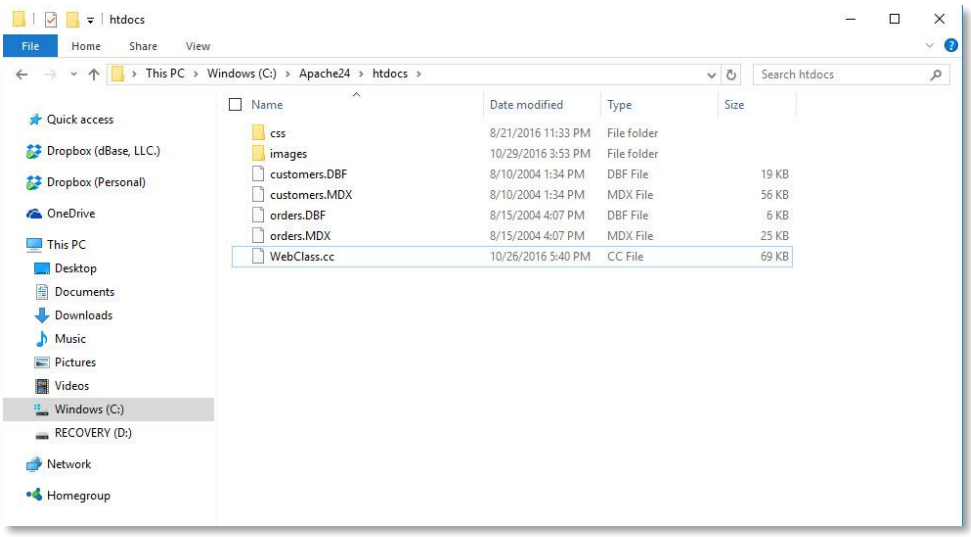
Exploring the dBASE™ PLUS 11 dbWeb™ Examples

dBASE™ PLUS 11 introduces a new dbWeb™ Wizard and five (5) dbWeb™ example programs provided in dBASE's abbrev.properties file which makes it very easy to use as a template for your dBASE™ Web programs.

This document will use one of those example programs for our exploration. There some additional assumptions we are making:

1) Our Windows Operating System is Microsoft Windows 10
2) dBASE™ PLUS 11 is installed in the C:\Program Files (x86)\dBASE\Plus11 directory
3) Our webserver will be Apache 2.4
4) Apache 2.4 is installed in the C:\Apache24 directory
5) The base webserver directory is C:\Apache24\htdocs

In the base webserver *htdocs* directory, we're going to need some additional files. The starting directory will look like this:

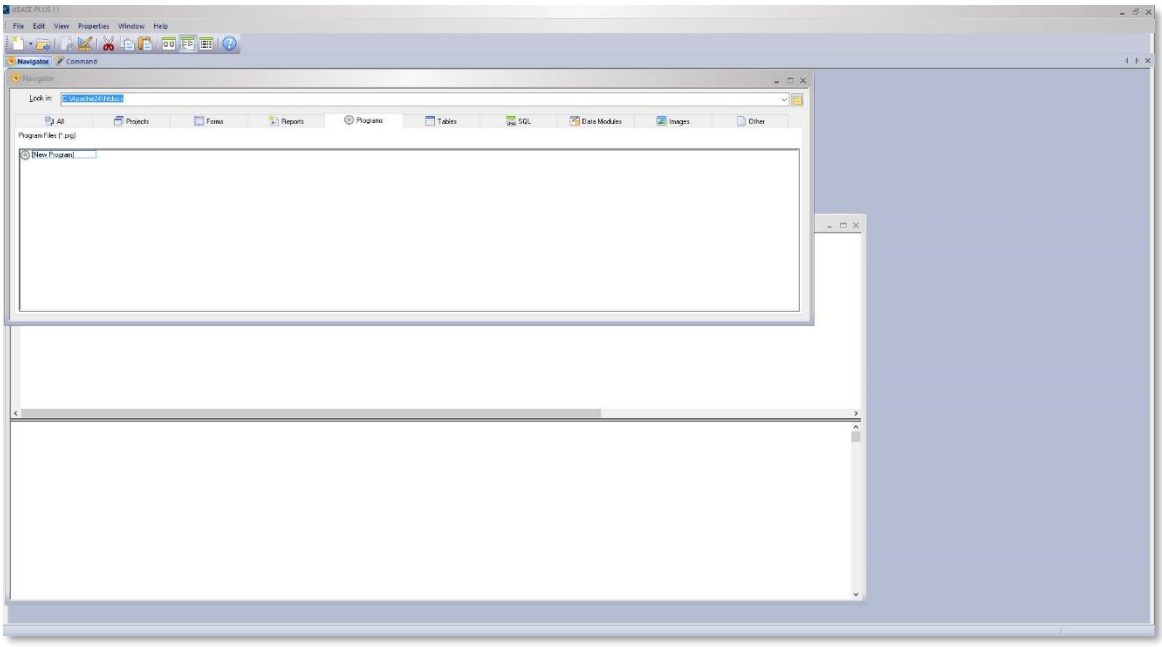


We will need to copy the *css* and *images* directories along with the *WebClass.cc* file from the C:\Program Files (x86)\dBASE\Plus11\Web\Wizards directory. We also need the Customers and Orders tables (.dbf) and indexes (.mdx) from the Samples directory, C:\Users\<username>\Documents\dBASE\Plus11\Samples.

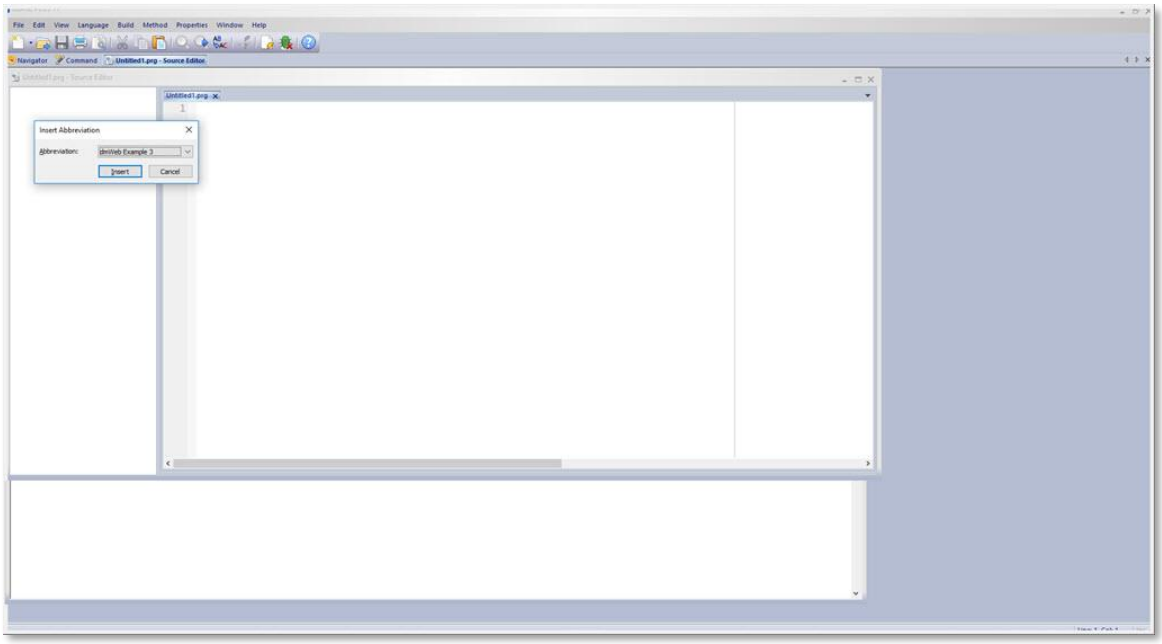
If you don't have Apache 2.4 setup as an automatic service, then start it in a CMD.exe window with the *httpd -w* command. Keep this window open until you are through using the webserver.

Copyright © 2016 dBase LLC.

Next you will want to start up dBASE™ PLUS 11 and change directories in the Navigator to the Apache *htdocs* directory.

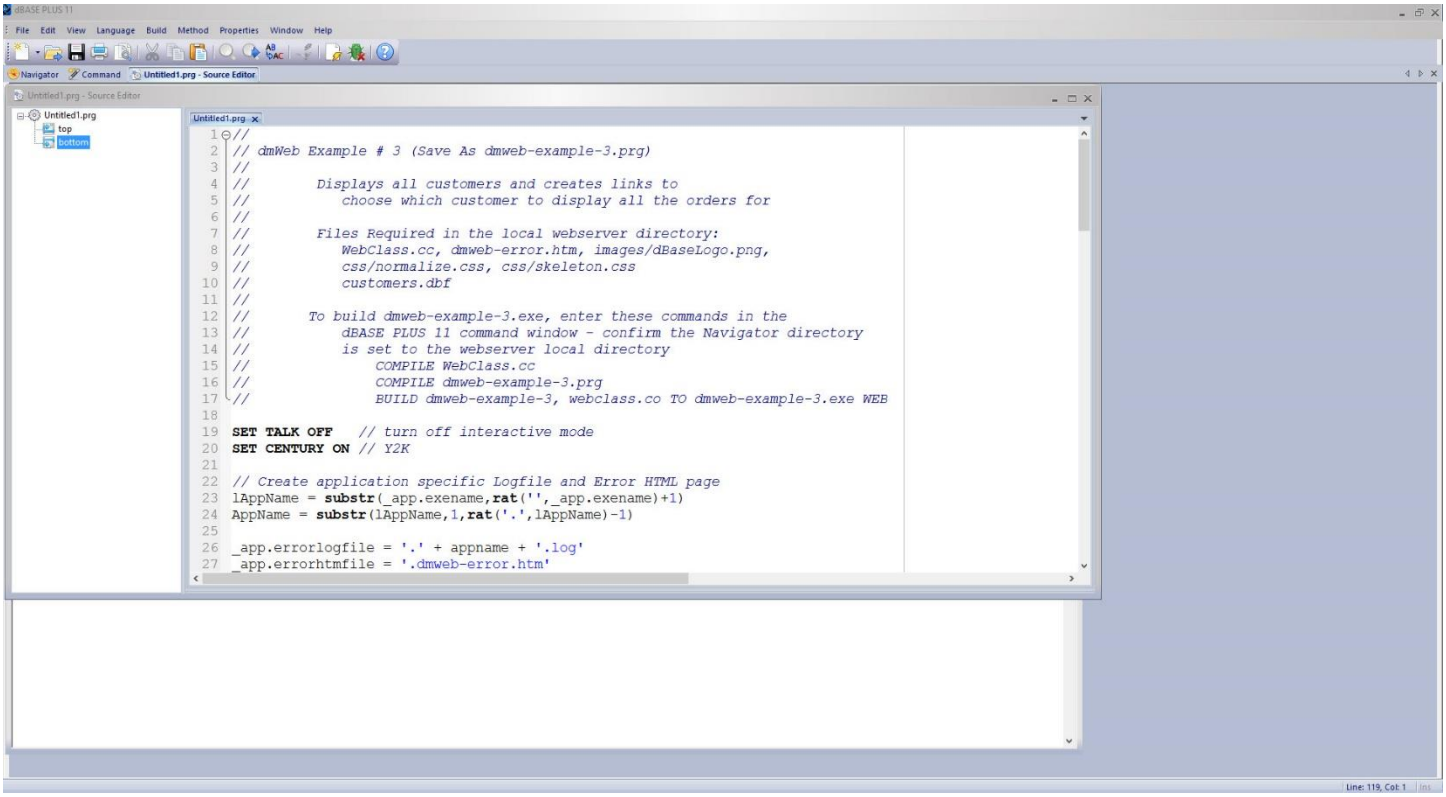


Click on the *[New Program]* link inside the Navigator and insert the dmWeb™ Example abbreviations by pressing *SHIFT-CTRL- B* or use the *Edit->Insert Abbreviation* pulldown menu item. Select the dbWeb™ Example 3 abbreviation.



Copyright © 2016 dBase LLC.

That will load the dbWeb™ example that reads the customer table and creates a customer listing with links to the *dmweb-example-4.exe* program which will list all the invoices for that company.



```
//
// dmWeb Example # 3 (Save As dmweb-example-3.prg)
//
//        Displays all customers and creates links to
//           choose which customer to display all the orders for
//
//        Files Required in the local webserver directory:
//           WebClass.cc, dmweb-error.htm, images/dBaseLogo.png,
//           css/normalize.css, css/skeleton.css
//           customers.dbf
//
//       To build dmweb-example-3.exe, enter these commands in the
//           dBASE PLUS 11 command window - confirm the Navigator directory
//           is set to the webserver local directory
//               COMPILE WebClass.cc
//               COMPILE dmweb-example-3.prg
//               BUILD dmweb-example-3, webclass.co TO dmweb-example-3.exe WEB

SET TALK OFF   // turn off interactive mode
SET CENTURY ON // Y2K

// Create application specific Logfile and Error HTML page
lAppName = substr(_app.exename,rat('',_app.exename)+1)
AppName = substr(lAppName,1,rat('.',lAppName)-1)

_app.errorlogfile = '.' + appname + '.log'
_app.errorhtmfile = '.dmweb-error.htm'
```

Now let's explore the content of this program and point the relevant code snippets:

First, because this program will run as a web program, we need a way to capture and display we'll setup to save the errors in a logfile and then display it as dynamic web page using the _app properties.

We override the default error log file with our own which is the application name with the .log extension.

```
_app.errorlogfile = '.\' + AppName + '.log'
```

We override the default error file template with our own (which is dmWeb Example 5 abbreviation).

```
_app.errorhtmfile = '.\dmweb-error.htm'
```

We set the errorAction to the value of 3 so it saves the error message to the logfile and also displays it as a web page.

```
_app.errorAction = 3
```

We load the *WebClass*.cc which contains dBASE's addon web functions that make creating dBASE web applications.

```
SET PROCEDURE TO WebClass.cc ADDITIVE
```

You must create the dBASE CGISession class.

```
oCGI = new CGISession()
```

You must then initialize the dBASE CGISession class which also loads into named pair array any values that were passed to the program via a POST command or arguments supplied via a link.

```
oCGI.Connect()
```

Copyright © 2016 dBase LLC.

You can create the link to call the dmweb-example-4.exe program (*dmweb-example-4.exe?CustomerID=48*) with code similar to this:

```
custNames.add( '<a href="dmweb-example-4.exe?CustomerID=' +;
        custRows.fields["CustomerID"].value +;
        '"><strong>' +;
        trim(custRows.fields["Company"].value) +;
        '</strong></a>')
```

Data driven dBASE web applications generate web pages via the CGI web interface which is very specific about how the web page header section must be setup. We have a special function, *streamHeader*, to do this. Just provide the title that you want displayed in the web browser's title banner.

```
oCGI.streamHeader("dBASE Customer Listing")
```

This creates the CGI HTML which is **REQUIRED** for the web CGI interface to work:

```
<HTML>
<HEAD>
 <META HTTP-EQUIV="Content-Type" CONTENT="text/html">
 <TITLE>dBASE Customer Listing</TITLE>
</HEAD>
```

The *streamBody* function sets up the formatting of the entire web page. The defaults are defined in the WebClass.cc file. It also includes the CSS and JavaScript that will make this web page responsive. Responsive web pages automatically adjust to the display size of the device rendering the page. This will allow one page to work on desktop browsers, tablets and cell phones.

```
oCGI.streamBody(cBackImage, cBackColor, cTextColor, cLinkColor, cVLinkColor, cALinkColor)
```

If you want to display a logo centered at the top of the web page, then use the *streamLogo* function. WebClass.cc set the transparent dBASE™ logo located in the *images* subdirectory of the Apache webserver *htdocs* directory.

```
oCGI.streamLogo() // Default image is dBASE Logo
```

Likewise, the *streamTitle* function will display a web page title centered on the page.

```
oCGI.streamTitle("dBASE Realtime Customer List")
```

To display other HTML code on the page you can encapsulate it in the *streamDetail* function.

```
oCGI.streamDetail('       <div class="container">')
```

To close the webpage, use the *streamFooter* function.

```
oCGI.streamFooter()
```

We have enclosed the entire page creating in a TRY / CATCH handler to process any error that occur between TRY and CATCH. If any errors happen then the *errorPage* function will be called to display the error in the standard format. Any errors that occur outside of the TRY / CATCH handler will be handled by the custom Log file and Error HTML template we defined at the start.

```
oCGI.errorPage(e)
```

Copyright © 2016 dBase LLC.

If there is a problem that is not a programmatic error but an unexpected result, you can use the *sorryPage* function to display a webpage with the specified title and message. You will have to support both the message (msg) and the page title (title) that you want it to display.

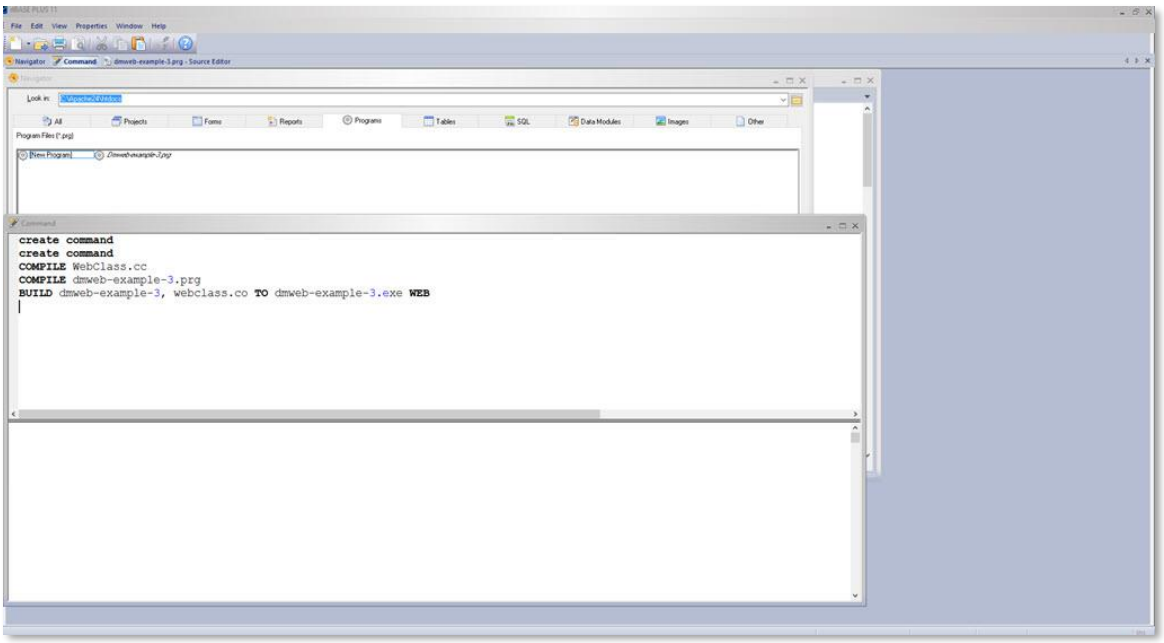
```
oCGI.sorryPage(msg, title)
```

Finally, you must specify the *QUIT* dBL command so the program completely exits and frees all resources.

```
QUIT
```

Once the program is completed, you must compile the source files, dmweb-example-4.prg and *WebClass*.cc, and then build the dmweb-example-4.exe program. Open the Command Window in dBASE™ PLUS 11 and enter the following commands:

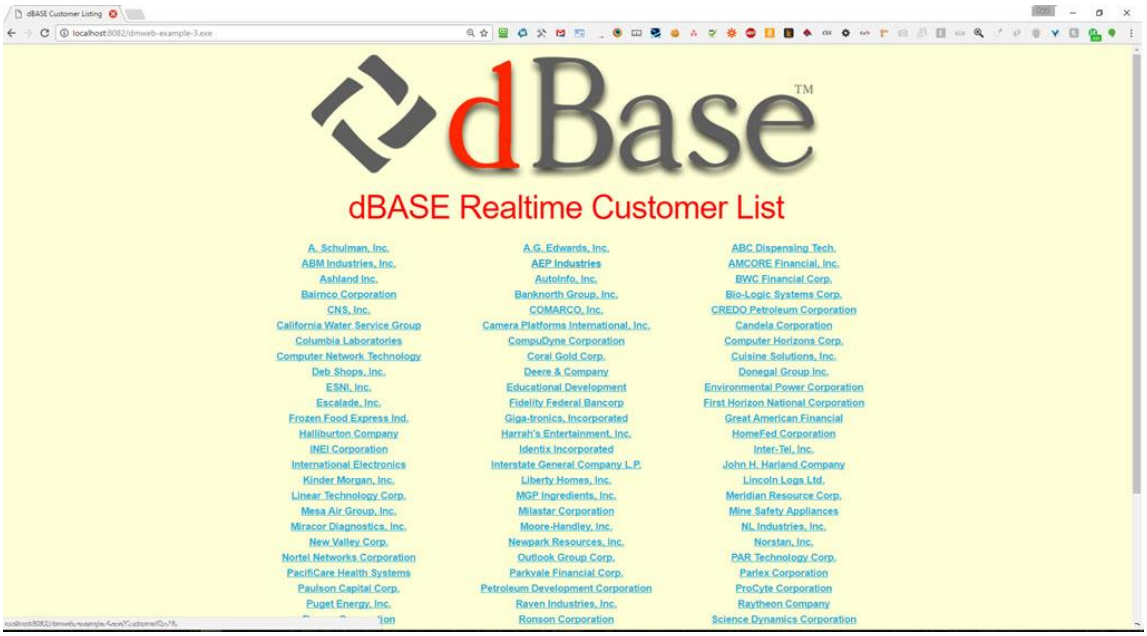
```
COMPILE WebClass.cc
COMPILE dmweb-example-3.prg
BUILD dmweb-example-3, WebClass.co TO dmweb-example-3.exe WEB
```



Copyright © 2016 dBase LLC.

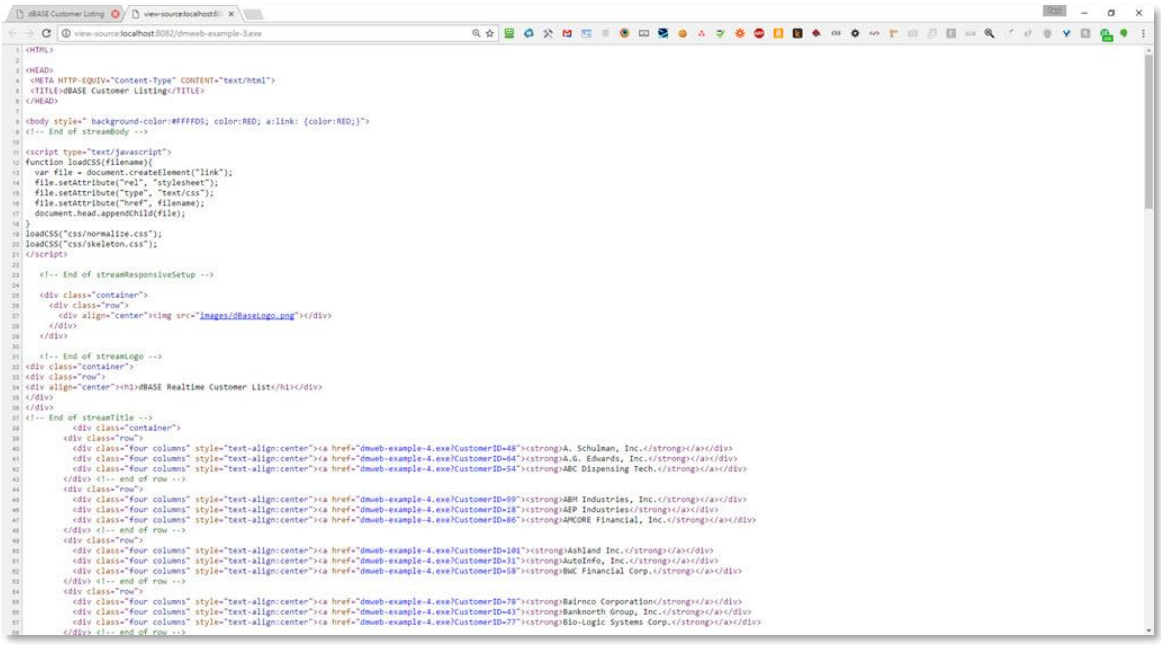
Finally, we get to run our compiled program in the Apache webserver using the URL below. The port number of *8082* is defined during the setup of the Apache 2.4 webserver.

http://localhost:8082/dmweb-example-4.exe



Hovering over any of the company names will display the destination link on the status bar at the bottom (only if your web browser supports a status bar).

This shows the actual HTML code generated from the dmweb-example-4.exe program.



Copyright © 2016 dBase LLC.

Finishing Up

There is a great deal more included in dBASE's WebClass.cc framework. Here are the functions included:

New Methods (some inspired by Ken Mayer):

errorMessage -- streams out a message on a page that has already started to display
streamTitle -- stream out the title at the top of the page <h1>
streamSubTitle -- stream out the title at the top of the page <h3>
streamDetail -- streams out individual lines of text and/or html
streamFormBegin -- steams out the <FORM> code with the cgi definition you specify for what to do with the form when it's submitted
streamFormEnd -- streams out the end tag for a form
streamText -- streams out an HTML entryfield
streamPassword -- streams out an HTML entryfield with a password mask
streamRadio -- streams out an HTML radio button
streamCheckbox -- streams out an HTML checkbox
streamReset -- streams out an HTML "Reset" button
streamSubmit -- streams out an HTML "Submit" button
streamTextArea -- streams out an HTML Editor control
streamSelectBegin -- streams out an HTML Combobox or Listbox (see details in documentation for this method)
streamOption -- stream out elements used in Select
streamSelectEnd -- end tag for a select object
streamHidden -- special object used to pass values in standard Name/Value pairs to an application when a form is posted, which do not have any UI (i.e., the user cannot see/interact with them!)
streamResponsiveHeader -- streams out responsive header including css and javascript for a responsive web page
streamResponsiveSetup -- streams out css and javascript for a responsive web page
streamLogo -- streams out logo (default is dBASE)
streamWebVars -- streams out Web data passed to CGI program
modSQLQuery -- modifies Existing SQL string to add a restriction clause (required) and to localize the data table (optional)

Revised Methods:

streamBody -- streams out the BODY tag, but with some parameters not in the original ...
sorryPage -- modified to use the methods defined here, and passing colors for the title ...
errorPage -- modified to handle passing colors through for the title ...

Existing Methods:

streamHeader -- streams out a header for CGI output
LoadArrayFromFields -- Load field info from dataset
loadDataModuleFromArray -- Load arrary info to update dataset
PassDataThrough -- Passes through all received CGI data to the next form

Copyright © 2016 dBase LLC.