promas" name="promas" dbId="findb" scope="session" />

    <adbc:turboBuffer id="product" dbId="findb" datasetId="promas"
dataset="promas" columns="pronum" fetchSize="200" scope="page"/>

    <adbc:read id="product" />

    <TABLE BORDER="1">
      <%      while ( product.next() ) { %>
        <TR><TD>
          <A
href="/adbc/productdetail.jsp?product=<%=product.getString("pronum")%>">
            <%=product.getString("pronum")%>
          </A>
        </TD></TR>
      <%      } %>
    </TABLE>

  </body>
</html>
```

This is just a very simple example of how to use a three-tier approach in web-enabling your HP e3000. You can create very sophisticated web pages using your HP e3000 as a back-end server.

The ADBC API is very robust and allows you to access IMAGE, MPE, KSAM and SPOOL files. It will also allow you to call existing subprograms located on your HP e3000.