Creating XBRL Instance Documents

By Efrem Litwin

Information Builders is currently in the early adopter stage of its XBRL (eXtensible Business Reporting Language) publishing option. Companies can use the Financial Reporting Platform extensions to create XBRL Instance Documents for submission to governmental agencies.

Before I get into further details of this endeavor, let me call your attention to another article I wrote for this issue of the WebFOCUS Newsletter, titled, “Using WebFOCUS to Report Against XBRL Documents,” on page 12. The article includes a little background on XBRL, some XBRL terminologies, and how to use WebFOCUS for reporting against XBRL. This solution is great for companies or governmental agencies that need to consume XBRL to consolidate financial information for reporting or data warehouse creation.

In December 2008, the U.S. Securities and Exchange Commission (SEC) mandated that public companies using U.S. GAAP (Generally Accepted Accounting Principles) with a worldwide public float of at least $5 billion could submit their financial statements in XBRL format, starting with their first fiscal period on or after June 15, 2009.

For a few years before that, the SEC had had an XBRL voluntary program through which companies could submit their financial results in XBRL format. Some countries already have various degrees of XBRL mandates in place. Other countries have XBRL initiatives in place whereby financial information can be submitted in XBRL format.

So what is Information Builders doing with XBRL? With Developer Studio 7.6.9 and WebFOCUS Reporting Server 7.7, WebFOCUS Financial Reports can be developed encompassing all of the mappings necessary for the XBRL Instance Document creation. One of the benefits of this solution is that an extract is not needed in order to perform the various XBRL mappings. Any data source accessible to WebFOCUS can be used. All XBRL mappings are performed within the Developer Studio environment.

The XBRL Instance Document creation process is as follows:

- Create a Financial Report in Developer Studio containing all the General Ledger account lines that are to be reported. Screen 1 on page 2 shows the initial financial report.

- Change the output format to XBRL and choose the XBRL taxonomy to be used in the XBRL Mapping process. XBRL taxonomies are the dictionaries that include the categorization schemes that define the specific tags (concepts) for individual items of data, such as "netprofit.” Screen 2 on page 2 shows the selected taxonomy for an output type of XBRL.
• Map the concepts from the taxonomy to the various General Ledger account lines within Financial Report Painter. **Screen 3** on page 3 shows the mapped concepts in red within the report matrix.
• Map report columns in Report Layout to the appropriate XBRL definition, such as timeframe, unit of measure, segment of business). **Screen 4** on page 3 shows the available XBRL mapping selections when the column is identified as a BY sort.

• When the output type is changed to XBRL, a WebFOCUS Validation Report is automatically inserted into the report flow.

This report accomplishes three things:
  - Creates a link to the created Instance Document
The report invokes UBmatrix’s XBRL Procession Engine (XPE), which is packaged with this solution to perform rollups and validation.

A WebFOCUS tabular report is displayed that uses the XBRL Adapter to read the XBRL Instance Document that was just created.

Screen 5 shows output of the WebFOCUS Validation Report.
Screen 6 shows the XBRL Instance Document that was just created.

Screen 7 shows the process flow of creating an XBRL Instance Document from within the WebFOCUS environment.
Active Technologies Update

By Yoshiko Akai

Our portable analytics platform, WebFOCUS Active Technologies has been evolving. We have not only added enhancements to the technology but also extended the platform to support interactive output formats so you can expand the use of the Active Technologies within your organization and introduce Business Intelligence to more users.

Active Technologies, which include Active Reports, Active Charts (Active Reports presented as charts), Active Dashboards (combining multiple Active Reports/Charts), Active Cache (connected mode to handle large datasets), and Active APIs (for standalone use), package both the data and the interactive functionality into a single file. Users have the ability to easily work with the data in either connected or disconnected mode.

While much of the emphasis of the technology has been on the portability of reports and dashboards for the mobile workforce, the real benefit of Active Technologies resides in the very simple-to-use Excel-like analytics capabilities that allow users to perform analytics by manipulating the report without causing any damage to the data.

With the support of Adobe Flash and PDF output formats introduced in the Active Technologies in WebFOCUS 7.6.9, you can provide rich content across your enterprise, and further enhance user productivity with the visual interactivity and the rapid response time provided by Rich Internet Applications (RIAs). RIAs are becoming more and more important because these applications offer Web-based interfaces that are as intuitive and responsive as the familiar desktop applications. They combine the advantages of the web with the usability of desktop applications.

Regardless of output format, Active Report menus provide guided navigation to the users to complete their tasks quickly without disruption. No technical knowledge is needed to interact with the report or the dashboard.

You can generate RIAs or, as we call them, Flash/Flex reports and dashboards, in a number of ways. By using WebFOCUS report development tools such as InfoAssist, you can convert any reports into Active Reports. Screen 1 on page 7 shows InfoAssist with different output format types.
An Active Dashboard is built from a group of Active Report components. Each Active Report can have the initial report view set to grid or pivot table, bar, line, or pie chart.

Screen 2 shows Active Report presentation options in Report Painter.

By combining these Active Report components into the dashboard view using the Document Composer, you can present complex data as a complete view of the business and allow users to analyze various business scenarios.
Screen 3 illustrates how to combine Active Reports to build an Active Dashboard in Document Composer.

You can easily convert the Active Dashboard into the different interactive output formats available in WebFOCUS, such as Active Flash/Flex.

Screen 4 shows an example of a compound document.

Users have options to view and analyze the data in many ways with the report or dashboard inside any browser or in Adobe Reader, using the menus with which they are already familiar.
Screen 5 shows Active Dashboard running inside Adobe Reader.

Screen 5

RIAs are rapidly becoming a standard for outward-facing applications to improve customer satisfaction and to increase business efficiencies over the Web and inside enterprises. You can leverage existing skills and knowledge of WebFOCUS Developer Studio to build RIAs without the need for any other tools or skills or third-party software. You can easily compile and deploy reports and dashboards using Active Technologies to enhance productivity and provide the values to the users who may have only received static reports and documents in the past.

For IT departments, Active Technologies provides the means for balancing server and client-side processing and computing. Even when a user runs an Active Report in the connected Active Cache mode, there is no trip to the backend systems; only back to the temporary dataset created on the WebFOCUS Reporting Server. This reduces traffic over the network and on the backend systems.

In WebFOCUS 7.6.9, Active Reports are now available as a trial without license. You can build new reports or dashboards or convert any existing HTML reports to share with the users and explore all of the functionalities available in the Active Technologies.
What Is zIIP, and Why Should I Care?

By Noreen Redden

This year at Information Builders’ Summit User Conference, CEO Gerry Cohen announced the coming of the zIIP. In FOCUS for z/OS 7.6.10, and WEBFOCUS release 8.0, FOCUS and WEBFOCUS on z/OS systems will be enabled to use the zIIP processors.

That’s good. But what exactly is it, and why should we care?

OK, first things first. What is it? IBM’s zIIP specialty engine is the newest of several specialty engines that IBM has implemented for mainframe consumption. IBM originally introduced it for DB2 and IP Security processing, both of which are IBM products, but recently the company has opened the processing doors to “enabled” software products.

zIIP stands for “system z Integrated Information Processor.” Not all processes can be sent to the zIIP. The Central Processor still must initialize jobs, control I/O interrupts, control user interaction with the operating system, etc. WorkLoad Manager, configured by the system programmers at a site, controls access and priority to the zIIP.

Knowing this will, of course, make your day and get you a seat on the train (so long as you pay full fare). But what does it really mean? What’s in it for your company? Well, the capacity of the zIIP engine does not count toward the overall MIPS rating of the mainframe image, and the CPU usage incurred on the zIIP is not chargeable in workload-based IBM software maintenance. So, with a zIIP processor, and enabled programs that exploit that engine wherever possible, there is more processing power available, at no additional cost, and in fact, with reduced costs from IBM.

Second, the zIIP engine always runs at 100 percent of the chip’s speed capacity, whereas the Central Processor is sometimes ramped down, based on the contract with IBM. More power, not more money. Sounds good to me.

Where does Information Builders come in? FOCUS or a WEBFOCUS server enabled for the zIIP means that some FOCUS functions can be “off-loaded” to the zIIP, (data conversions, calculations, formatting), while others remain on the central processor (I/O, external sort, etc.). As much as 80 percent of the FOCUS code is enabled to run on the zIIP, and the FOCUS program itself controls the switching.

Our programmers have put a lot of effort into ensuring that the usage of the zIIP is monitored to ensure that we only switch to zIIP processing when it would be most valuable; for instance, in a highly formatted output report. Additionally, program changes have been made to reduce I/O by using larger blocking factors to reduce CPU in the central processor. The customer will be able to display the CPU used by the central processor and the zIIP processor via &variables, ala &FOCCPU.

Check with your system programmers. You might already have a zIIP processor in house, especially if you are a large DB2 shop. Of course, to try it out, you will need to upgrade your current release of FOCUS or WEBFOCUS, but in these economic times, saving money is a good thing. If you need to upgrade your mainframe because of load, a zIIP might be a practical solution.

There are some caveats for FOCUS and WEBFOCUS, besides release version. First rule: All code must run from an authorized library. We send the FOCUS and server programs authorized, but it is up to the systems
programmer to authorize libraries. Therefore, if your company wants to take advantage of zIIP processing, all libraries that might be invoked must be authorized. That includes all of the load libraries distributed with FOCUS or WEBFOCUS (FOCLIB, EDALIB, FUSELIB, SQLLIB, etc.), as well as any other load libraries that might contain subroutines or functions.

If there are libraries in the STEPLIB that are not authorized, zIIP will automatically be turned off to prevent abends. Second rule: Access to the zIIP and priority to zIIP processing are controlled by WorkLoad Manager. Again, the systems programmer must give our programs access to the zIIP. So, make friends with those folks who configure WorkLoad Manager. We are all dependent here on the “kindness of strangers.” (Well, strangers to me!)

Much of this information has been gleaned from a white paper by Joel Starkman that will soon be available on our Web site, www.informationbuilders.com
Using WebFOCUS to Report Against XBRL Documents

By Efrem Litwin

XBRL (eXtensible Business Reporting Language) is an XML-type language for the electronic communication of business and financial data. It is becoming a standard around the world for companies to submit their financial results to various governmental agencies. WebFOCUS has extended its Financial Reporting Platform with the ability to create financial reports directly against XBRL. The way this is done is through the use of the Reporting Server’s XBRL adapter.

The XBRL adapter parses the XBRL documents to give a relational view to the data. Unlike the majority of our Reporting Server adapters, the current release does not require Create Synonym. There are a couple of XBRL Master File Descriptions packaged with the Reporting Server used for all XBRL documents. They are xbrlr.mas and xbrlrds.mas, which exist in the \home\catalog directory of the Reporting Server installation.

The XBRL Adapter functionality does exist in the 7.6 release of the WebFOCUS Reporting Server, but there are slight differences in the XBRLR and XBRLRD Master Files between the 7.6 and 7.7 versions.

The Allocation (FILEDEF) dialogue box is used to assign the Master File to a particular XBRL Instance Document. An XBRL Instance Document is a business report, such as a financial statement prepared to the XBRL specification. It is made up of a link to the taxonomy used in the creation of this document, the line from the taxonomy being reported (concept), the accuracy for each number (decimals or precision), and the context and units for each number. The context defines information such as time period for the data being reported. The unit defines the units for the data being reported, including USD and shares.

XBRL Taxonomies are the dictionaries that the language uses. These are the categorization schemes that define the specific tags (concepts) for individual items of data (such as "netprofit"). Included in the taxonomy is the long description for each concept as well as the relationship between one concept and another. In the XBRLR Master File, you will notice a FIELDNAME called PARENT. This field contains the value of the parent concept. For example, a concept of “CurrentAssets” might have a Parent Concept of “TotalAssets.” The LABEL_DESCRIPTION field contains the long description for each one of the concepts, such as “current assets.”

Below is a line from an Instance Document:

\<usfr-pte:OperatingRevenue contextRef="C0" decimals="-3" unitRef="U0">98166000</usfr-pte:OperatingRevenue>\n
The value being reported for the OperatingRevenue concept is 98166000

The contextRef refers to a place in the Instance Document that defines information about the number being reported like time period.

Below is an example of a context definition:

\<context id="C0"\>
  \<entity\>
    \<identifier scheme="http://www.amex.com/">ENG</identifier>\n  \</entity\>\n\</context\>
<period>
  <startDate>2008-01-01</startDate>
  <endDate>2008-03-31</endDate>
</period>
</context>

In this example, a concept with a contextRef equal to C0 states that the value for the concept is for a time period of January 1, 2008 to March 31, 2008.

The unitRef refers to a place in the Instance Document, which defines the units of the number being reported.

Below is an example of a unit definition:

<unit id="U0">
  <measure>iso4217:USD</measure>
</unit>

In this example, a concept with a unitRef equal to U0 states that the value for the concept is in US dollars.

To create a WebFOCUS report using XBRL documents as a data source, the allocation (FILEDEF) linking the XBRLR or XBRLRD Master to the XBRL Instance Document is first needed. See Screen 1.

Screen 1

The XBRLRD Master is used for Instance Documents where dimension references exist in the context definition.

Below is an example of a context definition containing a dimension reference:
Screen 2 shows all of the available fields that can be used in a WebFOCUS report that uses an XBRL Instance Document but that does not contain any dimensionality as a data source. When creating a financial report, the concept hierarchy from the taxonomy shows in a tree form on the right side of the Financial Report Painter. For the concept to appear in the financial report, the WebFOCUS report developer would need to click and drag the concept from the concept tree into the report matrix (Screen 3 on page 15).

Screen 4 on page 15 displays the output from the financial report.
Exiting a Maintain Application

By Mark Derwin

There are not a lot of ways that an application can end. The user can exit an application by clicking on an application event or by clicking on a browser control. Closing the browser, or changing the URL, can cause error or warning messages to appear.

This article discusses ways to exit an application without errors no matter what the user does. It’s important to note that the discussion in this article pertains to persistent applications.

Exiting the application
In a perfect world, your users would always click on the exit button on your form to leave an application. A standard exit button has the following Maintain code:

```maintain
self.WinExit();
```

This code tells Maintain to exit and return to the form that launched the application. This is usually the easiest and safest way to exit an application.

The next option is to redirect the user after the application ends. Using a simple command, when the user cleanly exits the Maintain application, any URL you want can be displayed:

```javascript
document.referrer = 'http://www.ibi.com';
self.WinExit();
```

This Maintain code displays the Information Builders home page when the Exit button is pressed.

Another technique is exiting or cancelling when you are using IBIVallidation. If you set IBIVallidation for a field to “required,” you cannot leave the form, even to cancel, unless the user enters a value. If you try, this is what you will see:

This is a required field. You must enter data.

But what if you are trying to cancel or exit without saving? Then you can use the following technique. For your cancel button, create a JavaScript trigger. Then you will need a line for each of the required fields:

```javascript
Form.Field.value='';
IWCTrigger("Closer");
```

Here, Form is the name of your form and Field is the name of your field. Remember, since this is JavaScript, it is case-sensitive. This technique assigns a value to the field and bypasses the required parameter. If you are trying to exit the application, Case Closer should look like this:

```javascript
Case Closer
Form.WinExit();
Endcase
```
Here, Form is the name of the form from which you are exiting. The case can be named anything that you want. If instead of exiting the application, you are just trying to return to a previous form, use the code:

Form.WinClose();

Where, again, Form is the name of the form you are closing.

**Exiting the application and the browser**
If the user leaves the application by exiting or redirecting the browser, you get the following warning:

This application has been disconnected from the server.

If you wish to disable this message so either the user or the application can close the browser, edit the EDASPROF.PRF and include the line:

`MNTCON EXIT_WARNING OFF`

Finally, what if when the user clicks the exit button, you want the application to close cleanly and close the browser? This is actually the safest way to make sure that your application and agents are closed properly. In the trigger behind your exit button, you need this JavaScript code:

```javascript
IWCTrigger("WINEXIT");
window.opener = self;
window.close();
```

Remember, this is JavaScript so case matters. You do not need a WINEXIT case in your code. This is a default. This code, combined with turning EXIT_WARNING off, causes a clean exit from the application and closes the browser without any error or warning messages.

One last note.: The above code works for IE6, but if you are using IE7 you have to add the following code:

```javascript
function OnExitButton_Click () {
    IWCTrigger("WINEXIT");
    var self = window.open('', '_self');
    self.focus();
    self.opener = self;
    self.close();
}
```

Using these techniques, you can be assured that your application and the agents on the server are terminated properly. You may need to combine more than one for your particular situation, but you will always get the desired results.
Developer Studio 7.6.9: Preserving Cookie Information

By Maria Volant

When working against a remote WebFOCUS environment in Developer Studio, cookie information is removed upon logging off from the remote environment, or when shutting down and restarting Developer Studio. Based on a customer request to preserve cookie information, we have implemented a solution in Developer Studio 7.6.9 that allows users to specify which cookies should be preserved.

Users can specify the names of the cookies they would like to preserve by creating and using what we call a “cookie exception list” in the dssso.xml template file. The code for the cookie exception list would look like this:

```xml
<cookie_exclude_list>
  <variable name="ParameterName" default="CookieName" visible="true"/>
  ...
</cookie_exclude_list>
```

Here, ParameterName is the parameter name for the cookie; CookieName is the name of the cookie; and visible="true" enables the display of the cookie parameter name and its value in the Authentication Settings dialog accessible through the WebFOCUS Environments Properties dialog.

Users can specify as many cookies as they wish. The list can be added to an existing authentication template block in the dssso.xml template file, just before the corresponding </authentication> closing authentication tag, or it can be added as a new template block. In the latter case, it would be listed as a new option in the dropdown list available under the Web Component Authentication area in the WebFOCUS Environments Properties dialog in Developer Studio.

For example, let’s say we add the code below as a new template block in the dssso.xml file for the purpose of creating a cookie exception list, which can be used with a WebFOCUS Environment that does not require a web server signon:

```xml
<authentication name="Preserve_Cookies_Template" desc="Cookie Exception List">
  <sso_logon_resource desc="Logon Resource" read_only="false" visible="false"/>
  <user name="user" desc="User's Name" default="%%environment%%" read_only="true" visible="true"/>
  <password name="password" desc="User's Password" default="%%environment%%" read_only="true" visible="true"/>
  <cookie_exclude_list>
    <variable name="ExcludeMR_COOKIE" default="MR_COOKIE" visible="true"/>
    <variable name="ExcludeWF_COOKIE" default="WF_COOKIE" visible="true"/>
  </cookie_exclude_list>
</authentication>
```

If we then add a WebFOCUS Environment in Developer Studio, the descriptive name of the template, which is shown in red bold text at the beginning of the sample code above, is displayed in
the WebFOCUS Environments Properties dialog along with the different authentication options, as shown toward the right bottom corner of **Screen 1**.

Screen 1

If we select **CookieExceptionList** from the dropdown list and click the Settings button in the above dialog, the Authentication Settings dialog comes up with the list of cookies we specified (highlighted in red bold text toward the bottom of the sample code above), as shown in **Screen 2** on page 20.
In the Authentication Settings dialog we have the ability to edit or remove the value of a cookie parameter. However, changes made in the Authentication Settings dialog would only apply to the selected WebFOCUS environment.

Any new environment we create will reflect the information entered in the dssso.xml template file. Thus, if we would like to make a change that would apply to any new WebFOCUS environment, we need to make the change in the dssso.xml file. If we want the change to be reflected for existing WebFOCUS environments, we need to recreate these environments once the dssso.xml file has been updated.

If we were to specify the cookie exception list above as part of an existing authentication template, such as SiteMinder, the code would look like this in the dssso.xml template file:

```xml
<authentication name="ibi_sm" desc="SiteMinder">
  <sso_logon_resource desc="Logon Resource" read_only="false" visible="false">
    <protocol default="%%environment%%" />
    <host default="%%environment%%" />
    <port default="%%environment%%" />
    <path desc="" default="/siteminderagent/forms/login.fcc" />
  </sso_logon_resource>
  <user name="user" desc="User Id" default="%%environment%%" read_only="true" visible="true" />
  <password name="password" desc="Password" default="%%environment%%" read_only="true" visible="true" />
  <logon_result name="SMSESSION" type="cookie" />
  <variable name="SMAUTHREASON" default="0" read_only="true" visible="true" />
  <variable name="TARGET" read_only="false" visible="true">
    <protocol default="%%environment%%" />
    <host default="%%environment%%" />
    <port default="%%environment%%" />
    <path default="/ibi_html/index.html" />
  </variable>
  <cookie_exclude_list>
    <variable name="ExcludeMR_COOKIE" default="MR_COOKIE" visible="true"/>
    <variable name="ExcludeWF_COOKIE" default="WF_COOKIE" visible="true"/>
  </cookie_exclude_list>
</authentication>
```
Screen 3 below shows how this information is reflected in the WebFOCUS Environments Properties dialog and the Authentication Settings dialog when configuring a SiteMinder environment.

Screen 3

When configuring Developer Studio to access a WebFOCUS Environment while using a cookie exception list, the product will not remove the file that contains the cookies for the WebFOCUS Environment. This is the case when performing a logoff for that environment or when shutting or starting Developer Studio, and the cookies specified in the exception list will be preserved.

For Windows XP, the cookies reside under `C:\Documents and Settings\UserName\Cookies` and have the form `UserName@MachineName.txt`. The cookies folder is hidden by default.

Things you should know

Here are some important tips regarding this implementation:
• This implementation is not currently supported for WebSEAL

• When performing a Service Pack installation, the dssso.xml template file is replaced with the default file, but it is backed up under the \...\DevStudio76\769_backup\bin directory. Thus, if you install DevStudio 7.6.9 as a Service Pack upgrade, you would need to either re-apply any changes previously made to the dssso.xml file, or we may restore the changes from the backed-up dssso.xml. Note that existing WebFOCUS configurations will not be affected by this, as the information for existing WebFOCUS configurations is stored in the wfcom.xml file. Changes in the dssso.xml file are only applied when creating a new WebFOCUS environment configuration.

For more information on how to preserve cookies in Developer Studio, please see the section “How to Add User-Specified Cookies” in the “WebFOCUS Security and Administration” guide.
Writing Code: Why Did It Do That?

By Art Greenhaus

For those of you who write perfect code every time, stop right here. There's no need to continue. Why mess with a good thing?

For the rest of us, we write code to do something, and something goes wrong. Now what? First, you have to consider what went wrong. I'm going to limit this discussion to two possibilities:

1. A language error (the server didn’t like our code as produced)
2. A retrieval error (what came back isn't what I wanted)

With a language error, you typically get something, indicating an error code (FOCxxx), which tells why the server didn't like your code. In fact, there are two areas that can contribute to errors in code:

1. Dialogue Manager execution
2. Command language

For those not familiar with the Dialogue Manager, it controls what command language gets executed. All Dialogue Manager control lines begin with a dash (-) in column 1. All Dialogue Manager variables begin with an ampersand (&).

Dialogue Manager functions like a .bat file in DOS, a shell script in Unix, or a CLIST or REXX procedure on IBM mainframes. It pre-processes the lines, substituting values for variables, doing comparisons and branching as specified. What the receiving facility sees, be it the OS or the FOCUS language processor, it no longer contains any of the pre-processing information; all the pre-processing is done.

As a simplified example. Consider the following:

```
TABLE FILE CAR
PRINT CAR
&SELECTION
END
-RUN
-IF &LINES EQ 0 GOTO EXIT;
```

&SELECTION is a Dialogue Manager variable. A value must be supplied for this procedure to execute (how it gets supplied is irrelevant here). Suppose it had the value WHERE SEATS EQ 2, the Focus language processor would see this:

```
TABLE FILE CAR
PRINT CAR
WHERE SEATS EQ 2
END
```

The '-RUN' is a Dialogue Manager command, saying to execute all preceding FOCUS code, and then continue. The next line (beginning '-IF') is another Dialogue Manager command, using the
statistical variable &LINES, whose value is set by the language processor. In this case, it's the number of lines of output from the previous FOCUS command. If the value of &LINES is 0, a branch to a Dialogue Manager label (EXIT) is performed.

With that basic explanation, one might ask how can I see what the Dialogue Manager is doing, since Dialogue Manager execution is normally not displayed. This can be controlled by using the SET command, to set the value for &ECHO. &ECHO can take any of three values:

- OFF (default) don't display anything
- ON display execution of Language Processor commands
- ALL display execution of D.M. and language

For debugging, it's a good idea to set &ECHO to 'ALL' to see everything. This can be done with the following:

-DEFAULT &ECHO='ALL'

(Sets the default value of &ECHO to 'ALL' for this procedure, but allow it to be overridden.)

Or

-SET &ECHO='ALL';

(Sets the value of &ECHO to 'ALL'.)

Or

SET DEFECHO=ALL

(Sets the default value of &ECHO to 'ALL' for any procedure where it's not otherwise SET.)

By viewing the execution in this way, you can see if there was an error in Dialogue Manager code.

Now, if there was no D.M. code in your procedure, and an error still occurs, the code may have an issue. Most error conditions are fairly self-explanatory, and should tell why your execution failed.

A more difficult issue is when the procedure did execute, but the results are not as expected. Then, you have to see what was executed. If your output is HTML (the default), you can right-click on the output, and choose “view source.” This will let you see the output of the ECHO of execution, as discussed above.

If we're using another output format (for example, PDF), then “view source” is not available. In this case, we just need to see what commands went to the language processor. The simplest way to do this is to suppress execution, so no output gets produced.

If we add the following to the procedure:

SET XRETRIEVAL=OFF

All language processing is done, but no data access is performed. This is useful, when the code being debugged is time consuming, or changes databases, which should not be done until the debugging is done. Unfortunately, any Dialogue Manager use of statistical variables (such as &LINES, discussed above) results in zeros, since no real execution is done. Sometimes, just temporarily changing the output format to HTML will suffice.
If we're using Developers Studio, there is another option. If we turn on the Message Viewer (click the down arrow to the right of the RUN button), our output now consists of two windows: the output, and the expansion of what was sent to the server.

Another reason for not getting expected results may be erroneous logic (that's up to the designer to determine), or because hidden (NOPRINTed) fields do not have what was expected. How can you tell what values the NOPRINTed fields have? The brute force method would be to remove all the NOPRINTs, and re-run the request. However, then you have to re-insert the NOPRINTs, for production. Invariably, one gets missed, or put in the wrong place.

An easier way is to turn off the word NOPRINT. This can be done by inserting the following at the beginning of the procedure:

```
LET NOPRINT = ;
```

This will replace the NOPRINT token with nothing, effectively removing the word NOPRINT from the language. Any execution will then display the results in any field previously NOPRINTed, allowing the values to be examined.

To reverse the action, issue this command:

```
LET CLEAR NOPRINT
```

Finally, what do you do when you change a procedure and the execution doesn't show your changes? Or worse, it shows execution of something entirely different from what you coded? Was something cached or broken? Probably not. When you have an APP PATH containing multiple directories, or you're using filenames that may appear in system paths, how do you tell from where the procedure is running? The answer is to use a command called WHENCE, which tells you from which directory a file was found. The syntax is as follows:

```
WHENCE filename datatype
```

Some valid datatypes are as follows:

```
FOCEXEC
MASTER
ACCESS
```

Thus, if I'm debugging a procedure called TROUBLE, but not seeing my changes take effect, I can issue the following:

```
WHENCE TROUBLE FOCEXEC
```

In this way you can find the source of the procedure being run.
Join the Focal Point Community Today!
Four thousand developers from almost every profession and industry are collaborating on original, elegant solutions to complex, technical issues right now.

You can join this exclusive group by registering for Focal Point, Information Builders' online community. Focal Point isn’t just a Web-based bulletin board, it’s an interactive network for problem-solving and sharing tips and techniques with your peers.

By becoming a Focal Point member you can:

- Peruse more than 11,000 different topics for answers to your questions on FOCUS, WebFOCUS, and iWay Software solutions
- Create a new topic for discussion and address your specific issue
- Display your technical prowess by posting a solution
- Stay informed on the latest developments in Information Builders’ solutions

Membership is free, but the information is priceless. Register today by visiting: http://forums.informationbuilders.com

Contact Kathleen_Butler@ibi.com for more information.

The WebFOCUS Newsletter is available online!
Visit http://www.informationbuilders.com/new/webfocus_news.html for the latest issue as well as links to back issues!